# Pontificia Universidad Católica del Ecuador

# Facultad de Ingeniería

**TEMA:**

ANÁLISIS DEL COMPORTAMIENTO DE LA MORTALIDAD EN EL ECUADOR, MEDIANTE EL USO DE MACHINE LEARNING.


**AUTOR:**

Adriana Nataly Jiménez Torres

**DIRECTOR:**

Henry N. Roa, PhD.


TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE MAGÍSTER EN SISTEMAS DE INFORMACIÓN MENCIÓN DATA SCIENCE

**Quito, febrero - 2024**

## DEDICATORIA

Este proyecto de fin de titulación está dedicado a mi héroe, mi mentor, al que siempre me motivó a superarme profesional y personalmente que, aunque ya no este físicamente, su amor, sabiduría y guía continúan iluminando mi camino cada día. Gracias papito por ser mi ejemplo de perseverancia y por enseñarme el valor del trabajo arduo y la dedicación, sé que me estás mirando desde arriba con orgullo mientras alcanzo este logro.

A mi querida mamita Carmen, que con su amor incondicional y sus palabras de aliento han sido un faro de luz en mi vida, esta titulación es un testimonio de su amor y apoyo inquebrantable, siempre estaré agradecida por el cariño y sabiduría.

A mi querido esposo Cris, por su amor, comprensión y apoyo incondicional, durante este viaje. Gracias por estar siempre a mi lado, animándome y celebrando cada paso de mi camino hacia esta meta, tu amor y aliento son el impulso para alcanzar mis metas y sueños, y estoy eternamente agradecida por tenerte a mi lado.

A mi mami Carmen, por su sacrificio y dedicación que han sido la fuerza motriz detrás de mi viaje académico.

A mis hermanas Ale y Caro, así como a mi hermano Steven, por el amor, apoyo y la inspiración que me han brindado en cada paso del camino, a mi hermosa familia Torres Castillo e Iñiguez Castro agradezco profundamente su amor, comprensión y apoyo constante, este logro también es de ustedes, y lo comparto con gratitud y alegría.

## AGRADECIMIENTO

Agradezco a Dios por la dicha de la vida y por haberme concedido la fortaleza, la sabiduría y la perseverancia necesaria para completar este proyecto de fin de titulación.

A mi director de proyecto PhD. Henry Roa, agradezco profundamente su orientación y su dedicación constante, su guía y liderazgo han sido fundamentales para alcanzar este logro.

A mis queridos amigos y amigas, quiero expresar mi sincero agradecimiento por su constante apoyo, aliento y amistad durante este emocionante viaje de mi proyecto de fin de titulación.

## RESUMEN

El presente trabajo tiene la finalidad analizar el comportamiento de la mortalidad en el Ecuador, mediante el uso de métodos de pronóstico de series temporales como el método de Holt con tendencia, el método multiplicativo de Holt Winters con estacionalidad y tendencia, así como los métodos autorregresivos ARIMA y SARIMA, además se emplea la técnica de agrupación k modes, para identificar patrones sobre la mortalidad.

Para llevar a cabo este análisis, se utiliza la metodología CRISP-DM, que permite desarrollar y evaluar los modelos, adicionalmente para los métodos de series de tiempo se utilizan diversas métricas de evaluación como el MSE, RMSE, MAE y MAPE, mientras que para la técnica de k-modes se utiliza el Silhouette Score.

Basándose en las métricas utilizadas, se concluye que el método SARIMA es el más adecuado para predecir la causa de muerte tanto por enfermedades isquémicas del corazón como por enfermedades cerebrovasculares, por otro lado, para la diabetes mellitus, el método de suavizado exponencial de Holt con tendencia se destaca como el más preciso entre los modelos evaluados. En el análisis de agrupación se identifican tres grupos distintos que presentan variaciones notables en la edad, causas de muerte y características demográficas.

# ÍNDICE


1. Introducción ..... 14
   1.1. Antecedentes ..... 14
   1.2. Planteamiento del Problema ..... 14
   1.3. Justificación ..... 15
   1.4. Objetivos ..... 16
      1.4.1. Objetivo General ..... 16
      1.4.2. Objetivos Específicos ..... 16
   1.5. Alcance ..... 16
2. Marco Teórico ..... 18
   2.1. Importancia de los Datos de Mortalidad ..... 18
      2.1.1. Sistema de Estadísticas Vitales ..... 18
      2.1.2. Factores Determinantes en la Mortalidad ..... 19
      2.1.3. Políticas Públicas de Salud ..... 19
   2.2. Machine Learning ..... 20
      2.2.1. Aprendizaje Supervisado ..... 20
      2.2.2. El aprendizaje No Supervisado ..... 21
      2.2.3. El Aprendizaje de Refuerzo ..... 21
   2.3. Series de Tiempo ..... 21
      2.3.1. Métodos y Técnicas de Pronóstico de Series Temporales ..... 22
      2.3.2. Métricas para Evaluar los Modelos de Series de Tiempo ..... 23
   2.4. Clustering ..... 24
      2.4.1. K-means ..... 24
      2.4.2. Clustering jerárquico ..... 25
      2.4.3. DBSCAN ..... 25
      2.4.4. K-Modes ..... 25
3. Metodología CRISP – DM ..... 27
4. Análisis de Series de Tiempo ..... 31
      4.1.1. Comprensión del Negocio ..... 31
      4.1.2. Comprensión de los Datos ..... 33
      4.1.3. Preparación de los Datos ..... 37
      4.1.4. Modelado ..... 41
5. Clustering ..... 74
      5.1.1. Comprensión del Negocio ..... 74
      5.1.2. Comprensión de los Datos ..... 75
      5.1.3. Preparación de los Datos ..... 80
      5.1.4. Modelado ..... 84
6. Análisis de Resultados ..... 90

6.1. Series de Tiempo ..... 90
6.1.1. Despliegue ..... 91
6.2. Clustering ..... 96
6.2.1. Despliegue ..... 97
7. Conclusiones ..... 98
8. Recomendaciones ..... 100

# ÍNDICE DE FIGURAS

Figura 1. Metodología CRISP-DM ..... 27
Figura 2. Tipos de datos base defunciones generales. ..... 34
Figura 3. Verificar valores nulos. ..... 35
Figura 4. Causas de muerte codificadas. ..... 36
Figura 5. Provincia de defunción. ..... 37
Figura 6. Tipos de datos de las variables de interés. ..... 37
Figura 7. Ejecución de código para eliminar defunciones ocurridas en el exterior. ..... 37
Figura 8. Ejecución de código para eliminar las muertes violentas ..... 38
Figura 9. Ejecución de código para eliminar por segunda ocasión las muertes violentas ..... 38
Figura 10. Ejecución de código para asignar la categoría que corresponde. ..... 38
Figura 11. Ejecución de código para eliminar causas de muertes atípicas y mal definidas. ..... 38
Figura 12. Ejecución de código para reemplazar valores vacíos. ..... 39
Figura 13. Ejecución de código para eliminar valores vacíos de lc1. ..... 39
Figura 14. Principales causas de muerte en el Ecuador. ..... 39
Figura 15. Ejecución de código para el teorema de Pareto. ..... 40
Figura 16. Ejecución de código para crear la fecha de defunción. ..... 40
Figura 17. Ejecución de código para filtrar las tres principales causas de muerte. ..... 40
Figura 18. Ejecución de código para agrupar la cantidad de defunciones por enfermedad. ..... 41
Figura 19. Ejecución de código para remplazar el mes de defunción ..... 41
Figura 20. Ejecución de código para modificar la fecha de defunción. ..... 41
Figura 21. Ejecución de código para creación de la función para eliminar valores atípicos. ..... 44
Figura 22. Ejecución de código para eliminación de valores atípicos. ..... 44
Figura 23. Ejecución de código para importan librerías para descomposición. ..... 45
Figura 24. Ejecución de código para dividir la data en entrenamiento y evaluación. ..... 46
Figura 25. Ejecución de código para el método de suavizado exponencial de Holt con tendencia. ..... 47
Figura 26. Ejecución de código para el método Holt Winter con tendencia y estacionalidad. ..... 47
Figura 27. Ejecución de código para crear la función para la prueba de Dickey-Fuller. ..... 48
Figura 28. Ejecución de código para obtener los resultados de la prueba de Dickey-Fuller. ..... 48
Figura 29. Ejecución de código para la transformación Box Cox. ..... 49
Figura 30. Ejecución de código para la diferenciación. ..... 49
Figura 31. Ejecución de código para verificar estacionariedad posterior transformación y diferenciación. ..... 49
Figura 32. Ejecución de código para dividir la data transformada en entrenamiento y evaluación. ..... 50
Figura 33. Ejecución de código modelo ARIMA. ..... 50

Figura 34. Ejecución de código para las predicciones. ..... 51
Figura 35. Ejecución de código para las invertir la transformación y diferenciación. ..... 51
Figura 36. Ejecución de código para el modelo SARIMA. ..... 52
Figura 37. Ejecución de código para las predicciones del modelo SARIMA ..... 52
Figura 38. Ejecución de código para invertir las transformaciones, modelo SARIMA. ..... 53
Figura 39. Ejecución de código para eliminar valores atípicos causa de muerte diabetes. ..... 54
Figura 40. Ejecución de código para dividir la data en entrenamiento y evaluación, causa de muerte diabetes. ..... 56
Figura 41. Ejecución de código para el método de suavizado exponencial de Holt con tendencia, causa de muerte diabetes. ..... 57
Figura 42. Ejecución de código para el método Holt Winter con tendencia y estacionalidad, causa de muerte diabetes. ..... 57
Figura 43. Ejecución de código para obtener los resultados de la prueba de Dickey-Fuller, causa de muerte diabetes. ..... 58
Figura 44. Ejecución de código para la transformación Box Cox, causa de muerte diabetes. .... 58
Figura 45. Ejecución de código para la diferenciación, causa de muerte diabetes. ..... 58
Figura 46. Ejecución de código para verificar estacionariedad posterior transformación y diferenciación, causa de muerte diabetes. ..... 59
Figura 47. Ejecución de código para dividir la data transformada en entrenamiento y evaluación, causa de muerte diabetes. ..... 59
Figura 48. Ejecución de código para búsqueda del mejor modelo ARIMA. ..... 60
Figura 49. Ejecución de código para las predicciones modelo ARIMA, causa de muerte diabetes. ..... 60
Figura 50. Ejecución de código para las invertir la transformación y diferenciación en el modelo ARIMA, causa de muerte diabetes. ..... 60
Figura 51. Ejecución de código modelo SARIMA, causa de muerte diabetes. ..... 61
Figura 52. Ejecución de código para las predicciones del modelo SARIM, causa de muerte diabetes. ..... 62
Figura 53. Ejecución de código para invertir las transformaciones, modelo SARIMA, causa de muerte diabetes. ..... 62
Figura 54. Ejecución de código para eliminar valores atípicos causa de muerte enfermedades cerebrovasculares. ..... 63
Figura 55. Ejecución de código para dividir la data en entrenamiento y evaluación, causa de muerte enfermedades cerebrovasculares. ..... 66
Figura 56. Ejecución de código para el método de suavizado exponencial de Holt, causa de muerte enfermedades cerebrovasculares. ..... 66

Figura 57. Ejecución de código para el método Holt Winter con tendencia y estacionalidad, causa de muerte enfermedades cerebrovasculares. ..... 67
Figura 58. Ejecución de código para obtener los resultados de la prueba de Dickey-Fuller, causa de muerte enfermedades cerebrovasculares ..... 67
Figura 59. Ejecución de código para la transformación Box Cox, causa de muerte enfermedades cerebrovasculares. ..... 67
Figura 60. Ejecución de código para la diferenciación, causa de muerte enfermedades cerebrovasculares. ..... 68
Figura 61. Ejecución de código para verificar estacionariedad posterior transformación y diferenciación, causa de muerte enfermedades cerebrovasculares. ..... 68
Figura 62. Ejecución de código para dividir la data transformada en entrenamiento y evaluación, causa de muerte enfermedades cerebrovasculares. ..... 69
Figura 63. Ejecución de código para búsqueda del mejor modelo ARIMA, causa de muerte enfermedades cerebrovasculares. ..... 69
Figura 64. Ejecución de código para las predicciones modelo ARIMA, causa de muerte enfermedades cerebrovasculares. ..... 69
Figura 65. Ejecución de código para las invertir la transformación y diferenciación en el modelo ARIMA, causa de muerte de enfermedades cerebrovasculares. ..... 70
Figura 66. Ejecución de código modelo SARIMA, causa de muerte enfermedades cerebrovasculares. ..... 70
Figura 67. Ejecución de código para las predicciones del modelo SARIMA, causa de muerte enfermedades cerebrovasculares. ..... 71
Figura 68. Ejecución de código para invertir las transformaciones, modelo SARIMA, causa de muerte enfermedades cerebrovasculares. ..... 71
Figura 69. Resultados de las métricas de los métodos aplicados para la causa de muerte enfermedades isquémicas del corazón. ..... 72
Figura 70. Resultados de las métricas de los métodos aplicados para la causa de muerte diabetes mellitus. ..... 72
Figura 71. Resultados de las métricas de los métodos aplicados para la causa de muerte enfermedades cerebrovasculares. ..... 72
Figura 72. Tipos de datos. ..... 76
Figura 73. Verificar valores nulos ..... 79
Figura 74. Causas de muerte. ..... 80
Figura 75. Provincia de defunción. ..... 80
Figura 76. Ejecución de código para seleccionar variables de interés. ..... 80
Figura 77. Ejecución de código para filtrar defunciones ocurridas en el Ecuador. ..... 81
Figura 78. Ejecución de código para eliminar muertes violentas. ..... 81

Figura 79. Ejecución de código para eliminar valores vacíos en la lista corta de causas de muerte. 81
Figura 80. Ejecución de código para asignar categorías a causas de muerte. 81
Figura 81. Ejecución de código para eliminar causas de muerte por COVID-19. 82
Figura 82. Ejecución de código para eliminar causas de muertes violentas. 82
Figura 83. Ejecución de código para cambiar el tipo de dato a la fecha de defunción 82
Figura 84. Ejecución de código para crear la fecha de defunción. 83
Figura 85. Ejecución de código para eliminar datos faltantes en edad. 83
Figura 86. Ejecución de códigos para eliminar valores vacíos. 83
Figura 87. Ejecución de código para filtrar los registros de menores de 60 años de edad. 83
Figura 88. Ejecución de código para crear categorías de la edad del fallecido y eliminar variable edad. 84
Figura 89. Verificar tipos de datos. 84
Figura 90. Ejecutar código para modificar el tipo de dato de las variables causa de muerte y fecha de defunción. 84
Figura 91. Ejecución de código para instalar librerías para k-modes. 85
Figura 92. Ejecución de código para determinar el número óptimo de clusters. 85
Figura 93. Ejecución de código para clustering. 86
Figura 94. Ejecución de código para insertar etiquetas del cluster 86
Figura 95. Ejecución de código para importar librerías. 86
Figura 96. Ejecución de código para obtener una muestra y modificar el tipo de datos. 86
Figura 97. Ejecución de código para obtener la distancia de Gower y calcular el Silhouette Score. 87
Figura 98. Ejecución de código para seleccionar nuevas variables. 88
Figura 99. Ejecución de código para determinar el número óptimo de clusters. 88
Figura 100. Ejecución de código para transformar las variables categóricas a dummies. 89
Figura 101. Ejecución de código para instalar e importar librerías para reducción de dimensionalidad. 89
Figura 102. Ejecución de código para inicializar y ajustar el objeto MCA y para crear las coordenadas. 89
Figura 103. Prueba de estacionariedad, método SARIMA. 91
Figura 104. Ejecución código modelo SARIMA. 91
Figura 105. Ejecución de código para entrenamiento del modelo SARIMA. 92
Figura 106. Ejecución de código para predicciones futuras. 92
Figura 107. Ejecución de código para crear el rango de fechas. 92
Figura 108. Ejecución de código para revertir la transformación y diferenciación. 93
Figura 109. Ejecución de código para definir el número de periodos usando el modelo previo. 93

Figura 110. Prueba de estacionariedad, causa de muerte enfermedades cerebrovasculares..... 94
Figura 111. Ejecución código modelo SARIMA, causa de muerte enfermedades cerebrovasculares. ..... 94
Figura 112. Ejecución de código para entrenamiento del modelo SARIMA, causa de muerte enfermedades cerebrovasculares. ..... 95
Figura 113. Ejecución de código para predicciones futuras, causa de muerte enfermedades cerebrovasculares. ..... 95
Figura 114. Ejecución de código para crear el rango de fechas, causa de muerte enfermedades cerebrovasculares. ..... 95
Figura 115. Ejecución de código para revertir la transformación y diferenciación, causa de muerte enfermedades cerebrovasculares. ..... 96

# ÍNDICE DE GRÁFICOS

Gráfico 1. Evolución de las defunciones en el Ecuador, periodo 1990 - 2021 ..... 34
Gráfico 2. Principales causas de muerte en el Ecuador. ..... 35
Gráfico 3. Evolución de la causa de muerte de enfermedades isquémicas del corazón. ..... 43
Gráfico 4. Boxplot causa de muerte enfermedades isquémicas del corazón. ..... 43
Gráfico 5. Boxplot de causa de muerte enfermedades isquémicas al corazón posterior al tratamiento de valores atípicos. ..... 44
Gráfico 6. Evolución de la causa de muerte de enfermedades isquémicas del corazón posterior al tratamiento de valores atípicos. ..... 44
Gráfico 7. Descomposición multiplicativa, causa de muerte enfermedades isquémicas del corazón. ..... 45
Gráfico 8. Descomposición aditiva, causa de muerte enfermedades isquémicas del corazón. ..... 46
Gráfico 9. Pronóstico de suavizamiento exponencial de Holt con tendencia. ..... 47
Gráfico 10. Pronóstico multiplicativo de Holt Winters. ..... 47
Gráfico 11. Transformación Box Cox. ..... 49
Gráfico 12. Transformación Box Cox y diferenciación. ..... 49
Gráfico 13. Pronóstico ARIMA ..... 51
Gráfico 14. Pronóstico SARIMA ..... 53
Gráfico 15. Evolución de la causa de muerte enfermedades diabetes mellitus. ..... 53
Gráfico 16. Boxplot causa de muerte enfermedades diabetes mellitus ..... 54
Gráfico 17. Boxplot de causa de muerte diabetes mellitus previo tratamiento de valores atípicos. ..... 54
Gráfico 18. Evolución de la causa de muerte diabetes mellitus previo tratamiento de valores atípicos. ..... 55
Gráfico 19. Descomposición multiplicativa, causa de muerte diabetes mellitus. ..... 55
Gráfico 20. Descomposición aditiva, causa de muerte diabetes mellitus. ..... 56
Gráfico 21. Pronóstico de suavizamiento exponencial de Holt con tendencia, causa de muerte diabetes. ..... 57
Gráfico 22. Pronóstico multiplicativo de Holt Winters, causa de muerte diabetes. ..... 57
Gráfico 23. Transformación Box Cox, causa de muerte diabetes. ..... 58
Gráfico 24. Transformación Box Cox y diferenciación, causa de muerte diabetes. ..... 59
Gráfico 25. Pronóstico ARIMA, causa de muerte diabetes. ..... 61
Gráfico 26. Pronóstico SARIMA, causa de muerte diabetes. ..... 62
Gráfico 27. Evolución de la causa de muerte enfermedades cerebrovasculares. ..... 63
Gráfico 28. Boxplot causa de muerte enfermedades cerebrovasculares. ..... 63
Gráfico 29. Boxplot de causa de muerte enfermedades cerebrovasculares previo tratamiento de valores atípicos. ..... 64

Gráfico 30. Evolución de la causa de muerte enfermedades cerebrovasculares previo tratamiento de valores atípicos. ..... 64
Gráfico 31. Descomposición multiplicativa, causa de muerte enfermedades cerebrovasculares. ..... 65
Gráfico 32. Descomposición aditiva, causa de muerte enfermedades cerebrovasculares. ..... 65
Gráfico 33. Pronóstico de suavizamiento exponencial de Holt con tendencia, causa de muerte enfermedades cerebrovasculares. ..... 66
Gráfico 34. Pronóstico multiplicativo de Holt Winters, causa de muerte enfermedades cerebrovasculares. ..... 67
Gráfico 35. Transformación Box Cox, causa de muerte enfermedades cerebrovasculares. ..... 68
Gráfico 36. Transformación Box Cox y diferenciación, causa de muerte enfermedades cerebrovasculares. ..... 68
Gráfico 37. Pronóstico ARIMA, causa de muerte enfermedades cerebrovasculares. ..... 70
Gráfico 38. Pronóstico SARIMA, causa de muerte enfermedades cerebrovasculares. ..... 71
Gráfico 39. Cantidad de defunciones por sexo, periodo 1990 – 2021. ..... 76
Gráfico 40. Cantidad de defunciones a nivel provincial, periodo 1990 – 2021. ..... 77
Gráfico 41. Principales causas de muerte en el Ecuador, periodo 1990 – 2021. ..... 78
Gráfico 42. Cantidad de defunciones por nivel de instrucción del fallecido, periodo 1990 – 2021. ..... 78
Gráfico 43. Distribución edad del fallecido. ..... 79
Gráfico 44. Método Elbow para determinar el número óptimo de clusters. ..... 85
Gráfico 45. Distribución de los clusters a nivel del género del fallecido. ..... 87
Gráfico 46. Método Elbow para determinar el número óptimo de clusters. ..... 88
Gráfico 47. Visualizar los clusters. ..... 90
Gráfico 48. Pronóstico SARIMA, valores futuros. ..... 93
Gráfico 49. Pronóstico usando el método de suavizado exponencial de Holt, incluyendo 60 periodos. ..... 94
Gráfico 50. Pronóstico SARIMA, valores futuros, causa de muerte enfermedades cerebrovasculares. ..... 96

## ÍNDICE DE TABLAS

Tabla 1. Plan del proyecto de fin de titulación, series de tiempo. ..... 32
Tabla 2. Plan del proyecto de fin titulación, sección clustering ..... 75

# 1. Introducción

## 1.1. Antecedentes

En América Latina y el Caribe, la tasa de mortalidad por cada 1.000 personas, a partir del año 2013 ha mantenido una tendencia creciente alcanzando al 2021 el 8,43; el crecimiento registrado ascendió a 2,29 puntos. En el caso específico del Ecuador la tasa de mortalidad por cada 1.000 habitantes a partir del año 2014 ha experimentado un aumento de 2,78 hasta el año 2020 (7,53), posteriormente al año 2021 la tasa de mortalidad se ubicó en 6,72; obteniendo así un resultado menor que el registrado en el año 2020 (Banco Mundial, n.d.).

El reciente impacto global del SARS-CoV-2 ha resaltado la importancia de una planificación eficaz por parte de las entidades gubernamentales y los organismos de salud de todo el mundo, este tipo de planificación implica la generación de proyecciones más precisas para el futuro y como resultado, existe la necesidad de utilizar mejores técnicas y métodos de pronóstico. En este contexto, el uso de técnicas de machine learning, han permitido optimizar los parámetros de los modelos, permitiendo predecir datos futuros (Ordoñez, 2023) (Mathonsi & Van Zyl, 2022).

En esta misma línea, el análisis y pronóstico de la mortalidad, son herramientas fundamentales en el sistema de salud, para la toma de decisiones informadas, vinculadas al desarrollo e implementación de políticas públicas, asignación de recursos económicos para el sistema de salud y para establecer directrices en el sector de la salud (Giraldo et al., 2017).

En la actualidad, en el Ecuador no existe información de acceso público que haga uso de técnicas de machine learning para determinar el comportamiento de la mortalidad en el futuro, la información disponible se limita a datos sobre la cantidad de defunciones ocurridas y la tasa de mortalidad, esta limitación constituye un obstáculo para la planificación nacional y para la ejecución de políticas públicas sociales y de salud, por ende, existe la necesidad imperante de incorporar métodos y técnicas de machine learning que permitan predecir la mortalidad y mejorar la toma de decisiones a nivel gubernamental.

## 1.2. Planteamiento del Problema

Según datos publicados por el Instituto Nacional de Estadística y Censos (INEC, n.d.) el número de defunciones ocurridas en el Ecuador ha mantenido una tendencia creciente a

partir del año 2014 hasta el año 2019, de esta manera en el año 2014 las defunciones ascendieron a 63.788 y aumentaron en 16,70% al 2019 (74.439), posteriormente para el año 2020 a partir de la pandemia por el SARS-CoV-2, esta cifra se incrementó en 57,44%, lo cual evidencia un crecimiento significativo en la mortalidad en el Ecuador.

La mortalidad históricamente ha desempeñado un papel esencial en la determinación del crecimiento demográfico, y ha sido una variable que influye en la calidad de vida de la población, los datos de mortalidad comúnmente son utilizados como indicadores del estado de salud y subrayan la importancia de que la reducción de enfermedades y tasas de mortalidad constituyen funciones fundamentales del gobierno (Naciones Unidas, 1978).

El modelado de la mortalidad es crucial para la planificación de las economías mundiales y para el sector de la salud, la capacidad de identificar patrones futuros proporciona herramientas para el diseño de estrategias (Mathonsi & Van Zyl, 2021).

En el contexto actual del crecimiento de la mortalidad en el Ecuador, es crucial comprender el comportamiento futuro de la mortalidad para la toma de decisiones informadas a nivel gubernamental, considerando que no solo refleja el estado de salud de la población sino también proporciona un aporte valioso para la planificación y ejecución de políticas públicas.

El análisis del comportamiento de la mortalidad a través del uso de métodos de machine learning, puede proporcionar a las instituciones y al Estado información valiosa para la toma de decisiones, los cuales requieren de información veraz para el diseño de programas y políticas de salud efectivas, siendo parte fundamental el pronóstico de la mortalidad y la identificación patrones.

### 1.3. Justificación

Los datos vinculados con defunciones, nacimientos, migración poblacional y la información obtenida a partir de los censos de población, son elementos fundamentales en el análisis de la estructura poblacional, considerando que permiten elaborar indicadores relevantes a nivel sociodemográfico y de salud, en este sentido la información sobre la mortalidad desempeña un papel crucial en la evaluación de los programas de salud, así como en la definición de acciones o estrategias, vinculadas a las políticas de salud (Organización Panamericana de la Salud, 2017).

Los modelos de predicción se han convertido en una importante área de estudio, sirven de base para la elaboración de políticas públicas enfocadas en el sistema de salud, estos modelos permiten crear herramientas de análisis y monitoreo de las tendencias de mortalidad (Giraldo et al., 2017), el análisis cuantitativo de la mortalidad y sus causas desempeñan un elemento esencial para evaluar la incidencia en las condiciones sanitarias de los países (Castañeda & González, 2014).

El Estado y las instituciones públicas requieren de información oportuna y veraz para la planificación nacional, la construcción de indicadores y de manera fundamental para el diseño y la ejecución de programas y políticas enfocadas en la salud pública, por ende es imperativo contar con métodos que permitan analizar los diferentes temas de interés, obtener relaciones entre los datos y predecir valores futuros para la ejecución de políticas públicas efectivas que permitan mejorar las condiciones de vida de la población.

La aplicación y selección de métodos y técnicas de machine learning permitirán construir modelos basados en datos históricos, con la finalidad de anticipar el comportamiento de la mortalidad, comprender sus causas e identificar patrones relevantes, esta información en la actualidad es clave para desarrollar políticas públicas y programas de salud más efectivos, que reflejen las necesidades de la población.

## 1.4. Objetivos

### 1.4.1. Objetivo General

Analizar el comportamiento de la mortalidad en el Ecuador, mediante el uso de métodos de pronóstico de series temporales y técnicas de agrupación, de modo que se proporcione información clave para el diseño y ejecución de políticas públicas.

### 1.4.2. Objetivos Específicos

- Analizar la evolución y las causas de mortalidad en el Ecuador.
- Aplicar métodos y técnicas de pronóstico de series de tiempo para determinar cuál es el más efectivo para predecir el comportamiento de la mortalidad en el futuro.
- Evaluar la precisión y la calidad de los modelos de series temporales.
- Aplicar técnicas de agrupación para identificar patrones sobre la mortalidad.

## 1.5. Alcance

El presente trabajo tiene como objetivo analizar el comportamiento de la mortalidad en el Ecuador, mediante el uso de machine learning, para el efecto, se utilizarán métodos de pronóstico de series de tiempo, con la finalidad de seleccionar el método más efectivo para predecir las tres principales causas de muerte en el Ecuador, adicionalmente se aplicará una técnica de agrupación para identificar patrones en los datos de la mortalidad.

Para llevar a cabo este análisis se aplicará cada una de las fases de la metodología CRISP-DM, en todas sus fases que incluyen desde el entendimiento del negocio, entendimiento de los datos, preparación de los datos, modelado, evaluación y despliegue, posteriormente, se analizarán los resultados obtenidos y se formularán conclusiones y recomendaciones basadas en estos hallazgos.

# 2. Marco Teórico

## 2.1. Importancia de los Datos de Mortalidad

Las estadísticas de mortalidad desempeñan un papel fundamental en el proceso de toma de decisiones referentes a la salud pública, contar con una medición precisa permitirá comprender los determinantes de la variación en la relación enfermedades y mortalidad en las diversas poblaciones, estas estadísticas constituyen un insumo directo para abordar y evaluar los programas de salud, realizar pronósticos, establecer acciones enfocadas en las políticas públicas de salud (Wang et al., 2022) (Organización Panamericana de la Salud, 2017), a partir de estos datos se han creado diversos métodos estadísticos para analizar la mortalidad, que van desde los enfoques simples hasta el análisis multivariado y los modelos de series de tiempo (Calderón, 2021).

En este contexto, las estadísticas de defunciones aportan de manera significativa en la planificación, seguimiento y evaluación, se destacan como herramientas esenciales para la generación de proyecciones poblacionales, en consecuencia, la información proveniente de las defunciones se utiliza para el planteamiento de objetivos y para la evaluación de los programas sociales y económicos (Naciones Unidas, 2014).

### 2.1.1. Sistema de Estadísticas Vitales

La principal fuente de información para el estudio de la mortalidad y la natalidad, son los sistemas de estadísticas vitales, en su mayoría estos sistemas están basados en la información capturada por el Registro Civil, entidad encargada de la inscripción de los hechos vitales como nacimientos, matrimonios y defunciones (Organización Panamericana de la Salud, 2017).

Los datos de estadísticas vitales principalmente se obtienen a través del sistema del Registro Civil, el cual proporciona información precisa, exhaustiva, oportuna y continua sobre los hechos vitales (Naciones Unidas, 2014).

En general, el marco legal y normativo de los países determina que las defunciones deben ser registradas a través del informe estadístico de defunción proporcionado por el profesional de la salud, que además de evidenciar la muerte, brinda información adicional sobre las causas de la muerte y proporciona información demográfica del fallecido (Organización Panamericana de la Salud, 2017).

En el Ecuador los datos sobre las defunciones se obtienen a partir del registro estadístico de defunciones generales, que recopila la información de los registros de las defunciones certificadas por los profesionales de la salud particulares, de establecimientos de salud públicos y privados, del servicio de medicina legal y ciencias forenses, y de las inscripciones que se realizan en las agencias del Registro Civil a nivel nacional (INEC, 2022).

### 2.1.2. Factores Determinantes en la Mortalidad

Los factores que afectan la mortalidad, como la educación, ocupación, comportamientos de salud, medio ambiente, violencia, condiciones físicas y socioeconómicas, son fundamentales para entender y abordar eficazmente este fenómeno, comprender estos factores es esencial para asignar adecuadamente los recursos disponibles, tanto en el contexto de las políticas públicas como en la planificación (Carracedo, 2017) (Cristancho, 2017).

Los factores conductuales como el consumo de tabaco y alcohol, así como los factores estructurales como la pobreza y las condiciones económicas, políticas y sociales, son determinantes en las disparidades de la mortalidad (Mackenbach et al., 2017).

En este escenario, la reducción de la mortalidad está asociada tanto a la mejora de las condiciones socioeconómicas de la población como a los avances en medicina, la implementación de métodos de prevención, la adopción de técnicas de saneamiento ambiental y el progreso en tecnología médica (Alba & Morelos, 2008), estos componentes permiten abordar y contrarrestar los factores causantes de la mortalidad.

### 2.1.3. Políticas Públicas de Salud

La información de la mortalidad se ha posicionado como una herramienta esencial para determinar el estado de salud de la población, conocer su nivel de vida y las condiciones sobre el acceso a servicios de salud, y consecuentemente esta información sirve de insumo para establecer políticas enfocadas en las necesidades de la población y vinculadas con la reducción de la desigualdad en el acceso a servicios públicos de salud (Organización Panamericana de la Salud, 2017).

Las tendencias actuales sobre la mortalidad y morbilidad resaltan la necesidad de enfocarse de manera global en la prevención y tratamiento de los traumatismos y las enfermedades cardiovasculares, cáncer, diabetes y neumopatías crónicas (Organización Panamericana de la Salud, 2020). Por lo tanto, es imperativo ejecutar programas de salud

enfocados en la prevención, diagnóstico y tratamiento de las principales enfermedades causantes de la mortalidad.

En el ámbito de las políticas públicas de la salud, se implementan programas destinados a ampliar la cobertura, reducir la desigualdad y mejorar los sistemas de salud, estos programas se organizan en base a diferentes componentes, incluyendo la provisión de medicamentos, accesibilidad a tratamientos, mejorar el enfoque de atención primaria y el desarrollo de políticas de salud participativas (Arriagada et al., 2005).

En este contexto, la información de la mortalidad sirve de base para el desarrollo de políticas públicas que permitan abordar las necesidades en el sector de la salud, siendo el Estado, el actor principal y gestor de las políticas públicas de salud.

## 2.2. Machine Learning

El machine learning es el campo de estudio que confiere a las computadoras la capacidad de aprender sin requerir de un proceso de programación, permite aprender de datos de entrenamiento específicos para automatizar el proceso de construcción de modelos analíticos y resolver tareas asociadas, su aplicación está enfocada en enseñar a las máquinas como gestionar los datos de manera eficiente (Mahesh, 2018) (Janiesch et al., 2021).

El machine learning es una herramienta fundamental en el campo de la ciencia de datos, mediante la utilización de metodologías estadísticas, los algoritmos son entrenados para realizar clasificaciones o predicciones, así como para descubrir información relevante en proyectos de minería de datos (IBM, n.d.), en otras palabras el machine learning tiene como finalidad gestionar el aprendizaje autónomo de las maquinas o sistemas, dicho aprendizaje puede definirse en tres tipos de algoritmos que se describen a continuación.

### 2.2.1. Aprendizaje Supervisado

Para el aprendizaje supervisado se requiere disponer de un conjunto de datos de entrenamiento que incluyan entrada de datos etiquetadas, en este tipo de aprendizaje se pueden distinguir entre problemas de regresión, donde se predice un valor numérico, y problemas de clasificación, donde el resultado de la predicción es una afiliación de clase categórica (Janiesch et al., 2021).

Las técnicas comunes de aprendizaje supervisado incluyen la regresión lineal y logística, así como muchos de los algoritmos de aprendizaje automático más populares, por ejemplo, árboles de decisión, máquinas de vectores de soporte (Bi et al., 2019).

### 2.2.2. El aprendizaje No Supervisado

El aprendizaje no supervisado se aplica cuando algún conjunto de datos no se encuentra etiquetado y por lo tanto la única manera de organizarlo sea a través de la identificación de similitudes o diferencias (Hinestroza, 2018). El algoritmo busca identificar relaciones naturales y agrupaciones dentro de los datos prescindiendo de cualquier referencia a un resultado específico o la "respuesta correcta", ejemplos de algoritmos de agrupación incluyen por ejemplo k means y agrupamiento de maximización de expectativas (Bi et al., 2019).

### 2.2.3. El Aprendizaje de Refuerzo

En el aprendizaje por refuerzo, se describe el estado actual del sistema, se establece un objetivo y se proporciona una lista de acciones permitidas junto con las restricciones para los resultados, el modelo se basa en el experimento de la maximización de la recompensa a partir del principio de prueba y error (Janiesch et al., 2021), este tipo de aprendizaje sigue un enfoque iterativo en función de los resultados positivos o en base a la retroalimentación negativa basada en el desempeño de una tarea explícita en algunos datos (Bi et al., 2019).

## 2.3. Series de Tiempo

Una serie de tiempo es una secuencia ordenada de datos recopilados a lo largo de un periodo de tiempo (Daza & García, 2021). El propósito del análisis de series temporales consiste en explicar las variaciones observadas en la secuencia pasada, tratando de determinar si estos son el resultado de un comportamiento específico, una vez identificado ese patrón, se busca pronosticar el comportamiento de la variable (Florencia, 2018).

Una serie temporal se puede caracterizar de acuerdo con sus componentes:

a. **Tendencia:** Constituye el componente de largo plazo que establece la base de crecimiento (o decrecimiento) de la serie. En caso de que la serie sea estacionaria, su media y varianza permanecen constantes.
b. **Estacionalidad:** Se refiere al comportamiento de una serie en un período específico. Las series temporales pueden presentar patrones que se repiten de un período a otro.
c. **Ciclos:** Representan desviaciones de la tendencia subyacente causada por diversos factores (generalmente externos), los cuales son distintos de la estacionalidad. El tiempo y duración de los ciclos no necesariamente es constante.

**d. Aleatoriedad:** Son fluctuaciones impredecibles o no periódicas que subyacen en la serie (Gambini & López, 2018).

### 2.3.1. Métodos y Técnicas de Pronóstico de Series Temporales

Los métodos de proyección histórica o de series de tiempo emplean un análisis minucioso de los patrones en los datos a lo largo del tiempo, extrapolando estos patrones hacia el futuro, este proceso se basa únicamente en los valores pasados de la variable objetivo (Corres et al., 2009) (Villarreal, 2016).

Existen diferentes métodos que se utilizan en el análisis de series temporales, que comprenden desde enfoques muy simples hasta enfoques más complejos.

**Métodos Simples de Pronóstico de Series Temporales:** En este tipo de métodos podemos encontrar al método ingenuo o Naïve, el promedio simple y el promedio móvil. Las predicciones del método Naïve, se basan en el valor de la última observación (Guerrero & Medina, 2016), mientras, que el promedio simple utiliza el promedio de todas las observaciones para realizar pronósticos (Pathak, 2021), finalmente el método de promedio móvil utiliza el promedio de los k valores de datos más recientes en la serie de tiempo (Villarreal, 2016).

**Técnicas de Suavizado Exponencial:** La técnica de suavización exponencial calcula el promedio ponderado de los valores de una serie temporal pasada para pronosticar valores futuros (Villarreal, 2016), estos métodos generan pronósticos confiables rápidamente y para una amplia gama de series temporales, lo cual es una gran ventaja (Pathak, 2021).

Existen tres técnicas, suavizado exponencial simple, método Holt con tendencia y el método Holt Winters.

El método de suavizado exponencial simple realiza las predicciones basándose en un valor del período anterior, es adecuado cuando la serie no es estacional y no tiene una tendencia constante, mientras que el método Holt realiza las predicciones en función de la tendencia, por otro lado, el método Winters modela el nivel general de la serie, así como la tendencia y la estacionalidad (Bello & Martínez, 2007).

El método Winters, puede ser aplicado de maneta aditiva o multiplicativa, el enfoque aditivo es recomendado para las series con una tendencia lineal y con un patrón estacional que no está relacionado con el nivel de la serie, mientras que el modelo multiplicativo es

más apropiado para series con tendencia lineal y con un patrón estacional que esta influenciado por el nivel de la serie (Florencia, 2018).

**Métodos Autoregresivos:** En los métodos autorregresivos, la técnica de regresión se utiliza para pronosticar observaciones futuras, utilizando una combinación lineal de observaciones pasadas, pero para ello la serie temporal debe seguir los supuestos de estacionalidad y autocorrelación (Pathak, 2021).

- **Método de Regresión Automática (AR):** En este enfoque el proceso se representa como la suma ponderada de las observaciones pasadas (Ríos & Hurtado, 2008).
- **Método de Media Móvil (MA):** En estos modelos la predicción se realiza en función de los errores de pronóstico pasados en un modelo similar a una regresión (Pathak, 2021).
- **Método de Media Móvil de Regresión Automática (ARMA):** En estos modelos, el pronóstico se realiza en función de las observaciones pasadas y de los valores actuales y rezagados (Ríos & Hurtado, 2008).
- **Media Móvil Integrada Regresiva Automática (ARIMA):** Este método permite representar un valor como una combinación lineal de datos previos y errores aleatorios y además incluye la posibilidad de incorporar un componente cíclico o estacional (Florencia, 2018), son los modelos de series de tiempo aditivos más utilizados, especialmente cuando la serie temporal a modelar no es estacionaria a causa de una clara tendencia (Guerrero & Medina, 2016).
- **Media Móvil Integrada Autoregresiva Estacional (SARIMA):** SARIMA es esencialmente lo mismo que ARIMA, pero tiene un componente adicional de estacionalidad (Pathak, 2021), pueden manejar datos con patrones estacionales y los parámetros asociados permiten capturar las fluctuaciones estacionales en los datos (Subramanian et al., 2023).

### 2.3.2. Métricas para Evaluar los Modelos de Series de Tiempo

La evaluación del rendimiento de los modelos de series de tiempo depende principalmente de la diferencia entre el valor predicho y el valor real, cuanto menor sea el valor, mejor será el efecto de predicción del modelo (Zhao et al., 2023).

La elección de métricas de desempeño para el pronóstico de series de tiempo depende del problema específico y de los objetivos del análisis. Las métricas de evaluación comunes

para la predicción de series de tiempo, incluyen el error cuadrático medio (MSE), el MSE raíz (RMSE) y el error absoluto medio (MAE) (Subramanian et al., 2023).

Adicionalmente también se utilizan como métricas de evaluación a las siguientes: MAPE (Mean Absolute Percentage Error), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) (El Amrani, 2020), (Sánchez, 2020).

**Mean Square Error (MSE):** Se calcula en base al promedio de las diferencias al cuadrado entre los valores predichos y los valores reales, esta métrica penaliza más los errores mayores debido a la operación de cuadratura (Subramanian et al., 2023).

**Root Mean Squared Error (RMSE):** Se obtiene a partir de la raíz cuadrada de la media de las diferencias al cuadrado entre los valores predichos y los observados (Subramanian et al., 2023).

**Mean Absolute Error (MAE):** El error absoluto medio se calcula a partir del promedio de las diferencias absolutas entre los valores reales y las predicciones, esta métrica representa el promedio de los residuos en los datos (Chugh, 2020).

**Mean Absolute Percent Error (MAPE):** Es la diferencia del promedio absoluto entre los valores ajustados por el modelo y los valores observados (Chugh, 2020).

## 2.4. Clustering

Las técnicas de agrupamiento no supervisadas se basan en la agrupación de conjuntos de datos o instancias con características similares, a través de clusters, lo cual contribuye al proceso de clasificación (Mattiev et al., 2023) (Bhatt et al., 2023), un clúster representa una colección de instancias, donde cada miembro es más similar a otros miembros del mismo clúster que a aquellos fuera de él, esta técnica ha encontrado numerosas aplicaciones en diversos dominios, que van desde la investigación de mercado y los motores de búsqueda hasta la psicología, la medicina, la biología y más (Gratsos et al., 2023).

### 2.4.1. K-means

K-means es un algoritmo de agrupamiento que funciona con restricciones, divide los datos en k grupos para minimizar la distancia euclidiana de los centros de los grupos, recibe como parámetro el número de grupos que se deben formar y está diseñado para trabajar con conjuntos de datos continuos, en otras palabras, este algoritmo es aplicable

únicamente a objetos que se describen mediante un conjunto de atributos numéricos (Mattiev et al., 2023) (López, 2007).

Este tipo de algoritmo es el más común, pero no es efectivo cuando se trabaja con datos categóricos (Bhatt et al., 2023).

### 2.4.2. Clustering Jerárquico

El método de agrupación jerárquica se basa en la asignación de grupos se lleva a cabo a través de la construcción de una jerarquía, que puede ser implementada a través de un enfoque ascendente o descendente, se genera una jerarquía de puntos, representada por un árbol, conocida como dendrograma (Nissa, 2020), se construyen grupos que se fundamentan en la idea de que los elementos de un conjunto tienden a tener una relación más estrecha con los elementos más cercanos que con los más lejanos, este método conecta a los elementos para formar grupos basados en la presencia de características comunes (Battaglia et al., 2016).

### 2.4.3. DBSCAN

Es un algoritmo de agrupación basado en densidad, su objetivo es localizar regiones de alta densidad que están separadas entre sí por regiones de baja densidad, puede identificar grupos en un gran conjunto de datos espaciales observando la densidad local de los elementos correspondientes (Nissa, 2020), este método permite generar un número desconocido de grupos dentro de un conjunto de datos, al mismo tiempo que realiza un filtrado del ruido y de los valores atípicos presentes en el conjunto de datos (Navarro & Alencastre, 2016).

### 2.4.4. K-modes

El algoritmo de K-modes para clustering es una variante del algoritmo K-Means diseñado para el agrupamiento de datos categóricos, aunque sigue el mismo enfoque que el algoritmo K-Means y mantiene su estructura general, la principal diferencia radica en la medida de similitud utilizada para cotejar objetos. Este tipo de algoritmo utiliza modas y se basa en un método de frecuencias para actualizar las modas (López, 2007).

Ese método de agrupamiento posibilita la identificación de patrones ocultos y de subgrupos naturales en los datos (Atif et al., 2023), su objetivo es identificar grupos que comparten modos similares en las categorías, permite capturar la estructura presente en los conjuntos de elementos contenidos en las reglas, revelando patrones y asociaciones

que podrían permanecer ocultos en los métodos de agrupamiento basados en números, ofrecen eficiencia computacional y escalabilidad para grandes conjuntos de datos con variables categóricas, puede manejar datos de alta dimensión y una gran cantidad de categorías dentro de cada atributo (Mattiev et al., 2023).

El proceso de este método se basa en la selección inicial de los centroides, luego, cada instancia se asigna al centroide más cercano según una métrica de distancia para los datos categóricos, la métrica de distancia más utilizada para la agrupación de modos k es la distancia de Hamming, que calcula el porcentaje de atributos que difieren entre dos instancias (Gratsos et al., 2023).

## 3. Metodología CRISP DM

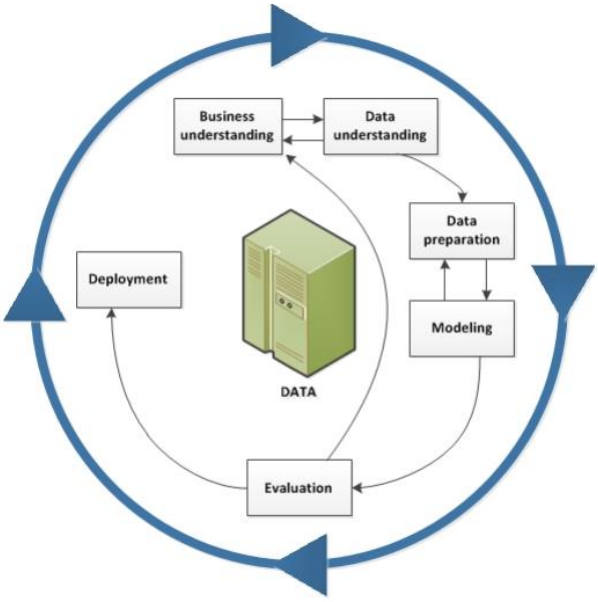
CRISP-DM contiene seis fases que muestran las dependencias más significativas y más frecuentes entre fases, la secuencia de las fases no es rígida, los proyectos pueden avanzar y retroceder según sea necesario, esto hace que el modelo sea flexible y personalizable en función de los requisitos del proyecto (IBM, n.d.).

Esta metodología es actualmente la guía de referencia más utilizada en el desarrollo de proyectos de Data Mining (Mishra et al., 2023), proporciona un enfoque estructurado para guiar todo el proceso de extracción de datos, desde la comprensión del problema empresarial hasta la implementación de la solución final (Mustapha et al., 2023).

Existen metodologías que describen minuciosamente las actividades de cada fase del proceso de minería de datos, un ejemplo claro es la metodología CRISP-DM, por otro lado, algunas solo ofrecen una guía general de las actividades a realizar en cada fase, como el proceso KDD o SEMMA. La metodología SEMMA se centra especialmente en aspectos técnicos, dejando de lado actividades de análisis y comprensión del problema abordado, mientras que KDD hace referencia al proceso completo de descubrimiento de conocimiento (Moine et al., 2011).

Las fases son: comprensión del negocio, comprensión de los datos, preparación de datos, modelado, evaluación y despliegue.

*Figura 1. Metodología CRISP-DM*




Fuente: (IBM, n.d.)

a) **Comprensión del negocio:**
   - **Determinación de los objetivos comerciales:** En esta fase se lleva a cabo una recopilación de la información que permitirá conocer el contexto de la organización, la problemática y la solución, luego se definen los objetivos comerciales enfocados en las soluciones identificadas y se detallan los criterios de rendimiento comercial.
   - **Evaluación de la situación:** El proceso de evaluación implica generar un inventario de recursos (hardware, personal), requisitos, supuestos, restricciones, riesgos, contingencias, que influyan en el cumplimiento de los objetivos del proyecto, también se realiza un análisis de costes operativos y de despliegue, así como los beneficios potenciales del rendimiento.
   - **Determinación de los objetivos de minería de datos:** En esta fase se especifican los objetivos de minería de datos y los criterios de rendimiento enfocados en los métodos de evaluación.
   - **Plan del proyecto:** Se detallan las fases del proyecto incluyendo información relacionada con los objetivos, recursos y riesgos vinculados al proyecto, además se establece un cronograma con las fases de minería de datos.

b) **Comprensión de los datos**
   - **Recopilación de datos:** Se analizan los datos existentes, datos adquiridos, y datos adicionales, identificando atributos prometedores y aquellos que no son relevantes para el modelado.
   - **Descripción de los datos:** Se examina la cantidad de datos y tipos de datos, así como los esquemas de codificación.
   - **Exploración de datos:** Se realiza un análisis exploratorio de los datos mediante tablas y gráficos enfocados en la obtención de estadísticos y en la formulación de hipótesis.
   - **Verificación de la calidad de los datos:** Se evalúa la calidad de los datos incluyendo la identificación de datos faltantes, errores en datos, metadatos, mediciones y codificación.

c) **Preparación de los datos:** Es una fase crucial y la que requiere de mayor tiempo.
   - **Selección de datos:** Se eligen los atributos y elementos relevantes para dar cumplimiento a los objetivos de minería de datos.

- **Limpieza de datos:** Este proceso implica dar solución a los datos faltantes, errores en datos e incoherencias de codificación.
- **Construcción de nuevos datos:** En función de la necesidad se pueden derivar atributos o generar nuevos registros, considerando los requisitos del modelado.
- **Integración de datos:** Proceso utilizado cuando se disponen de varias fuentes de datos para el mismo objetivo.
- **Formato de datos:** Se verifican los requerimientos de formato o clasificación en los datos.

**d) Modelado:**

- **Selección de técnicas de modelado:** Se determinan los modelos más adecuados para cumplir con los objetivos, según los tipos de datos disponibles y los requisitos del modelo.
- **Generación de un diseño de comprobación:** Se define el proceso para evaluar los resultados del modelo, centrándose en los criterios de bondad.
- **Generación de los modelos:** Se experimenta con varios modelos, considerando la configuración de los parámetros y la descripción del modelo.
- **Evaluación del modelo:** Se seleccionan los modelos más precisos ajustando parámetros o seleccionando diferentes modelos según sea necesario.

**e) Evaluación**

- **Evaluación de resultados:** Se formaliza la evaluación en función del cumplimiento de los criterios del rendimiento comercial.
- **Proceso de revisión:** Se examinan tanto los aciertos como los errores del proceso.
- **Determinación de los pasos siguientes:** Se define la fase de despliegue con el propósito de integrar los resultados del modelo en el proceso organizativo, teniendo en cuenta tanto la precisión como la relevancia de los resultados del modelado.

**f) Despliegue**.

- **Planificación del despliegue:** Se planifican actividades para integrar los modelos o presentar descubrimientos.
- **Panificación del control y del mantenimiento:** Se planifica la evaluación periódica para asegurar la eficacia del modelo y realizar mejoras continuas.

- **Creación de un informe final:** Se documentan los resultados para su comunicación.
- **Revisión final del proyecto:** Con el objetivo de integrar los resultados del modelo en el proceso organizativo, se considera tanto la precisión como la relevancia de los resultados del modelado (IBM, n.d.).

# 4. Análisis de Series de Tiempo

## 4.1.1. Comprensión del Negocio

### *Determinación de los Objetivos Comerciales*

#### *4.1.1.1.1. Contexto*

La predicción de la mortalidad es un aspecto crucial para la planificación y asignación de recursos por parte del Estado, en base al conocimiento anticipado de la cantidad de defunciones, se pueden ejecutar programas o políticas públicas enfocadas en las principales causas de mortalidad, así mismo esto se traducirá en la toma de decisiones estratégicas que permitan abordar las necesidades de la población, por lo cual, desarrollar un modelo de predicción se convierte en una herramienta clave a nivel gubernamental.

#### *4.1.1.1.2. Definición de los Objetivos del Negocio*

Desarrollar un modelo para predecir las principales causas de muerte en el Ecuador, con la finalidad de proporcionar al Estado herramientas efectivas para la planificación y ejecución de intervenciones focalizadas en el sistema de la salud.

#### *4.1.1.1.3. Criterios de Rendimiento*

Obtener una precisión efectiva en los modelos de series de tiempo para la predicción de las principales causas de muerte en el conjunto de datos de prueba.

Asegurar que el modelo sea interpretable, facilitando la comprensión de las relaciones entre variables.

### *Evaluación de la Situación*

El Instituto Nacional de Estadística y Censos, como ente rector y coordinador del Sistema Estadístico Nacional, anualmente publica el Registro Estadístico de Defunciones Generales, el cual corresponde a las defunciones ocurridas y/o inscritas en el territorio nacional, esta información se obtiene de los registros realizados por los profesionales de la salud en el sistema REVIT Defunciones, de la base de los formularios físicos recolectados, base de la mortalidad materna del Ministerio de Salud Pública y base de defunciones de la Dirección General de Registro Civil, Identificación y Cedulación.

En el presente proyecto no existen restricciones legales con respecto al acceso de la información que se utilizará, la fuente de información es de acceso público y los datos están anonimizados.

Entre los riesgos que podrían presentarse en la ejecución del proyecto se destaca la ausencia de datos en las variables de interés y el hecho de que los mismos no puedan imputarse a través del algún estadístico, adicionalmente la presencia de sesgo en los datos o en el modelo podría afectar la toma de decisiones y a la planificación gubernamental.

***Determinación de los Objetivos de Minería de Datos***

*4.1.1.1.4. Objetivos de los Modelos de Predicción de Series de Tiempo*

- Desarrollar modelos de pronóstico de series de tiempo para predecir las tres principales causas de muerte en el Ecuador.
- Utilizar métricas de evaluación para seleccionar el modelo más efectivo y preciso.

*4.1.1.1.5. Criterios de rendimiento.*

En el marco del presente proyecto, las métricas de evaluación utilizadas para los métodos de pronóstico de series temporales comprenden el MAPE (Mean Absolute Percentage Error), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) y Mean Absolute Error (MAE).

La selección del modelo óptimo para la predicción de la causa de muerte objeto de análisis se basará en la obtención de los valores más bajos en dichas métricas.

***Plan del proyecto***

*Tabla 1. Plan del proyecto de fin de titulación, series de tiempo.*

| Fase | Duración | Recursos | Riesgos |
|---|---|---|---|
| Comprensión del negocio | 1 semana | Responsable del proyecto | |
| Comprensión de los datos | 2 semanas | Acceso a datos históricos sobre las defunciones | |
| Preparación de los datos | 4 semanas | | Excesivos datos faltantes |
| Modelado | 6 semanas | Plataforma para el modelado (Jupyter Notebook) | Dificultad para encontrar el modelo adecuado |
| Evaluación | 2 semanas | | |

Fuente: Autoría Propia.

### 4.1.2. Comprensión de los Datos

***Recopilación de los Datos***

Los datos de las defunciones fueron proporcionados por el Instituto Nacional de Estadística y Censos, en formato SPSS y cargados en la Jupyter Notebook, para su revisión y procesamiento, el periodo comprende desde el año 1990 hasta el año 2021.

Previo al proceso de carga de la base de datos se instala la biblioteca pyreadstat, y se importan las librerías de pandas, numpy, seaborn y matplotlib.pyplot, para los siguientes procesos.

***Descripción de los Datos***

La base de datos de las defunciones generales en el Ecuador consta de 1.982.281 registros y 49 variables: provincia de inscripción, cantón de inscripción, parroquia de inscripción, año de inscripción, mes de inscripción, día de inscripción, fecha de inscripción, nacionalidad del fallecido, código país, sexo del fallecido, año de nacimiento del fallecido, mes de nacimiento del fallecido, día de nacimiento del fallecido, año de defunción, mes de defunción, día de defunción, código edad, edad del fallecido, provincia de residencia, cantón de residencia, parroquia de residencia, estado civil, sabe leer, etnia, lugar de ocurrencia del fallecimiento, provincia de fallecimiento, cantón de fallecimiento, parroquia de fallecimiento, mujer en edad fértil, muerte violenta, lugar de muerte violenta, autopsia, nivel de instrucción, fecha de nacimiento, causa de muerte, certificado por, año base, total, área de fallecimiento, área de residencia, lista corta de causas, causa básica de defunción, causa básica de defunción (3 caracteres), causa básica de defunción (4 caracteres), lista condensada de causas de muerte, lista de tabulación 2 para la mortalidad subcategorías, lista condensada (67 causas) A, lista condensada desagregada (67 causas) B.

Para desarrollar de los modelos de predicción de series de tiempo univariantes, es fundamental incluir el componente temporal, en este contexto, dicho componente se define por el mes y el año en el cual ocurrió el fallecimiento, el día no se considera debido a que algunos registros no cuentan con esta información.

Adicionalmente, se consideran las causas de muerte a partir de la lista corta de causas de muerte, la cual corresponde a una lista que agrupa las principales causas de muerte.

Se lleva a cabo una revisión de los tipos de datos asignados a las variables, y según lo representado en la Figura 2, se observa que la mayoría de las variables son de tipo categórico.

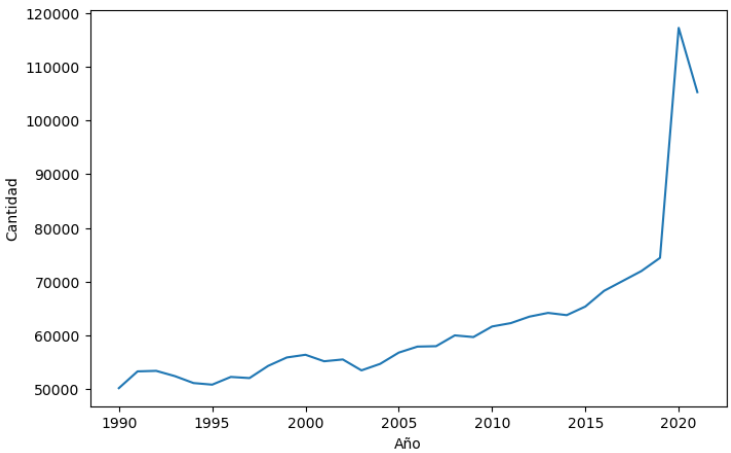
*Figura 2. Tipos de datos base defunciones generales.*

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1982281 entries, 0 to 1982280
Data columns (total 49 columns):
 #   Column      Dtype
---  ------      -----
 0   prov_insc   category
 1   cant_insc   category
 2   parr_insc   category
 3   anio_insc   float64
 4   mes_insc    category
 5   dia_insc    float64
 6   fecha_insc  object
 7   nac_fall    category
 8   cod_pais    category
 9   sexo        category
 10  anio_nac    float64
 11  mes_nac     category
 12  dia_nac     float64
 13  anio_fall   float64
 14  mes_fall    category
 15  dia_fall    float64
 16  fecha_fall  object
 17  cod_edad    category
 18  edad        float64
 19  prov_res    category
 20  cant_res    category
 21  parr_res    category
 22  est_civil   category
 23  sabe_leer   category
 24  etnia       category
 24  etnia       category
 25  lugar_ocur  category
 26  prov_fall   category
 27  cant_fall   category
 28  parr_fall   category
 29  muj_fertil  category
 30  mor_viol    category
 31  lug_viol    category
 32  autopsia    category
 33  niv_inst    category
 34  fecha_nac   object
 35  causa9      object
 36  cer_por     category
 37  anio_base   float64
 38  total       float64
 39  area_fall   category
 40  area_res    category
 41  causa3      category
 42  causa       category
 43  causa103    category
 44  causa80     category
 45  causa67A    category
 46  causa67B    category
 47  causa4      category
 48  lc1         category
dtypes: category(36), float64(9), object(4)
memory usage: 282.1+ MB
```

### *Exploración de los Datos*

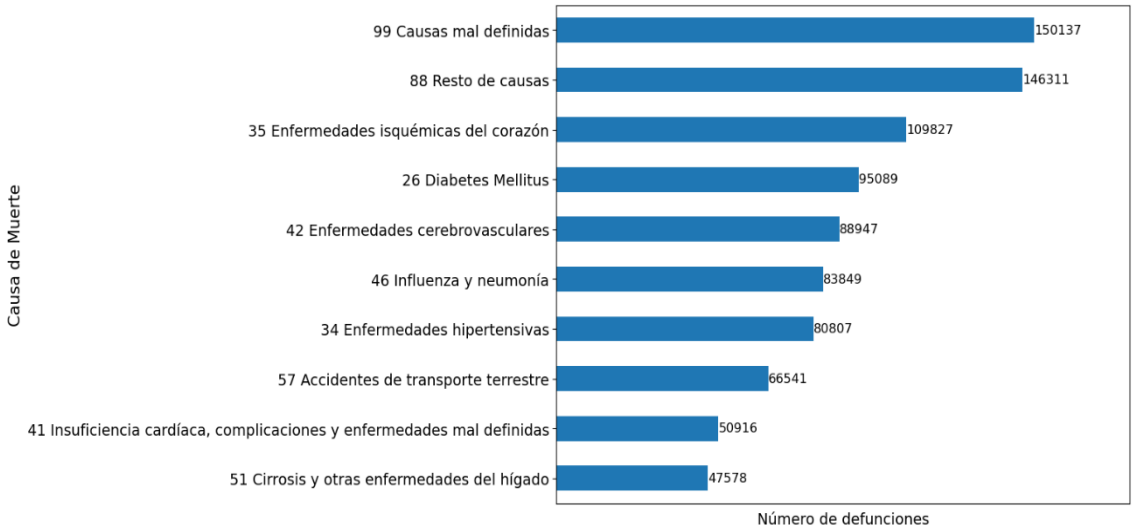
En el gráfico Nro. 1, se aprecia que las defunciones en Ecuador han experimentado un aumento constante desde 1990, llegando a registrar 50.140 fallecimientos, sin embargo, a partir de 2020, debido al impacto del SARS-CoV-2, las defunciones aumentaron de manera significativa, alcanzando un total de 117.200; finalmente, en 2021, esta cifra disminuyó a 105.248 defunciones.

*Gráfico 1. Evolución de las defunciones en el Ecuador, periodo 1990 - 2021*



Según el gráfico Nro. 2 en el Ecuador existe una importante cantidad de defunciones que poseen causas mal definidas, las cuales afectan la calidad de los datos, adicionalmente las causas predominantes de la mortalidad son las enfermedades isquémicas del corazón, diabetes mellitus, enfermedades cerebrovasculares, influenza, neumonía y enfermedades hipertensivas, la principal causa de muerte corresponde a enfermedades isquémicas del corazón y participa con el 5,54% respecto al total de defunciones ocurridas, lo cual evidencia la necesidad de abordar y gestionar adecuadamente el ámbito de la salud en el país.

*Gráfico 2. Principales causas de muerte en el Ecuador.*




### *Verificación de la Calidad de los Datos*

En la figura 3, se verifica que existe una importante cantidad de valores nulos en las variables de interés como, por ejemplo: día de fallecimiento, fecha de fallecimiento y lista corta de causas de muerte.

*Figura 3. Verificar valores nulos.*

| | |
|---|---|
| prov_fall | 0 |
| cant_fall | 0 |
| parr_fall | 0 |
| muj_fertil | 1956221 |
| mor_viol | 1769791 |
| lug_viol | 1896116 |
| autopsia | 1199368 |
| niv_inst | 181739 |
| fecha_nac | 1282136 |
| causa9 | 0 |
| cer_por | 0 |
| anio_base | 0 |
| total | 0 |
| area_fall | 0 |
| area_res | 0 |
| causa3 | 0 |
| causa | 0 |
| causa103 | 0 |
| causa80 | 0 |
| causa67A | 0 |
| causa67B | 0 |
| causa4 | 0 |
| lc1 | 363765 |
| dtype: int64 | |

| | |
|---|---|
| prov_insc | 0 |
| cant_insc | 0 |
| parr_insc | 0 |
| anio_insc | 2867 |
| mes_insc | 2867 |
| dia_insc | 1284446 |
| fecha_insc | 1284862 |
| nac_fall | 1281579 |
| cod_pais | 0 |
| sexo | 0 |
| anio_nac | 0 |
| mes_nac | 0 |
| dia_nac | 1281579 |
| anio_fall | 0 |
| mes_fall | 0 |
| dia_fall | 1281579 |
| fecha_fall | 1281579 |
| cod_edad | 0 |
| edad | 0 |
| prov_res | 0 |
| cant_res | 0 |
| parr_res | 0 |
| est_civil | 199729 |
| sabe_leer | 185659 |
| etnia | 1036048 |
| lugar_ocur | 0 |

Se han identificado causas de muerte codificadas, no se detalla que causa de muerte corresponde, esto se visualiza en la figura 4.

*Figura 4. Causas de muerte codificadas.*

| | |
|---|---|
| COVID-19, virus no identificado | 8303 |
| 50 Apendicitis, hernia y obstrucción intestinal | 8183 |
| 49 Insuficiencia respiratoria | 7931 |
| 61 Accidentes que obstruyen la respiración | 7184 |
| 29 Demencia y enfermedad de Alzheimer | 7180 |
| 23 Neoplasia maligna del encéfalo | 6809 |
| 52 Enfermedades del sistema osteomuscular y tejido conjuntivo | 6331 |
| 32 Epilepsia y estado de mal epiléptico | 5760 |
| 12 Neoplasia maligna de la vesícula biliar y de otras | 5637 |
| 40 Arritmias cardíacas | 5134 |
| 54 Embarazo, parto y puerperio | 4572 |
| 28 Trastornos de los líquidos, electrolitos, y del equilibrio ácido básico | 4569 |
| 30 Trastornos mentales y del comportamiento por uso de sustancias psicoactivas | 4336 |
| 101.0 | 4321 |

Se ha constatado que la base de datos incluye defunciones ocurridas en el extranjero, como se puede observar en la figura 5.

*Figura 5. Provincia de defunción.*

```
Cañar                          32761
Santa Elena                    31505
Bolívar                        29782
Carchi                         22410
Sucumbíos                      13964
Morona Santiago                11454
Napo                            9877
Orellana                        9658
Zamora Chinchipe                7901
Pastaza                         6865
Exterior                        1199
Galápagos                       1047
Name: count, dtype: int64
```

Se verifica a través de la figura 6, que la fecha de defunción posee un tipo de dato inconsistente con el formato de fecha.

*Figura 6. Tipos de datos de las variables de interés.*

```
10  anio_nac     float64
11  mes_nac      category
12  dia_nac      float64
13  anio_fall    float64
14  mes_fall     category
15  dia_fall     float64
16  fecha_fall   object
```

### 4.1.3. Preparación de los Datos

***Seleccionar los Datos***

Para el desarrollo de los modelos de predicción de series de tiempo se requieren las siguientes variables:

- **Año de defunción (anio_fall):** Corresponde al año en el cual ocurrió la defunción.
- **Mes de defunción (mes_fall):** Corresponde al mes en el cual ocurrió la defunción.
- **Lista corta de causas de defunción (lc1):** Corresponde a la lista condensada de las principales causas de muerte.

***Limpieza de los Datos***

Se procede a la eliminación de los registros correspondientes a defunciones ocurridas fuera del Ecuador, ya que el análisis se centra únicamente en datos a nivel nacional, esto se refleja en la figura 7.

*Figura 7. Ejecución de código para eliminar defunciones ocurridas en el exterior.*

```
valor_a_eliminar = "Exterior"
df2 = df_dropped.drop(df_dropped[df_dropped['prov_fall'] == valor_a_eliminar].index)
df2['prov_fall'].value_counts(dropna=False)
```

En la figura 8, se muestra el proceso de eliminación de los registros relacionados con defunciones resultantes de muertes violentas, dado que el objetivo del análisis se centra en el comportamiento típico para predecir la mortalidad.

*Figura 8. Ejecución de código para eliminar las muertes violentas*

```
df3 = df2[df2['mor_viol'].isna()]
df3['mor_viol'].value_counts(dropna=False)
```

Previamente se verificó que aún existen causas de muerte vinculadas a muertes violentas, las cuales no han sido catalogadas en la variable de muertes violentas, de modo que, se procede a eliminarlas, esto se detalla en la figura 9.

*Figura 9. Ejecución de código para eliminar por segunda ocasión las muertes violentas*

```
df3 = df3[~df3['lc1'].str.contains(
    '62 Envenenamiento accidental|65 Eventos de intención no determinada|63 Lesiones autoinflingidas intencionalmente \(Suicidio'
    '|61 Accidentes que obstruyen la respiración|57 Accidentes de transporte terrestre|60 Ahogamiento y sumersión accidentales'
    '|58 Caídas accidentales|64 Agresiones \(Homicidios\)|59 Disparo de arma de fuego no intencional', case=False)]
```

Se han identificado códigos en la variable lc1 que no tienen la categoría correspondiente, por lo tanto, es necesario asignarla, este proceso se muestra en la figura 10.

*Figura 10. Ejecución de código para asignar la categoría que corresponde.*

```
df4['lc1'] = df4['lc1'].astype(str)
codigos_a_reemplazar1 = ["100.0"]
df4['lc1'] =df4['lc1'].replace(codigos_a_reemplazar1, "COVID-19, virus identificado")
codigos_a_reemplazar2 = ["101.0"]
df4['lc1'] =df4['lc1'].replace(codigos_a_reemplazar2, "COVID-19, virus no identificado")
```

Se eliminan los registros que tienen como causa de muerte COVID-19, debido a que corresponde a un periodo atípico, adicionalmente se eliminan las causas especificadas como mal definidas y las detalladas como resto de causas, esto se puede observar en la figura 11.

*Figura 11. Ejecución de código para eliminar causas de muertes atípicas y mal definidas.*

```
df3 = df3[~df3['lc1'].str.contains
          ('COVID-19, virus identificado|COVID-19, virus no identificado|99 Causas mal definidas|88 Resto de causas',
           case=False)]
df3.info()
```

El la figura 12, se visualiza el reemplazo de los valores vacíos de la columna “1c1” (lista corta de las causas de muerte) por N/A.

*Figura 12. Ejecución de código para reemplazar valores vacíos.*

```
def reemplazar_vacios_con_nan(valor):
    if isinstance(valor, str) and valor.strip() == "":
        return np.nan
    return valor

df4 = df3.copy()
df4['lc1'] = df4['lc1'].apply(reemplazar_vacios_con_nan)
df4.head()
```

En la figura 13, se proceden a eliminar todos los registros que tienen valores vacíos en la columna “c1” (lista corta de las causas de muerte), con la finalidad de no generar sesgos en los resultados de las predicciones, tomando en cuenta que existe una importante cantidad de valores vacíos.

*Figura 13. Ejecución de código para eliminar valores vacíos de lc1.*

```
df4 = df4.dropna(subset=['lc1'])
df4['lc1'].info()
```

***Construir los Datos***

En la figura 14, se proceden a identificar cuáles son las principales causas de muerte en el país, determinadas por la variable 'lc1', que hace referencia a la lista corta de causas.

*Figura 14. Principales causas de muerte en el Ecuador.*

```
lc1
35 Enfermedades isquémicas del corazón                                    109781
26 Diabetes Mellitus                                                       95078
42 Enfermedades cerebrovasculares                                          88920
46 Influenza y neumonía                                                    83837
34 Enfermedades hipertensivas                                              80800
41 Insuficiencia cardíaca, complicaciones y enfermedades mal definidas     50903
51 Cirrosis y otras enfermedades del hígado                                47571
53 Enfermedades del sistema urinario                                       45341
55 Ciertas afecciones originadas en el período prenatal                    43442
9 Neoplasia maligna del estómago                                           39478
47 Enfermedades crónicas de las vías respiratorias inferiores              31522
24 Neoplasia maligna del tejido linfático, hematopoyético y afinies        23351
20 Neoplasia maligna de la próstata                                        19452
56 Malformaciones congénitas, deformidades y anomalías cromosómicas        18529
18 Neoplasia maligna del útero                                             18184
27 Desnutrición y anemias nutricionales                                    17895
6 Septicemia                                                               17746
2 Tuberculosis                                                             16896
11 Neoplasia maligna del hígado y de las vías biliares                     16108
48 Edema pulmonar y otras enfermedades respiratorias que afectan al intersticio  15983
15 Neoplasia maligna de la tráquea, bronquios y pulmón                     15946
7 Enfermedad por virus de la inmunodeficiencia (VIH)                       14773
10 Neoplasia maligna del colon, sigmoide, recto y ano                      14072
```

Se aplica el teorema de Pareto para determinar las causas de muerte más críticas, se calcula el 80%, esto se detalla en la figura 15.

*Figura 15. Ejecución de código para el teorema de Pareto.*

```
total_causas = len(conteo_causas)
ochenta_por_ciento = int(0.8 * total_causas)
ochenta_por_ciento
```

```
44
```

44 causas de muerte son demasiadas para realizar el presente análisis, por lo cual nos enfocaremos únicamente en las tres principales que son: Enfermedades isquémicas del corazón, diabetes mellitus y enfermedades cerebrovasculares.

En la figura 16, se muestra el proceso de creación de la variable temporal, para el efecto se cambia el tipo de dato a entero, se genera una nueva variable que corresponde al día de la defunción, con la finalidad de contar con una fecha tipo d/m/a, y finalmente se crea la variable temporal que corresponde al año, mes y día de defunción.

*Figura 16. Ejecución de código para crear la fecha de defunción.*

```
bdefunciones['anio_fall'] = bdefunciones['anio_fall'].astype(int)
bdefunciones['mes_fall'] = bdefunciones['mes_fall'].astype(int)
bdefunciones['dia_def'] = 1
bdefunciones['dia_def'] = bdefunciones['dia_def'].astype(int)
bdefunciones['anio_mes_fall'] = bdefunciones['anio_fall'].astype(str) + \
                                '-' +bdefunciones['mes_fall'].astype(str)+ \
                                '-' +bdefunciones['dia_def'].astype(str)
bdefunciones['anio_mes_fall'] = pd.to_datetime(bdefunciones['anio_mes_fall'])
```

En la figura 17, se crea un data frame con las variables de interés y posterior para las tres principales causas de muerte.

*Figura 17. Ejecución de código para filtrar las tres principales causas de muerte.*

```
causas_muerte = bdefunciones[['anio_mes_fall', 'lc1']]
df_enfer_corazon = causas_muerte [causas_muerte['lc1'].str.contains('35 Enfermedades isquémicas del corazón', case=False)]
df_diabetes_org = causas_muerte [causas_muerte['lc1'].str.contains('26 Diabetes Mellitus', case=False)]
df_enfer_cerebrovasc_org = causas_muerte [causas_muerte['lc1'].str.contains('42 Enfermedades cerebrovasculares', case=False)]
```

En la figura 18, se muestra el proceso para ordenar los datos en función de la variable fecha de defunción, la creación de las variables dummies, agrupación de los datos por fecha y la operación de agregación de las enfermedades.

*Figura 18. Ejecución de código para agrupar la cantidad de defunciones por enfermedad.*

```
df_enfer_corazon = df_enfer_corazon.copy()
df_enfer_corazon.sort_values(by='anio_mes_fall', inplace=True)
dummies = pd.get_dummies(df_enfer_corazon['lc1'], prefix='lc1')
df_enfer_corazon = pd.concat([df_enfer_corazon, dummies], axis=1)
df_enfer_corazon = df_enfer_corazon.drop('lc1', axis=1)
grupo_por_fecha = df_enfer_corazon.groupby('anio_mes_fall')
df_3c = grupo_por_fecha.sum(numeric_only=False)
df_3c.head()
```

| anio_mes_fall | lc1_35 Enfermedades isquémicas del corazón |
|---|---|
| 1997-01-01 | 184 |
| 1997-02-01 | 160 |
| 1997-03-01 | 148 |
| 1997-04-01 | 144 |
| 1997-05-01 | 167 |

***Formateo de Datos***

Para la generación de la variable fecha de defunción, se requiere modificar el mes de defunción a número, debido a que consta como texto, esto se detalla en la figura 19.

*Figura 19. Ejecución de código para remplazar el mes de defunción*

```
codigos_a_reemplazar = ["Enero"]
bdefunciones['mes_fall'] =bdefunciones['mes_fall'].replace(codigos_a_reemplazar, "01")
codigos_a_reemplazar1 = ["Febrero"]
bdefunciones['mes_fall'] =bdefunciones['mes_fall'].replace(codigos_a_reemplazar1, "02")
codigos_a_reemplazar2 = ["Marzo"]
bdefunciones['mes_fall'] =bdefunciones['mes_fall'].replace(codigos_a_reemplazar2, "03")
codigos_a_reemplazar3 = ["Abril"]
bdefunciones['mes_fall'] =bdefunciones['mes_fall'].replace(codigos_a_reemplazar3, "04")
codigos_a_reemplazar4 = ["Mayo"]
bdefunciones['mes_fall'] =bdefunciones['mes_fall'].replace(codigos_a_reemplazar4, "05")
codigos_a_reemplazar5 = ["Junio"]
bdefunciones['mes_fall'] =bdefunciones['mes_fall'].replace(codigos_a_reemplazar5, "06")
codigos_a_reemplazar6 = ["Julio"]
bdefunciones['mes_fall'] =bdefunciones['mes_fall'].replace(codigos_a_reemplazar6, "07")
codigos_a_reemplazar7 = ["Agosto"]
bdefunciones['mes_fall'] =bdefunciones['mes_fall'].replace(codigos_a_reemplazar7, "08")
codigos_a_reemplazar8 = ["Septiembre"]
bdefunciones['mes_fall'] =bdefunciones['mes_fall'].replace(codigos_a_reemplazar8, "09")
codigos_a_reemplazar9 = ["Octubre"]
bdefunciones['mes_fall'] =bdefunciones['mes_fall'].replace(codigos_a_reemplazar9, "10")
codigos_a_reemplazar10 = ["Noviembre"]
bdefunciones['mes_fall'] =bdefunciones['mes_fall'].replace(codigos_a_reemplazar10, "11")
codigos_a_reemplazar11 = ["Diciembre"]
bdefunciones['mes_fall'] =bdefunciones['mes_fall'].replace(codigos_a_reemplazar11, "12")
```

En la figura 20, se convierte a la variable fecha de defunción a tipo fecha.

*Figura 20. Ejecución de código para modificar la fecha de defunción.*

```
bdefunciones['anio_mes_fall'] = pd.to_datetime(bdefunciones['anio_mes_fall'])
```

### 4.1.4. Modelado

Para la construcción de los modelos de series de tiempo univariados existen diversos métodos y técnicas, entre las cuales tenemos: métodos simples de pronóstico de series de tiempo, técnicas de suavizado exponencial y métodos autorregresivos.

***Selección de Técnicas de Modelado***

Considerando las ventajas de cada una de las técnicas y métodos de series de tiempo, se han seleccionado cuatro, de los métodos de suavizamiento exponencial el método Holt con tendencia y el método multiplicativo de Holt Winters con estacionalidad y tendencia, y de los métodos autorregresivos se han seleccionado al ARIMA y SARIMA.

Para la ejecución de los métodos seleccionados se dispone de suficientes datos temporales que permitan capturar el comportamiento de las defunciones en el tiempo, los datos se dividen en entrenamiento y evaluación para la validación y estimación del rendimiento de los modelos.

*4.1.4.1.1. Modelado de Supuestos*

Se asume que los métodos de suavizamiento exponencial, como Holt con tendencia y Holt Winter con estacionalidad y tendencia, son adecuados para capturar patrones y tendencias en los datos de series temporales.

Los métodos autorregresivos, específicamente ARIMA y SARIMA, son robustos frente a datos atípicos y permite captar patrones en los datos.

Los métodos ARIMA y SARIMA requieren de series estacionarias, previo al pronóstico de la serie de tiempo.

***Diseño de Comprobación***

Se divide el conjunto de datos en dos partes: entrenamiento y evaluación.

Se experimenta con varios modelos, incluyendo ARIMA, SARIMA, Holt con tendencia y Holt-Winters multiplicativo.

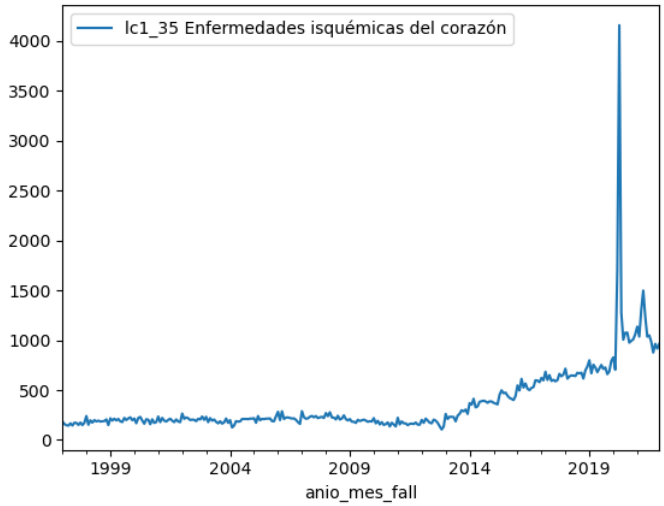
Se evalúa el rendimiento de cada modelo utilizando métricas específicas como el Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) y MAPE (Mean Absolute Percentage Error).

***Generación de los Modelos***

*4.1.4.1.2. Causa de Muerte Enfermedades Isquémicas del Corazón*

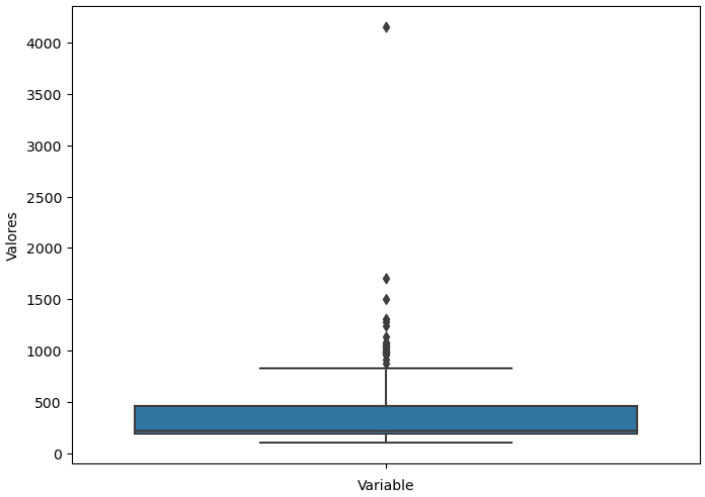
En el gráfico 3, se visualiza la cantidad de muertes por enfermedades isquémicas al corazón, destacando valores atípicos que superan el promedio normal.

*Gráfico 3. Evolución de la causa de muerte de enfermedades isquémicas del corazón.*




En el gráfico 4, se genera un diagrama de caja para visualizar la distribución de los datos, confirmando la presencia de valores atípicos.

*Gráfico 4. Boxplot causa de muerte enfermedades isquémicas del corazón.*




En los gráficos generados, se evidencian múltiples valores atípicos, los cuales requieren de tratamiento, con tal propósito, se emplea la técnica del rango intercuartílico, la cual consiste en la identificación y posterior tratamiento de los datos atípicos a través de la distancia intercuartílica definida por el espacio entre el primer cuartil (Q1) y el tercer cuartil (Q3) de la distribución de los datos, se elige el umbral común que es 1.5, para el efecto se genera la siguiente función, detallada en la figura 21.

*Figura 21. Ejecución de código para creación de la función para eliminar valores atípicos.*

```
def eliminar_outliers(df, variable):
    Q1 = df[variable].quantile(0.25)
    Q3 = df[variable].quantile(0.75)
    IQR = Q3 - Q1
    umbral = 1.5
    sin_outliers = df[(df[variable] >= Q1 - umbral * IQR) & (df[variable] <= Q3 + umbral * IQR)]
    return sin_outliers
```

Posteriormente se eliminan los datos atípicos, a partir de la función eliminar outliers, esto se visualiza en la figura 22.

*Figura 22. Ejecución de código para eliminación de valores atípicos.*

```
df_sin_outliers = eliminar_outliers(df_3c, 'lc1_35 Enfermedades isquémicas del corazón')
df_sin_outliers.head()
```

*Gráfico 5. Boxplot de causa de muerte enfermedades isquémicas al corazón posterior al tratamiento de valores atípicos.*

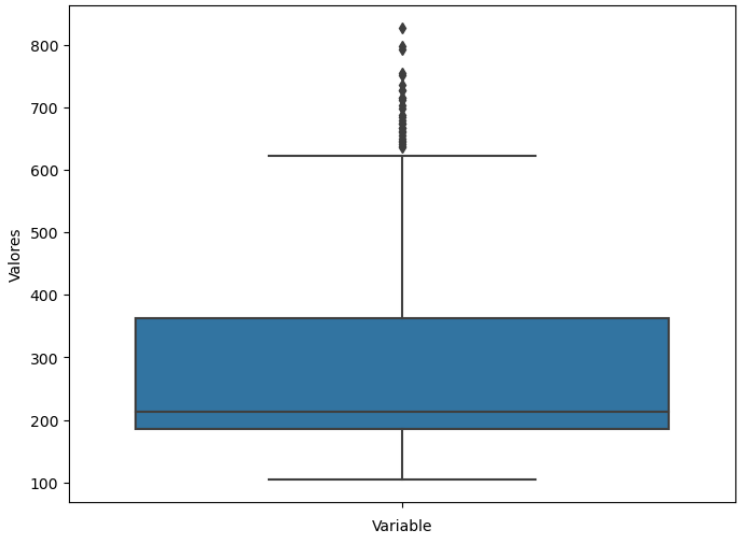
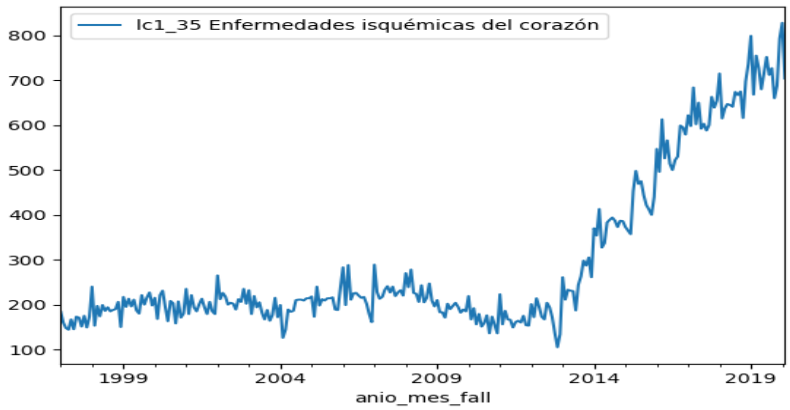



*Gráfico 6. Evolución de la causa de muerte de enfermedades isquémicas del corazón posterior al tratamiento de valores atípicos.*



En los gráficos 5 y 6, se verifica la eliminación de determinados valores atípicos, esta acción busca optimizar la capacidad de los modelos de predicción para entrenar y evaluar los datos a través de un comportamiento más consistente con la distribución normal, excluir valores atípicos permite mitigar potenciales distorsiones en la capacidad predictiva de los modelos.

Con la finalidad de observar la tendencia, estacionalidad y residuos de la serie temporal, se realiza la descomposición multiplicativa y aditiva, para el efecto se importan las librerías detalladas en la figura 2

*Figura 23. Ejecución de código para importan librerías para descomposición.*

```
from statsmodels.tsa.seasonal import seasonal_decompose
from dateutil.parser import parse
```

*Gráfico 7. Descomposición multiplicativa, causa de muerte enfermedades isquémicas del corazón.*

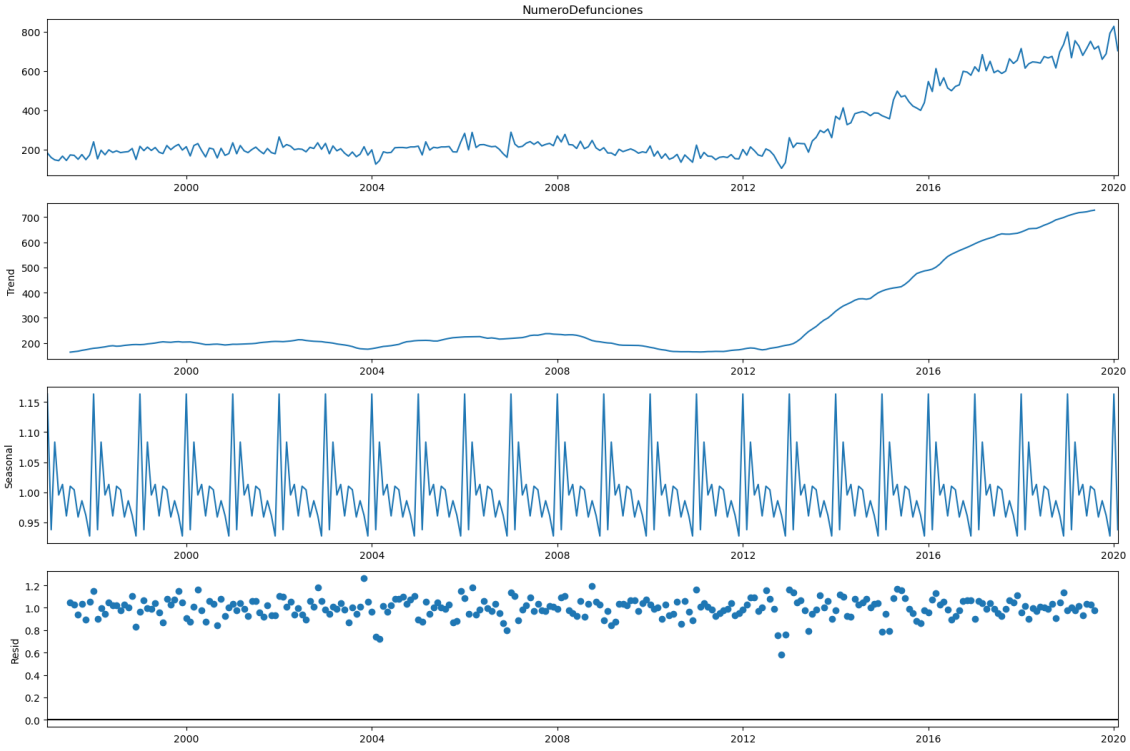
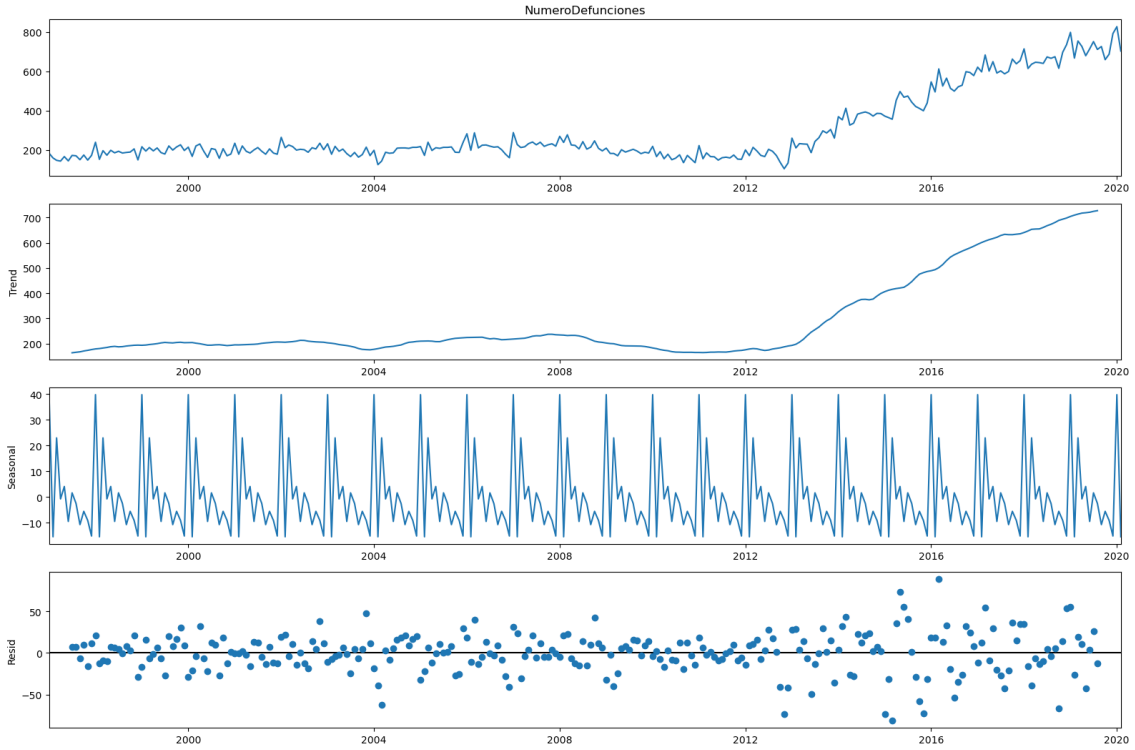


*Gráfico 8. Descomposición aditiva, causa de muerte enfermedades isquémicas del corazón.*




En los gráficos 7 y 8 se realiza la descomposición multiplicativa y aditiva, técnicas utilizadas para separar la serie temporal en componentes clave como la tendencia, estacionalidad y comportamiento de los residuos.

Como se puede observar en los gráficos la tendencia de las defunciones por enfermedades isquémicas del corazón, sugiere un aumento constante a lo largo del tiempo. Adicionalmente se observan picos en ciertos meses del año, de modo que, existe un patrón estacional en los datos.

Para la aplicación de los modelos se divide la base de datos en 80% para entrenamiento y 20% para evaluación, el proceso se detalla en la figura 24.

*Figura 24. Ejecución de código para dividir la data en entrenamiento y evaluación.*

```
total_len = len(df2_enfer_corazon)
train_len = round(total_len*0.8)
train_len
train = df2_enfer_corazon[0 : train_len]
test = df2_enfer_corazon[train_len : ]
```

### a. Método Suavizado Exponencial de Holt con Tendencia

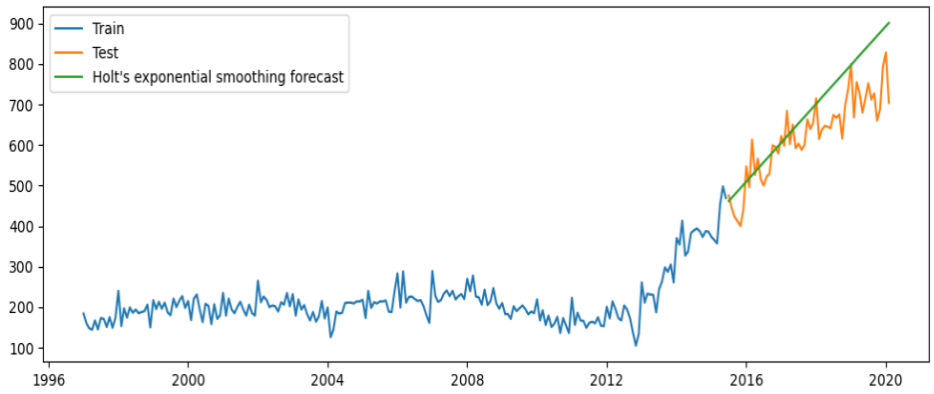
Para la aplicación del método de suavizado exponencial de Holt con tendencia, se especifican los parámetros del modelo, incluyendo la cantidad de períodos estacionales (12 meses), el tipo de tendencia (aditiva), y la ausencia de estacionalidad en los datos,

debido a que este método captura de manera principal la tendencia, este proceso se visualiza en la figura 25 y las predicciones se visualizan en el gráfico 9.

*Figura 25. Ejecución de código para el método de suavizado exponencial de Holt con tendencia.*

```
from statsmodels.tsa.holtwinters import ExponentialSmoothing
model = ExponentialSmoothing(np.asarray(train['NumeroDefunciones']) ,seasonal_periods=12 ,trend='additive', seasonal=None)
model_fit = model.fit(optimized=True)
print(model_fit.params)
y_hat_holt = test.copy()
y_hat_holt['holt_forecast'] = model_fit.forecast(len(test))
```

*Gráfico 9. Pronóstico de suavizamiento exponencial de Holt con tendencia.*



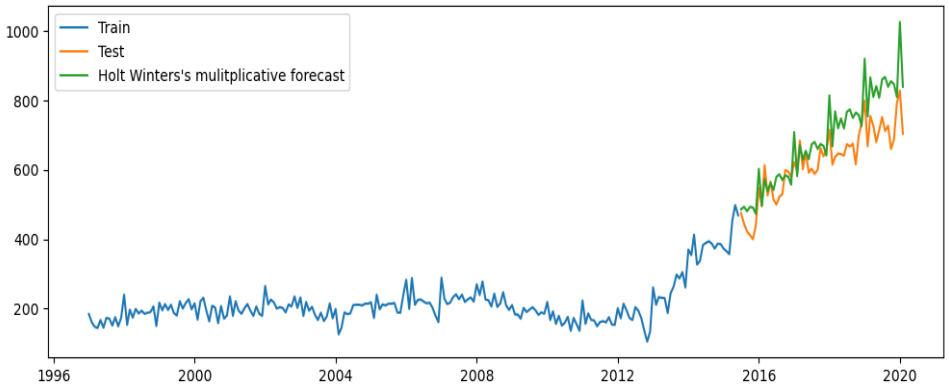

### b. Método Multiplicativo de Holt Winters con Tendencia y Estacionalidad

Para el método multiplicativo de Holt Winters con tendencia y estacionalidad, se especifican los parámetros del modelo, incluyendo la cantidad de períodos estacionales (12), el tipo de tendencia (aditiva), y la estacionalidad multiplicativa, este proceso se detalla en la figura 26 y las predicciones se visualizan en el gráfico 10.

*Figura 26. Ejecución de código para el método Holt Winter con tendencia y estacionalidad.*

```
y_hat_hwm = test.copy()
model = ExponentialSmoothing(np.asarray(train['NumeroDefunciones']) ,seasonal_periods=12 ,trend='add', seasonal='mul')
model_fit = model.fit(optimized=True)
print(model_fit.params)
y_hat_hwm['hw_forecast'] = model_fit.forecast(len(test))
```

*Gráfico 10. Pronóstico multiplicativo de Holt Winters.*



### c. Método ARIMA

Para trabajar con métodos autorregresivos como ARIMA o SARIMA, se debe verificar que la serie sea estacionaria, debido que una de las propiedades de este tipo de métodos es la suposición de estacionariedad en la serie de tiempo.

Para comprobar si la serie es estacionaria se utiliza la prueba de Dickey-Fuller, para el efecto se crea función adfuller_test, detallada en la figura 27.

*Figura 27. Ejecución de código para crear la función para la prueba de Dickey-Fuller.*

```
from statsmodels.tsa.stattools import adfuller
#Se valida si la series estacionaria o no
#Ho: no es estacionaria
#H1: es estacionaria

def adfuller_test(NumeroDefunciones):
    result=adfuller(NumeroDefunciones)
    labels = ['ADF Test Statistic','p-value','#Lags Used','Number of Observations Used']
    for value,label in zip(result,labels):
        print(label+' : '+str(value) )
    if result[1] <= 0.05:
        print("Evidencia fuerte contra la hipótesis nula (Ho), rechace la hipótesis nula. Los datos no tienen raíz unitaria y
    else:
        print("evidencia débil contra la hipótesis nula, la serie de tiempo tiene una raíz unitaria, lo que indica que no es
```

Se ejecuta la prueba, y los resultados se muestran en la figura 28.

*Figura 28. Ejecución de código para ejecutar la prueba de Dickey-Fuller.*

```
adfuller_test(df2_enfer_corazon['NumeroDefunciones'])
```

```
ADF Test Statistic : 0.8398808493692832
p-value : 0.9922533939140432
#Lags Used : 14
Number of Observations Used : 263
evidencia débil contra la hipótesis nula, la serie de tiempo tiene una raíz unitaria, lo que indica que no es estacionaria
```

Debido a que la serie no es estacionaria se deben aplicar técnicas para convertir la serie en estacionaria previo a la aplicación de los métodos de ARIMA y SARIMA.

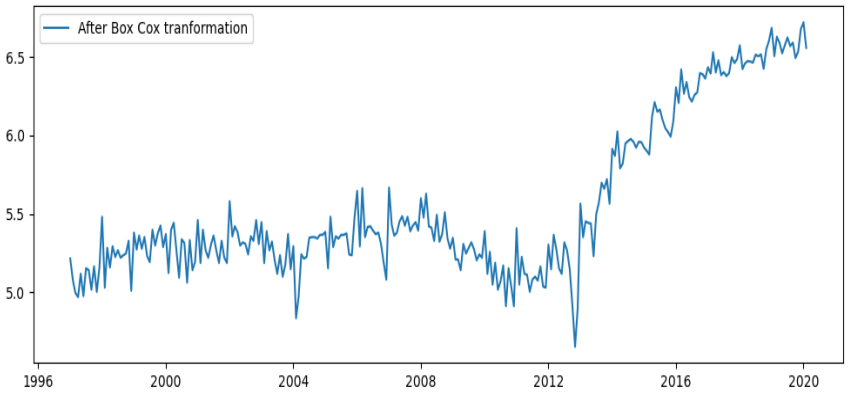
Es importante considerar que existen diversos métodos para convertir una serie en estacionaria, en este caso se aplicó la transformación de Box Cox, la cual se utiliza para estabilizar la varianza y hacer que los datos se aproximen más a una distribución normal, adicionalmente para determinar el método de transformación también se probó con diferenciación (2) y transformación logarítmica (1 diferenciación), pero los valores de la predicción se vuelven demasiado pequeños, lo cual afecta al resultado de las métricas de evaluación en este caso MSE, RMSE, MAE y MAPE.

En la figura 29, se importa la función boxcox y posterior se ejecuta el proceso de transformación en la serie de tiempo, la serie temporal transformada se visualiza en el gráfico 11.

*Figura 29. Ejecución de código para la transformación Box Cox.*

```
from scipy.stats import boxcox
data_boxcox = pd.Series(boxcox(df2_enfer_corazon['NumeroDefunciones'], lmbda=0), index = df2_enfer_corazon.index)
```

*Gráfico 11. Transformación Box Cox.*

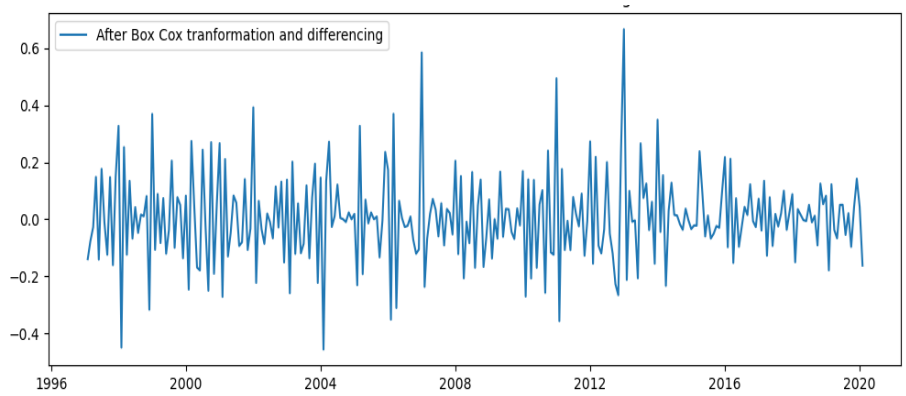



En la figura 30 se ejecuta el proceso de diferenciación de la serie temporal y en el gráfico 12 se visualiza la serie temporal diferenciada.

*Figura 30. Ejecución de código para la diferenciación.*

```
data_boxcox_diff = pd.Series(data_boxcox - data_boxcox.shift(), df2_enfer_corazon.index)
```

*Gráfico 12. Transformación Box Cox y diferenciación.*




Una vez aplicada la transformación y diferenciación, se aplica nuevamente la prueba de estacionariedad para verificar si la serie es estacionaria o no, los resultados se muestran en la figura 31.

*Figura 31. Ejecución de código para verificar estacionariedad posterior transformación y diferenciación.*

```
adfuller_test(data_boxcox_diff)
```

```
ADF Test Statistic : -3.732640677713971
p-value : 0.0036767683929460316
#Lags Used : 13
Number of Observations Used : 263
Evidencia fuerte contra la hipótesis nula (Ho), rechace la hipótesis nula. Los datos no tienen raíz unitaria y son estacionario
s.
```

Se verifica que la serie es estacionaria (p-value menor al 5%), de modo que, se pueden aplicar los métodos autorregresivos, para continuar con el proceso se genera la data de entrenamiento y evaluación a partir de la data diferenciada, esto se detalla en la figura 32.

*Figura 32. Ejecución de código para dividir la data transformada en entrenamiento y evaluación.*

```
train_data_boxcox = data_boxcox[:train_len]
test_data_boxcox = data_boxcox[train_len:]
train_data_boxcox_diff = data_boxcox_diff[:train_len-1]
test_data_boxcox_diff = data_boxcox_diff[train_len-1:]

y_pred=test_data_boxcox_diff.copy()
```

Para aplicar el modelo ARIMA, es necesario importar la función auto_arima desde la biblioteca pmdarima. Luego, se procede a realizar una búsqueda automática para encontrar el mejor modelo ARIMA, este proceso se detalla en la figura 33.

*Figura 33. Ejecución de código modelo ARIMA.*

```
model_arima = auto_arima(train_data_boxcox_diff, trace=True, error_action='ignore',
                         start_p=1, start_q=1, max_p=3, max_q=3, suppress_warnings=True,
                         stepwise=False, seasonal=True)
model_arima.fit(train_data_boxcox_diff)
```

```
 ARIMA(0,0,0)(0,0,0)[1] intercept   : AIC=-157.675, Time=0.34 sec
 ARIMA(0,0,1)(0,0,0)[1] intercept   : AIC=-241.130, Time=0.17 sec
 ARIMA(0,0,2)(0,0,0)[1] intercept   : AIC=-239.443, Time=0.25 sec
 ARIMA(0,0,3)(0,0,0)[1] intercept   : AIC=-238.434, Time=0.39 sec
 ARIMA(1,0,0)(0,0,0)[1] intercept   : AIC=-220.118, Time=0.10 sec
 ARIMA(1,0,1)(0,0,0)[1] intercept   : AIC=-239.499, Time=0.21 sec
 ARIMA(1,0,2)(0,0,0)[1] intercept   : AIC=-237.817, Time=0.59 sec
 ARIMA(1,0,3)(0,0,0)[1] intercept   : AIC=-236.468, Time=0.55 sec
 ARIMA(2,0,0)(0,0,0)[1] intercept   : AIC=-229.143, Time=0.09 sec
 ARIMA(2,0,1)(0,0,0)[1] intercept   : AIC=-238.136, Time=0.20 sec
 ARIMA(2,0,2)(0,0,0)[1] intercept   : AIC=-236.178, Time=0.48 sec
 ARIMA(2,0,3)(0,0,0)[1] intercept   : AIC=-237.661, Time=0.52 sec
 ARIMA(3,0,0)(0,0,0)[1] intercept   : AIC=-230.102, Time=0.17 sec
 ARIMA(3,0,1)(0,0,0)[1] intercept   : AIC=-236.329, Time=0.45 sec
 ARIMA(3,0,2)(0,0,0)[1] intercept   : AIC=-234.411, Time=0.67 sec

Best model:  ARIMA(0,0,1)(0,0,0)[1] intercept
Total fit time: 5.221 seconds

ARIMA(order=(0, 0, 1), scoring_args={}, seasonal_order=(0, 0, 0, 1),
      suppress_warnings=True)
```

Se definen los siguientes parámetros:

- trace=True: Permite obtener información detallada durante el proceso de búsqueda de parámetros, lo cual es útil para entender cómo se seleccionan los valores óptimos.
- error_action='ignore': Ignora los errores para evitar interrupciones en el caso de que el proceso de ajuste encuentre problemas con algunos conjuntos de parámetros.
- start_p=1, start_q=1: Inicia la búsqueda de los parámetros p (componente auto regresivo) y q (promedio móvil) desde 1.

- max_p=3, max_q=3: Limita la búsqueda de p y q hasta 3 para evitar modelos demasiado complejos.
- suppress_warnings=True: Suprime las advertencias para mejorar la legibilidad del resultado.
- stepwise=False: Realiza una búsqueda exhaustiva en lugar de una búsqueda paso a paso, evalúa todos los modelos posibles dentro del rango de los hiperparámetros.
- seasonal=True: Considera componentes estacionales en el modelo ARIMA, lo cual es común en datos temporales.

En la figura 34 se obtiene las predicciones.

*Figura 34. Ejecución de código para las predicciones.*

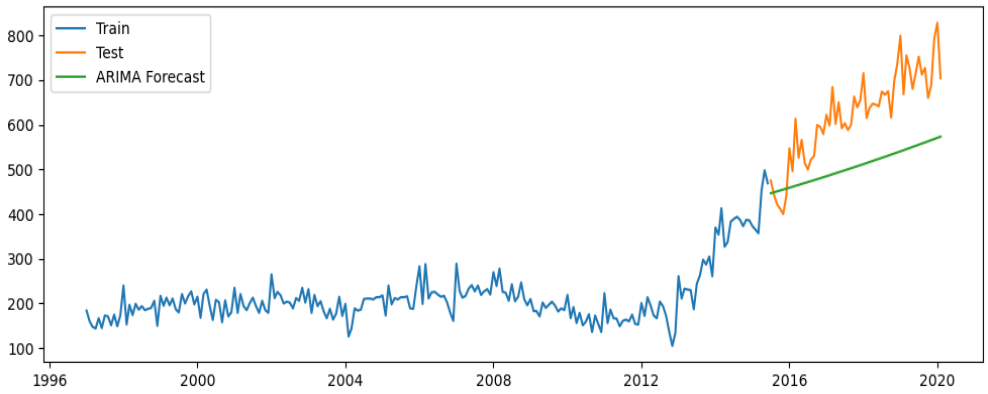
```
prediction_arima=model_arima.predict(len(test_data_boxcox_diff))
y_pred["ARIMA Model Prediction"]=prediction_arima
```

En la figura 35, se invierten la transformación y diferenciación para obtener las predicciones en la escala original y en el gráfico 13 se visualizan las predicciones del método ARIMA.

*Figura 35. Ejecución de código para las invertir la transformación y diferenciación.*

```
y_hat_arima = data_boxcox_diff.copy()
y_hat_arima['arima_forecast_boxcox_diff'] = prediction_arima
y_hat_arima['arima_forecast_boxcox'] = y_hat_arima['arima_forecast_boxcox_diff'].cumsum()
y_hat_arima['arima_forecast_boxcox'] = y_hat_arima['arima_forecast_boxcox'].add(data_boxcox[train_len-1])
y_hat_arima['arima_forecast'] = np.exp(y_hat_arima['arima_forecast_boxcox'])
```

*Gráfico 13. Pronóstico ARIMA*




### d. Método SARIMA

En la figura 36, se realiza la búsqueda automática para encontrar el mejor modelo SARIMA.

*Figura 36. Ejecución de código para el modelo SARIMA.*

```
model_sarima= auto_arima(train_data_boxcox_diff,trace=True, error_action='ignore',
                         start_p=0,d=0,start_q=0, max_p=3,max_d=2,max_q=3, m=12,
                         start_P=0, D=0, start_Q=0, max_P=3,max_Q=3, max_D=2,
                         suppress_warnings=True,stepwise=False,seasonal=True)
model_sarima.fit(train_data_boxcox_diff)
```

```
 ARIMA(2,0,2)(0,0,1)[12] intercept   : AIC=-255.135, Time=1.08 sec
 ARIMA(2,0,2)(1,0,0)[12] intercept   : AIC=-261.403, Time=0.80 sec
 ARIMA(2,0,3)(0,0,0)[12] intercept   : AIC=-237.661, Time=0.52 sec
 ARIMA(3,0,0)(0,0,0)[12] intercept   : AIC=-230.102, Time=0.16 sec
 ARIMA(3,0,0)(0,0,1)[12] intercept   : AIC=-250.539, Time=0.35 sec
 ARIMA(3,0,0)(0,0,2)[12] intercept   : AIC=-251.944, Time=0.83 sec
 ARIMA(3,0,0)(1,0,0)[12] intercept   : AIC=-256.451, Time=0.49 sec
 ARIMA(3,0,0)(1,0,1)[12] intercept   : AIC=inf, Time=0.86 sec
 ARIMA(3,0,0)(2,0,0)[12] intercept   : AIC=-257.327, Time=1.15 sec
 ARIMA(3,0,1)(0,0,0)[12] intercept   : AIC=-236.329, Time=0.29 sec
 ARIMA(3,0,1)(0,0,1)[12] intercept   : AIC=-255.199, Time=0.93 sec
 ARIMA(3,0,1)(1,0,0)[12] intercept   : AIC=-261.581, Time=0.94 sec
 ARIMA(3,0,2)(0,0,0)[12] intercept   : AIC=-234.411, Time=0.51 sec

Best model:  ARIMA(0,0,2)(1,0,2)[12] intercept
Total fit time: 118.930 seconds

ARIMA(order=(0, 0, 2), scoring_args={}, seasonal_order=(1, 0, 2, 12),
      suppress_warnings=True)
```

- trace=True: Muestra información detallada durante el proceso de búsqueda del modelo. Esto incluye los modelos que se están evaluando y las métricas de ajuste asociadas.
- error_action=ignore: Especifica que, si ocurre algún error durante el ajuste del modelo, se debe ignorar y el proceso continuará.
- start_p=0, d=0, start_q=0, max_p=3, max_d=2, max_q=3: Estos parámetros definen los rangos de búsqueda para los componentes no estacionales (p, d, q) del modelo SARIMA.
- m=12: Indica una estacionalidad mensual.
- start_P=0, D=0, start_Q=0, max_P=3, max_Q=3, max_D=2: Definen los componentes estacionales (P, D, Q) del modelo SARIMA.
- suppress_warnings=True: Suprime los mensajes de advertencia durante el proceso de ajuste del modelo.
- stepwise=False: Indica que se realizará una búsqueda exhaustiva en lugar de una búsqueda secuencial para encontrar el mejor modelo SARIMA.
- seasonal=True: Indica que se ajustará un modelo SARIMA estacional.

En la figura 37, se realizan las predicciones del método SARIMA.

*Figura 37. Ejecución de código para las predicciones del modelo SARIMA*

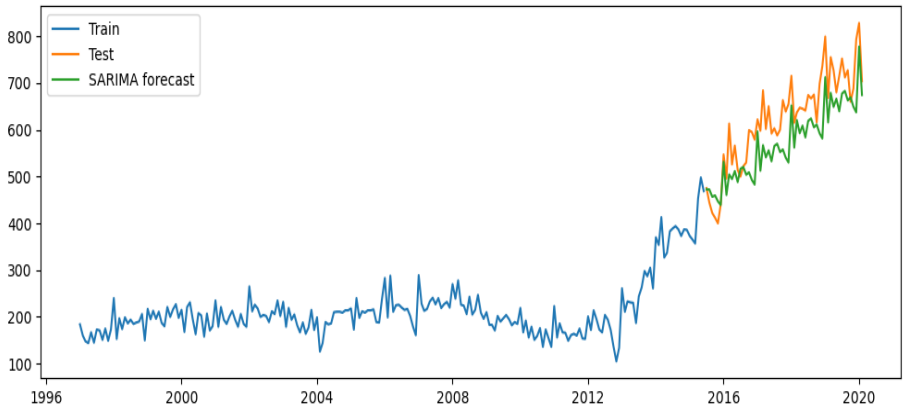
```
prediction_sarima=model_sarima.predict(len(test_data_boxcox_diff))
y_pred["SARIMA Model Prediction"]=prediction_sarima
```

En la figura 38, se invierte la transformación y la diferenciación, para obtener las predicciones en la escala original y en el grafico 14 se visualizan las predicciones.

*Figura 38. Ejecución de código para invertir las transformaciones, modelo SARIMA.*

```
y_hat_sarima = data_boxcox_diff.copy()
y_hat_sarima['sarima_forecast_boxcox_diff'] = prediction_sarima
y_hat_sarima['sarima_forecast_boxcox'] = y_hat_sarima['sarima_forecast_boxcox_diff'].cumsum()
y_hat_sarima['sarima_forecast_boxcox'] = y_hat_sarima['sarima_forecast_boxcox'].add(data_boxcox[train_len-1])
y_hat_sarima['sarima_forecast'] = np.exp(y_hat_sarima['sarima_forecast_boxcox'])
y_pred["SARIMA Model Prediction"] = y_hat_sarima['sarima_forecast']
```

*Gráfico 14. Pronóstico SARIMA*




### *4.1.4.1.3. Causa de Muerte Enfermedades Diabetes Mellitus*

En el gráfico 15, se visualiza la cantidad de defunciones que poseen como causa de muerte diabetes mellitus, donde se verifica la presencia de datos atípicos.

*Gráfico 15. Evolución de la causa de muerte enfermedades diabetes mellitus.*

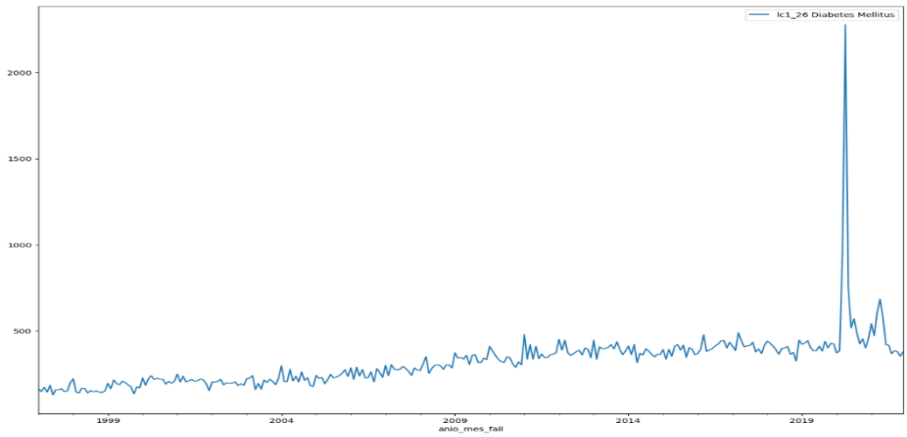
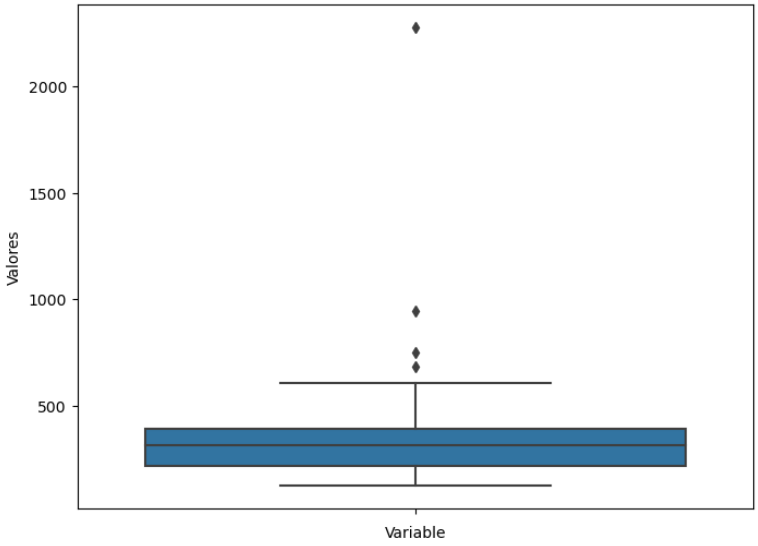


*Gráfico 16. Boxplot causa de muerte enfermedades diabetes mellitus*




Como se puede observar en los gráficos 15 y 16, se evidencian múltiples valores atípicos, los cuales requieren de tratamiento, de esta manera se emplea la técnica del rango intercuartílico para eliminar este tipo de datos, debido a que pueden generar sesgos en los resultados, se utiliza la función definida anteriormente y se eliminan los valores atípicos, esto se detalla en la figura 39.

*Figura 39. Ejecución de código para eliminar valores atípicos causa de muerte diabetes.*

```
df_sin_outliers_d = eliminar_outliers(df_4c, 'lc1_26 Diabetes Mellitus')
df_sin_outliers_d.head()
```

En los gráficos 17 y 18, se verifica el proceso de eliminación de valores atípicos.

*Gráfico 17. Boxplot de causa de muerte diabetes mellitus previo tratamiento de valores atípicos.*

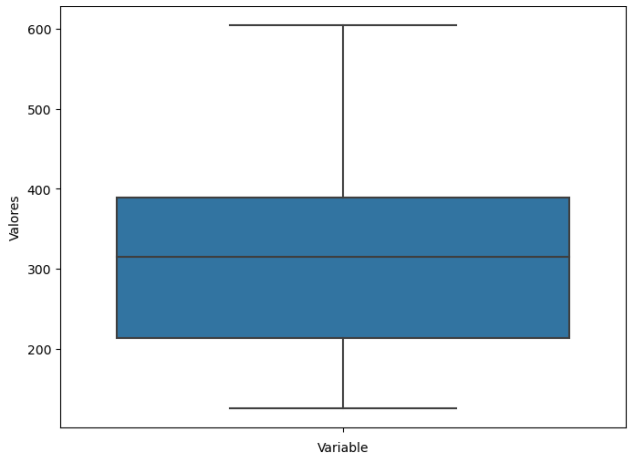
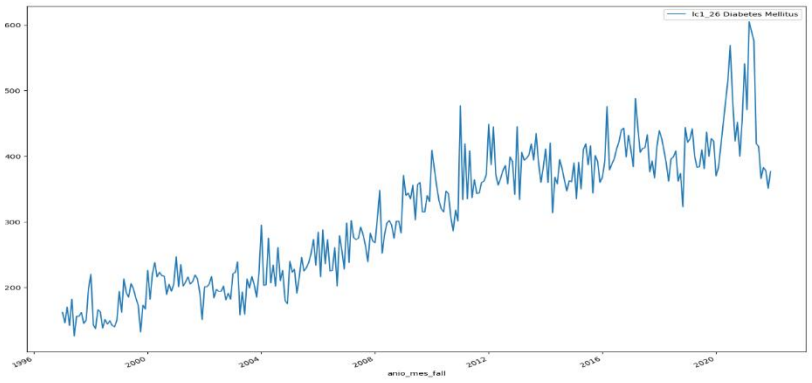


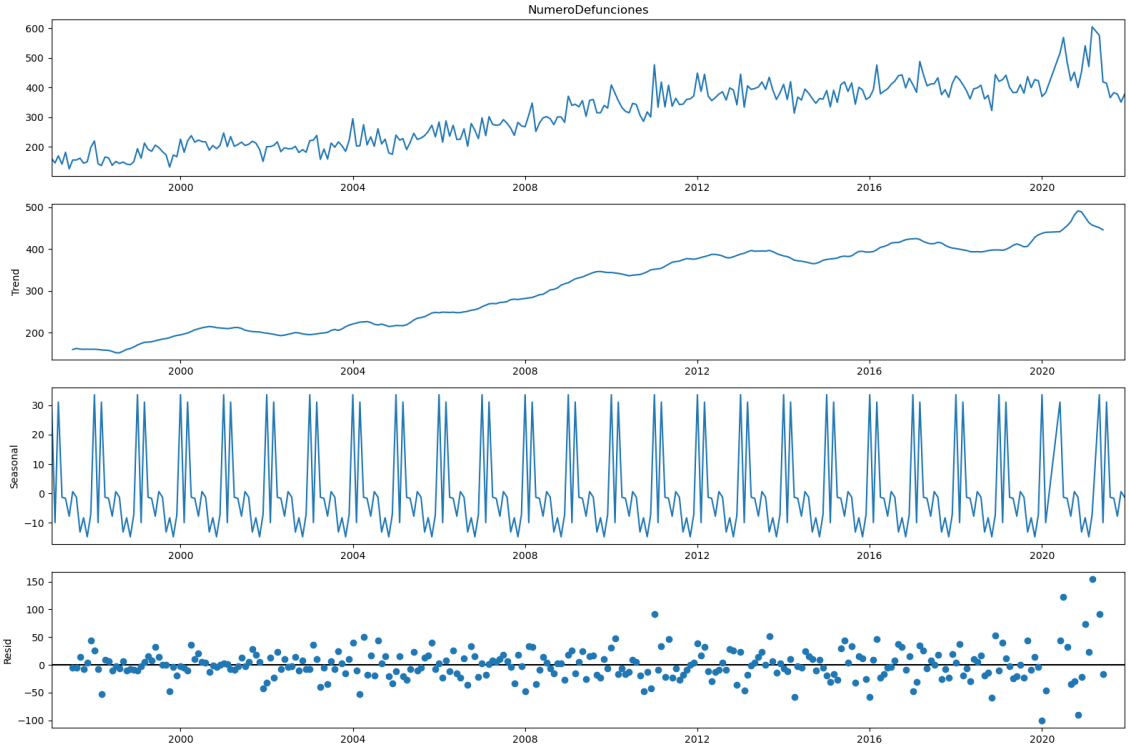
*Gráfico 18. Evolución de la causa de muerte diabetes mellitus previo tratamiento de valores atípicos.*




Posterior a la ejecución del código de la función de rango intercuartílico, se ha observado una reducción significativa en la cantidad de observaciones anómalas en los gráficos, esta depuración de datos atípicos contribuirá a mejorar la integridad de la serie temporal, facilitando así la generación de pronósticos más precisos y ajustados a las condiciones reales.

*Gráfico 19. Descomposición multiplicativa, causa de muerte diabetes mellitus.*

*Gráfico 20. Descomposición aditiva, causa de muerte diabetes mellitus.*




La descomposición aditiva y multiplicativa indica que las causas de muerte ocurridas a partir de la diabetes han mantenido una tendencia creciente, además existe un componente estacional en los datos.

Para aplicar los modelos de series de tiempo se procede a separar la base de datos en 80% para entrenamiento y 20% para evaluación, este proceso se detalla en la figura 40.

*Figura 40. Ejecución de código para dividir la data en entrenamiento y evaluación, causa de muerte diabetes.*

```
total_lend = len(df2_diabetes)
train_lend = round(total_lend*0.8)
train_lend
traind = df2_diabetes[0 : train_lend]
testd = df2_diabetes[train_lend : ]
```

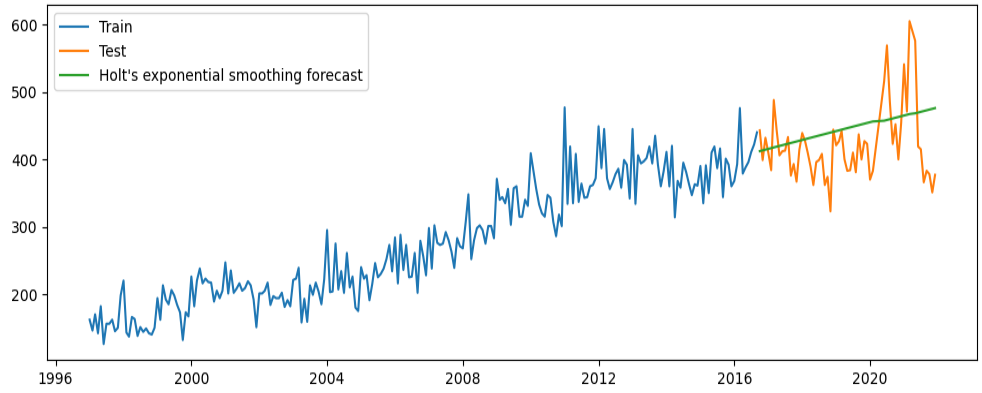
### a. Método de Suavizado Exponencial de Holt con Tendencia

En la figura 41, se establecen los parámetros del modelo, incluyendo la cantidad de períodos estacionales (12 meses), el tipo de tendencia (aditiva), y la ausencia de estacionalidad en los datos y en el gráfico 21 se visualizan las predicciones del método.

*Figura 41. Ejecución de código para el método de suavizado exponencial de Holt con tendencia, causa de muerte diabetes.*

```
from statsmodels.tsa.holtwinters import ExponentialSmoothing
model_d = ExponentialSmoothing(np.asarray(traind['NumeroDefunciones']) ,seasonal_periods=12 ,trend='additive', seasonal=None)
model_fit_d = model_d.fit(optimized=True)
print(model_fit_d.params)
y_hat_holt_d = testd.copy()
y_hat_holt_d['holt_forecast_d'] = model_fit_d.forecast(len(testd))
```

*Gráfico 21. Pronóstico de suavizamiento exponencial de Holt con tendencia, causa de muerte diabetes.*



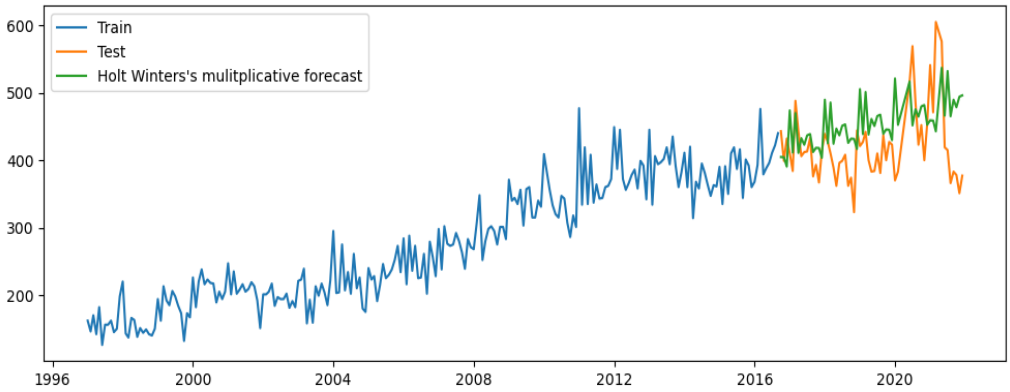

## b. Método multiplicativo de Holt Winter con tendencia y estacionalidad

En la figura 42, se especifican los parámetros del método, incluyendo la cantidad de períodos estacionales (12), el tipo de tendencia aditiva y la estacionalidad multiplicativa, posteriormente se visualizan las predicciones en el gráfico 22.

*Figura 42. Ejecución de código para el método Holt Winter con tendencia y estacionalidad, causa de muerte diabetes.*

```
y_hat_hwm_d = testd.copy()
model_d = ExponentialSmoothing(np.asarray(traind['NumeroDefunciones']) ,seasonal_periods=12 ,trend='add', seasonal='mul')
model_fit_d = model_d.fit(optimized=True)
print(model_fit_d.params)
y_hat_hwm_d['hw_forecast_d'] = model_fit_d.forecast(len(testd))
```

*Gráfico 22. Pronóstico multiplicativo de Holt Winters, causa de muerte diabetes.*




## c. Método ARIMA

Para aplicar los métodos autorregresivos se debe cumplir con el supuesto de estacionariedad, de modo que, se procede a verificar, si la serie de tiempo es estacionaria o no, a través de la prueba de Dickey Fuller, este proceso se detalla en la figura 43.

*Figura 43. Ejecución de código para obtener los resultados de la prueba de Dickey-Fuller, causa de muerte diabetes.*

```
adfuller_test(df2_diabetes['NumeroDefunciones'])
```

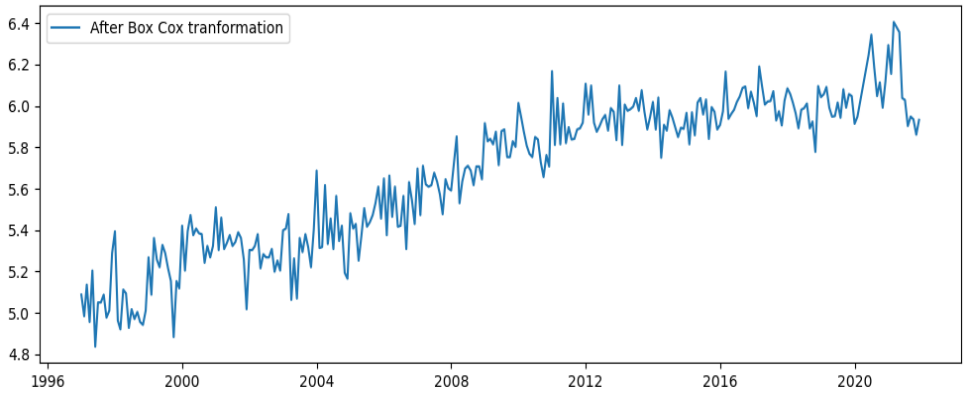
```
ADF Test Statistic : -1.0157350341285853
p-value : 0.7474821343767004
#Lags Used : 11
Number of Observations Used : 284
evidencia débil contra la hipótesis nula, la serie de tiempo tiene una raíz unitaria, lo que indica que no es estacionaria
```

Considerando que la serie no es estacionaria, aplicamos la transformación Box Cox, detallada en la figura 44 y visualizamos los resultados en el gráfico 23.

*Figura 44. Ejecución de código para la transformación Box Cox, causa de muerte diabetes.*

```
from scipy.stats import boxcox
data_boxcox_d= pd.Series(boxcox(df2_diabetes['NumeroDefunciones'], lmbda=0), index = df2_diabetes.index)
```

*Gráfico 23. Transformación Box Cox, causa de muerte diabetes.*

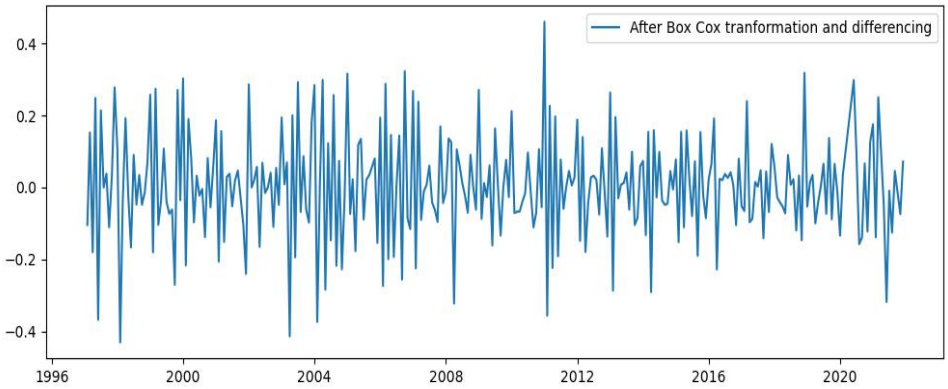



En la figura 45, se ejecuta el proceso de diferenciación de la serie temporal y en el gráfico 24 se visualizan los resultados.

*Figura 45. Ejecución de código para la diferenciación, causa de muerte diabetes.*

```
data_boxcox_diff_d= pd.Series(data_boxcox_d - data_boxcox_d.shift(), df2_diabetes.index)
```

*Gráfico 24. Transformación Box Cox y diferenciación, causa de muerte diabetes.*




En la figura 46 se comprueba que la serie de tiempo es estacionaria, de modo que, se puede proceder a aplicar los modelos autorregresivos,

*Figura 46. Ejecución de código para verificar estacionariedad posterior transformación y diferenciación, causa de muerte diabetes.*

```
adfuller_test(data_boxcox_diff_d)
```

```
ADF Test Statistic : -8.691260000808029
p-value : 4.0347992413967566e-14
#Lags Used : 10
Number of Observations Used : 284
Evidencia fuerte contra la hipótesis nula (Ho), rechace la hipótesis nula. Los datos no tienen raíz unitaria y son estacionario
```

Para continuar con el modelado se divide la base de datos en entrenamiento y evaluación, esto se detalla en la figura 47.

*Figura 47. Ejecución de código para dividir la data transformada en entrenamiento y evaluación, causa de muerte diabetes.*

```
train_data_boxcox_d = data_boxcox_d[:train_lend]
test_data_boxcox_d = data_boxcox_d[train_lend:]
train_data_boxcox_diff_d = data_boxcox_diff_d[:train_lend-1]
test_data_boxcox_diff_d = data_boxcox_diff_d[train_lend-1:]

y_pred_d=test_data_boxcox_diff_d.copy()
```

Se realiza la búsqueda automática del modelo ARIMA, esto se detalla en la figura 48.

*Figura 48. Ejecución de código para búsqueda del mejor modelo ARIMA.*

```
model_arima_d = auto_arima(train_data_boxcox_diff_d, trace=True, error_action='ignore',
                           start_p=0, start_q=0, max_p=3, max_q=3, suppress_warnings=True,
                           stepwise=False, seasonal=True)
model_arima_d.fit(train_data_boxcox_diff_d)
```

```
 ARIMA(0,0,0)(0,0,0)[1] intercept   : AIC=-219.220, Time=0.06 sec
 ARIMA(0,0,1)(0,0,0)[1] intercept   : AIC=-346.275, Time=0.14 sec
 ARIMA(0,0,2)(0,0,0)[1] intercept   : AIC=-344.334, Time=0.17 sec
 ARIMA(0,0,3)(0,0,0)[1] intercept   : AIC=-349.357, Time=0.34 sec
 ARIMA(1,0,0)(0,0,0)[1] intercept   : AIC=-312.363, Time=0.06 sec
 ARIMA(1,0,1)(0,0,0)[1] intercept   : AIC=-344.364, Time=0.21 sec
 ARIMA(1,0,2)(0,0,0)[1] intercept   : AIC=-355.323, Time=0.38 sec
 ARIMA(1,0,3)(0,0,0)[1] intercept   : AIC=-349.130, Time=0.34 sec
 ARIMA(2,0,0)(0,0,0)[1] intercept   : AIC=-319.052, Time=0.09 sec
 ARIMA(2,0,1)(0,0,0)[1] intercept   : AIC=inf, Time=0.42 sec
 ARIMA(2,0,2)(0,0,0)[1] intercept   : AIC=-348.689, Time=0.54 sec
 ARIMA(2,0,3)(0,0,0)[1] intercept   : AIC=-343.696, Time=0.56 sec
 ARIMA(3,0,0)(0,0,0)[1] intercept   : AIC=-329.813, Time=0.17 sec
 ARIMA(3,0,1)(0,0,0)[1] intercept   : AIC=-348.646, Time=0.47 sec
 ARIMA(3,0,2)(0,0,0)[1] intercept   : AIC=-346.593, Time=0.55 sec

Best model:  ARIMA(1,0,2)(0,0,0)[1] intercept
Total fit time: 4.503 seconds

ARIMA(order=(1, 0, 2), scoring_args={}, seasonal_order=(0, 0, 0, 1),
      suppress_warnings=True)
```

En el código se establece la búsqueda inicial de los parámetros p y q desde 0 hasta 3, se suprimen las advertencias, se permite la obtención de información detalladas durante la búsqueda de los parámetros, se realiza una búsqueda exhaustiva para evaluar todos los modelos posibles y se considera el componente estacional de la serie de tiempo.

En la figura 49 se realizan las predicciones.

*Figura 49. Ejecución de código para las predicciones modelo ARIMA, causa de muerte diabetes.*

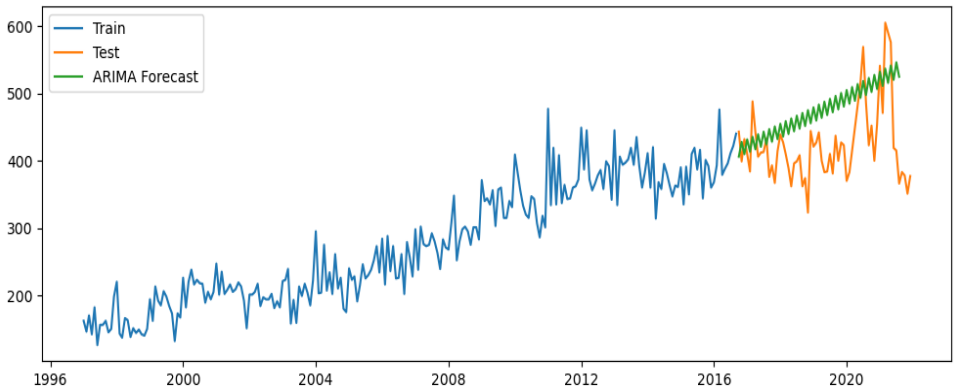
```
prediction_arima_d=model_arima_d.predict(len(test_data_boxcox_diff_d))
y_pred_d["ARIMA Model Prediction"]=prediction_arima_d
```

En la figura 50 se invierten las transformaciones para obtener las predicciones en la escala original.

*Figura 50. Ejecución de código para las invertir la transformación y diferenciación en el modelo ARIMA, causa de muerte diabetes.*

```
y_hat_arima_d = data_boxcox_diff_d.copy()
y_hat_arima_d['arima_forecast_boxcox_diff_d'] = prediction_arima_d
y_hat_arima_d['arima_forecast_boxcox_d'] = y_hat_arima_d['arima_forecast_boxcox_diff_d'].cumsum()
y_hat_arima_d['arima_forecast_boxcox_d'] = y_hat_arima_d['arima_forecast_boxcox_d'].add(data_boxcox_d[train_lend-1])
y_hat_arima_d['arima_forecast_d'] = np.exp(y_hat_arima_d['arima_forecast_boxcox_d'])
```

*Gráfico 25. Pronóstico ARIMA, causa de muerte diabetes.*




## d. Método SARIMA

En la figura 51, se realiza la búsqueda automática para encontrar el mejor modelo SARIMA.

*Figura 51. Ejecución de código modelo SARIMA, causa de muerte diabetes.*

```
model_sarima_d= auto_arima(train_data_boxcox_diff_d,trace=True, error_action='ignore',
                           start_p=0,d=0,start_q=0, max_p=3,max_d=2,max_q=3, m=12,
                           start_P=0, D=0, start_Q=0, max_P=3,max_Q=3, max_D=2,
                           suppress_warnings=True,stepwise=False,seasonal=True)
model_sarima.fit(train_data_boxcox_diff_d)
```

```
 ARIMA(2,0,0)(2,0,0)[12] intercept   : AIC=-343.226, Time=0.82 sec
 ARIMA(2,0,0)(2,0,1)[12] intercept   : AIC=-356.723, Time=2.32 sec
 ARIMA(2,0,0)(3,0,0)[12] intercept   : AIC=-345.855, Time=2.85 sec
 ARIMA(2,0,1)(0,0,0)[12] intercept   : AIC=inf, Time=0.46 sec
 ARIMA(2,0,1)(0,0,1)[12] intercept   : AIC=-293.615, Time=1.23 sec
 ARIMA(2,0,1)(0,0,2)[12] intercept   : AIC=inf, Time=2.21 sec
 ARIMA(2,0,1)(1,0,0)[12] intercept   : AIC=-263.726, Time=0.64 sec
 ARIMA(2,0,1)(1,0,1)[12] intercept   : AIC=-285.862, Time=1.18 sec
 ARIMA(2,0,1)(2,0,0)[12] intercept   : AIC=-326.324, Time=2.87 sec
 ARIMA(2,0,2)(0,0,0)[12] intercept   : AIC=-348.689, Time=0.43 sec
 ARIMA(2,0,2)(0,0,1)[12] intercept   : AIC=-353.568, Time=1.01 sec
 ARIMA(2,0,2)(1,0,0)[12] intercept   : AIC=-356.545, Time=1.21 sec
 ARIMA(2,0,3)(0,0,0)[12] intercept   : AIC=-343.696, Time=0.61 sec
 ARIMA(3,0,0)(0,0,0)[12] intercept   : AIC=-329.813, Time=0.22 sec
 ARIMA(3,0,0)(0,0,1)[12] intercept   : AIC=-336.885, Time=0.67 sec
 ARIMA(3,0,0)(0,0,2)[12] intercept   : AIC=-350.475, Time=1.77 sec
 ARIMA(3,0,0)(1,0,0)[12] intercept   : AIC=-341.449, Time=0.63 sec
 ARIMA(3,0,0)(1,0,1)[12] intercept   : AIC=inf, Time=2.03 sec
 ARIMA(3,0,0)(2,0,0)[12] intercept   : AIC=-355.899, Time=1.69 sec
 ARIMA(3,0,1)(0,0,0)[12] intercept   : AIC=-348.646, Time=0.55 sec
 ARIMA(3,0,1)(0,0,1)[12] intercept   : AIC=-353.994, Time=1.50 sec
 ARIMA(3,0,1)(1,0,0)[12] intercept   : AIC=-357.269, Time=1.62 sec
 ARIMA(3,0,2)(0,0,0)[12] intercept   : AIC=-346.593, Time=0.67 sec

Best model:  ARIMA(0,0,1)(1,0,1)[12] intercept
Total fit time: 165.655 seconds

ARIMA(order=(0, 0, 1), scoring_args={}, seasonal_order=(1, 0, 1, 12),
      suppress_warnings=True)
```

Se definen los siguientes parámetros: Información detallada del proceso de búsqueda de los parámetros, se ignoran los errores durante el proceso de búsqueda, se definen los rangos de búsqueda de los parámetros a través de los componentes estacionales y no estacionales, para p y q desde cero hasta tres, y d desde cero hasta dos, se suprimen los

mensajes de advertencias, se realiza una búsqueda exhaustiva y se considera el componente estacional de la serie.

En la figura 52 se realizan las predicciones.

*Figura 52. Ejecución de código para las predicciones del modelo SARIM, causa de muerte diabetes.*

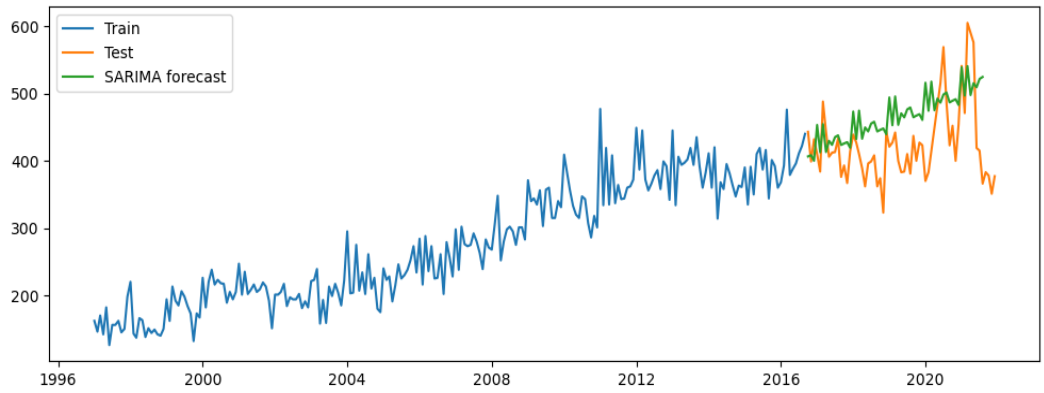
```
prediction_sarima_d=model_sarima_d.predict(len(test_data_boxcox_diff_d))
y_pred_d["SARIMA Model Prediction"]=prediction_sarima_d
```

En la figura 53 se invierten la transformación y la diferenciación para obtener las predicciones en la escala original y en el gráfico 26 se visualizan las predicciones del método SARIMA.

*Figura 53. Ejecución de código para invertir las transformaciones, modelo SARIMA, causa de muerte diabetes.*

```
y_hat_sarima_d = data_boxcox_diff_d.copy()
y_hat_sarima_d['sarima_forecast_boxcox_diff_d'] = prediction_sarima_d
y_hat_sarima_d['sarima_forecast_boxcox_d'] = y_hat_sarima_d['sarima_forecast_boxcox_diff_d'].cumsum()
y_hat_sarima_d['sarima_forecast_boxcox_d'] = y_hat_sarima_d['sarima_forecast_boxcox_d'].add(data_boxcox_d[train_lend-1])
y_hat_sarima_d['sarima_forecast_d'] = np.exp(y_hat_sarima_d['sarima_forecast_boxcox_d'])
y_pred_d["SARIMA Model Prediction"] = y_hat_sarima_d['sarima_forecast_d']
```

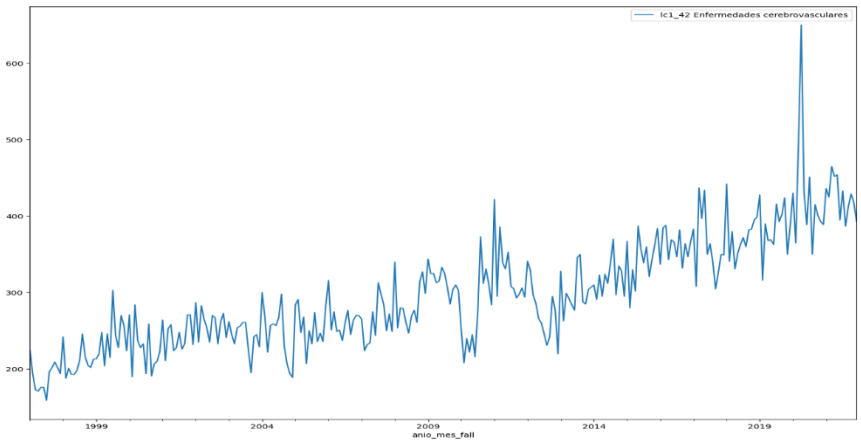
*Gráfico 26. Pronóstico SARIMA, causa de muerte diabetes.*




#### *4.1.4.1.4. Causa de muerte de muerte enfermedades cerebrovasculares.*

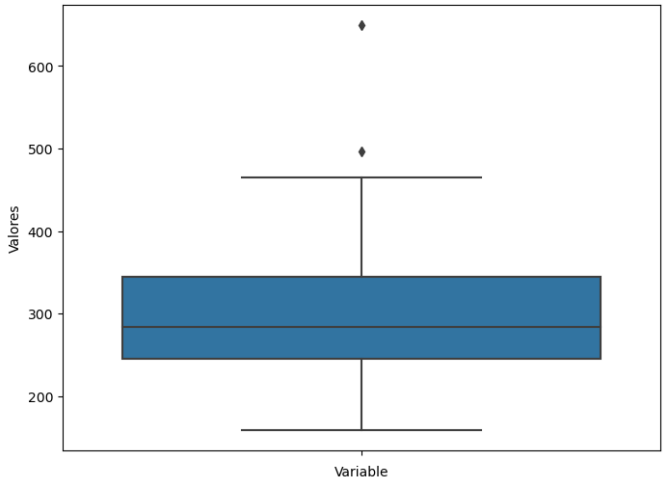
En el gráfico 27, se visualiza el comportamiento de la cantidad de muertes ocurridas a causa de enfermedades cerebrovasculares.

*Gráfico 27. Evolución de la causa de muerte enfermedades cerebrovasculares.*




Adicionalmente en el gráfico 28, se genera un diagrama de caja, para conocer como están distribuidos los datos.

*Gráfico 28. Boxplot causa de muerte enfermedades cerebrovasculares.*




Como se puede observar en el diagrama de caja, existen valores atípicos, de modo que, a través de método de rango intercuartílico se proceden a eliminar estos datos, esto se detalla en la figura 54.

*Figura 54. Ejecución de código para eliminar valores atípicos causa de muerte enfermedades cerebrovasculares.*

```
df_sin_outliers_c = eliminar_outliers(df_5c, 'lc1_42 Enfermedades cerebrovasculares')
```

*Gráfico 29. Boxplot de causa de muerte enfermedades cerebrovasculares previo tratamiento de valores atípicos.*

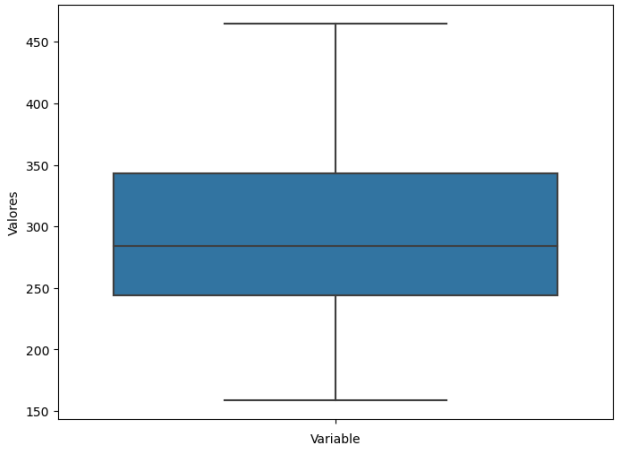
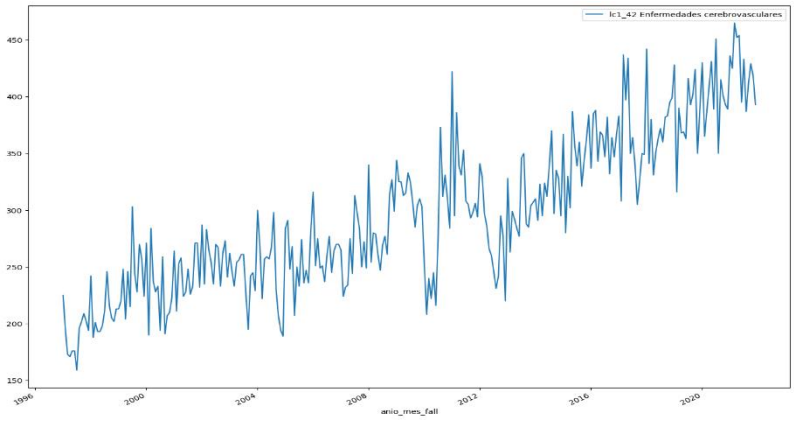


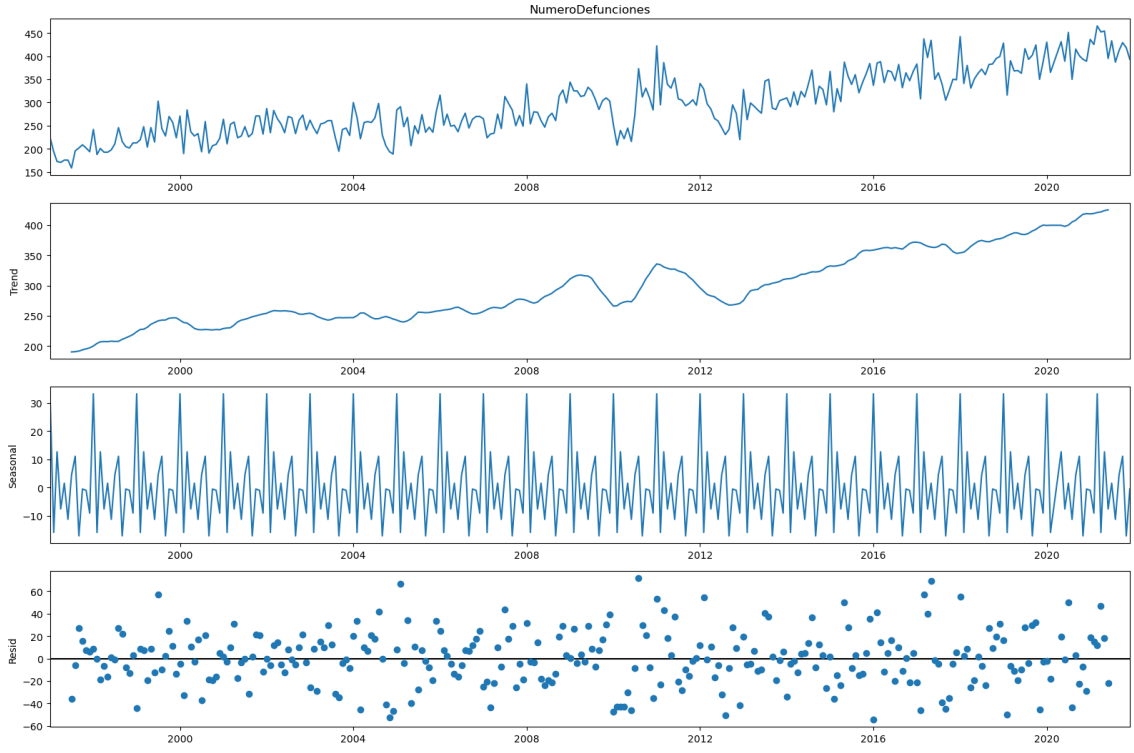

*Gráfico 30. Evolución de la causa de muerte enfermedades cerebrovasculares previo tratamiento de valores atípicos.*




Se procede a verificar en los gráficos 29 y 30, la eliminación de los valores atípicos en la serie temporal de la causa de muerte de enfermedades cerebrovasculares, lo cual permitirá reducir los sesgos en los métodos de series de tiempo que serán aplicados.

*Gráfico 31. Descomposición multiplicativa, causa de muerte enfermedades cerebrovasculares.*


*Gráfico 32. Descomposición aditiva, causa de muerte enfermedades cerebrovasculares.*




La descomposición aditiva y multiplicativa para la causa de muerte de enfermedades cerebrovasculares muestra que existe un comportamiento creciente con algunos picos en

ciertos años, adicionalmente el gráfico de estacionalidad indica que existe un componente estacional en la serie de tiempo.

Para aplicar los modelos de series de tiempo se procede a separar la base de datos en 80% para entrenamiento y 20% para evaluación, esto se detalla en la figura 55.

*Figura 55. Ejecución de código para dividir la data en entrenamiento y evaluación, causa de muerte enfermedades cerebrovasculares.*

```
total_len_c = len(df2_enfer_cerebrovas)
train_len_c = round(total_len_c*0.8)
train_len_c
train_c = df2_enfer_cerebrovas[0 : train_len_c]
test_c = df2_enfer_cerebrovas[train_len_c : ]
```

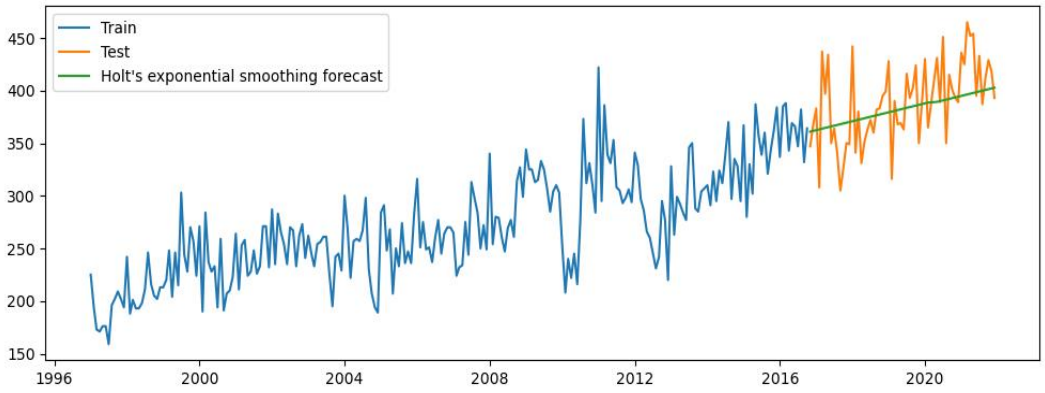
### a. Método de Suavizado Exponencial de Holt con Tendencia

Para aplicar el método de suavizado exponencial de Holt con tendencia, en la figura 56 se establece la cantidad de períodos estacionales (12 meses), el tipo de tendencia (aditiva), y la ausencia de estacionalidad en los datos y en el gráfico 33 se visualizan las predicciones.

*Figura 56. Ejecución de código para el método de suavizado exponencial de Holt, causa de muerte enfermedades cerebrovasculares.*

```
from statsmodels.tsa.holtwinters import ExponentialSmoothing
model_c = ExponentialSmoothing(np.asarray(train_c['NumeroDefunciones']) ,seasonal_periods=12 ,trend='additive', seasonal=None)
model_fit_c = model_c.fit(optimized=True)
print(model_fit_c.params)
y_hat_holt_c = test_c.copy()
y_hat_holt_c['holt_forecast_c'] = model_fit_c.forecast(len(test_c))
```

*Gráfico 33. Pronóstico de suavizamiento exponencial de Holt con tendencia, causa de muerte enfermedades cerebrovasculares.*



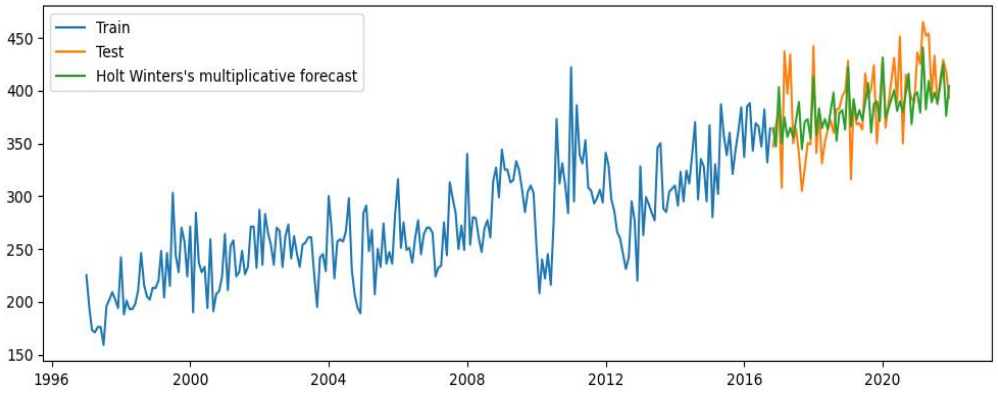

### b. Método multiplicativo de Holt Winters con tendencia y estacionalidad.

En el código de este método se incluye la cantidad de períodos estacionales (12 meses), el tipo de tendencia aditiva y la estacionalidad multiplicativa, esto se detalla en la figura 57, y las predicciones se visualizan en el gráfico 34.

*Figura 57. Ejecución de código para el método Holt Winter con tendencia y estacionalidad, causa de muerte enfermedades cerebrovasculares.*

```
y_hat_hwm_c = test_c.copy()
model_c = ExponentialSmoothing(np.asarray(train_c['NumeroDefunciones']) ,seasonal_periods=12 ,trend='add', seasonal='mul')
model_fit_c = model_c.fit(optimized=True)
print(model_fit_c.params)
y_hat_hwm_c['hw_forecast_c'] = model_fit_c.forecast(len(test_c))
```

*Gráfico 34. Pronóstico multiplicativo de Holt Winters, causa de muerte enfermedades cerebrovasculares.*




## c. Método ARIMA

Para proceder con la aplicación de los métodos ARIMA y SARIMA, es fundamental cumplir con el supuesto de estacionariedad. En este proceso, se verifica utilizando la prueba de Dickey-Fuller, tal como se detalla en la figura 58.

*Figura 58. Ejecución de código para obtener los resultados de la prueba de Dickey-Fuller, causa de muerte enfermedades cerebrovasculares*

```
adfuller_test(df2_enfer_cerebrovas['NumeroDefunciones'])
```

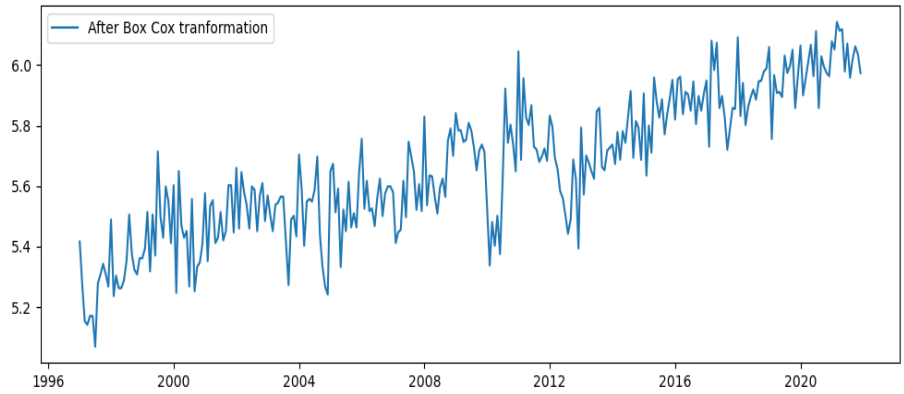
```
ADF Test Statistic : -0.9127781022418786
p-value : 0.7837352846000083
#Lags Used : 11
Number of Observations Used : 286
evidencia débil contra la hipótesis nula, la serie de tiempo tiene una raíz unitaria, lo que indica que no es estacionaria
```

Debido a que la series no es estacionaria, se aplica la transformación Box Cox, el código ejecutado se detalla en la figura 59 y los resultados se visualizan en el gráfico 35.

*Figura 59. Ejecución de código para la transformación Box Cox, causa de muerte enfermedades cerebrovasculares.*

```
data_boxcox_c = pd.Series(boxcox(df2_enfer_cerebrovas['NumeroDefunciones'], lmbda=0), index = df2_enfer_cerebrovas.index)
```

*Gráfico 35. Transformación Box Cox, causa de muerte enfermedades cerebrovasculares.*

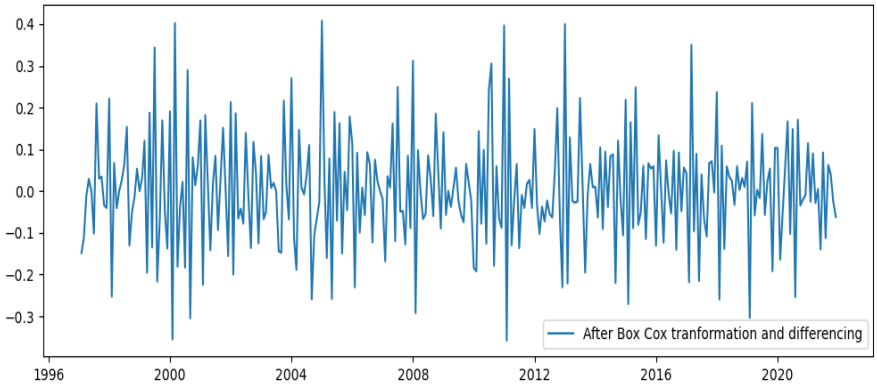



Posteriormente se aplica la diferenciación a la serie temporal transformada, el código de específica en la figura 60 y los resultados se visualizan en el grafico 36.

*Figura 60. Ejecución de código para la diferenciación, causa de muerte enfermedades cerebrovasculares.*

```
data_boxcox_diff_c = pd.Series(data_boxcox_c - data_boxcox_c.shift(), df2_enfer_cerebrovas.index)
```

*Gráfico 36. Transformación Box Cox y diferenciación, causa de muerte enfermedades cerebrovasculares.*




Se verifica la estacionariedad, en la figura 61.

*Figura 61. Ejecución de código para verificar estacionariedad posterior transformación y diferenciación, causa de muerte enfermedades cerebrovasculares.*

```
adfuller_test(data_boxcox_diff_c)
```

```
ADF Test Statistic : -6.8362385495026015
p-value : 1.8394745536828662e-09
#Lags Used : 16
Number of Observations Used : 280
Evidencia fuerte contra la hipótesis nula (Ho), rechace la hipótesis nula. Los datos no tienen raíz unitaria y son estacionario
```

En base a la serie transformada y diferenciada, y posterior a la verificación de estacionariedad en la serie, se generan los datos de entrenamiento y evaluación, tal como se detalla en la figura 62.

*Figura 62. Ejecución de código para dividir la data transformada en entrenamiento y evaluación, causa de muerte enfermedades cerebrovasculares.*

```
train_data_boxcox_c = data_boxcox_c[:train_len_c]
test_data_boxcox_c = data_boxcox_c[train_len_c:]
train_data_boxcox_diff_c = data_boxcox_diff_c[:train_len_c-1]
test_data_boxcox_diff_c = data_boxcox_diff_c[train_len_c-1:]

y_pred_c=test_data_boxcox_diff_c.copy()
```

En la figura 63 se realiza la búsqueda del mejor modelo ARIMA.

*Figura 63. Ejecución de código para búsqueda del mejor modelo ARIMA, causa de muerte enfermedades cerebrovasculares.*

```
model_arima_c = auto_arima(train_data_boxcox_diff_c, trace=True, error_action='ignore',
                           start_p=0, start_q=0, max_p=3, max_q=3, suppress_warnings=True,
                           stepwise=False, seasonal=True)
model_arima_c.fit(train_data_boxcox_diff_c)
```

```
 ARIMA(0,0,0)(0,0,0)[1] intercept   : AIC=-259.944, Time=0.68 sec
 ARIMA(0,0,1)(0,0,0)[1] intercept   : AIC=-353.628, Time=0.14 sec
 ARIMA(0,0,2)(0,0,0)[1] intercept   : AIC=-351.629, Time=0.15 sec
 ARIMA(0,0,3)(0,0,0)[1] intercept   : AIC=-359.001, Time=0.39 sec
 ARIMA(1,0,0)(0,0,0)[1] intercept   : AIC=-333.099, Time=0.08 sec
 ARIMA(1,0,1)(0,0,0)[1] intercept   : AIC=-351.630, Time=0.21 sec
 ARIMA(1,0,2)(0,0,0)[1] intercept   : AIC=-350.008, Time=0.37 sec
 ARIMA(1,0,3)(0,0,0)[1] intercept   : AIC=-362.827, Time=0.55 sec
 ARIMA(2,0,0)(0,0,0)[1] intercept   : AIC=-342.362, Time=0.14 sec
 ARIMA(2,0,1)(0,0,0)[1] intercept   : AIC=inf, Time=0.60 sec
 ARIMA(2,0,2)(0,0,0)[1] intercept   : AIC=-361.073, Time=0.72 sec
 ARIMA(2,0,3)(0,0,0)[1] intercept   : AIC=-348.658, Time=0.49 sec
 ARIMA(3,0,0)(0,0,0)[1] intercept   : AIC=-341.264, Time=0.31 sec
 ARIMA(3,0,1)(0,0,0)[1] intercept   : AIC=inf, Time=0.53 sec
 ARIMA(3,0,2)(0,0,0)[1] intercept   : AIC=inf, Time=0.54 sec

Best model:  ARIMA(1,0,3)(0,0,0)[1] intercept
Total fit time: 5.957 seconds

ARIMA(order=(1, 0, 3), scoring_args={}, seasonal_order=(0, 0, 0, 1),
      suppress_warnings=True)
```

La búsqueda inicial de los parámetros p y q se establece desde 0 hasta 3, se suprimen las advertencias, se habilita la obtención de información detallada durante la búsqueda de los parámetros, se lleva a cabo una búsqueda exhaustiva para evaluar todos los modelos posibles, teniendo en cuenta el componente estacional de la serie de tiempo.

En la figura 64 se realizan las predicciones.

*Figura 64. Ejecución de código para las predicciones modelo ARIMA, causa de muerte enfermedades cerebrovasculares.*

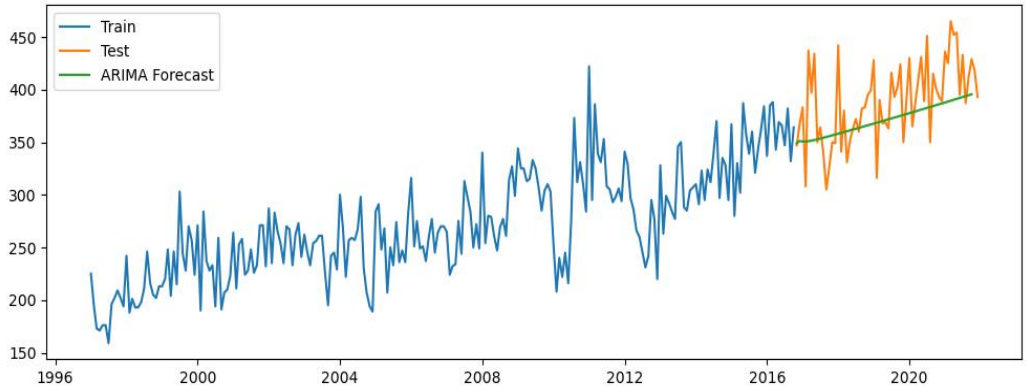
```
prediction_arima_c=model_arima_c.predict(len(test_data_boxcox_diff_c))
y_pred_c["ARIMA Model Prediction"]=prediction_arima_c
```

En la figura 65 se invierten la transformación y diferenciación para obtener las predicciones en la escala original y en el gráfico 37 se observan las predicciones obtenidas a través del mejor modelo ARIMA.

*Figura 65. Ejecución de código para las invertir la transformación y diferenciación en el modelo ARIMA, causa de muerte de enfermedades cerebrovasculares.*

```
y_hat_arima_c = data_boxcox_diff_c.copy()
y_hat_arima_c['arima_forecast_boxcox_diff_c'] = prediction_arima_c
y_hat_arima_c['arima_forecast_boxcox_c'] = y_hat_arima_c['arima_forecast_boxcox_diff_c'].cumsum()
y_hat_arima_c['arima_forecast_boxcox_c'] = y_hat_arima_c['arima_forecast_boxcox_c'].add(data_boxcox_c[train_len_c-1])
y_hat_arima_c['arima_forecast_c'] = np.exp(y_hat_arima_c['arima_forecast_boxcox_c'])
```

*Gráfico 37. Pronóstico ARIMA, causa de muerte enfermedades cerebrovasculares.*




## d. Método SARIMA

En la figura 66, se realiza la búsqueda del mejor modelo SARIMA, para la búsqueda del se establecen las siguientes condiciones: presentar en los resultados la información detallada durante el proceso de búsqueda de los parámetros, ignorar errores durante el proceso de búsqueda, se establecen los rangos de búsqueda de parámetros para componentes estacionales y no estacionales, con p y q desde cero hasta tres, y d desde cero hasta dos, se eliminan los mensajes de advertencia, se realiza una búsqueda exhaustiva, y se considera el componente estacional de la serie.

*Figura 66. Ejecución de código modelo SARIMA, causa de muerte enfermedades cerebrovasculares.*

```
model_sarima_c= auto_arima(train_data_boxcox_diff_c,trace=True, error_action='ignore',
                           start_p=0,d=0,start_q=0, max_p=3,max_d=2,max_q=3, m=12,
                           start_P=0, D=0, start_Q=0, max_P=3,max_Q=3, max_D=2,
                           suppress_warnings=True,stepwise=False,seasonal=True)
model_sarima_c.fit(train_data_boxcox_diff_c)
```

```
 ARIMA(2,0,2)(0,0,1)[12] intercept   : AIC=-354.841, Time=2.15 sec
 ARIMA(2,0,2)(1,0,0)[12] intercept   : AIC=-358.583, Time=1.86 sec
 ARIMA(2,0,3)(0,0,0)[12] intercept   : AIC=-348.658, Time=0.52 sec
 ARIMA(3,0,0)(0,0,0)[12] intercept   : AIC=-341.264, Time=0.31 sec
 ARIMA(3,0,0)(0,0,1)[12] intercept   : AIC=-342.413, Time=0.39 sec
 ARIMA(3,0,0)(0,0,2)[12] intercept   : AIC=-342.081, Time=2.04 sec
 ARIMA(3,0,0)(1,0,0)[12] intercept   : AIC=-343.088, Time=0.72 sec
 ARIMA(3,0,0)(1,0,1)[12] intercept   : AIC=inf, Time=1.62 sec
 ARIMA(3,0,0)(2,0,0)[12] intercept   : AIC=-344.664, Time=1.78 sec
 ARIMA(3,0,1)(0,0,0)[12] intercept   : AIC=inf, Time=0.61 sec
 ARIMA(3,0,1)(0,0,1)[12] intercept   : AIC=-361.333, Time=1.06 sec
 ARIMA(3,0,1)(1,0,0)[12] intercept   : AIC=-361.366, Time=1.04 sec
 ARIMA(3,0,2)(0,0,0)[12] intercept   : AIC=inf, Time=0.56 sec

Best model:  ARIMA(2,0,1)(1,0,1)[12] intercept
Total fit time: 172.218 seconds

ARIMA(order=(2, 0, 1), scoring_args={}, seasonal_order=(1, 0, 1, 12),
      suppress_warnings=True)
```

En la figura 67 se realizan las predicciones utilizando el mejor modelo SARIMA.

*Figura 67. Ejecución de código para las predicciones del modelo SARIMA, causa de muerte enfermedades cerebrovasculares.*

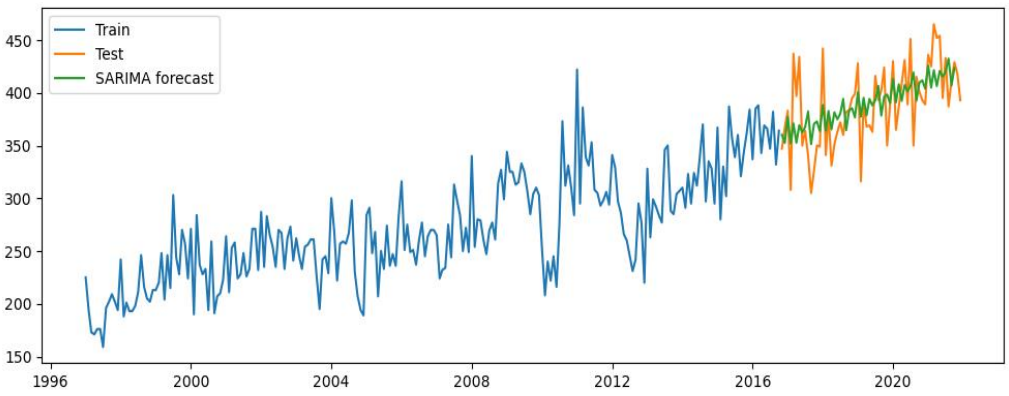
```
prediction_sarima_c=model_sarima_c.predict(len(test_data_boxcox_diff_c))
y_pred_c["SARIMA Model Prediction"]=prediction_sarima_c
```

En la figura 68 se invierten la transformación y diferenciación, para obtener los datos en la escala original y finalmente en el gráfico 38 se observan los resultados de las predicciones.

*Figura 68. Ejecución de código para invertir las transformaciones, modelo SARIMA, causa de muerte enfermedades cerebrovasculares.*

```
y_hat_sarima_c = data_boxcox_diff_c.copy()
y_hat_sarima_c['sarima_forecast_boxcox_diff_c'] = prediction_sarima_c
y_hat_sarima_c['sarima_forecast_boxcox_c'] = y_hat_sarima_c['sarima_forecast_boxcox_diff_c'].cumsum()
y_hat_sarima_c['sarima_forecast_boxcox_c'] = y_hat_sarima_c['sarima_forecast_boxcox_c'].add(data_boxcox_c[train_len_c-1])
y_hat_sarima_c['sarima_forecast_c'] = np.exp(y_hat_sarima_c['sarima_forecast_boxcox_c'])
y_pred_c["SARIMA Model Prediction"] = y_hat_sarima_c['sarima_forecast_c']
```

*Gráfico 38. Pronóstico SARIMA, causa de muerte enfermedades cerebrovasculares.*




***Evaluación y Validación de los Modelos***

**a. Causa de Muerte Enfermedades Isquémicas del Corazón**

*Figura 69. Resultados de las métricas de los métodos aplicados para la causa de muerte enfermedades isquémicas del corazón.*

| Method | MSE | RMSE | MAE | MAPE |
|---|---|---|---|---|
| Holt's exponential smoothing method | 7267.91 | 85.25 | 69.44 | 11.05 |
| Holt Winters' multiplicative method | 7462.81 | 86.39 | 70.19 | 11.09 |
| Autoregressive integrated moving average (ARIMA) method | 17863.17 | 133.65 | 120.19 | 18.33 |
| Seasonal autoregressive integrated moving average (SARIMA) method | 4616.49 | 67.94 | 56.89 | 8.90 |

En función de los métodos aplicados y en base a las métricas utilizadas se observa que los errores más bajos se registran en el método SARIMA, de modo que, es el método que mejor predice la causa de defunción de enfermedades isquémicas del corazón, por ende, las predicciones futuras se realizarán a través de este método.

### b. Causa de Muerte Enfermedades Diabetes Mellitus

*Figura 70. Resultados de las métricas de los métodos aplicados para la causa de muerte diabetes mellitus.*

| Method | MSE | RMSE | MAE | MAPE |
|---|---|---|---|---|
| Holt's exponential smoothing method | 3421.20 | 58.49 | 46.85 | 11.42 |
| Holt Winters' multiplicative method | 4433.32 | 66.58 | 53.25 | 13.10 |
| Autoregressive integrated moving average (ARIMA) method | 6373.91 | 79.84 | 59.17 | 14.71 |
| Seasonal autoregressive integrated moving average (SARIMA) method | 5330.14 | 73.01 | 52.55 | 13.00 |

Se puede observar que el método de suavizado exponencial de Holt con tendencia tiene el MSE más bajo de todos los modelos evaluados, lo que indica que tiene la menor variación cuadrática media entre las predicciones y los valores reales. Además, tiene el menor RMSE, MAE y MAPE, lo que sugiere que tiene el mejor ajuste a los datos y las predicciones más precisas en comparación con los otros modelos.

### c. Causa de Muerte Enfermedades Cerebrovasculares

*Figura 71. Resultados de las métricas de los métodos aplicados para la causa de muerte enfermedades cerebrovasculares.*

| Method | MSE | RMSE | MAPE | MAE | $R^2$ |
|---|---|---|---|---|---|
| Holt's exponential smoothing method | 1159.03 | 34.04 | 6.99 | 27.15 | 0.214925 |
| Holt Winters' multiplicative method | 938.40 | 30.63 | 6.35 | 24.58 | 0.364368 |
| Autoregressive integrated moving average (ARIMA) method | 1379.60 | 37.14 | 7.30 | 29.14 | 0.065517 |
| Seasonal autoregressive integrated moving average (SARIMA) method | 878.81 | 29.64 | 6.70 | 25.52 | 0.404736 |

En la causa de muerte enfermedades cerebrovasculares se aplican 5 métricas para evaluar los resultados de los métodos aplicados, debido a que entre MSE, RMSE, MAPE y MAE

no permiten elegir un método debido a que MSE y RMSE tiene valores bajos para el modelo SARIMA, y MAPE junto con MAE tiene valores bajos en el método multiplicativo de Holt Winters.

En función de las métricas utilizadas se observa que el mejor modelo es el SARIMA (Seasonal Autoregressive Integrated Moving Average). Este modelo tiene un buen desempeño en términos de MSE y RMSE, y además tiene el mayor $R^2$ entre los modelos evaluados. Esto sugiere que el modelo SARIMA proporciona predicciones precisas y al mismo tiempo explica una mayor proporción de la variabilidad en los datos en comparación con los otros modelos.

Los resultados se proceden a analizar en la sección 6.

# 5. Clustering

## 5.1.1. Comprensión del Negocio

***Determinación de los Objetivos Comerciales***

*5.1.1.1.1. Contexto*

El análisis de patrones de agrupación relacionados con la mortalidad en Ecuador es crucial para proporcionar información estratégica que guíe la formulación de políticas específicas en el sistema de salud, permitiendo identificar a los grupos de población más afectados, lo que a su vez facilita la asignación eficiente de recursos y la implementación de estrategias de prevención y control de enfermedades.

*5.1.1.1.1. Definición de los Objetivos del Negocio*

Identificar patrones de agrupación relacionados con la mortalidad en el Ecuador, con el fin de proporcionar insights que guíen la formulación de políticas específicas y eficaces para el sistema de salud.

*5.1.1.1.2. Criterios de Rendimiento.*

Descubrir patrones significativos y pertinentes que destaquen las características distintivas de cada grupo demográfico.

***Evaluación de la Situación.***

A través de la información obtenida del Registro Estadístico de Defunciones Generales, se busca identificar patrones efectivos en cada grupo demográfico, en este sentido, la comparación entre segmentos es crucial para evaluar el impacto en las decisiones de las entidades públicas y del Estado.

Entre los principales riesgos que se pueden presentar en la ejecución del presente proyecto están la elección incorrecta del número de clusters, la presencia de valores faltantes y erróneos puede afectar en la calidad de los resultados, adicionalmente la interpretación subjetiva de los resultados puede variar en función de la perspectiva del analista.

La dependencia entre variables categorías puede afectar la precisión del modelo, además puede existir dificultad para evaluar los resultados del modelo debido a que las métricas no aplican a todos los escenarios.

***Determinación de los Objetivos de Minería de Datos***

#### *5.1.1.1.3. Objetivos Aplicados a Clustering*

Aplicar técnicas de clustering con el objetivo de descubrir patrones, correlaciones y tendencias significativas en la mortalidad en el Ecuador.

#### *5.1.1.1.4. Criterios de Rendimiento*

Para evaluar el rendimiento de la técnica de clustering se utilizará la métrica de Silhouette Score, un valor más alto indica una mejor calidad de los agrupamientos, donde los objetos están más cerca de los miembros de su propio grupo y más alejados de los miembros de otros grupos.

***Plan de Proyecto***

*Tabla 2. Plan del proyecto de fin titulación, sección clustering*

| Fase | Duración | Recursos | Riesgos |
|---|---|---|---|
| Comprensión del negocio | 1 semana | Responsable del proyecto | |
| Comprensión de los datos | 1 semana | Acceso a datos históricos sobre las defunciones | |
| Preparación de los datos | 1 semana | | |
| Modelado | 4 semanas | Plataforma para el modelado (Jupyter Notebook) | Valores faltantes puedes sesgar los resultados |
| Evaluación | 1 semanas | | Dificultad en la interpretabilidad de los resultados |

Fuente: Autoría Propia.

### 5.1.2. Comprensión de los Datos

***Recopilación de Datos Iniciales***

Los datos sobre las defunciones en el Ecuador fueron suministrados por el Instituto Nacional de Estadística y Censos en formato SPSS, estos datos se importaron y revisaron en la Jupyter Notebook con el fin de procesarlos, el período utilizado comprende desde 1990 hasta el año 2021.

***Descripción de los Datos***

La base de datos de las defunciones generales en el Ecuador tiene 1.982.281 filas y 49 columnas, cada fila corresponde a una defunción ocurrida, y posee información asociada al fallecido.

Como se puede observar en la figura 72, la mayor parte de los tipos de datos son categóricos.

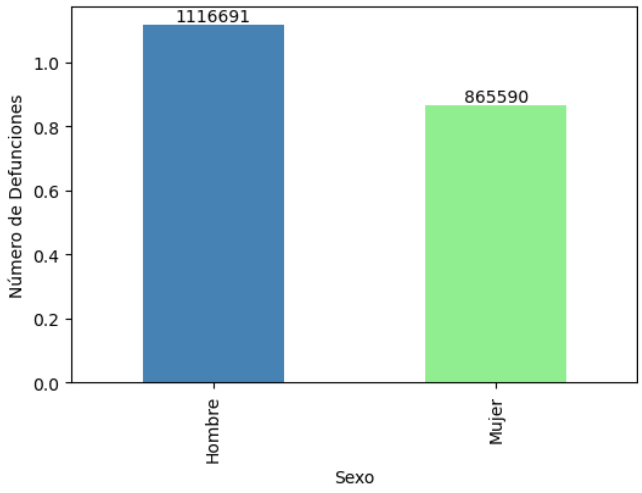
*Figura 72. Tipos de datos.*

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1982281 entries, 0 to 1982280
Data columns (total 49 columns):
 #   Column      Dtype
---  ------      -----
 0   prov_insc   category
 1   cant_insc   category
 2   parr_insc   category
 3   anio_insc   float64
 4   mes_insc    category
 5   dia_insc    float64
 6   fecha_insc  object
 7   nac_fall    category
 8   cod_pais    category
 9   sexo        category
 10  anio_nac    float64
 11  mes_nac     category
 12  dia_nac     float64
 13  anio_fall   float64
 14  mes_fall    category
 15  dia_fall    float64
 16  fecha_fall  object
 17  cod_edad    category
 18  edad        float64
 19  prov_res    category
 20  cant_res    category
 21  parr_res    category
 22  est_civil   category
 23  sabe_leer   category
 24  etnia       category
 24  etnia       category
 25  lugar_ocur  category
 26  prov_fall   category
 27  cant_fall   category
 28  parr_fall   category
 29  muj_fertil  category
 30  mor_viol    category
 31  lug_viol    category
 32  autopsia    category
 33  niv_inst    category
 34  fecha_nac   object
 35  causa9      object
 36  cer_por     category
 37  anio_base   float64
 38  total       float64
 39  area_fall   category
 40  area_res    category
 41  causa3      category
 42  causa       category
 43  causa103    category
 44  causa80     category
 45  causa67A    category
 46  causa67B    category
 47  causa4      category
 48  lc1         category
dtypes: category(36), float64(9), object(4)
memory usage: 282.1+ MB
```

### *Exploración de Datos*

En el Ecuador, en el periodo comprendido entre 1990 y 2021, se registró que el 56,33% de los fallecimientos correspondieron al género masculino, mientras que el 43,67% restante correspondió al género femenino.

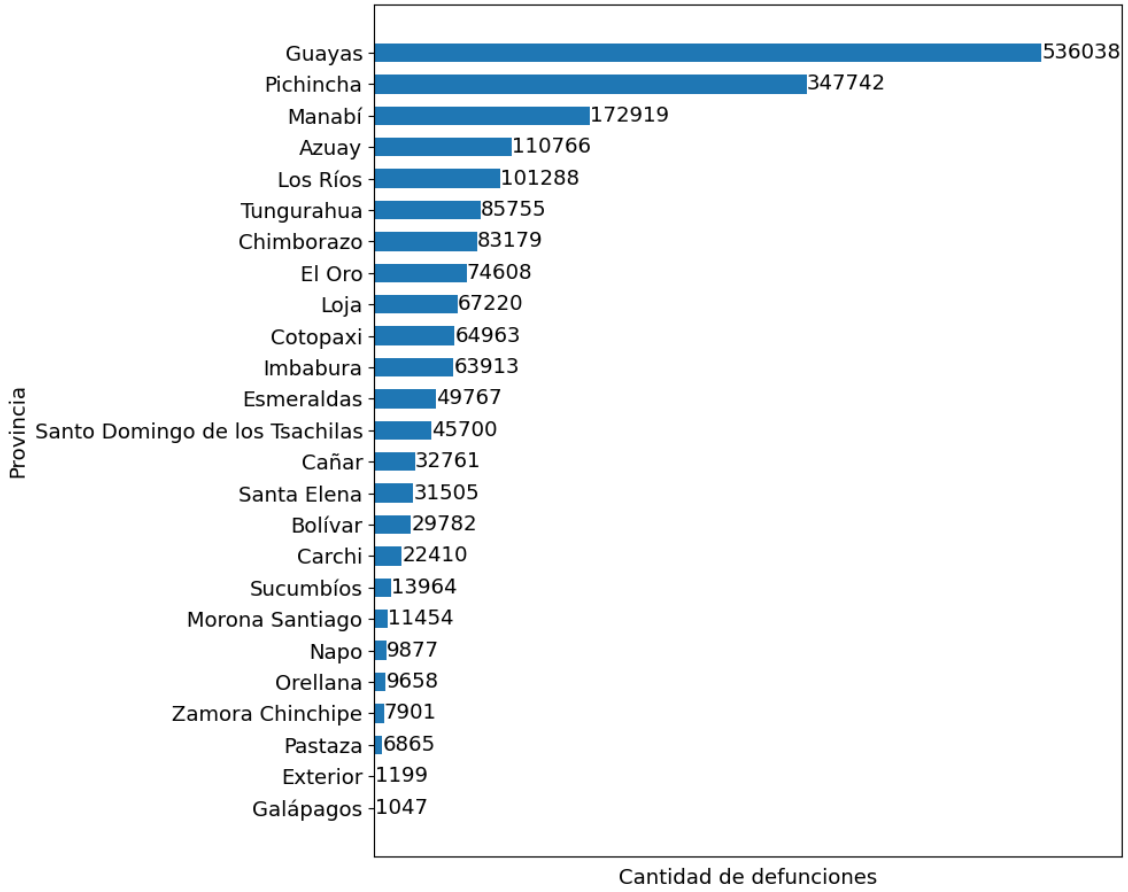
*Gráfico 39. Cantidad de defunciones por sexo, periodo 1990 – 2021.*




En el Ecuador las defunciones se concentran históricamente en las 4 provincias más grandes esto es Guayas, Pichincha, Manabí y Azuay. La provincia de Guayas registra la

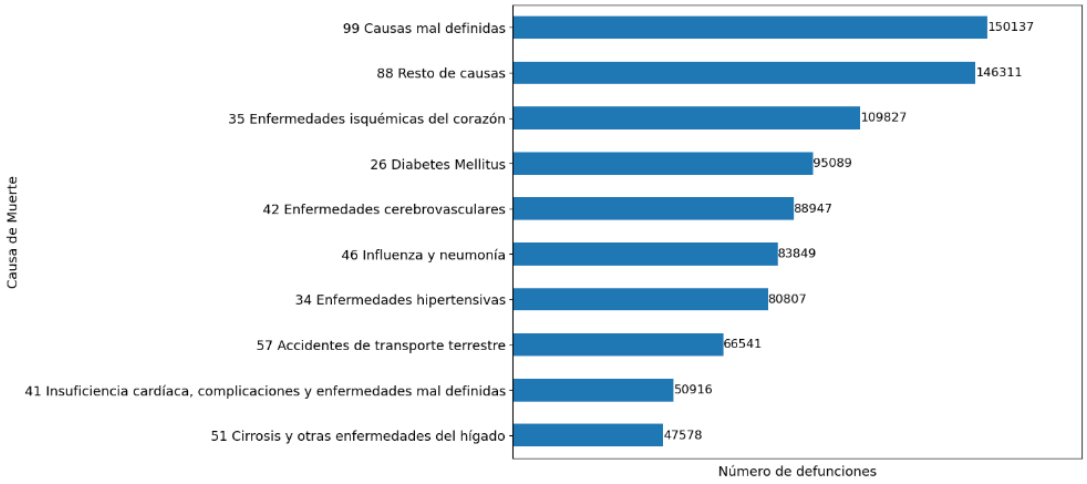
cifra más alta de defunciones, alcanzando 536.038 desde 1990 hasta 2021, seguida por la provincia de Pichincha con 347.742 y Manabí con 172.919.

*Gráfico 40. Cantidad de defunciones a nivel provincial, periodo 1990 – 2021.*



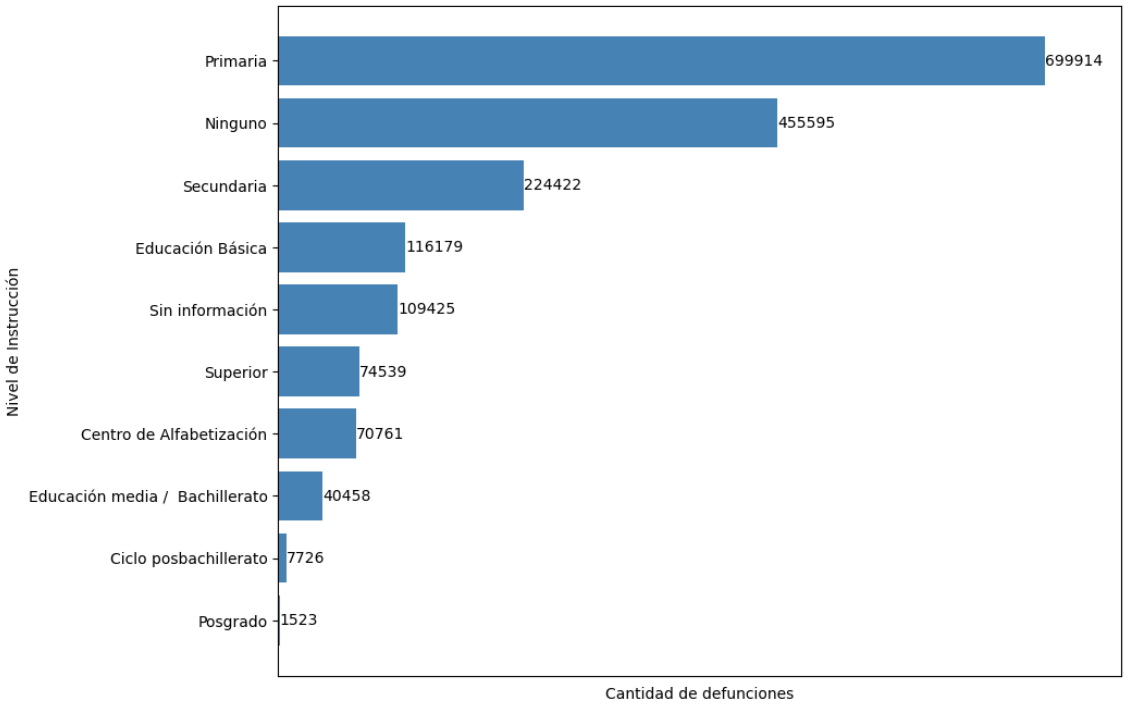

Como se puede evidenciar en el gráfico 40 existen 150.137 casos de defunciones que poseen causas mal definidas, 146.311 registros que corresponden al resto de causas y entre las enfermedades que son principales causas de muerte están: enfermedades isquémicas al corazón, diabetes mellitus, enfermedades cerebrovasculares, influenza y neumonía, y enfermedades hipertensivas.

*Gráfico 41. Principales causas de muerte en el Ecuador, periodo 1990 – 2021.*




En Ecuador, entre 1990 y 2021, la mayoría de los fallecidos tenían educación primaria, con un total de 699.914 casos, seguidos por aquellos sin nivel de instrucción, que sumaron 455.595. Por último, se registraron 224.422 fallecidos con educación secundaria. En cuanto al nivel educativo más alto, el posgrado, se registraron solo 1.523 casos durante ese periodo.

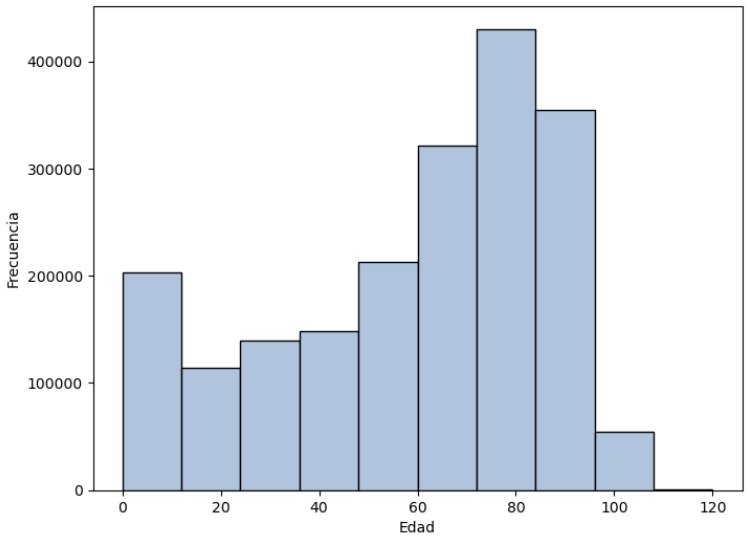
*Gráfico 42. Cantidad de defunciones por nivel de instrucción del fallecido, periodo 1990 – 2021.*




La distribución de edades de los fallecidos se concentra mayormente entre los 60 y los 95 años aproximadamente. Además, se observa una notable presencia de fallecimientos en

el rango de edad de cero a diez años. A partir de esta etapa, la cantidad de fallecidos aumenta progresivamente hasta alcanzar su punto máximo alrededor de los 85 años.

*Gráfico 43. Distribución edad del fallecido.*




***Verificación de la Calidad de los Datos***

En la figura 73, se verifica que existe una importante cantidad de valores faltantes.

*Figura 73. Verificar valores nulos*

| | |
|---|---|
| prov_fall | 0 |
| cant_fall | 0 |
| parr_fall | 0 |
| muj_fertil | 1956221 |
| mor_viol | 1769791 |
| lug_viol | 1896116 |
| autopsia | 1199368 |
| niv_inst | 181739 |
| fecha_nac | 1282136 |
| causa9 | 0 |
| cer_por | 0 |
| anio_base | 0 |
| total | 0 |
| area_fall | 0 |
| area_res | 0 |
| causa3 | 0 |
| causa | 0 |
| causa103 | 0 |
| causa80 | 0 |
| causa67A | 0 |
| causa67B | 0 |
| causa4 | 0 |
| lc1 | 363765 |
| dtype: int64 | |

| | |
|---|---|
| prov_insc | 0 |
| cant_insc | 0 |
| parr_insc | 0 |
| anio_insc | 2867 |
| mes_insc | 2867 |
| dia_insc | 1284446 |
| fecha_insc | 1284862 |
| nac_fall | 1281579 |
| cod_pais | 0 |
| sexo | 0 |
| anio_nac | 0 |
| mes_nac | 0 |
| dia_nac | 1281579 |
| anio_fall | 0 |
| mes_fall | 0 |
| dia_fall | 1281579 |
| fecha_fall | 1281579 |
| cod_edad | 0 |
| edad | 0 |
| prov_res | 0 |
| cant_res | 0 |
| parr_res | 0 |
| est_civil | 199729 |
| sabe_leer | 185659 |
| etnia | 1036048 |
| lugar_ocur | 0 |

En la figura 74, se visualiza que existen ciertas causas de muerte que están únicamente codificadas y no se detalla la causa de muerte a la cual pertenece el código.

*Figura 74. Causas de muerte.*

```
COVID-19, virus no identificado                                          8303
50 Apendicitis, hernia y obstrucción intestinal                          8183
49 Insuficiencia respiratoria                                            7931
61 Accidentes que obstruyen la respiración                               7184
29 Demencia y enfermedad de Alzheimer                                    7180
23 Neoplasia maligna del encéfalo                                        6809
52 Enfermedades del sistema osteomuscular y tejido conjuntivo            6331
32 Epilepsia y estado de mal epiléptico                                  5760
12 Neoplasia maligna de la vesícula biliar y de otras                    5637
40 Arritmias cardíacas                                                   5134
54 Embarazo, parto y puerperio                                           4572
28 Trastornos de los líquidos, electrolitos, y del equilibrio ácido básico 4569
30 Trastornos mentales y del comportamiento por uso de sustancias psicoactivas 4336
101.0                                                                    4321
```

En la base de datos constan defunciones del exterior, esto se verifica en la figura 75.

*Figura 75. Provincia de defunción.*

```
Cañar            32761
Santa Elena      31505
Bolívar          29782
Carchi           22410
Sucumbíos        13964
Morona Santiago  11454
Napo              9877
Orellana          9658
Zamora Chinchipe  7901
Pastaza           6865
Exterior          1199
Galápagos         1047
Name: count, dtype: int64
```

### 5.1.3. Preparación de los Datos

#### *Selección de Datos*

Para seleccionar los atributos, se considera el enfoque de clustering basado en k modes el cual requiere únicamente de variables categóricas, el resto de variables no se consideran relevantes para el presente análisis.

*Figura 76. Ejecución de código para seleccionar variables de interés.*

```
columnas_deseadas = ['sexo', 'edad', 'prov_fall','niv_inst','lc1','anio_mes_fall']
df_study = df7[columnas_deseadas]
df_study.head()
```

| | sexo | edad | prov_fall | niv_inst | lc1 | anio_mes_fall |
|---|---|---|---|---|---|---|
| 363766 | Hombre | 57.0 | El Oro | Primaria | 9 Neoplasia maligna del estómago | 1997-5-1 |
| 363768 | Mujer | 2.0 | Azuay | NaN | 5 Meningitis | 1997-3-1 |
| 363770 | Hombre | 76.0 | Azuay | Primaria | 2 Tuberculosis | 1997-12-1 |
| 363772 | Mujer | 28.0 | El Oro | Primaria | 9 Neoplasia maligna del estómago | 1997-8-1 |
| 363774 | Mujer | 74.0 | Azuay | Secundaria | 53 Enfermedades del sistema urinario | 1997-11-1 |

#### *Limpieza de Datos*

Dado que se han identificado valores nulos dentro del conjunto de datos, es imperativo implementar un procedimiento de tratamiento específico para abordar dichos valores.

Debido a que existen registros del exterior en la provincia de fallecimiento, se proceden a filtrar las defunciones sin esta condición, tal como se detalla en la figura 77.

*Figura 77. Ejecución de código para filtrar defunciones ocurridas en el Ecuador.*

```
valor_a_eliminar = "Exterior"
df2 = df_dropped.drop(df_dropped[df_dropped['prov_fall'] == valor_a_eliminar].index)
df2['prov_fall'].value_counts(dropna=False)
```

En la figura 78 se proceden a eliminar los registros provenientes de las muertes violentas, debido a que no corresponde a un comportamiento típico de la mortalidad.

*Figura 78. Ejecución de código para eliminar muertes violentas.*

```
df3 = df2[df2['mor_viol'].isna()]
```

Con respecto a la variable causa de muerte, se proceden a eliminar 363.766 filas vacías, con la finalidad de no generar sesgos en los resultados del modelo a partir de la imputación de la moda, este valor corresponde el 18,4% del total de los datos, lo cual puede representar un porcentaje importante para imputación, el proceso de eliminación se detalla en la figura 79.

*Figura 79. Ejecución de código para eliminar valores vacíos en la lista corta de causas de muerte.*

```
def reemplazar_vacios_con_nan(valor):
    if isinstance(valor, str) and valor.strip() == "":
        return np.nan
    return valor

df4 = df3.copy()
df4['lc1'] = df4['lc1'].apply(reemplazar_vacios_con_nan)
df4.head()
```

Existen códigos en la variable lc1 que no poseen la categoría, por ende, debemos asignar la que corresponde, esto se detalla en la figura 80.

*Figura 80. Ejecución de código para asignar categorías a causas de muerte.*

```
df4['lc1'] = df4['lc1'].astype(str)
codigos_a_reemplazar1 = ["100.0"]
df4['lc1'] =df4['lc1'].replace(codigos_a_reemplazar1, "COVID-19, virus identificado")
codigos_a_reemplazar2 = ["101.0"]
df4['lc1'] =df4['lc1'].replace(codigos_a_reemplazar2, "COVID-19, virus no identificado")
```

Se eliminan los registros que tienen como causa de muerte COVID debido a que son datos atípicos que afectan al comportamiento normal de los datos, adicionalmente se eliminan

las causas mal definidas y el resto de causas que no brindan información relevante sobre las muertes en el Ecuador, este proceso se detalla en la figura 81.

*Figura 81. Ejecución de código para eliminar causas de muerte por COVID-19.*

```
df5 = df4[~df4['lc1'].str.contains('COVID-19, virus identificado|COVID-19, virus no identificado
                                   |99 Causas mal definidas|88 Resto de causas', case=False)]
df5.info()
```

En la figura 82 se verifica que aún constan causas de muerte violentas, por lo cual se procede a eliminarlas.

*Figura 82. Ejecución de código para eliminar causas de muertes violentas.*

```
df6 = df5[~df5['lc1'].str.contains('62 Envenenamiento accidental|65 Eventos de intención no determinada
                                   |63 Lesiones autoinflingidas intencionalmente \(Suicidio\)
                                   |61 Accidentes que obstruyen la respiración|57 Accidentes de transporte terrestre
                                   |60 Ahogamiento y sumersión accidentales|58 Caídas accidentales|58 Caidas accidentales
                                   |64 Agresiones \(Homicidios\)|59 Disparo de arma de fuego no intencional', case=False)]
```

En la figura 83 se procede a asignar el número al mes de defunción para crear la variable fecha de defunción.

*Figura 83. Ejecución de código para cambiar el tipo de dato a la fecha de defunción*

```
df7 = df6.copy()

codigos_a_reemplazar = ["Enero"]
df7  ['mes_fall'] =df7  ['mes_fall'].replace(codigos_a_reemplazar, "01")
codigos_a_reemplazar1 = ["Febrero"]
df7  ['mes_fall'] =df7  ['mes_fall'].replace(codigos_a_reemplazar1, "02")
codigos_a_reemplazar2 = ["Marzo"]
df7  ['mes_fall'] =df7  ['mes_fall'].replace(codigos_a_reemplazar2, "03")
codigos_a_reemplazar3 = ["Abril"]
df7  ['mes_fall'] =df7  ['mes_fall'].replace(codigos_a_reemplazar3, "04")
codigos_a_reemplazar4 = ["Mayo"]
df7  ['mes_fall'] =df7  ['mes_fall'].replace(codigos_a_reemplazar4, "05")
codigos_a_reemplazar5 = ["Junio"]
df7  ['mes_fall'] =df7  ['mes_fall'].replace(codigos_a_reemplazar5, "06")
codigos_a_reemplazar6 = ["Julio"]
df7  ['mes_fall'] =df7  ['mes_fall'].replace(codigos_a_reemplazar6, "07")
codigos_a_reemplazar7 = ["Agosto"]
df7  ['mes_fall'] =df7  ['mes_fall'].replace(codigos_a_reemplazar7, "08")
codigos_a_reemplazar8 = ["Septiembre"]
df7  ['mes_fall'] =df7  ['mes_fall'].replace(codigos_a_reemplazar8, "09")
codigos_a_reemplazar9 = ["Octubre"]
df7  ['mes_fall'] =df7  ['mes_fall'].replace(codigos_a_reemplazar9, "10")
codigos_a_reemplazar10 = ["Noviembre"]
df7  ['mes_fall'] =df7  ['mes_fall'].replace(codigos_a_reemplazar10, "11")
codigos_a_reemplazar11 = ["Diciembre"]
df7  ['mes_fall'] =df7 ['mes_fall'].replace(codigos_a_reemplazar11, "12")
```

En la figura 84, se muestra el proceso de creación de la variable temporal, para el efecto se cambia el tipo de dato a entero, se genera una nueva variable que corresponde al día de la defunción, con la finalidad de contar con una fecha tipo d/m/a, y finalmente se crea la variable temporal que corresponde al año, mes y día de defunción.

*Figura 84. Ejecución de código para crear la fecha de defunción.*

```
bdefunciones['anio_fall'] = bdefunciones['anio_fall'].astype(int)
bdefunciones['mes_fall'] = bdefunciones['mes_fall'].astype(int)
bdefunciones['dia_def'] = 1
bdefunciones['dia_def'] = bdefunciones['dia_def'].astype(int)
bdefunciones['anio_mes_fall'] = bdefunciones['anio_fall'].astype(str) + \
                                '-' +bdefunciones['mes_fall'].astype(str)+ \
                                '-' +bdefunciones['dia_def'].astype(str)
bdefunciones['anio_mes_fall'] = pd.to_datetime(bdefunciones['anio_mes_fall'])
```

Se verifica que existe el código 999 en la edad, lo cual indica que existen registros que no poseen edad, de modo que, se proceden a eliminar, esto se detalla en la figura 85.

*Figura 85. Ejecución de código para eliminar datos faltantes en edad.*

```
df7 = df6[df6['edad'] != 999.0]
```

A la base de datos que contiene las variables de interés se proceden a eliminar los valores vacíos, esto se verifica en la figura 86.

*Figura 86. Ejecución de códigos para eliminar valores vacíos.*

```
df_study = df_study.dropna()
df_study.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1014672 entries, 363766 to 1946893
Data columns (total 6 columns):
 #   Column         Non-Null Count    Dtype
---  ------         --------------    -----
 0   sexo           1014672 non-null  category
 1   edad           1014672 non-null  float64
 2   prov_fall      1014672 non-null  category
 3   niv_inst       1014672 non-null  category
 4   lc1            1014672 non-null  object
 5   anio_mes_fall  1014672 non-null  object
dtypes: category(3), float64(1), object(2)
memory usage: 33.9+ MB
```

En la figura 87 se eliminan todos los registros de fallecidos de mayores a 60 años, debido a que en un primer análisis no se identificaron clusters significativos considerando a este grupo de la población, posteriormente se probó eliminando este grupo de población y se observaron mejoras sustanciales en los resultados del análisis, así como interpretaciones más congruentes con la realidad. Además, se destaca la importancia de obtener información detallada sobre grupos de población que abarquen desde niños hasta aquellos en la mediana edad, con el propósito de disponer de información relevante para la elaboración de políticas públicas.

*Figura 87. Ejecución de código para filtrar los registros de menores de 60 años de edad.*

```
df_filtered = df_study[df_study['edad'] < 60]
```

***Construcción de Datos***

En la figura 88 se proceden a generar rangos para edad para cada registro.

- Niños (0-12 años)
- Adolescentes (13-19 años)
- Adultos jóvenes (20-39 años)
- Adultos de mediana edad (40-59 años)

*Figura 88. Ejecución de código para crear categorías de la edad del fallecido y eliminar variable edad.*

```
edades = [0, 12, 19, 39, float('inf')]
categorias = ['Niño', 'Adolescente', 'Adulto jóven', 'Adulto de mediana edad']
df_filtered['edad_categorica'] = pd.cut(df_filtered['edad'], bins=edades, labels=categorias, right=False)
df_filtered = df_filtered.drop('edad', axis=1)
df_filtered
```

***Formateo de Datos***

En la figura 89, se verifica que existen variables que no posee el tipo de dato que corresponde, por lo cual se procede a modificar, este proceso se detalla en la figura 90.

*Figura 89. Verificar tipos de datos.*

```
df_filtered.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 295126 entries, 363766 to 1946893
Data columns (total 6 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   sexo             295126 non-null  category
 1   prov_fall        295126 non-null  category
 2   niv_inst         295126 non-null  category
 3   lc1              295126 non-null  object
 4   anio_mes_fall    295126 non-null  object
 5   edad_categorica  295126 non-null  category
dtypes: category(4), object(2)
memory usage: 7.9+ MB
```

*Figura 90. Ejecutar código para modificar el tipo de dato de las variables causa de muerte y fecha de defunción.*

```
df_filtered['lc1'] = df_filtered['lc1'].astype('category')
df_filtered['anio_mes_fall'] = df_filtered['anio_mes_fall'].astype('category')
```

### 5.1.4. Modelado

***Selección de Técnicas de Modelado***

Con la finalidad de identificar patrones, correlaciones y tendencias relevantes en la mortalidad en Ecuador, y dado que la mayoría de los datos son de naturaleza categórica, se debe emplear el algoritmo k-modes, el cual usa este tipo de variables.

***Diseño de Comprobación***

Se utilizará la métrica de Silhouette Score para evaluar el desempeño de la técnica de clustering.

### *Generación de Modelos*

Se instala k-modes y luego se importan las librerías que se requieren para ejecución del algoritmo k modes, esto se puede observar en la figura 91.

*Figura 91. Ejecución de código para instalar librerías para k-modes*

```
from kmodes.kmodes import KModes
import matplotlib.pyplot as plt
```

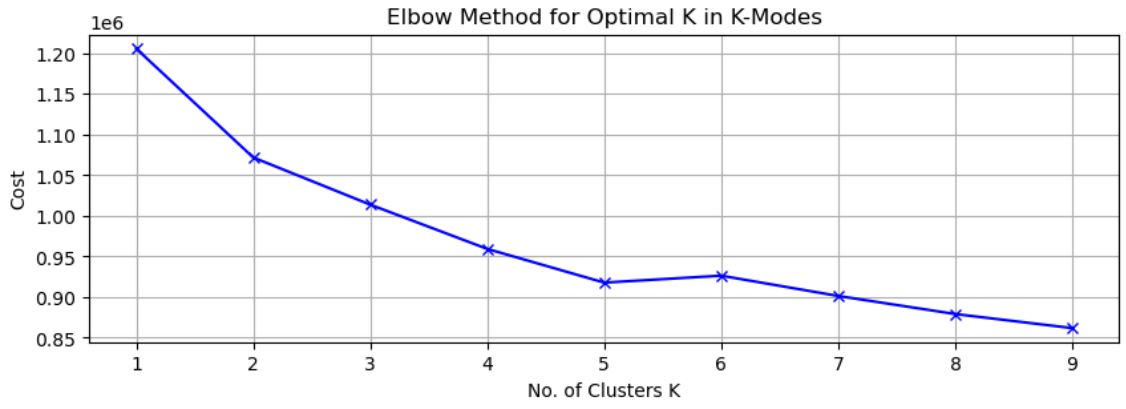
En la figura 92, se calcula el valor óptimo de clusters, a través del método Elbow

*Figura 92. Ejecución de código para determinar el número óptimo de clusters.*

```
cost = []
K = range(1,10)
for num_clusters in list(K):
    kmode = KModes(n_clusters=num_clusters, init = "Huang", n_init = 10, verbose=1, random_state=0)
    kmode.fit_predict(df)
    cost.append(kmode.cost_)
```

Para la búsqueda del número óptimo de clusters se define un rango de números de clusters desde 1 hasta 9, los centroides se inicializan a través del método Huang, se especifica la cantidad de veces que se ejecutará el algoritmo Kmodes con diferentes centroides iniciales en este caso 10, se configura el nivel de verbosidad para mostrar información sobre el proceso de clustering y se fija una semilla aleatoria.

*Gráfico 44. Método Elbow para determinar el número óptimo de clusters.*




En función del gráfico 44 se puede observar que existe un ligero codo en el cluster número 2 y otro codo en el cluster número 5, adicionalmente se evidencia que a partir del número 6 el coste se ralentiza, por lo cual se procederá a probar con estos distintos valores para posteriormente evaluar los resultados.

En el gráfico 93 se construye el modelo en base a k = 2

*Figura 93. Ejecución de código para clustering.*

```
kmode2 = KModes(n_clusters=2, init = "Huang", n_init = 10, verbose=1, random_state=0)
clusters2 = kmode2.fit_predict(df)
clusters2
```

```
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 1, iteration: 1/100, moves: 17842, cost: 1148849.0
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 2, iteration: 1/100, moves: 33259, cost: 1071928.0
Run 2, iteration: 2/100, moves: 3, cost: 1071928.0
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 3, iteration: 1/100, moves: 61888, cost: 1125211.0
Run 3, iteration: 2/100, moves: 780, cost: 1125211.0
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 4, iteration: 1/100, moves: 49892, cost: 1099034.0
Run 4, iteration: 2/100, moves: 3354, cost: 1099034.0
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 5, iteration: 1/100, moves: 30539, cost: 1079016.0
Run 5, iteration: 2/100, moves: 0, cost: 1079016.0
```

En la figura 94 se insertan las etiquetas en el data set.

*Figura 94. Ejecución de código para insertar etiquetas del cluster*

```
df_labels2 = df.copy()
df_labels2.insert(0,"cluster_labels", clusters2, True)
```

Para evaluar los clusters se utilizará el coeficiente de Gower, el cual determinar la medida de similitud en el conjunto de datos respecto de la variable numérica (cluster) y variables categóricas.

Posterior a la instalación de Gower, se importan las librerías detalladas en la figura 95.

*Figura 95. Ejecución de código para importar librerías.*

```
import gower
from sklearn.metrics import silhouette_score
```

Debido al coste computacional para obtener el Silhouette Score, se debe utilizar una muestra de los datos de 10.000 registros, se fija una semilla aleatoria que sea reproducible y previamente se convierten los valores a cadenas de texto, este proceso se puede observar en la figura 96.

*Figura 96. Ejecución de código para obtener una muestra y modificar el tipo de datos.*

```
df_str = df_labels2.applymap(str)
df_str = df_str.sample(n=10000, random_state=1)
```

En la figura 97 se ejecuta el proceso y se obtiene el resultado de la métrica.

*Figura 97. Ejecución de código para obtener la distancia de Gower y calcular el Silhouette Score.*

```
gower_distances = gower.gower_matrix(df_str.drop('cluster_labels', axis=1))
silhouette_avg_gower = silhouette_score(gower_distances, df_str['cluster_labels'], metric='precomputed')
print("Average Categorical Silhouette Score:", silhouette_avg_gower)
```

```
Average Categorical Silhouette Score: 0.18437394
```

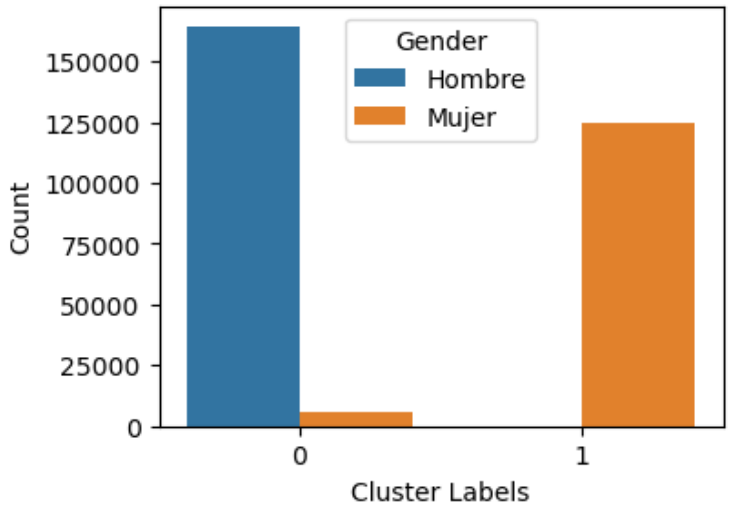
Se realiza el mismo proceso para entrenar el modelo con 5 y 3 clusters y se obtienen los siguientes resultados en la métrica del Average Silhouette Score.

Cluster 5 = 0.1408416

Cluster 3 = 0.116577536

Por ende, en función de los resultados se evidencia que el valor más alto para el Silhouette Score se obtiene usando 2 clusters, ahora debemos revisar si los resultados tienen una interpretación válida.

*Gráfico 45. Distribución de los clusters a nivel del género del fallecido.*




El gráfico 45 se muestra que el clustering básicamente está dividiendo los datos en un grupo de hombres y uno de mujeres, por ende, estos resultados no poseen una interpretación válida, por lo cual se procederá a considerar menos variables para el análisis.

En la figura 98, se selecciona la provincia de fallecimiento, la causa de muerte y los rangos de edad.

*Figura 98. Ejecución de código para seleccionar nuevas variables.*

```
selected_columns = ['prov_fall','lc1','edad_categorica']
df_red = df[selected_columns]
```

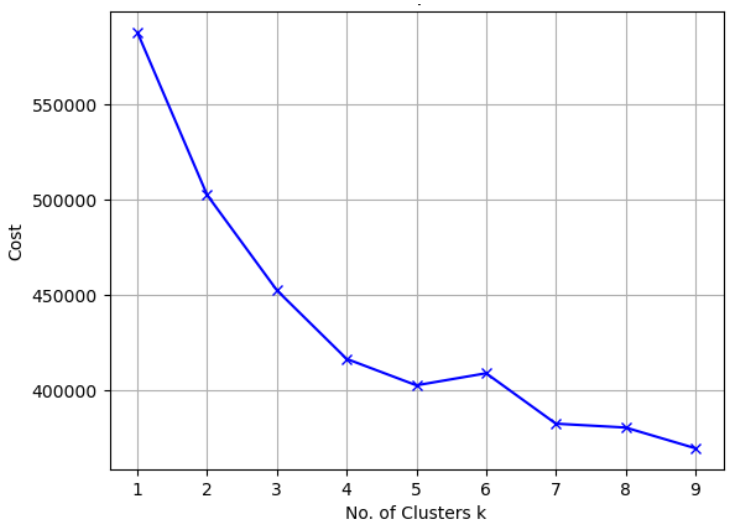
En la figura 99, se determina el número óptimo de clusters.

*Figura 99. Ejecución de código para determinar el número óptimo de clusters.*

```
cost_red = []
K_red = range(1,10)
for num_clusters in list(K_red):
    kmode_red = KModes(n_clusters=num_clusters, init = "Huang", n_init = 5, verbose=1, random_state=0)
    kmode_red.fit_predict(df_red)
    cost_red.append(kmode_red.cost_)
```

En base al gráfico 46 se puede observar que existe un ligero codo en el cluster número 2, y otro en el cluster número 4, posterior al cluster número 5 aumenta el coste, por ende, se procederá a probar diferentes números de clusters.

*Gráfico 46. Método Elbow para determinar el número óptimo de clusters.*




***Evaluación de los Modelos***

Una vez realizado el proceso de entrenamiento de los modelos con cada cluster, se obtienen los siguientes resultados del Silhouette Score.

- Cluster 2 = 0.28260353
- Cluster 3 = 0.332463
- Cluster 4 = 0.15647021
- Cluster 5 = 0.18919127

En función de los resultados el cluster número 3 obtiene el valor más alto para el Average Silhouette Score.

Para visualizar el cluster se reduce la dimensionalidad del dataset a dos variables usando MCA (Análisis de Correspondencias Múltiples), esta técnica es usada para analizar y visualizar datos categóricos, de modo que, se puedan identificar patrones y relaciones entre las categorías de múltiples variables.

Para el proceso de MCA en la figura 100 se codifican las categorías de las variables, luego se calculan las distancias para medir la similaridad o diferencia, posteriormente se utiliza la descomposición en valores singulares para reducir la dimensionalidad en los datos, y finalmente se generan los gráficos que permitan visualizar las distancias.

*Figura 100. Ejecución de código para transformar las variables categóricas a dummies.*

```
df_categorical = pd.get_dummies(df_categorical)
df_categorical
```

En la figura 101 se instala MCA y se importa la librería prince.

*Figura 101. Ejecución de código para instalar e importar librerías para reducción de dimensionalidad.*

```
!pip install prince
import prince
```

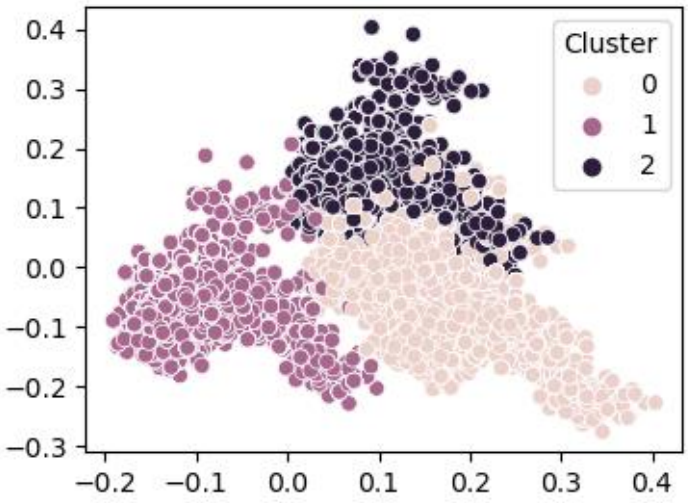
Se inicializa el objeto MCA y se especifica 2 componentes principales, se ajusta el objeto MCA a los datos, esto calcula las coordenadas de las categorías, finalmente se transforma el conjunto de datos utilizando el modelo MCA ajustado, lo cual proporciona las coordenadas de cada registro en los nuevos ejes definidos por los componentes principales, este proceso se detalla en la figura 102.

*Figura 102. Ejecución de código para inicializar y ajustar el objeto MCA y para crear las coordenadas.*

```
mca = prince.MCA(n_components=2)
mca = mca.fit(df_categorical)
mca_coordinates = mca.transform(df_categorical)
```

En el gráfico 47 se evidencia que los clusters están muy cerca unos de otros, lo cual podría indicar que las muestras comparten características comunes.

*Gráfico 47. Visualizar los clusters.*




## 6. Análisis de Resultados

### 6.1. Series de Tiempo

En base a las métricas utilizadas, se destaca que el método SARIMA se establece como el mejor modelo para predecir la causa de defunción por enfermedades isquémicas del corazón. Este método presenta los errores más bajos entre los métodos evaluados, lo que indica una mayor precisión en sus predicciones.

En el caso de las enfermedades relacionadas con la diabetes mellitus, se observa que el método de suavizado exponencial de Holt con tendencia presenta los errores más bajos de todos los modelos analizados, lo que indica un ajuste óptimo a los datos y una precisión superior en sus predicciones en comparación con otros modelos.

En cuanto a las enfermedades cerebrovasculares, debido a que las métricas MSE, RMSE, MAPE y MAE no permiten seleccionar un único método de manera clara, se aplican 5 métricas para evaluar los resultados de los métodos aplicados. Entre ellas, se destaca que el modelo SARIMA muestra un buen desempeño en términos de MSE y RMSE, y además exhibe el mayor $R^2$ entre los modelos evaluados. Esto indica que el modelo SARIMA proporciona predicciones precisas y, al mismo tiempo, explica una mayor proporción de la variabilidad en los datos en comparación con los otros.

Los resultados obtenidos indican una mayor precisión en las predicciones realizadas a través del modelo SARIMA, el cual captura las tendencias y variaciones en los datos de

las enfermedades cerebrovasculares, por lo tanto, puede ser valioso en la toma de decisiones y la planificación relacionada con la salud pública en el contexto de las enfermedades cerebrovasculares.

A nivel de los gráficos generados para visualizar los resultados de las predicciones en comparación con los datos de validación para cada uno de los métodos utilizados, se evidencia que para el caso de las enfermedades isquémicas del corazón el método SARIMA captura las tendencias y los periodos de estacionalidad, asimismo las predicciones se encuentran por debajo de los datos reales utilizados para evaluación. Para el caso de la causa de muerte de diabetes mellitus el método de suavizamiento exponencial del Holt con tendencia captura el comportamiento creciente pero no captura la variabilidad. Para la causa de muerte proveniente de enfermedades cerebrovasculares el método SARIMA captura la tendencia creciente y los periodos de estacionalidad, los picos altos o bajos no son capturados por este tipo de método y los resultados de las predicciones son más pequeños que los resultados de los datos reales.

### 6.1.1. Despliegue

#### a. Causa de Muerte Enfermedades Isquémicas del Corazón

Para el desarrollo de las predicciones de la causa de muerte ocurridas por enfermedades isquémica del corazón se utiliza el modelo autorregresivo SARIMA, el cual posee las mejores métricas de evaluación con respecto al resto de modelos, para el efecto se procede a ejecutar la transformación Box Cox y se aplica la diferenciación a la serie de tiempo, posteriormente se comprueba que la serie es estacionaria, este proceso se verifica en la figura 103.

*Figura 103. Prueba de estacionariedad, método SARIMA.*

```
adfuller_test(serie_transformada_diferenciada)
```

```
ADF Test Statistic : -3.732640677713971
p-value : 0.0036767683929460316
#Lags Used : 13
Number of Observations Used : 263
Evidencia fuerte contra la hipótesis nula (Ho), rechace la hipótesis nula. Los datos no tienen raíz unitaria y son estacionario
s.
```

En la figura 104 se buscan los parámetros del mejor modelo SARIMA,

*Figura 104. Ejecución código modelo SARIMA.*

```
best_model_sarima= auto_arima(serie_transformada_diferenciada,trace=True, error_action='ignore',
                        start_p=0,d=0,start_q=0, max_p=3,max_d=2,max_q=3, m=12,
                        start_P=0, D=0, start_Q=0, max_P=3,max_Q=3, max_D=2,
                        suppress_warnings=True,stepwise=False,seasonal=True)
```

```
 ARIMA(2,0,1)(1,0,1)[12] intercept   : AIC=-397.272, Time=1.38 sec
 ARIMA(2,0,1)(2,0,0)[12] intercept   : AIC=-378.043, Time=2.48 sec
 ARIMA(2,0,2)(0,0,0)[12] intercept   : AIC=-340.442, Time=0.57 sec
 ARIMA(2,0,2)(0,0,1)[12] intercept   : AIC=-364.307, Time=1.80 sec
 ARIMA(2,0,2)(1,0,0)[12] intercept   : AIC=-371.959, Time=2.24 sec
 ARIMA(2,0,3)(0,0,0)[12] intercept   : AIC=-342.436, Time=0.72 sec
 ARIMA(3,0,0)(0,0,0)[12] intercept   : AIC=-331.301, Time=0.13 sec
 ARIMA(3,0,0)(0,0,1)[12] intercept   : AIC=-356.395, Time=0.80 sec
 ARIMA(3,0,0)(0,0,2)[12] intercept   : AIC=-356.467, Time=1.49 sec
 ARIMA(3,0,0)(1,0,0)[12] intercept   : AIC=-362.735, Time=1.32 sec
 ARIMA(3,0,0)(1,0,1)[12] intercept   : AIC=inf, Time=2.56 sec
 ARIMA(3,0,0)(2,0,0)[12] intercept   : AIC=-365.109, Time=1.68 sec
 ARIMA(3,0,1)(0,0,0)[12] intercept   : AIC=-340.670, Time=0.38 sec
 ARIMA(3,0,1)(0,0,1)[12] intercept   : AIC=-364.502, Time=1.42 sec
 ARIMA(3,0,1)(1,0,0)[12] intercept   : AIC=-372.112, Time=2.76 sec
 ARIMA(3,0,2)(0,0,0)[12] intercept   : AIC=inf, Time=0.67 sec

Best model:  ARIMA(0,0,2)(2,0,1)[12] intercept
Total fit time: 208.778 seconds
```

Se procede a entrenar el modelo, en la figura 105.

*Figura 105. Ejecución de código para entrenamiento del modelo SARIMA.*

```
best_model_sarima = best_model_sarima.fit(serie_transformada_diferenciada)
best_model_sarima
```

```
ARIMA(order=(0, 0, 2), scoring_args={}, seasonal_order=(2, 0, 1, 12),
      suppress_warnings=True)
```

Se calculan las predicciones para los siguientes 60 meses, este proceso se detalla en la figura 106.

*Figura 106. Ejecución de código para predicciones futuras.*

```
periods = 60
sarima_prediction = best_model_sarima.predict(periods)
sarima_prediction
```

```
array([ 0.12928986, -0.04443   ,  0.01486149, -0.02862791,  0.04608572,
       -0.00690758, -0.01072399, -0.00438065, -0.00369385,  0.02328742,
        0.14167612, -0.13155186,  0.10330391, -0.04894392,  0.01896886,
       -0.03202849,  0.0450039 , -0.00411974, -0.01202357,  0.00069488,
       -0.00586185,  0.0166808 ,  0.14396051, -0.12700189,  0.09986529,
       -0.04800606,  0.01891098, -0.03139299,  0.04411788, -0.0037544 ,
       -0.01170443,  0.00107773, -0.0057198 ,  0.01610866,  0.14114028,
       -0.12392241,  0.09768501, -0.04680169,  0.01863543, -0.03055893,
        0.04326105, -0.00352417, -0.01130702,  0.0012058 , -0.00545938,
        0.01586587,  0.13811194, -0.12098591,  0.09561737, -0.04560901,
        0.01835431, -0.02973223,  0.04242403, -0.00330596, -0.01091398,
        0.00131773, -0.00519829,  0.01564556,  0.13513728, -0.11811956])
```

En la figura 107 se crea en rango de fechas para las predicciones.

*Figura 107. Ejecución de código para crear el rango de fechas.*

```
forecast_index = pd.date_range(start=df_sarima.index[-1] + pd.DateOffset(months=1), periods=periods, freq='MS')
forecast_df = pd.DataFrame(index=forecast_index)
forecast_df['sarima_prediction_diff'] = sarima_prediction
forecast_df['sarima_prediction_diff'].head()
```

```
2020-03-01    0.129290
2020-04-01   -0.044430
2020-05-01    0.014861
2020-06-01   -0.028628
2020-07-01    0.046086
Freq: MS, Name: sarima_prediction_diff, dtype: float64
```

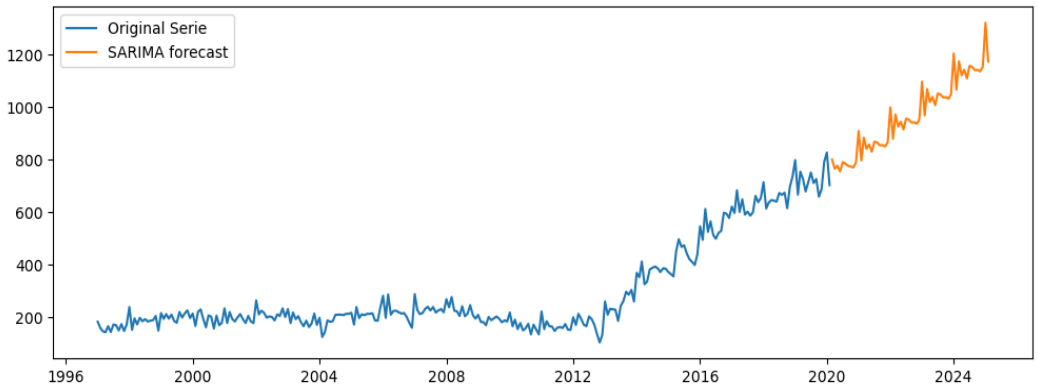
En la figura 108 se invierte la transformación y la diferenciación para obtener valores originales y en el gráfico 48 se visualizan las predicciones.

*Figura 108. Ejecución de código para revertir la transformación y diferenciación.*

```
forecast_df['sarima_forecast_boxcox'] = forecast_df['sarima_prediction_diff'].cumsum()
forecast_df['sarima_forecast_boxcox'] = forecast_df['sarima_forecast_boxcox'].add(serie_transformada[278-1])
forecast_df['sarima_forecast'] = np.exp(forecast_df['sarima_forecast_boxcox'])
forecast_df['sarima_forecast'].head()
```

```
2020-03-01    801.166038
2020-04-01    766.349408
2020-05-01    777.823552
2020-06-01    755.871807
2020-07-01    791.521871
Freq: MS, Name: sarima_forecast, dtype: float64
```

*Gráfico 48. Pronóstico SARIMA, valores futuros.*



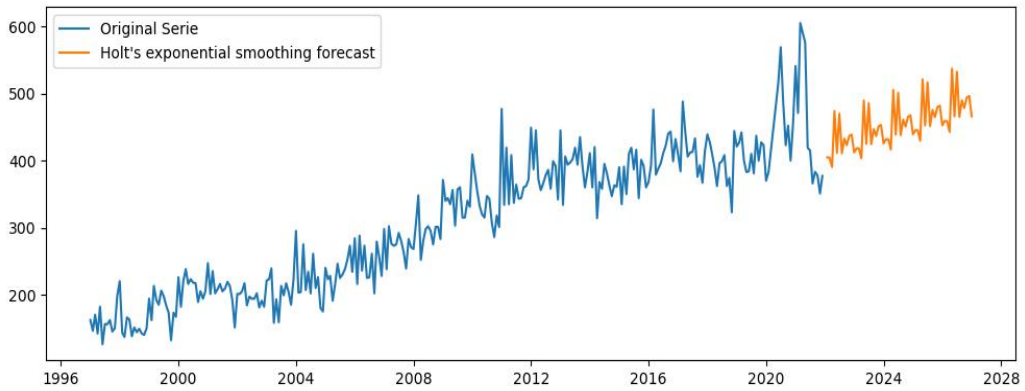

## b. Causa de Muerte Enfermedades Diabetes Mellitus

Para el caso de la causa de muerte por diabetes mellitus se utiliza el método de suavizado exponencial de Holt, para ello se define el número de periodos que se desea predecir utilizando el modelo previamente ajustado, crear un índice para las fechas futuras, se crea un data frame para almacenar las predicciones en base al método Holt con tendencia y se combinan las predicciones con la base de datos de prueba. Este proceso se detalla en la figura 109, finalmente se visualizan las predicciones en el gráfico 49.

*Figura 109. Ejecución de código para definir el número de periodos usando el modelo previo.*

```
forecast_periods_d = 60
forecast_values_d = model_fit_d.forecast(steps=forecast_periods_d)
future_index_d = pd.date_range(start=testd.index[-1] + pd.DateOffset(months=1), periods=forecast_periods_d, freq='M')
forecast_df_d = pd.DataFrame(forecast_values_d, index=future_index_d, columns=['holt_forecast_d'])
y_hat_holt_d = testd.copy()
y_hat_holt_d = pd.concat([y_hat_holt_d, forecast_df_d])
```

*Gráfico 49. Pronóstico usando el método de suavizado exponencial de Holt, incluyendo 60 periodos.*




## c. Causa de Muerte Enfermedades Cerebrovasculares

Para el desarrollo de las predicciones de la causa de muerte de enfermedades cerebrovasculares se utiliza el modelo autorregresivo SARIMA, el cual posee las mejores métricas de evaluación con respecto al resto de modelos, para el efecto se procede a ejecutar la transformación Box Cox y se aplica la diferenciación a la serie de tiempo y se aplica la prueba de Dickey Fuller, para comprobar estacionariedad, esto se detalla en la figura 110.

*Figura 110. Prueba de estacionariedad, causa de muerte enfermedades cerebrovasculares.*

```
adfuller_test(serie_transformada_diferenciada)
```

```
ADF Test Statistic : -3.732640677713971
p-value : 0.0036767683929460316
#Lags Used : 13
Number of Observations Used : 263
Evidencia fuerte contra la hipótesis nula (Ho), rechace la hipótesis nula. Los datos no tienen raíz unitaria y son estacionario
s.
```

En la figura 111 se buscan los parámetros para el mejor modelo SARIMA.

*Figura 111. Ejecución código modelo SARIMA, causa de muerte enfermedades cerebrovasculares.*

```
best_model_sarima_c= auto_arima(serie_transformada_diferenciada_c,trace=True, error_action='ignore',
                        start_p=0,d=0,start_q=0, max_p=3,max_d=2,max_q=3, m=12,
                        start_P=0, D=0, start_Q=0, max_P=3,max_Q=3, max_D=2,
                        suppress_warnings=True,stepwise=False,seasonal=True)
```

```
 ARIMA(2,0,1)(1,0,1)[12] intercept   : AIC=inf, Time=1.38 sec
 ARIMA(2,0,1)(2,0,0)[12] intercept   : AIC=-486.085, Time=2.35 sec
 ARIMA(2,0,2)(0,0,0)[12] intercept   : AIC=-480.168, Time=0.48 sec
 ARIMA(2,0,2)(0,0,1)[12] intercept   : AIC=-478.109, Time=1.12 sec
 ARIMA(2,0,2)(1,0,0)[12] intercept   : AIC=-467.831, Time=1.11 sec
 ARIMA(2,0,3)(0,0,0)[12] intercept   : AIC=-477.066, Time=0.58 sec
 ARIMA(3,0,0)(0,0,0)[12] intercept   : AIC=-451.115, Time=0.38 sec
 ARIMA(3,0,0)(0,0,1)[12] intercept   : AIC=-453.729, Time=0.37 sec
 ARIMA(3,0,0)(0,0,2)[12] intercept   : AIC=-454.270, Time=0.87 sec
 ARIMA(3,0,0)(1,0,0)[12] intercept   : AIC=-454.846, Time=0.89 sec
 ARIMA(3,0,0)(1,0,1)[12] intercept   : AIC=inf, Time=1.29 sec
 ARIMA(3,0,0)(2,0,0)[12] intercept   : AIC=-458.309, Time=1.77 sec
 ARIMA(3,0,1)(0,0,0)[12] intercept   : AIC=inf, Time=0.50 sec
 ARIMA(3,0,1)(0,0,1)[12] intercept   : AIC=-482.638, Time=1.13 sec
 ARIMA(3,0,1)(1,0,0)[12] intercept   : AIC=-483.633, Time=1.52 sec
 ARIMA(3,0,2)(0,0,0)[12] intercept   : AIC=-480.370, Time=0.70 sec

Best model:  ARIMA(1,0,1)(1,0,1)[12] intercept
Total fit time: 155.845 seconds
```

Se entrena el modelo, en función de los resultados de la búsqueda del modelo SARIMA, este proceso se detalla en la figura 112.

*Figura 112. Ejecución de código para entrenamiento del modelo SARIMA, causa de muerte enfermedades cerebrovasculares.*

```
best_model_sarima_c = best_model_sarima_c.fit(serie_transformada_diferenciada_c)
best_model_sarima_c
```

```
ARIMA(order=(1, 0, 1), scoring_args={}, seasonal_order=(1, 0, 1, 12),
      suppress_warnings=True)
```

En la figura 113 se calculan las predicciones para los siguientes 60 meses.

*Figura 113. Ejecución de código para predicciones futuras, causa de muerte enfermedades cerebrovasculares.*

```
periods_c = 60
sarima_prediction_c = best_model_sarima_c.predict(periods_c)
sarima_prediction_c
```

```
array([ 0.02768572,  0.00527786,  0.09036849, -0.10718205,  0.0809571 ,
       -0.05785014,  0.04336945, -0.04874284,  0.04587799,  0.00945896,
       -0.04527119,  0.02705397,  0.00469094, -0.0048219 ,  0.08257204,
       -0.10516694,  0.07710556, -0.05603876,  0.04165148, -0.04693564,
        0.04421816,  0.00917308, -0.04351472,  0.02612868,  0.00459819,
       -0.00456044,  0.07958811, -0.1011781 ,  0.07432493, -0.05387441,
        0.04018765, -0.04510935,  0.04265902,  0.00891551, -0.04181548,
        0.02524138,  0.00451053, -0.00430796,  0.07671534, -0.09733726,
        0.07164763, -0.05179042,  0.03877821, -0.04335089,  0.04115779,
        0.0086675 , -0.04017935,  0.02438704,  0.00442613, -0.00406484,
        0.07394927, -0.09363906,  0.06906977, -0.04978382,  0.03742111,
       -0.04165773,  0.03971231,  0.00842871, -0.03860398,  0.02356443])
```

Se crea en rango de fecha para las predicciones, en la figura 114.

*Figura 114. Ejecución de código para crear el rango de fechas, causa de muerte enfermedades cerebrovasculares.*

```
forecast_index_c = pd.date_range(start=df_sarima_c.index[-1] + pd.DateOffset(months=1), periods=periods_c, freq='MS')
forecast_df_c = pd.DataFrame(index=forecast_index_c)
forecast_df_c['sarima_prediction_diff_c'] = sarima_prediction_c
forecast_df_c['sarima_prediction_diff_c'].head()
```

```
2022-01-01    0.027686
2022-02-01    0.005278
2022-03-01    0.090368
2022-04-01   -0.107182
2022-05-01    0.080957
Freq: MS, Name: sarima_prediction_diff_c, dtype: float64
```

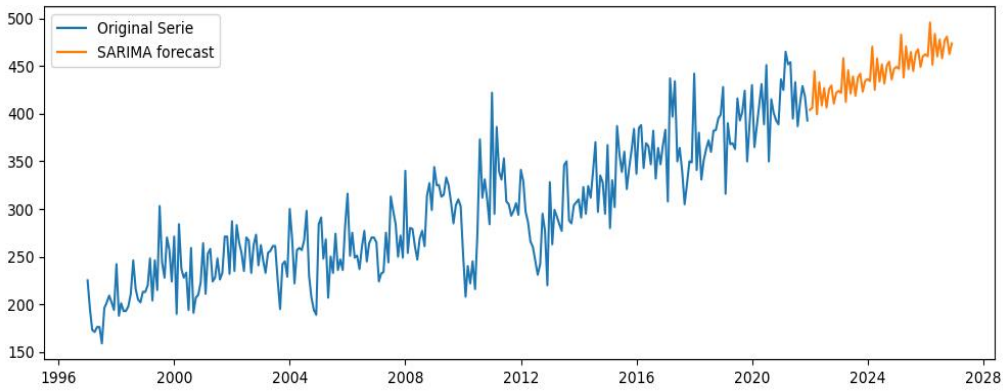
Se invierte la transformación y la diferenciación para obtener valores originales, esto se detalla en la figura 115, y finalmente se muestran las predicciones de los 60 meses en el gráfico 50.

*Figura 115. Ejecución de código para revertir la transformación y diferenciación, causa de muerte enfermedades cerebrovasculares.*

```
forecast_df_c['sarima_forecast_boxcox_c'] = forecast_df_c['sarima_prediction_diff_c'].cumsum()
forecast_df_c['sarima_forecast_boxcox_c'] = forecast_df_c['sarima_forecast_boxcox_c'].add(serie_transformada_c[298-1])
forecast_df_c['sarima_forecast_c'] = np.exp(forecast_df_c['sarima_forecast_boxcox_c'])
forecast_df_c['sarima_forecast_c'].head()
```

```
2022-01-01    404.032504
2022-02-01    406.170570
2022-03-01    444.585189
2022-04-01    399.398490
2022-05-01    433.077517
Freq: MS, Name: sarima_forecast_c, dtype: float64
```

*Gráfico 50. Pronóstico SARIMA, valores futuros, causa de muerte enfermedades cerebrovasculares.*




## 6.2. Clustering

Los resultados del análisis de clusters revelan tres grupos distintos en el conjunto de datos

**Cluster 0:**

Este cluster está compuesto por 58.224 instancias, el género más frecuente es el masculino, la provincia de muerte que predomina es la provincia del Guayas, el nivel de educación más común entre las instancias es “Ninguno”, la causa de muerte más frecuente son las provenientes de afecciones originadas en el período prenatal, adicionalmente el rango de edad más común pertenece a los niños (0 a 12 años de edad).

Este grupo se caracteriza por la mortalidad infantil ocurrida en la provincia de Guayas, con una alta incidencia en los niños de sexo masculino que no poseen ningún nivel de instrucción, probablemente debido a su corta edad.

**Cluster 1:**

Este cluster comprende 171.474 instancias, el género más común es el masculino, la provincia de muerte más frecuente es la provincia del Guayas, el nivel de educación más común entre las instancias es la educación primaria, la causa de muerte más frecuente es

diabetes mellitus, adicionalmente el rango de edad más común pertenece a los adultos de mediana edad.

Este grupo se caracteriza por la mortalidad ocurrida en la provincia del Guayas a partir de una de las principales causas de muerte en este caso diabetes mellitus, afectando a los adultos de género masculino, con un nivel de instrucción de primaria, que se hallan en un rango de edad de 40 a 59 años de edad.

**Cluster 2:**

Este cluster comprende un total de 65.428 instancias, predomina el género masculino, la provincia de muerte más frecuente es la provincia del Guayas, el nivel de educación más habitual es la primaria, además la causa de muerte más común es la enfermedad por virus de la inmunodeficiencia (VIH), finalmente el rango de edad más común son los adultos jóvenes.

Este grupo se caracteriza por la mortalidad en los adultos jóvenes de género masculino, con un nivel de instrucción de primaria y que se hallan en un rango de edad de 20 a 39 años de edad, al igual que los otros clusters, el mayor número de muertes se presenta en la provincia del Guayas.

En términos generales, estos tres grupos revelan patrones únicos sobre la mortalidad en Ecuador, con variaciones significativas en los grupos de edad, causas de muerte y características demográficas, de modo que, la provincia del Guayas se destaca como un factor común en todos los grupos, lo que indica un énfasis regional en estos patrones de mortalidad, adicionalmente el nivel de instrucción más frecuente es la primaria y el género más afectado es el masculino.

### 6.2.1. Despliegue

Los resultados del análisis de clusters revelan tres grupos distintos en el conjunto de datos, lo cual puede utilizarse para generar una nueva base de datos actualizada que permita generar modelos predictivos, y en función de las características y patrones de mortalidad, las autoridades estatales pueden tomar decisiones informadas para diseñar y ejecutar políticas públicas más efectivas y dirigidas a las necesidades específicas de cada grupo demográfico.

## 7. Conclusiones

- A través de la aplicación de métodos de series de tiempo a las principales causas de muerte en Ecuador, la evaluación y selección de los modelos de pronóstico, se ha logrado cumplir los objetivos establecidos, además se ha analizado la evolución y las principales causas de la mortalidad, identificado tendencias y periodos estacionales, finalmente a través de la aplicación de la técnica de agrupación k-modes se identificaron patrones en grupos específicos de la población.
- Mediante la implementación de los métodos de series de tiempo a las tres principales causas de muerte en el Ecuador: Enfermedades isquémica del corazón, diabetes mellitus y enfermedades cerebrovasculares, se identificaron tendencias crecientes y la existencia de periodos estacionales, lo que proporciona información crucial sobre la evolución de la mortalidad en el país.
- En función de la aplicación de los métodos de predicción se series de tiempo como suavizamiento exponencial Holt con tendencia, método multiplicativo de Holt Winters con tendencia y estacionalidad, ARIMA y SARIMA se concluye lo siguiente: para enfermedades isquémicas del corazón, el método SARIMA es el más preciso, en el caso de diabetes mellitus, el método de suavizado exponencial de Holt con tendencia ofrece las predicciones más precisas, finalmente para enfermedades cerebrovasculares, aunque hay discrepancias en algunas métricas, el modelo SARIMA generalmente proporciona las predicciones más precisas y explica mejor la variabilidad en los datos.
- El modelo SARIMA muestra un buen desempeño en dos de las tres causas de muerte evaluadas, proporcionando las predicciones más precisas y explicando una mayor proporción de la variabilidad en los datos en comparación con otros modelos. Para el caso de causa de muerte de enfermedades isquémicas del corazón el método SARIMA logra capturar las tendencias y variaciones en los datos, sin embargo las predicciones se ubican por debajo de los datos reales, en el caso de la diabetes mellitus, el método de suavizamiento exponencial de Holt con tendencia es capaz de capturar el comportamiento creciente pero no la variabilidad, finalmente para la causa de muerte relacionada con las enfermedades cerebrovasculares el método SARIMA no logra capturar los picos altos o bajos en las predicciones, y los resultados predichos para estos picos son inferiores a los valores reales.

- Comparando los resultados de las métricas evaluadas para las tres causas de muerte, se puede mencionar que en la causa de muerte de enfermedades cerebrovasculares se registran los valores más bajos para MSE, RMSE, MAPE y MAE, lo cual indica una mayor precisión entre las predicciones y los valores reales.
- Los clusters identificados muestran patrones sobre la distribución en términos de género, nivel de instrucción, provincia de fallecimiento, rango de edad y causa de muerte en el Ecuador. En el cluster 0, se identifica la incidencia de la mortalidad en la provincia del Guayas en niños de sexo masculino, los cuales no poseen nivel de instrucción alguno, para el cluster 1 la mortalidad afecta a los adultos de mediana edad, de sexo masculino, con un nivel de educación primaria, cuya causa de muerte más común es la diabetes mellitus, y nuevamente la mayor parte de muertes ocurre en la provincia del Guayas. Por otro lado, el cluster 2, está compuesto por adultos jóvenes hombres con educación primaria, en donde la causa de muerte más común es la enfermedad por virus de la inmunodeficiencia (VIH) y predomina como lugar de ocurrencia la provincia del Guayas.
- Los grupos identificados a través de los clusters destacan variaciones notables en la edad, causas de muerte y características demográficas, siendo la provincia del Guayas un factor común de ocurrencia de muertes, lo que sugiere un énfasis regional en estos patrones de mortalidad. Además, se observa que el género masculino es el más afectado y que la educación primaria es el nivel más común entre los fallecidos. Finamente un patrón relevante identificado es la incidencia de la mortalidad infantil a partir de la causa de muerte proveniente de afecciones originadas en el período prenatal. Estos hallazgos resaltan la importancia de abordar las enfermedades a través de programas y políticas públicas efectivas enfocadas en los grupos poblacionales antes descritos.

## 8. Recomendaciones

- Realizar un monitoreo de manera continua a las tendencias registradas en el comportamiento de las principales causas de mortalidad en Ecuador utilizando métodos de series de tiempo, lo cual contribuirá a la detección temprana de cambios significativos en los patrones de mortalidad y facilitará la planificación gubernamental y por ende la aplicación de políticas o programas de salud preventivos.
- Realizar una validación regular de los modelos de predicción utilizando datos actualizados, para garantizar la precisión y fiabilidad de las predicciones a medida que cambian las condiciones y los factores que influyen en la mortalidad por estas enfermedades.
- Es importante integrar los resultados de las predicciones en la planificación y desarrollo de políticas de salud pública, estos insumos pueden proporcionar información valiosa para la asignación de recursos, la implementación de programas de prevención, y para la toma de decisiones informada.
- Aunque el modelo SARIMA fue el más preciso en la mayoría de los casos, podría ser útil explorar otro métodos y técnicas de series de tiempo como LSTM, Random Forest, Prohet, entre otros, también se podrían utilizar modelos incluyendo variables exógenas que afecten a cada una de las enfermedades, estos modelos pueden permitir identificar factores clave para la ejecución de políticas y programas de salud.
- En función de los patrones identificados es necesario examinar más a fondo las características de cada cluster identificado, con la finalidad de comprender mejor las diferencias y similitudes entre ellos. Además del algoritmo utilizado previamente, se puede considera explorar otras técnicas de clustering, se pueden incorporar variables adicionales relacionadas a las condiciones socioeconómicas.
- Dado que los clusters identificados muestran patrones específicos en términos de género, nivel educativo, provincia de fallecimiento, rango de edad y causa de muerte, es crucial implementar políticas públicas enfocadas en el desarrollo de programas de intervención que aborden las necesidades particulares de estos grupos, específicamente programas de salud pública enfocados en la prevención, detección temprana y manejo de afecciones originadas en el período prenatal y de la diabetes mellitus en la provincia del Guayas.

## BIBLIOGRAFÍA


Alba, F., & Morelos, J. (2008). *Población y grandes tendencias demográficas en América Latina y El Caribe*. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=c945ea8dc9ba34b0cb4fe1ce286f51090f0255f4

Arriagada, Irma., Miranda, Francisca., Aranda, V., & United Nations. Economic Commission for Latin America and the Caribbean. Social Development Division. (2005). *Políticas y programas de salud en América Latina : programas y propuestas*. Comisión Económica para América Latina y el Caribe, División de Desarrollo Social.

Atif, M., Farooq, M., Shafiq, M., Ayub, G., & Ilyas, M. (2023). The impact of partner's behaviour on pregnancy related outcomes and safe child-birth in Pakistan. *BMC Pregnancy and Childbirth*, *23*(1). https://doi.org/10.1186/s12884-023-05814-z

Banco Mundial. (n.d.). *Tasa de mortalidad en un año (por cada 1.000 personas)*. Retrieved August 31, 2023, from https://datos.bancomundial.org/indicador/SP.DYN.CDRT.IN

Battaglia, O. R., Paola, B. Di, & Fazio, C. (2016). A New Approach to Investigate Students' Behavior by Using Cluster Analysis as an Unsupervised Methodology in the Field of Education. *Applied Mathematics*, *07*(15), 1649–1673. https://doi.org/10.4236/am.2016.715142

Bello, L., & Martínez, S. (2007). *Una metodología de series de tiempo para el área de la salud; caso práctico*. *25*, 117–122.

Bhatt, C. M., Patel, P., Ghetia, T., & Mazzeo, P. L. (2023). Effective Heart Disease Prediction Using Machine Learning Techniques. *Algorithms*, *16*(2). https://doi.org/10.3390/a16020088

Bi, Q., Goodman, K. E., Kaminsky, J., & Lessler, J. (2019). What is machine learning? A primer for the epidemiologist. *American Journal of Epidemiology*, *188*(12), 2222–2239. https://doi.org/10.1093/aje/kwz189

Carracedo, P. (2017). *Metodología espacio-temporal con datos de panel. Estudio de la mortalidad europea*. https://riunet.upv.es/handle/10251/89080?show=full

Castañeda, C. D., & González, V. M. (2014). *Estudio diagnóstico de la mortalidad humana en el Ecuador, en la Provincia de Tungurahua, el Cantón capital Ambato y los principales cantones con el mayor porcentaje de mortalidad. 3*. https://dialnet.unirioja.es/servlet/articulo?codigo=6756301

Chugh, A. (2020). *MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better?* https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e

Corres, G., Esteban, A., García, J. C., & Zárate, C. (2009). *Análisis de series temporales*. https://dialnet.unirioja.es/descarga/articulo/3998101.pdf

Cristancho, C. (2017). *Niveles, tendencias y determinantes de la mortalidad reciente en Colombia*.

Daza, K., & García, M. (2021). *Predicción, análisis y pronóstico de COVID-19 utilizando un modelo de machine learning basado en el análisis forecasting sobre series temporales.*

El Amrani, E. H. (2020). *Análisis de series temporales: Uso del transporte público en Barcelona.* [Universidad Politécnica de Cataluña]. https://upcommons.upc.edu/bitstream/handle/2117/340448/Memoria_Ana%CC%81lisis%20de%20series%20temporales.pdf?sequence=1&isAllowed=y

Florencia, M. (2018). *Exploración y análisis exhaustivo de los métodos estadísticos aplicados a las series de tiempo.* https://fce.unl.edu.ar/jornadasdeinvestigacion/trabajos/uploads/trabajos/45.pdf

Gambini, M., & López, J. (2018). *Análisis de Series de Tiempo*. https://ri.itba.edu.ar/server/api/core/bitstreams/46f84c43-9130-49c0-bf7e-4b41ecb051ef/content

Giraldo, D., Atehortúa, A., García-Arteaga, J. D., Romero, E., & Rodríguez, J. (2017). Modelo para el análisis de la mortalidad en Colombia 2000-2012. *Revista de Salud Pblica*, *19*(2), 241–248. https://doi.org/10.15446/rsap.v19n2.66239

Gratsos , K., Ougiaroglou , S., & Margaris , D. (2023). kClusterHub: An AutoML-Driven Tool for Effortless Partition-Based Clustering over Varied Data Types. *Future Internet*, *15*(10). https://doi.org/10.3390/fi15100341

Guerrero, M., & Medina, S. (2016). Modelos de series de tiempo. *Science Direct*, *398*, 89–99.

Hinestroza, D. (2018). *El machine learning a través de los tiempos y los aportes a la humanidad.* https://repository.unilibre.edu.co/bitstream/handle/10901/17289/EL%20MACHINE%20LEARNING.pdf?sequence=1&isAllowed=y

IBM. (n.d.). *¿Qué es machine learning?* Retrieved November 14, 2023, from https://www.ibm.com/es-es/topics/machine-learning

INEC. (n.d.). *Defunciones Generales*. Retrieved July 12, 2023, from https://www.ecuadorencifras.gob.ec/defunciones-generales/

INEC. (2022). *Boletín Técnico. Registro Estadístico de Defunciones Generales*. www.ecuadorencifras.gob.ec

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Springer*, *31*, 685–695. https://doi.org/10.1007/s12525-021-00475-2/Published

López, S. (2007). *Algoritmos de agrupamiento global para datos mezclados*. https://inaoe.repositorioinstitucional.mx/jspui/bitstream/1009/628/1/LopezES.pdf

Mackenbach, J. P., Bopp, M., Deboosere, P., Kovacs, K., Martikainen, P., Menvielle, G., Regidor, E., & De Gelder, R. (2017). *Determinants of the magnitude of socioeconomic inequalities in mortality: A study of 17 European countries*. https://doi.org/10.1016/j.healthplace.2017.07.005

Mahesh, B. (2018). Machine Learning Algorithms-A Review. *International Journal of Science and Research*. https://doi.org/10.21275/ART20203995

Mathonsi, T., & van Zyl, T. L. (2022). A Statistics and Deep Learning Hybrid Method for Multivariate Time Series Forecasting and Mortality Modeling. *Forecasting*, *4*(1), 1–25. https://doi.org/10.3390/forecast4010001

Mattiev, J., Davityan, M., & Kavsek, B. (2023). ACMKC: A Compact Associative Classification Model Using K-Modes Clustering with Rule Representations by Coverage. *Mathematics*, *11*(18). https://doi.org/10.3390/math11183978

Mishra, S., Singh, T., Kumar, M., & Satakshi. (2023). Multivariate time series short term forecasting using cumulative data of coronavirus. *Evolving Systems*. https://doi.org/10.1007/s12530-023-09509-w

Moine, J., Haedo, A., & Gordillo, S. (2011). *Estudio comparativo de metodologías para minería de datos*.

Mustapha, M. F., Zulkifli, A. N. I., Kairan, O., Zizi, N. N. S. M., Yahya, N. N., & Mohamad, N. M. (2023). The prediction of student's academic performance using RapidMiner. *Indonesian Journal of Electrical Engineering and Computer Science*, *32*(1), 363–371. https://doi.org/10.11591/ijeecs.v32.i1.pp363-371

Naciones Unidas. (1978). *Mortalidad factores determinantes y consecuencias de las tendencias demográficas: Vol. I.* https://repositorio.cepal.org/server/api/core/bitstreams/db97144e-00c5-4bba-a619-e73371918a05/content

Naciones Unidas. (2014). *Principios y recomendaciones para un sistema de estadísticas vitales*. http://unstats.un.org/unsd/

Navarro, O., & Alencastre, M. (2016). DBSCAN modificado con Octrees para agrupar nubes de puntos en tiempo real. *Research in Computing Science*, *114*(1), 173–186. https://doi.org/10.13053/rcs-114-1-14

Nissa, N. K. (2020). *Clustering Method using K-Means, Hierarchical and DBSCAN (using Python) | by Nuzulul Khairu Nissa | Medium*. Medium. https://nzlul.medium.com/clustering-method-using-k-means-hierarchical-and-dbscan-using-python-5ca5721bbfc3

Ordoñez, J. (2023). *Modelo predictivo aplicando algoritmos de machine learning para la producción lechera en la hacienda el Prado del InstItuto Agropecuario Superior Andino (IASA)*. https://repositorio.puce.edu.ec/server/api/core/bitstreams/e9d4e417-0ba9-4752-a250-9b257b6dcc1a/content

Organización Panamericana de la Salud. (2017). *Lineamientos básicos para el análisis de la mortalidad*. www.paho.org

Organización Panamericana de la Salud. (2020). *La OMS revela las principales causas de muerte y discapacidad en el mundo: 2000-2019*. https://www.paho.org/es/noticias/9-12-2020-oms-revela-principales-causas-muerte-discapacidad-mundo-2000-2019

Pathak, P. (2021). *Time Series Forecasting — A Complete Guide*. https://medium.com/analytics-vidhya/time-series-forecasting-a-complete-guide-d963142da33f

Ríos, G., & Hurtado, C. (2008). *Series de tiempo*. https://gc.scalahed.com/recursos/files/r161r/w24113w/Semana%2011/Series_de_Tiempo.pdf

Sánchez, N. (2020). *Estudio comparativo de modelos de predicción estocásticos y heurísticos aplicados a la estimación de la calidad del aire*.

Subramanian, M., Cho, J., Veerappampalayam Easwaramoorthy, S., Murugesan, A., & Chinnasamy, R. (2023). Enhancing Sustainable Transportation: AI-Driven Bike Demand Forecasting in Smart Cities. *Sustainability (Switzerland)*, *15*(18). https://doi.org/10.3390/su151813840

Villarreal, F. (2016). *Introducción a los Modelos de Pronósticos*. https://www.matematica.uns.edu.ar/uma2016/material/Introduccion_a_los_Modelos_de_Pronosticos.pdf

Wang, H., Paulson, K. R., Pease, S. A., Watson, S., Comfort, H., Zheng, P., Aravkin, A. Y., Bisignano, C., Barber, R. M., Alam, T., Fuller, J. E., May, E. A., Jones, D. P., Frisch, M. E., Abbafati, C., Adolph, C., Allorant, A., Amlag, J. O., Bang-Jensen, B., … Murray, C. J. L. (2022). Estimating excess mortality due to the COVID-19 pandemic: a systematic analysis of COVID-19-related mortality, 2020–21. *The Lancet*, *399*(10334), 1513–1536. https://doi.org/10.1016/S0140-6736(21)02796-3

Zhao, R., Liu, J., Zhao, Z., Zhai, M., Ren, H., Wang, X., Li, Y., Cui, Y., Qiao, Y., Ren, J., Chen, L., & Qiu, L. (2023). A hybrid model for tuberculosis forecasting based on empirical mode decomposition in China. *BMC Infectious Diseases*, *23*(1). https://doi.org/10.1186/s12879-023-08609-x