ANÁLISIS PREDICTIVO DE CASOS DE COVID-19 BASADO EN CONDICIONES METEOROLÓGICAS

DANILO PLAZAS IRREÑO
DNI: 1024538287

UNIVERSIDAD INTERNACIONAL DE VALENCIA
FACULTAD ESCUELA DE CIENCIA Y TECNOLOGÍA
MÁSTER EN BIG DATA Y CIENCIA DE DATOS
BOGOTÁ D.C.
2022

ANÁLISIS PREDICTIVO DE CASOS DE COVID-19 BASADO EN CONDICIONES METEOROLÓGICAS

DANILO PLAZAS IRREÑO
viudanilo0221p@gmail.com
DNI: 1024538287

Trabajo de máster para optar al titulo de:
Máster en Big Data y Ciencia de Datos

DIRECTOR:
MSc. BENJAMÍN ARROQUIA CUADROS
Docente Universidad Internacional de Valencia

UNIVERSIDAD INTERNACIONAL DE VALENCIA
FACULTAD ESCUELA DE CIENCIA Y TECNOLOGÍA
MÁSTER EN BIG DATA Y CIENCIA DE DATOS
BOGOTÁ D.C.
2022

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

Tras la aparición del Covid-19 a nivel mundial a comienzos del 2020, han surgido varios estudios para identificar los factores que influyen en la propagación del virus, en donde el contacto cercano con personas infectadas es el principal factor, esto se debe a temas sociales como la movilidad, eventos, recreación, deporte, entre otros. En la actualidad no existen muchos estudios acerca de los factores climáticos, para este proyecto de TFM se sigue como hipótesis que las condiciones meteorológicas influyen en menor medida en la propagación del virus Covid-19. En este proyecto se seguirá una metodología KDD persiguiendo los pasos habituales para la obtención de conocimiento y entendimiento de los datos.

Se lleva a cabo un análisis de las bases de datos disponibles, así como la preparación y compresión de los datos y limpieza de los mismos. Los datos utilizados para este TFM han sido recopilados del Centro Nacional de Epidemiología (CNE) para los casos de Covid-19 reportados, los datos meteorológicos proporcionados por el portal datos abiertos AEMET, los datos de población proporcionados por el Instituto Nacional de Estadística (INE). Con estos datos se ha generado un único dataset para la realización de los modelos de machine learning (regresión lineal múltiple, RandomForest Regresión, series temporales y K-means) para predecir o encontrar patrones sobre la tasa de incidencia (normalización de los casos de Covid-19) por provincia con relación a los factores climáticos durante la pandemia.

Los resultados obtenidos con los distintos modelos se contrastaron basado en nuestras métricas definidas (R^2, MAE y RMSE), y se obtuvieron valores bajos en estas métricas, este resultado es el esperado, ya que las condiciones meteorológicas no son el factor principal en la propagación del virus, las variables no logran explicar nuestra variable a predecir. Sin embargo, el modelo de K-means evidencia una influencia de los factores climáticos sobre la tasa de incidencia, mediante una segmentación de los casos en clusters, en donde los clúster con ciertas características meteorológicas presentara una tasa de incidencia diferente.

Palabras clave: Covid-19, machine learning, KDD y clúster.

1. INTRODUCCIÓN

El COVID-19 es una enfermedad respiratoria causada por el virus SARS-CoV-2. Desde su aparición en Wuhan, China a finales de 2019, ha afectado a millones de personas en todo el mundo. Los gobiernos de todo el mundo han implementado diversas medidas para prevenir la propagación del virus y proteger la salud pública. (Ciotti y cols., 2020)

Algunas de las medidas más comunes fueron: cierre de fronteras, distanciamiento social (como el cierre de escuelas, lugares de trabajo y eventos públicos), uso de mascarillas, pruebas y rastreo de contactos (para identificar a las personas infectadas y rastrear a aquellos con los que han tenido contacto cercano), cierre de empresas y restricciones de actividades no esenciales (para reducir la cantidad de personas que se congregan en lugares públicos), campañas de concientización y educación pública. Todas estas medidas ayudan a mitigar la propagación del virus y aunque la transmisión del virus se produce principalmente por contacto cercano con personas infectadas, se ha investigado sobre la posible influencia de las condiciones meteorológicas en la propagación del virus. (Wang, Tang, Feng, Lv, y cols., 2020)

En general, se cree que el clima cálido y húmedo puede reducir la propagación del virus, ya que el calor y la humedad pueden debilitar la capacidad del virus para sobrevivir en el aire y en las superficies. Sin embargo, los expertos señalan que no hay suficiente evidencia científica para afirmar que las altas temperaturas y la humedad reducen significativamente la transmisión del virus. Por otro lado, el invierno y el clima frío pueden aumentar la transmisión del virus, ya que las personas tienden a pasar más tiempo en espacios cerrados y con poca ventilación, lo que facilita la propagación del virus de persona a persona. (Araujo y Naimi, 2020)

En este proyecto se desarrollará un estudio y análisis sobre el impacto de las condiciones meteorológicas en la propagación del virus covid-19 en España y determinar si existe algún factor relacionado con la transmisión.

2. OBJETIVOS

2.1. OBJETIVO GENERAL

- Identificar la posible influencia de las condiciones meteorológicas en la propagación del virus de COVID-19 en España.

2.2. OBJETIVOS ESPECÍFICOS

- Extraer, transformar y obtener conocimiento de las diferentes fuentes de información o bases de datos de COVID-19 en España por provincia, centrándonos en características meteorológicas.
- Crear, comparar y contrastar los diferentes modelos de predicción y/o clustering sobre el COVID-19 en España por provincia.
- Seleccionar el modelo que predice o explica lo más exacto posible la influencia de las características meteorológicas en los casos de virus de COVID-19.
- Automatizar procesos con datos manteniendo la metodología implementada en el desarrollo del proyecto.

3. MACHINE LEARNING, ESTUDIOS Y MODELOS ACTUALES DE COVID-19

El Machine Learning es una técnica de Inteligencia Artificial que permite a los sistemas informáticos aprender de manera automática a partir de datos y experiencias previas, sin ser programados explícitamente para cada tarea. En lugar de seguir un conjunto fijo de instrucciones, los sistemas de Machine Learning pueden aprender a partir de datos, identificando patrones y tendencias, y utilizando esta información para realizar predicciones o tomar decisiones.

El aprendizaje automático se basa en la idea de que los sistemas informáticos pueden aprender de manera similar a como lo hacen los seres humanos, mediante la identificación de patrones y la adaptación a nuevas situaciones. En lugar de requerir que se programen todas las posibles situaciones y resultados, también nos permite que los sistemas aprendan a partir de datos históricos y experiencias previas, y así puedan tomar decisiones informadas y precisas en tiempo real. Programar computadoras para aprender de la experiencia eventualmente debería eliminar la necesidad de gran parte de este esfuerzo de programación detallado. Conforme a la definición de ML de Tom M. Mitchell: "Se dice que un programa de computadora aprende de la experiencia E con respeto a alguna clase de tareas T y medida de rendimiento P, si su rendimiento en tareas en T, medido por P, mejora con la experiencia E". (Awad, Khanna, Awad, y Khanna, 2015) Existen varios tipos de técnicas de Machine Learning, incluyendo el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje por refuerzo. En el aprendizaje supervisado, el sistema aprende a partir de datos etiquetados previamente, mientras que, en el aprendizaje no supervisado, el sistema busca patrones y similitudes en los datos sin etiquetar. En el aprendizaje por refuerzo, el sistema aprende a partir de la retroalimentación del entorno.

Se aplica en una variedad de campos, como la detección de fraudes, la clasificación de imágenes, el análisis de sentimientos y la predicción de ventas, entre otros. A medida que los datos se vuelven cada vez más importantes y abundantes, el Machine Learning se está convirtiendo en una herramienta esencial para empresas e investigadores que buscan automatizar tareas y mejorar la toma de decisiones.

3.1. APRENDIZAJE SUPERVISADO

El aprendizaje supervisado es una técnica de ML que se basa en el uso de datos etiquetados previamente para entrenar un modelo de predicción o clasificación. En el aprendizaje supervisado, el modelo se entrena utilizando un conjunto de datos de entrenamiento que contiene ejemplos de entrada y salida esperada. El objetivo del modelo es aprender una función que pueda predecir la salida correcta para nuevas entradas nunca antes vistas.

Por ejemplo, en la clasificación de correos electrónicos como spam o no spam, el modelo se entrena con una gran cantidad de correos electrónicos etiquetados previamente como spam o no spam. Utilizando esta información, el modelo aprende a identificar patrones en los correos electrónicos que le permiten clasificarlos correctamente. Una de las principales ventajas del aprendizaje supervisado es que puede proporcionar predicciones precisas y confiables en una variedad de tareas. Sin embargo, el aprendizaje supervisado

también tiene algunas limitaciones. En particular, requiere grandes cantidades de datos etiquetados, lo que puede ser costoso y laborioso en algunos casos. Además, el modelo puede ser susceptible al sobreajuste si se entrena con demasiados datos, lo que significa que se adapta demasiado a los datos de entrenamiento y no generaliza bien a nuevos datos nunca antes vistos. Este a su vez se divide en problemas de regresión y clasificación.

3.1.1. REGRESIÓN

Los problemas de regresión en el aprendizaje supervisado son aquellos en los que se busca establecer una relación funcional entre una variable de entrada (también llamada variable independiente o predictor) y una variable de salida (también llamada variable dependiente o respuesta) que toma valores continuos en lugar de discretos. Es decir, se busca predecir un valor numérico continuo en lugar de una etiqueta o clase discreta. El objetivo de la regresión es encontrar una función que mejor se ajuste a los datos observados, de manera que pueda ser utilizada para predecir el valor de la variable de salida para nuevas observaciones de la variable de entrada.

Sin embargo, los problemas de regresión pueden presentar algunos desafíos, como:

- La presencia de valores atípicos o datos faltantes, que pueden afectar negativamente el ajuste del modelo y la precisión de las predicciones.
- La elección de la función de regresión adecuada y de los parámetros del modelo, que pueden depender de la distribución de los datos y del objetivo de la predicción.
- La evaluación de la calidad del modelo, que puede requerir el uso de medidas de error y de rendimiento específicas para problemas de regresión.

Para superar estos desafíos, se pueden utilizar técnicas de preprocesamiento de datos, selección de características, validación cruzada y ajuste de hiperparámetros, entre otras. Estos modelos pueden ser lineales o no lineales. Los modelos lineales, como la regresión lineal simple o múltiple, buscan establecer una relación lineal entre las variables de entrada y la variable de salida. (Rativa, 2020) Por otro lado, los modelos no lineales, como la regresión polinómica, la regresión logística, regresión de árbol de decisión, random forest o redes neuronales, permiten establecer relaciones no lineales entre las variables.

3.1.1.1. REGRESIÓN LINEAL MÚLTIPLE

Un modelo de regresión lineal múltiple es una técnica estadística que se utiliza para predecir la variable de respuesta (o variable dependiente) en función de dos o más variables predictoras (o variables independientes). Es una extensión del modelo de regresión lineal simple, que solo utiliza una variable predictora. La ecuación para un modelo de regresión lineal múltiple se puede escribir como:

$$y = b_o + b_1 x_1 + b_2 x_2 + b_n x_n + e \tag{1}$$

donde:

y : Variable de respuesta (o variable dependiente) que se quiere predecir.
$x_1, x_2, ..., x_n$: Variables predictoras (variables independientes) que se utilizan para predecir y.

$b_0, b_1, b_2, ..., b_n$: Coeficientes de regresión que representan la relación entre cada variable predictora y la variable de respuesta.
e : Error residual o término de error, que representa la variación de y que no se puede explicar por las variables predictoras.

El objetivo de un modelo de regresión lineal múltiple es estimar los coeficientes de regresión $(b_0, b_1, b_2, ..., b_n)$ de tal manera que la suma de los errores residuales sea lo más pequeña posible. Esto se logra mediante el método de mínimos cuadrados, que minimiza la suma de los cuadrados de los errores residuales. Para estimar los coeficientes de regresión, se utilizan técnicas como la matriz de diseño, el cálculo de la matriz inversa y la solución de sistemas de ecuaciones lineales. Una vez que se han estimado los coeficientes de regresión, se puede utilizar el modelo para hacer predicciones sobre la variable de respuesta para nuevos valores de las variables predictoras. (Vannieuwenhuyze, 2020) (Rodrigo, 2016)
Al igual que en el modelo de regresión lineal simple, el modelo de regresión lineal múltiple asume que hay una relación lineal entre las variables predictoras y la variable de respuesta, si la relación no es lineal, puede ser necesario utilizar otro tipo de modelo, por tal motivo es importante evaluar la calidad del modelo mediante técnicas como la validación cruzada y la evaluación de métricas como:

- **MAE (Error Absoluto Medio)**

  Es una métrica que calcula la media de los errores absolutos de las predicciones del modelo en relación con los valores reales de la variable dependiente. Es una medida de la magnitud promedio del error en las predicciones del modelo en términos absolutos. Un MAE bajo indica que las predicciones del modelo tienen un pequeño error promedio en relación con los valores reales de la variable dependiente, lo que indica que el modelo tiene un buen ajuste a los datos, el MAE se calcula mediante:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

  El MAE no considera la dirección del error, es decir, si el modelo está subestimando o sobrestimando los valores reales. Por lo tanto, el MAE debe interpretarse junto con otras métricas de evaluación del modelo, como el error cuadrático medio (MSE) y el coeficiente de determinación (R^2), para obtener una imagen completa del rendimiento del modelo.

- **RMSE (Raiz del Error Cuadratico Medio)**

  Se calcula como la raíz cuadrada del promedio de los errores cuadráticos de las predicciones del modelo en relación con los valores reales de la variable dependiente. El RMSE es similar al MAE, pero penaliza más fuertemente las predicciones que tienen un error mayor, el RMSE es particularmente útil cuando los errores de predicción son importantes y se desea minimizar el error promedio de predicción al cuadrado y igual que el MAE, el RMSE no considera la dirección del error, es decir, si el modelo está subestimando o sobrestimando los valores reales, el MAE se calcula mediante:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3}$$

Sin embargo, el RMSE es más sensible a los valores atípicos que el MAE, lo que significa que puede ser más apropiado en situaciones en las que los valores atípicos tienen un gran impacto en la precisión del modelo.

- R^2 **(Coeficiente de Determinación)**

  Indica la proporción de la varianza total de la variable dependiente (y) que es explicada por las variables independientes $(x_1, x_2, ..., x_n)$ incluidas en el modelo. En otras palabras, R^2 es una medida de qué tan bien las variables predictoras explican la variabilidad en la variable dependiente, se calcula mediante:

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4}$$

  El valor de R^2 varía entre 0 y 1, y se interpreta como sigue:

  $R^2 = 0$: el modelo no explica ninguna variabilidad en la variable dependiente.
  $R^2 = 1$: el modelo explica toda la variabilidad en la variable dependiente.

  En la práctica, R^2 generalmente toma valores intermedios, lo que significa que el modelo explica parte, pero no toda la variabilidad en la variable dependiente. Un valor alto de R^2 indica que el modelo ajusta bien los datos y que las variables predictoras son buenas para predecir la variable dependiente. Sin embargo, un valor alto de R^2 no necesariamente significa que el modelo sea bueno. Puede haber otras variables que no se hayan incluido en el modelo que también puedan explicar la variabilidad en la variable dependiente, es por esto importante evaluar el modelo en función de otras métricas.

3.1.1.2. RANDOM FOREST

El modelo de Random Forest es una extensión del modelo de Bagging (Bootstrap Aggregating) que utiliza múltiples árboles de decisión para mejorar la precisión de la predicción. Bagging es un enfoque de aprendizaje automático que implica la creación de múltiples muestras de entrenamiento a partir del conjunto de datos original, utilizando muestreo con reemplazo. Luego, se entrena un modelo separado para cada muestra y se promedian las predicciones de los modelos para obtener una predicción final, esto ayuda a reducir el sobreajuste y mejorar la precisión de la predicción.
Random Forest utiliza múltiples árboles de decisión para hacer predicciones de valores numéricos, la principal diferencia es que en Random Forest, en cada nodo de decisión se selecciona un subconjunto aleatorio de características para realizar la división en lugar de considerar todas las características disponibles, al seleccionar un subconjunto aleatorio de características, el modelo de Random Forest puede reducir la correlación entre los árboles y aumentar la diversidad de los árboles en el modelo, esto puede mejorar la precisión del modelo y reducir el sobreajuste. (R, 2023)

En Random Forest la idea es que cada árbol de decisión tenga una idea ligeramente diferente de cómo se relacionan las características con la variable objetivo. Para hacer una

predicción se evalúa cada muestra en cada árbol de decisión en el modelo y se toma la media de las predicciones resultantes, se puede expresar matemáticamente como:

$$y = \frac{1}{N} \sum_{i=1}^{n} y_i \tag{5}$$

donde y es la predicción de la variable objetivo para una muestra dada, y_i es la predicción de la variable objetivo para esa muestra en el i-ésimo árbol de decisión, y N es el número total de árboles en el modelo. (Holmäng y von Grothusen, 2021)
Las predicciones que se apartan demasiado de la media no son deseables, ya que pueden estar basadas en un árbol de decisión que tiene una idea atípica de cómo se relacionan las características con la variable objetivo. A continuación, se muestra el modelo de un random forest:

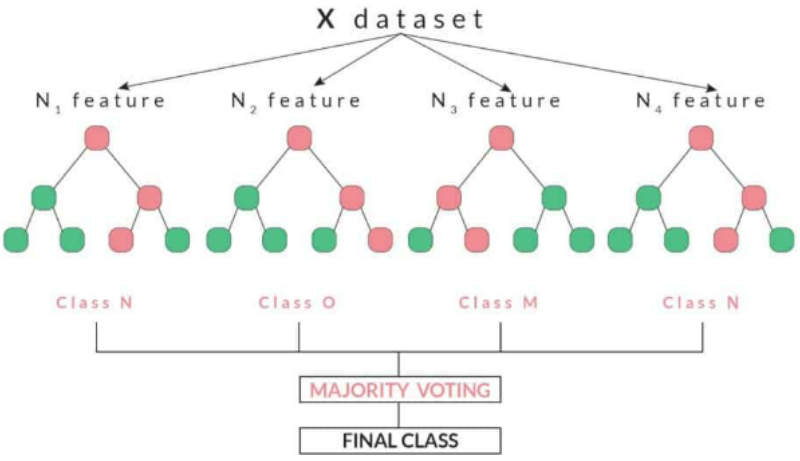

Figura 1: Modelo de funcionamiento random forest.

Fuente: cnvrg.io

El modelo de Random Forest para regresión tiene varias ventajas sobre otros modelos de regresión, incluyendo:

- Es capaz de manejar tanto características numéricas como categóricas.
- Es resistente al sobreajuste, lo que significa que es menos probable que se sobreajuste en comparación con otros modelos de regresión.
- Es capaz de manejar conjuntos de datos grandes y complejos.

Sin embargo, el modelo de Random Forest para regresión también tiene algunas limitaciones, incluyendo:

- Es más difícil de interpretar que algunos otros modelos de regresión.
- Puede ser más computacionalmente costoso que algunos otros modelos de regresión.
- No proporciona información sobre la forma en que se relacionan las características con la variable objetivo.

Las métricas de evaluación comunes para un modelo de Random Forest de regresión son similares a las utilizadas en otros modelos de regresión, como los descritos en el modelo de regresión lineal múltiple.

3.1.2. CLASIFICACIÓN

Los modelos de clasificación binaria son utilizados cuando se desea predecir una variable de salida que puede tomar únicamente dos valores posibles, como "sí" o "no", "verdadero" o "falso", etc. Ejemplos de modelos de clasificación binaria incluyen la regresión logística y la máquina de vectores de soporte. Por otro lado, los modelos de clasificación no binaria son utilizados cuando se desea predecir una variable de salida que puede tomar más de dos valores posibles, como la clasificación de imágenes en diferentes categorías o la predicción de resultados deportivos. Ejemplos de modelos de clasificación no binaria incluyen los árboles de decisión, los bosques aleatorios y las redes neuronales.

Aunque los modelos de clasificación binaria y no binaria utilizan diferentes técnicas y algoritmos, el proceso general de construcción de modelos es el mismo. Se trata de identificar las variables de entrada más importantes, elegir un modelo adecuado, ajustar sus parámetros y evaluar su rendimiento utilizando medidas de evaluación adecuadas. binaria como no binaria. La elección del modelo adecuado dependerá del tipo de datos y del objetivo de la predicción. (Valenzuela González y cols., 2022)

3.2. APRENDIZAJE NO SUPERVISADO

El aprendizaje no supervisado es un tipo de aprendizaje automático en el que el algoritmo se entrena con datos no etiquetados, es decir, sin información previa sobre las categorías a las que pertenecen los datos. A diferencia del aprendizaje supervisado, donde los algoritmos aprenden a partir de datos etiquetados, en el aprendizaje no supervisado, los algoritmos buscan patrones y estructuras en los datos sin ninguna orientación sobre lo que se debe buscar. Se utiliza para descubrir patrones ocultos y estructuras en los datos, como grupos de datos similares, tendencias en los datos y relaciones entre variables. Algunos de los algoritmos de aprendizaje no supervisado más comunes incluyen la agrupación (clustering), la reducción de dimensionalidad y la asociación.

3.2.1. CLUSTERING

El clustering, también conocido como agrupamiento, es una técnica de aprendizaje no supervisado en la que se agrupan datos similares en grupos o clústeres. El objetivo del clustering es dividir un conjunto de datos en grupos, donde los objetos en cada clúster son similares entre sí y diferentes de los objetos en otros clústeres. Esto se logra mediante el uso de medidas de similitud o distancia para medir la distancia entre objetos. Existen varios algoritmos de clustering, cada uno con sus propias fortalezas y debilidades. El algoritmo K-means es uno de los algoritmos más populares y ampliamente utilizados en el clustering. Funciona dividiendo el conjunto de datos en K clústeres y asignando cada objeto al clúster más cercano. Luego, se recalcula el centroide de cada clúster y se repite el proceso hasta que se alcanza una solución óptima.

Otro algoritmo popular de clustering es el clustering jerárquico. Este algoritmo construye una jerarquía de clústeres mediante la combinación iterativa de clústeres en subgrupos más grandes. El resultado es un árbol jerárquico que representa la estructura de agrupamiento de los datos. El clustering se utiliza en muchas aplicaciones, como la segmentación de clientes, la clasificación de imágenes y la agrupación de documentos. Por ejemplo, en la

segmentación de clientes, el clustering se puede utilizar para agrupar a los clientes en función de sus patrones de compra o preferencias. En la clasificación de imágenes, el clustering se puede utilizar para agrupar imágenes similares para su posterior análisis o clasificación.

En la agrupación de documentos, el clustering se puede utilizar para agrupar documentos similares en temas específicos.Sin embargo, es importante tener en cuenta que el clustering no siempre es la mejor opción para todos los conjuntos de datos y situaciones. Algunas limitaciones del clustering incluyen la necesidad de definir el número de clústeres de antemano, la sensibilidad a los valores atípicos y la dificultad de evaluar los resultados. Es importante seleccionar el algoritmo y los parámetros adecuados para obtener resultados precisos y significativos. (Bobadilla, 2021)

3.2.1.1. K-MEANS

El algoritmo k-means busca un número predeterminado de grupos dentro de un conjunto de datos multidimensional sin etiquetar. Logra esto utilizando una concepción simple de cómo se ve el agrupamiento óptimo:

- El “centro del grupo” es la media aritmética de todos los puntos que pertenecen al grupo.
- Cada punto está más cerca de su propio centro de conglomerado que de otros centros de conglomerados.

Esas dos suposiciones son la base del modelo de k-medias que funciona mediante el algoritmo de maximización de expectativas (E - M), es un algoritmo poderoso que surge en una variedad de contextos dentro de la ciencia de datos. En resumen, el enfoque de maximización de expectativas aquí consiste en el siguiente procedimiento:

- Adivinar algunos centros de los clúster.
- Repetir hasta converger.
- E-Step: asigna puntos al centro del clúster más cercano.
- M-Step: establezca los centros de los clústers en la media.

Aquí, el “paso E” o “paso de expectativa” se llama así porque implica actualizar nuestra expectativa de a qué grupo pertenece cada punto. El “paso M” o “paso de maximización” se llama así porque implica maximizar alguna función de aptitud que define la ubicación de los centros del conglomerado; en este caso, esa maximización se logra tomando una media simple de los datos en cada conglomerado.
La literatura sobre este algoritmo es amplia, pero se puede resumir de la siguiente manera: en circunstancias típicas, cada repetición del paso E y del paso M siempre dará como resultado una mejor estimación de las características del grupo.
Se puede visualizar el algoritmo como se muestra en la siguiente figura; para la inicialización particular que se muestra aquí, los clústeres convergen en solo tres iteraciones.

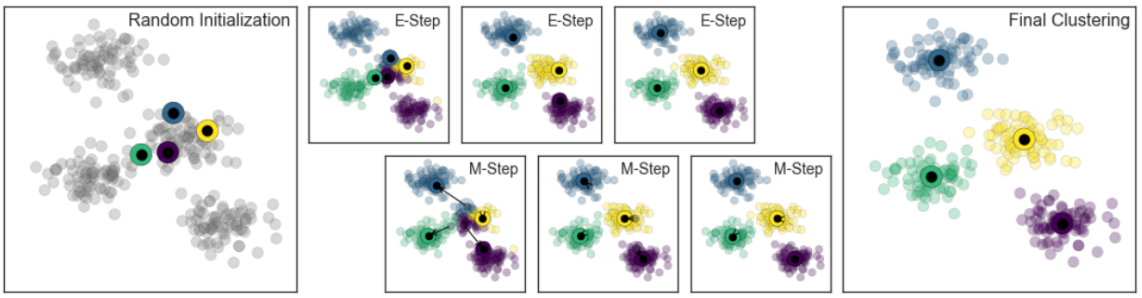

Figura 2: Visualización del algoritmo E-M en k-means.
Fuente: Python Data Science Handbook

El modelo K-means es un algoritmo de optimización y su objetivo es minimizar la suma de las distancias al cuadrado de cada punto de datos al centroide de su clúster. Esta métrica se llama "inercia" o "suma de los cuadrados de las distancias" y se utiliza para evaluar la calidad de los clústeres, así como Coeficiente de silueta que mide la similitud entre los puntos de datos dentro de su clúster en comparación con otros clústeres. Un coeficiente de silueta cercano a 1 indica que el punto de datos está correctamente asignado a su clúster, mientras que un coeficiente de silueta cercano a -1 indica que el punto de datos debería haber sido asignado a otro clúster.

3.2.2. REDUCCIÓN DE LA DIMENSIONALIDAD

La reducción de la dimensionalidad es una técnica que se utiliza para reducir el número de variables o características en un conjunto de datos, mientras se mantiene la mayor cantidad posible de información útil. La reducción de la dimensionalidad es importante ya que a menudo los conjuntos de datos pueden contener muchas variables o características, lo que puede hacer que los modelos sean complejos y difíciles de interpretar. Además, muchos algoritmos de aprendizaje automático pueden tener dificultades para manejar conjuntos de datos con un gran número de variables. Existen varios algoritmos de reducción de dimensionalidad, pero uno de los más populares es el análisis de componentes principales (PCA). PCA es un método lineal que utiliza una transformación matemática para encontrar una nueva representación de los datos en un espacio de menor dimensión, mientras se mantiene la mayor cantidad posible de información. La idea detrás de PCA es encontrar una nueva combinación de las variables originales que explique la mayor cantidad posible de la varianza en los datos.

Otro algoritmo popular de reducción de la dimensionalidad es el t-distributed stochastic neighbor embedding (t-SNE). t-SNE es una técnica no lineal que se utiliza para visualizar datos en un espacio de menor dimensión. t-SNE es especialmente útil para visualizar datos de alta dimensión en dos o tres dimensiones. La técnica funciona encontrando una representación de los datos en un espacio de menor dimensión que mantiene la estructura de similitud de los datos originales. La reducción de la dimensionalidad se utiliza en muchas aplicaciones, como la visualización de datos, la detección de anomalías y la clasificación de datos. Por ejemplo, en la visualización de datos, la reducción de la dimensionalidad se puede utilizar para visualizar datos de alta dimensión en dos o tres dimensiones. En la detección de anomalías, la reducción de la dimensionalidad se puede utilizar para identificar patrones o grupos de datos anómalos. En la clasificación de datos, la reducción de la

dimensionalidad se puede utilizar para mejorar la precisión de los modelos al reducir la complejidad del conjunto de datos.

Sin embargo, es importante tener en cuenta que la reducción de la dimensionalidad también puede tener algunas limitaciones. Por ejemplo, puede perder información importante durante el proceso de reducción de la dimensionalidad. Además, algunos algoritmos pueden ser sensibles a los valores atípicos o pueden ser difíciles de interpretar. (Benítez Iglesias, Escudero Bakx, Kanaan Izquierdo, Masip Rodó, y Cencerrado Barraqué, 2018)

3.3. SERIES TEMPORALES

Una serie temporal es una colección de datos de una variable recogidas secuencialmente en el tiempo, estos datos de series temporales siguen intervalos de tiempo periódicos que se midieron en intervalos de tiempo regulares o se recopilaron en intervalos de tiempo particulares. Estos datos se suelen recoger en instantes de tiempo equiespaciados, si los datos se recogen en instantes temporales de forma continua, se debe o bien digitalizar la serie, es decir, recoger sólo los valores en instantes de tiempo equiespaciados, o bien acumular los valores sobre intervalos de tiempo.
Los datos de series temporales siguen intervalos de tiempo periódicos que se midieron en intervalos de tiempo regulares o se recopilaron en intervalos de tiempo particulares. En decir una serie temporal es simplemente una serie de puntos de datos ordenados en el tiempo, y el análisis de series temporales es el proceso de dar sentido a dichos datos. (de la Fuente Fernández, 2013)

Las series temporales pueden ser descompuestas en varios componentes, que ayudan a entender mejor la estructura de los datos y a modelarlos de manera adecuada. Los componentes comunes de una serie temporal son los siguientes:

- **Tendencia:** Es la dirección general de los datos a largo plazo. Puede ser creciente, decreciente o constante. La tendencia indica la dirección en la que se mueven los datos en el largo plazo.
- **Estacionalidad:** Son patrones que se repiten en un intervalo de tiempo fijo, como las estaciones del año, días de la semana, horas del día, etc. La estacionalidad indica cómo los datos varían a corto plazo, en períodos fijos.
- **Cíclico:** Son patrones que se repiten en intervalos irregulares, generalmente más largos que los patrones estacionales. Los ciclos pueden ser causados por factores económicos, sociales o políticos.
- **Componente aleatorio:** Es la variación aleatoria en la serie temporal que no puede ser explicada por la tendencia, la estacionalidad o los ciclos. Este componente representa la variabilidad en la serie temporal que no puede ser explicada por otros factores.

La descomposición de una serie temporal en estos componentes se puede hacer utilizando técnicas estadísticas como el análisis de series de tiempo o métodos más avanzados como el análisis de componentes principales. La comprensión de estos componentes es importante para modelar adecuadamente la serie temporal y hacer predicciones precisas. Por ejemplo, si se espera que la tendencia y la estacionalidad se mantengan constantes, se puede utilizar

un modelo ARIMA para modelar la serie temporal. Si los datos tienen ciclos irregulares, un modelo de regresión de series temporales puede ser más apropiado.

3.3.1. FORECASTING AUTORREGRESIVO RECURSIVO (RAF)

Es un modelo de pronóstico de series temporales que utiliza un enfoque recursivo para predecir valores futuros, se basa en la idea de que los valores futuros de una serie temporal están altamente correlacionados con sus valores pasados, y se puede utilizar esta relación para predecir valores futuros de manera recursiva. El modelo RAF se compone de dos partes principales: la primera parte es un modelo autorregresivo (AR) que estima la relación entre los valores pasados de la serie temporal y su valor actual, mientras que la segunda parte es una función recursiva que utiliza el modelo AR para predecir valores futuros de manera recursiva. (Joaquín Amat Rodrigo, 2021)
El modelo AR se basa en la idea de que el valor actual de la serie temporal está relacionado linealmente con sus valores pasados, y se puede modelar mediante una ecuación matemática. La ecuación AR se puede escribir como:

$$y_t = c + \sum_{i=1}^{n} \varphi_i \cdot y_{t-i} + e_t \tag{6}$$

donde:

y_t es el valor actual de la serie temporal.
e_t es el término de error aleatorio.
c es una constante.
φ_i son los coeficientes de la ecuación AR que representan la relación entre los valores pasados y el valor actual.

Una vez que se ha ajustado el modelo AR a los datos de la serie temporal, se utiliza la función recursiva para predecir los valores futuros de la serie temporal. La función recursiva se define como:

$$y_{(t+h)} = c + \sum_{i=1}^{n} \varphi_i \cdot y_{(t+h-i)} \tag{7}$$

donde:

$y_{(t+h)}$ es el valor predicho de la serie temporal en el momento $t + h$.
y_{t-i} es el valor pasado de la serie temporal en el momento $t - i$.

Esta función se utiliza para predecir los valores futuros de la serie temporal recursivamente, utilizando los valores pasados de la serie temporal. El modelo RAF se puede utilizar para pronosticar diferentes horizontes de pronóstico, es decir, el modelo puede utilizarse para pronosticar los valores futuros de la serie temporal a corto plazo, a medio plazo o a largo plazo. El modelo es especialmente útil para pronósticos de corto plazo, ya que se basa en la información disponible en el momento actual y utiliza una función recursiva para predecir valores futuros.

3.4. ESTUDIOS Y MODELOS ACTUALES DE COVID-19

El aprendizaje automático ha experimentado un gran avance en las últimas décadas gracias al aumento de la cantidad y calidad de datos disponibles y a la mejora de los algoritmos uno de los más importante ha sido el desarrollo de algoritmos de aprendizaje semi-supervisado, redes neuronales y de transferencia de aprendizaje, que permiten entrenar modelos con conjuntos de datos más pequeños y reducir el tiempo y los costos de entrenamiento. Estos enfoques son especialmente útiles en áreas donde la recopilación de datos es costosa o difícil, como en la medicina (enfermedades, pandemias, etc.) o la astronomía. Desde que el COVID-19 comenzó a propagarse por todo el mundo, se han desarrollado una variedad de modelos de predicción para ayudar a los responsables de la toma de decisiones a entender mejor la propagación del virus y tomar decisiones informadas para proteger a la población. Estos estudios han surgido de diferentes áreas como lo son: Modelos epidemiológicos que se basan en la teoría de la propagación de enfermedades y utilizan datos de casos confirmados, fallecimientos, y otros factores como la edad, el género y las comorbilidades para predecir la propagación futura del virus, los modelos epidemiológicos más comunes son SIR (Susceptible-Infectado-Recuperado) y SEIR (Susceptible-Expuesto-Infectado-Recuperado); modelos de aprendizaje automático que utilizan algoritmos para analizar datos y hacer predicciones basadas en patrones; los modelos basados en la movilidad que utilizan datos de movilidad de los teléfonos móviles y otros dispositivos para predecir la propagación del COVID-19; entre otros. Haciendo un examen más profundo en el área de interés de este trabajo se encuentran investigaciones bastante interesantes detalladas a continuación:

- **Modelos basados en series de tiempo:** Estos modelos utilizan datos históricos del COVID-19 para predecir la propagación futura de la enfermedad. Un ejemplo es el modelo ARIMA (Autoregressive Integrated Moving Average) que se ha utilizado para predecir la propagación del COVID-19 en varios países.
  En donde los casos infectados, el número de muertes y los casos recuperados se pronostican con la media móvil integrada autorregresiva (ARIMA). Las técnicas se comparan en términos de coeficiente de correlación y error cuadrático medio (MSE), los autores encontraron que el modelo ARIMA tuvo una buena capacidad de predicción en los primeros días de la epidemia en China, pero la precisión disminuyó a medida que se acumularon más datos, esto debido a que se introdujeron diferentes factores que aumentaron la complejidad y la incertidumbre (medidas de control y prevención). (Toğa, Atalay, y Toksari, 2021)

- **Modelos basados en redes neuronales:** Las redes neuronales son capaces de aprender patrones complejos en los datos y han sido utilizadas para predecir la propagación del COVID-19. Por ejemplo, el modelo LSTM (Long Short-Term Memory) ha sido utilizado para predecir el número de casos confirmados en Canada, Estados Unidos e Italia. (Chimmula y Zhang, 2020)

- **Modelos basados en análisis de redes:** Estos modelos utilizan técnicas de análisis de redes para modelar la propagación del COVID-19. Un ejemplo es el modelo SEIR (Susceptible-Exposed-Infected-Recovered) que modela la propagación del virus como una red de interacciones entre individuos (Chen, Yang, Yang, Wang, y Bärnighausen, 2020). Los autores utilizaron datos públicos de COVID-19 en Jordania para calibrar y validar el modelo. Luego, el modelo se utilizó para predecir la propagación del virus bajo diferentes escenarios de intervención, como el cierre de escuelas, el cierre

de negocios y la cuarentena de casos sospechosos y confirmados. Los resultados del estudio muestran que el modelo puede predecir con precisión la propagación del COVID-19 en Jordania bajo diferentes escenarios de intervención. Los autores también encontraron que el cierre de escuelas y negocios, junto con la cuarentena de casos sospechosos y confirmados, puede reducir significativamente la propagación del virus en Jordania.

- **Modelos basados en aprendizaje automático:** Los modelos de aprendizaje automático utilizan técnicas de machine learning para predecir la propagación del COVID-19. Un ejemplo es el modelo XGBoost que ha sido utilizado para predecir la propagación del virus en varios países (Sakly y cols., 2023). Este propone un nuevo enfoque híbrido de aprendizaje profundo (DL) para estimar los patrones de transmisión de COVID-19 en Corea del Sur. El marco propuesto combina el aprendizaje profundo con el modelo de meta población susceptible-expuesto-infectado-recuperado (SEIR). Para mostrar su eficacia, el marco híbrido de aprendizaje profundo se comparó con el modelo de memoria a corto plazo (LSTM) y el modelo general de red neuronal profunda (DNN) para pronosticar patrones epidémicos en Corea del Sur en función del mismo conjunto de datos. El análisis numérico demostró que el marco híbrido de aprendizaje profundo que utiliza el modelo de meta población y el modelo LSTM exhibe el mejor rendimiento entre los métodos de prueba.

Fuera de los ejemplos expuestos anteriormente sobresale el proyecto **COVID-19 Projections** de Estados Unidos, que utiliza técnicas de aprendizaje automático para predecir la propagación del COVID-19 en línea (Gu, 2021a) y se actualiza regularmente con nuevos datos. Este proyecto está desarrollado un simulador basado en el modelo SEIR (He, Peng, y Sun, 2020), para simular la epidemia de COVID-19 en cada región. Luego, los parámetros/entradas de este simulador se aprenden mediante técnicas de aprendizaje automático que intentan minimizar el error entre las salidas proyectadas y los resultados reales. Utiliza los datos diarios de muertes informados por cada región para pronosticar futuras muertes informadas. Después de algunas técnicas de validación adicionales (para minimizar un fenómeno llamado sobre-ajuste), se usa los parámetros aprendidos para simular el futuro y hacer proyecciones.

Este modelo SEIR es de código abierto (Gu, 2021b). Las proyecciones se cargan diariamente en GitHub (Gu, 2021c). El objetivo de este proyecto es mostrar las fortalezas de la inteligencia artificial para abordar uno de los problemas más difíciles del mundo: predecir la trayectoria de una pandemia. En donde se utiliza un enfoque basado en datos puros al dejar que la máquina aprenda. Actualmente esta haciendo proyecciones para: Estados Unidos, los 50 estados de EE. UU. (más DC, PR, VI, GU, MP) y 70 países (incluidos los 27 países de la UE). Combinados, estos 71 países representan más del 95 % de todas las muertes globales por COVID-19.

Sin duda hay una gran variedad de estudios, artículos y publicaciones que tratan los factores principales en la propagación del Covid-19, pero se ha dejado de lado el estudio sobre otros factores que podrían influir en la propagación, como lo es el factor climático; centro de estudio de este trabajo.

4. DESARROLLO DEL PROYECTO Y RESULTADOS

Para el desarrollo de este proyecto se utilizó la metodología KDD (Knowledge Discovery in Databases) ya que proporciona una estructura sistemática para la extracción de conocimiento a partir de los datos para darle una solución al planteamiento del problema. Se realizará un análisis sobre los resultados obtenidos y las posibles mejoras en trabajos futuros.

Todo el desarrollo del proyecto se realizará en lenguaje de programación Python, a partir de notebooks de Jupyter, en estos se encontrará el desarrollo de todas las etapas del proceso de KDD y el porque se toman ciertas decisiones. Basado en los resultados de los notebooks se crean archivos Python con algunas etapas para realizar una CI/CD. Las fuentes de datos, los notebooks con los análisis de cada etapa y los archivos Python para la automatización de todo el flujo se encuentra en el repositorio personal para este TFM, a su vez se encontrará el código fuente de LaTeX, el cual se desarrolló el presente documento: REPOSITORIO TFM.

4.1. METODOLOGÍA

KDD (Knowledge Discovery in Databases), o Descubrimiento de Conocimiento en Bases de Datos, es un proceso integral que implica la identificación, extracción y transformación de patrones y conocimientos valiosos a partir de grandes conjuntos de datos. Es una disciplina que combina el uso de técnicas de minería de datos, estadísticas, aprendizaje automático y bases de datos para descubrir información útil y conocimientos ocultos en datos no estructurados o estructurados. (Fayyad, Piatetsky-Shapiro, y Smyth, 1996)

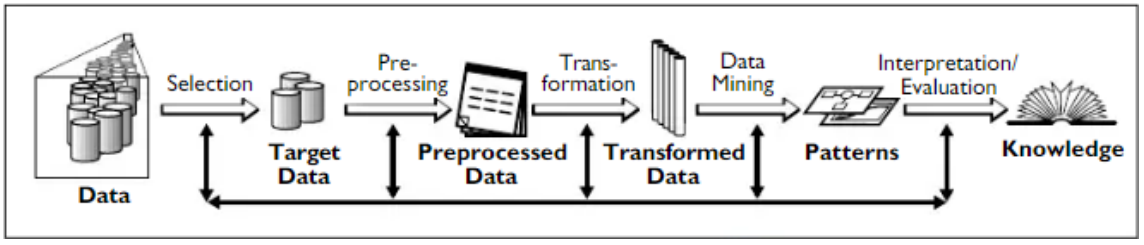

Figura 3: Descripción de los pasos que constituyen el proceso KDD

Fuente: The KDD Process for Extracting Useful Knowledge from Volumes of Data

El proceso de KDD consta de varias etapas, que incluyen:

- **Selección de datos:** Consiste en la identificación y recopilación de los datos relevantes para el análisis. Esto puede involucrar la obtención de datos de diversas fuentes, la limpieza y preprocesamiento de los datos para asegurar su calidad y consistencia.
- **Preprocesamiento de datos:** Implica la transformación y limpieza de los datos para prepararlos para el análisis. Esto puede incluir la eliminación de datos duplicados o inconsistentes, la normalización de los datos, la imputación de valores faltantes y la selección de características relevantes.

- **Transformación de datos:** Involucra la conversión de los datos preprocesados en formatos adecuados para el análisis. Esto puede incluir la transformación de datos categóricos en datos numéricos, la discretización de datos continuos, la reducción de dimensionalidad, entre otros.
- **Minería de datos:** Es la etapa central de KDD, donde se aplican técnicas y algoritmos de minería de datos para descubrir patrones y conocimientos en los datos. Esto puede incluir técnicas de clasificación, regresión, agrupamiento, asociación, entre otras.
- **Interpretación y Evaluación de resultados:** Implica la interpretación y comunicación de los resultados obtenidos a través de técnicas de visualización y presentación de datos. Así como la utilización de métricas de evaluación y validación para medir la precisión, el rendimiento y la utilidad de los resultados obtenidos. Esto puede ayudar a comprender y utilizar los patrones y conocimientos descubiertos para tomar decisiones informadas y mejorar la toma de decisiones en diversas áreas de aplicación.

4.2. PLANTEAMIENTO DEL PROBLEMA

El COVID-19 es una enfermedad infecciosa altamente contagiosa que ha afectado a millones de personas en todo el mundo y ha causado la muerte de cientos de miles de personas. Si bien se sabe que la propagación del virus se produce principalmente por contacto cercano con personas infectadas, también hay evidencia emergente que sugiere que las condiciones meteorológicas como la temperatura, la humedad y la luz solar, pueden influir en la propagación del virus. Un modelo de machine learning puede ser una herramienta útil para evaluar la relación entre las condiciones meteorológicas y la propagación del COVID-19. El objetivo de este planteamiento del problema es obtener valor y conocimiento a través de los datos mediante modelos de machine learning, que pueda mitigar o encontrar patrones de propagación del virus en función de factores climáticos como la temperatura, la humedad y la luz solar.
El modelo se entrenaría con datos históricos sobre la propagación del virus y las condiciones meteorológicas para predecir la propagación futura del virus en diferentes condiciones meteorológicas o más específicamente dependiendo del modelo utilizado una propuesta sería de la siguiente forma:

- **Modelo de regresión:** Utilizando datos históricos de propagación del virus y condiciones meteorológicas para predecir la propagación futura del virus. El modelo puede incluir variables relacionadas con la propagación del virus, como el número de casos confirmados y la tasa de reproducción.
- **Redes neuronales:** Se podría utilizar para analizar grandes conjuntos de datos de propagación del virus y condiciones meteorológicas. El modelo puede aprender patrones y relaciones entre las variables para predecir la propagación futura del virus en diferentes condiciones meteorológicas.
- **Análisis de series de tiempo:** Si contamos con datos históricos para identificar patrones y tendencias en la propagación del virus y las condiciones meteorológicas. El modelo puede predecir la propagación futura del virus en función de los patrones identificados.

- **Modelos de aprendizaje profundo:** Si tenemos datos de satélite y mapas climáticos para predecir la propagación del virus. Estos modelos pueden integrar datos climáticos con información sobre la densidad de población y la movilidad humana para predecir cómo se propagará el virus en diferentes áreas geográficas.

Lo anterior es solo un bosquejo de un posible uso de cada tipo de modelo, conforme vayamos avanzando y en base a la investigación realizada determinaremos cuál es el modelo que más se ajusta a nuestro requerimiento o que obtenga mejores resultados basado en sus métricas, el resultado de esto podría ayudar a informar la toma de decisiones sobre políticas de salud pública y permitir a las autoridades sanitarias tomar medidas preventivas antes de que se produzca un aumento en los casos de COVID-19. Al responder a esta pregunta, se pueden desarrollar mejores estrategias de prevención y mitigación para el COVID-19, y se pueden aplicar los hallazgos a futuras pandemias y enfermedades infecciosas.

4.3. DESARROLLO DEL PROYECTO

El proyecto se desarrollará siguiendo cada una de las etapas del KDD, ya que proporciona una estructura sistemática para la extracción de conocimiento a partir de los datos. Al seguir cada una de las etapas del KDD, se puede asegurar que el proceso es riguroso y que se obtienen resultados precisos y relevantes para el proyecto.

4.3.1. SELECCIÓN DE DATOS

Para la ejecución del proyecto se utilizaron distintas fuentes de datos las cuales se detallan a continuación:

- **Datos casos COVID-19 por provincias**

  Los resultados que se presentan se obtienen a partir de la declaración de los casos de COVID-19 a la Red Nacional de Vigilancia Epidemiológica (RENAVE) a través de la plataforma informática vía Web SiViES (Sistema de Vigilancia de España) que gestiona el Centro Nacional de Epidemiología (CNE). Esta información procede de la encuesta epidemiológica de caso que cada Comunidad Autónoma cumplimenta ante la identificación de un caso de COVID-19. Datos oficiales disponibles en el sitio web. (cnecovid, 2023)
  Se utilizan los siguientes sets de datos:

  casos_hosp_uci_def_sexo_edad_provres.csv: Datos desde el inicio de la pandemia, para todas las edades, hasta el 28 de marzo de 2022.
  casos_hosp_uci_def_sexo_edad_provres_60_mas.csv: Datos desde el inicio de la pandemia, para la población de 60 o más años.

  Para ambos casos cuentan con las mismas columnas, así como su descripción. Esta descripción se detalla en la tabla **15** del anexo I.

- **Datos códigos provincias**

  Archivo de elaboración propia que contiene el nombre de la provincia y el correspondiente código ISO de la provincia. Esta fuente de datos es necesaria para el cruce de

información o unión de los sets de datos, ya que algunas fuentes tienen el nombre de la provincia y otros tiene su correspondiente código ISO de cada provincia.

Tabla 1: Código ISO y nombre de provincia.

Variable	Descripción
PROVINCIA_ISO	Código ISO correspondiente a la provincia
PROVINCIA	Nombre de la provincia

Fuente: Elaboración Propia

- **Datos meteorológicos por provincia**

  Los datos meteorológicos son diarios por provincia, estos datos oficiales fueron extraídos desde el portal datos abiertos de AEMET, que permite la difusión y la reutilización de la información meteorológica y climatológica de la Agencia, en el sentido indicado en la Ley 18/2015, de 9 de julio (oficial del estado, 2015), por la que se modifica la Ley 37/2007, de 16 de noviembre, sobre reutilización de la información del sector público. Para poder acceder a AEMET OpenData, es necesario solicitar una API Key (https://opendata.aemet.es/centrodedescargas/altaUsuario?). Una API Key es un identificador, mediante el cual se contabilizan e imputan los accesos que un usuario realiza al API. Mediante el API KEY solicitado se obtiene la data. (opendata, 2023)

Tabla 2: Variables meteorológicas de provincia.

Variable	Descripción
fecha	fecha del dia (AAAA-MM-DD)
indicativo	indicativo meteorológico
nombre	nombre (ubicación) de la estación
provincia	provincia de la estación
altitud	altitud de la estación en m sobre el nivel del mar
tmed	Temperatura media diaria
prec	Precipitación diaria de 07 a 07
tmin	Temperatura Mínima del día
horatmin	Hora y minuto de la temperatura mínima
tmax	Temperatura Máxima del día
horatmax	Hora y minuto de la temperatura máxima
dir	Dirección de la racha máxima
velmedia	Velocidad media del viento
racha	Racha máxima del viento
horaracha	Hora y minuto de la racha máxima
sol	Insolación
presmax	Presión máxima al nivel de referencia de la estación
horapresmax	Hora de la presión máxima (redondeada a la hora entera más próxima)
presmin	Presión mínima al nivel de referencia de la estación
horapresmin	Hora de la presión mínima (redondeada a la hora entera más próxima)

Fuente: Elaboración Propia

- **Datos población por provincia**

  Los datos anuales demográficos por provincia fueron extraídos desde el instituto nacional de estadística de España (INE, 2023), esta fuente de datos es necesaria para realizar una normalización de los casos de covid-19 por provincia o lo que llamamos la tasa de incidencia.

Tabla 3: Población anual desde 1998 por provincia.

Variable	Descripción
Provincias	Nombre de la provincia
Sexo	H (hombre), M (mujer), Ambos sexos
Edad (año a año)	Rango de edad - total edades (contempla todas)
Españoles/Extranjeros	Origen de la persona - total (comtempla ambos)
Año	Año de recopilación de la información
Total	Cantidad de personas

Fuente: Elaboración Propia

4.3.2. PREPROCESAMIENTO DE DATOS

Para cada una de las fuentes de datos se realiza el procesamiento de su información, así como el dataset total que esta compuesto de la unión de estas fuentes mencionados en el apartado anterior.

- **Datos casos COVID-19 por provincias**

  Ambos sets de datos de covid-19 por provincia cuenta con 8 columnas con los mismos nombres que fueron descritas en el apartado anterior. Se realiza la unión de ambos set de datos (*casos_hosp_uci_def_sexo_edad_provres.csv* y *casos_hosp_uci_def_sexo_edad_provres_60_mas.csv*), ya que la única diferencia entre estos dos es el rango de edad que presenta en una de sus variables. Se realiza un perfilamiento de los datos mediante la librería de pandas_profiling el cual a partir de un dataframe de pandas genera el perfilamiento de la data en el siguiente archivo df_covid_prof.html. (Plazas, 2023)

  Realizamos un análisis de la cantidad de nulos mediante una lista de porcentaje:

Tabla 4: Porcentaje de nulos dataset covid-19.

Variable	Porcentaje de nulos
provincia_iso	0,0 %
Sexo	0,0 %
grupo_edad	0,0 %
fecha	0,0 %
num_casos	0,0 %
num_hosp	0,0 %
num_uci	0,0 %
num_def	0,0 %

Fuente: Elaboración Propia

Los estadísticos obtenidos para el dataset se representan en la siguiente tabla:

Tabla 5: Estadísticos dataset covid-19.

	num_casos	num_hosp	num_uci	num_def
count	1461210.00	1461210.00	1461210.00	1461210.00
mean	8.74	0.43	0.04	0.08
std	48.38	2.49	0.30	0.77
min	0.00	0.00	0.00	0.00
25 %	0.00	0.00	0.00	0.00
50 %	0.00	0.00	0.00	0.00
75 %	4.00	0.00	0.00	0.00
max	3749.00	271.00	35.00	100.00

Fuente: Elaboración Propia

La media de *num_casos* es de 8.74 y la mediana es 0 ya que en muchas fechas diarias no se reportaron casos, la media de *num_hosp* es de 0.43 y la mediana 0, la media de *num_uci* es 0.04 y la mediana 0, por último, la media de *num_def* es de 0.08 y la mediana 0. Las variables parecen ser asimétricas a la derecha dado que su media es mayor a su mediana.

Las desviaciones típicas son bajas, la mayoría de las observaciones se encuentran dispersas a no más de una desviación estándar a cada lado. La imagen anterior también muestra los valores máximos y mínimos que toman cada variable objeto de estudio además de los cuartiles calculados. Los datos menores al cuartil 1 (Q1) representan el 25 % de los datos, los que están por debajo del cuartil 2 (Q2) representan el 50 % de los datos y los que están por debajo del cuartil 3 (Q3) representan el 75 % de los datos.

Tras realizar un análisis de correlación en la figura **4** entre sus variables de estudio del conjunto de datos se evidencia en la siguiente gráfica que las variables *num_hosp*, *num_uci* y *num_def* tiene una correlación positiva y fuerte entre ellas. Es de esperarse este resultado ya por lo general son consecuentes una con la otra, es decir, por ejemplo, si hubo una difusión por covid-19 es muy probable que haya estado en uci y hospitalización previamente.

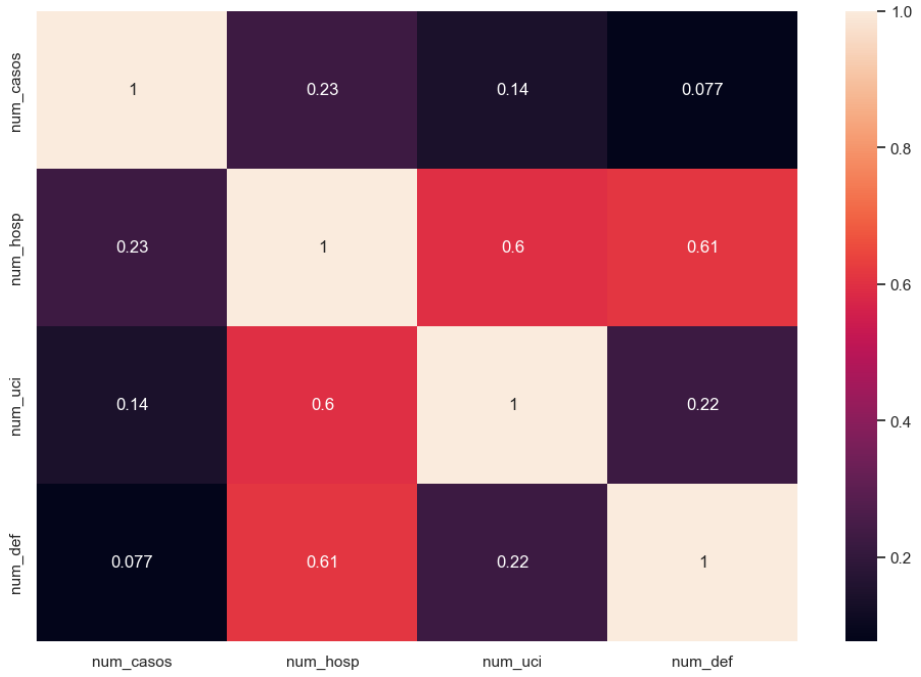

Figura 4: Correlación variables dataset covid-19.

Fuente: Elaboración propia

Se realizó un análisis de outliers para cada una de las variables, así como la distribución de los valores de las variables categóricas para identificar valores atípicos, para ningún caso se haya evidencia de alguno. Para evitar que existan palabras distintas y que simbolicen el mismo significado solo por el hecho de estar en minúscula o mayúsculas, para todas las variables tipo string las pasaremos a mayúscula, ya que por defecto todas viene así, también eliminaremos los espacios al principio y al final.

- **Datos códigos provincias**

  Esta fuente de datos fue de elaboración propia por lo cual se aseguro que no hubiera valores duplicado, valores nulos, valores atípicos, el nombre de las columnas está en mayúscula, los valores están en mayúscula sin ningún tipo de espacio por ende no necesita ningún tipo de transformación. Esta fuente cuenta con dos columnas y con 52 registros, el cual servirá para unir los datasets.

- **Datos meteorológicos por provincia**

  Se realiza la concatenación de cada uno de los archivos con la información meteorológica por provincia para poder obtener un dataset completo y su respectivo análisis, comenzando por la cantidad de nulos en todas sus variables mediante un heatmap.

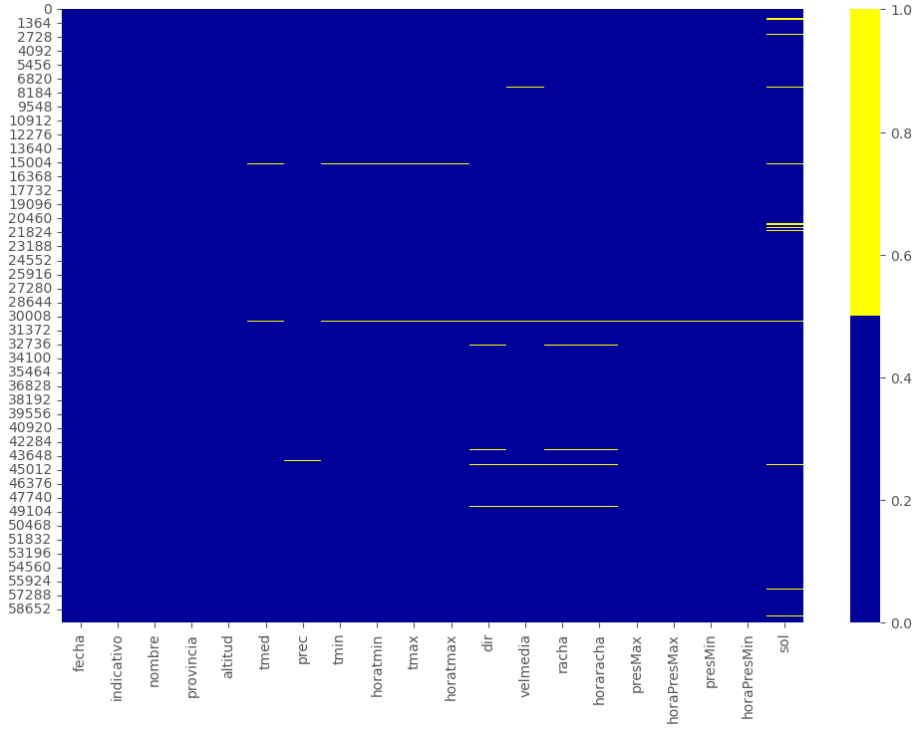

Figura 5: Headmap variables dataset meteorológico.

Fuente: Elaboración propia

En la figura **5** muestra los patrones de datos que faltan de todas las columnas, el eje horizontal muestra el nombre del atributo de entrada; el eje vertical muestra el número de observaciones/filas; el color amarillo representa los datos que faltan, mientras que el color azul, en caso contrario. Detallamos que todas las características tienen muy pocos valores perdidos o inclusive no tienen, para tener un valor exacto hacemos una lista de porcentaje de valores nulos:

Tabla 6: Población anual desde 1998 por provincia.

Variable	Porcentaje de nulos
fecha	0,0 %
indicativo	0,0 %
nombre	0,0 %
provincia	0,0 %
altitud	0,0 %
tmed	0,32 %
prec	0,31 %
tmin	0,32 %
horatmin	0,34 %
tmax	0,30 %
horatmax	0,32 %
dir	1,04 %
velmedia	0,71 %
racha	1,04 %
horaracha	1,05 %
presMax	0,41 %
horaPresMax	0,42 %
presMin	0,41 %
horaPresMin	0,43 %
sol	2,36 %

Fuente: Elaboración Propia

La imputación de los valores faltantes de las variables se estableció mediante fillna aplicado al dataframe por el método *ffill* como primera opción, como segunda opción el método *bfill*. El primer método usa la anterior observación válida para llenar el valor faltante y el segundo método usa la siguiente observación válida para llenar el valor faltante. La elección de estos métodos es debido a que son variables meteorológicas, en donde el clima entre estaciones meteorológicas y periodos de tiempo corto son similares, la granularidad de este set de datos es diaria por lo que la imputación de valores faltantes se hará sobre el valor del día anterior o el más próximo. Se elimina las variables indicativo y nombre, ya que estas hacen referencia netamente a la información de la estación meteorológica en donde se obtuvieron los datos, esta información no aporta a nuestro estudio.

Tras realizar un análisis de correlación entre sus variables de estudio del conjunto de datos se evidencia en la siguiente gráfica que las variables *PREX_MAX* y *PREX_MIN* tiene una correlación positiva y fuerte entre ellas; así como las variables *TEMP_MIN*, *TEMP_MAX* y *TEMP_MED* tienen una correlación positiva alta.

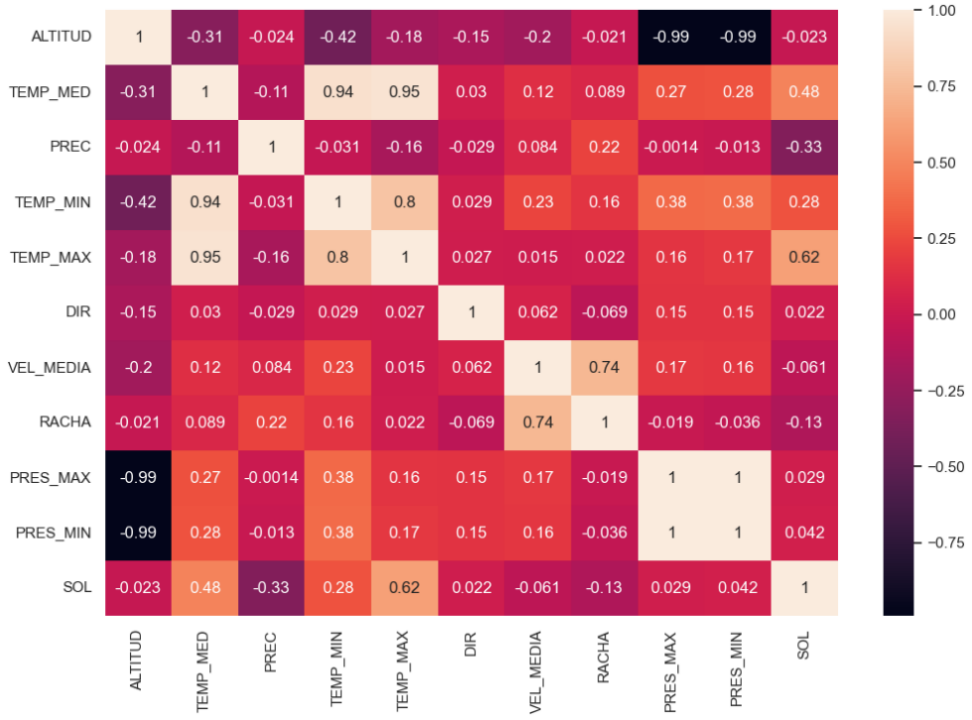

Figura 6: Correlación variables dataset meteorológico.

Fuente: Elaboración propia

Se realizó un análisis de outliers por medio de boxplots para cada una de las variables, así como la distribución de los valores de las variables categóricas para identificar valores atípicos, se encontraron valores alejados de la media y poco comunes, pero son valores meteorológicos posibles por esta razón para ningún caso se haya evidencia de outliers. Para evitar que existan palabras distintas y que simbolicen el mismo significado solo por el hecho de estar en minúscula o mayúsculas, para todas las variables tipo string las pasaremos a mayúscula, ya que por defecto todas viene así, también eliminaremos los espacios al principio y al final.

- **Datos población por provincia**

Para este set de datos se eliminó las variables Sexo, Edad (año a año) y Españoles/Extranjeros, ya que solo nos interesa la población anual por provincia para poder generar la tasa de incidencia de covid-19 mensual. Como primer paso se crea una columna equivalente a la tasa de crecimiento mensual ya que tenemos la población anual, se utiliza una de las formulas más utilizadas para cálculos poblacionales como lo es el modelo geométrico (Torres-Degró, 2011). A continuación, se describe la fórmula utilizada:

$$r = \left(\frac{P_f}{P_i}\right)^{\frac{1}{t}} - 1 \tag{8}$$

donde:

r Tasa de crecimiento mensual
P_f Población final
P_i Población inicial

t Distancia en tiempo entre las dos poblaciones de referencia

Tomaremos la población inicial del año 2019 y la población actual del año 2022, esto debido a que en los años anteriores en casi todos los casos aumento la población, pero en este periodo de tiempo el comportamiento fue diferente a razón del covid-19, el cálculo de la tasa de crecimiento mensual se utilizará para calcular el aproximado de la población mensual por provincia hasta el primer trimestre del 2023. Se realiza el calculo de la población mensual con proyección en un periodo t, mediante la siguiente ecuación (Vandermeer y Goldberg, 2013)

$$P_f = P_i(1+r)^t \tag{9}$$

donde:

P_f Población final en ese caso mes a mes
P_i Población Inicial en este caso del comienzo de cada año
r Tasa de crecimiento mensual calculada anteriormente
t La proyección en tiempo, en este caso el mes a calcular

El valor de esta población mensual tomara el nombre de *POB_MEN*

- **Dataset total**

Este dataset está conformado por el dataset meteorológico unido a la fuente de datos cod_iso_provincias por medio del campo *PROVINCIA* esto con el fin de añadir la columna *PROVINCIA_ISO*, a su vez este dataset se unirá a la fuente de datos de covid por medio de la columna *PROVINCIA_ISO* y la *FECHA*, para generar el dataset total, este proceso se describe mediante los siguientes comandos en Python:

```
df_clima_iso = df_clima.merge(df_iso, how="inner", on="PROVINCIA")
df_total = df_covid.merge(df_clima_iso, how="inner",
                          on=["FECHA", "PROVINCIA_ISO"])
```

A este dataset total se eliminaron las variables de Horas (*HORA_TEMP_MIN, HORA_TEMP_MAX, HORA_RACHA, HORA_PRES_MAX, HORA_PRES_MIN*) ya que no deseamos un nivel de granularidad tan bajo, por el contrario, se tomara en cuenta la demás variables meteorológicas diarias; se elimina la variable *PROVINCIA_ISO* ya que tenemos la variable *PROVINCIA* la cual hace referencia al mismo significado; se elimina las variables *NUM_HOSP, NUM_UCI, NUM_DEFU* esto debido a la explicación de la figura **4** y el objetivo principal del estudio es la propagación del virus covid-19 es decir el número de casos (*NUM_CASOS*) y no las defunciones y/o hospitalizaciones; se elimina las variables *TEMP_MIN, TEMP_MAX* esto debido a la explicación de la figura **6** y se conserva la variable *TEMP_MED*; por el momento se eliminan las variables *GRUPO_EDAD, SEXO* para centrar el estudio en la propagación del virus entorno a las variables meteorológicas. Tras esta serie de pasos se realiza una gráfica de tendencia de los casos de covid de todas las provincias para tener un panorama más amplio del caso de estudio, ver primera gráfica de la figura **7**.

Se crea la variable *TASA_INCIDENCIA* a partir de la normalización del número de casos, dada por la siguiente formula:

$$TASA\_INCIDENCIA = \left(\frac{NUM\_CASOS}{POB\_MEN}\right) \times 100000 \tag{10}$$

En donde *POB_MEN* (población mensual) se calculó en el ítem anterior para cada año desde el 2020 hasta el primer trimestre del 2023 por provincia. De igual forma se realiza una gráfica de tendencia de la tasa de incidencia de todas las provincias, como se puede observar en la siguiente figura la tasa de incidencia es una muy buena normalización con respecto a los casos covid de la figura **7**.

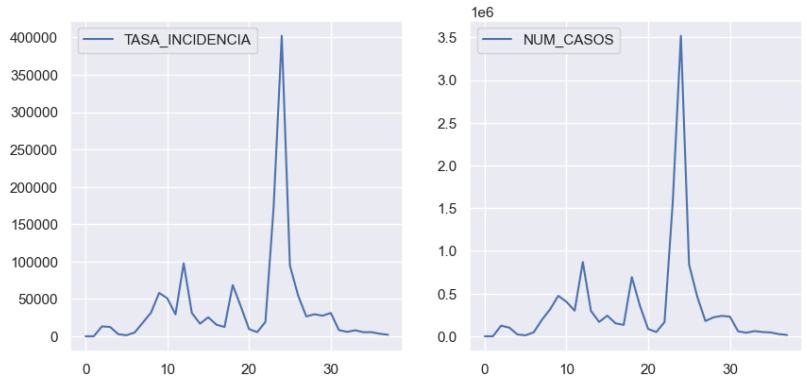

Figura 7: Tendencia tasa de incidencia.

Fuente: Elaboración propia

4.3.3. TRANSFORMACIÓN DE DATOS

Para cada una de las fuentes de datos descrita anteriormente se realiza la transformación de su información, así como el dataset total que esta compuesto de la unión de estas fuentes.

- **Datos casos COVID-19 por provincias**

  Se realizó la transformación del nombre de todas las columnas a mayúscula, de esta forma se trabajará en todos los dataset para manejar un estándar de nombramiento, se realiza conversión de la variable de tiempo *fecha* a tipo "datetime", las variables que estén tipo float y no tengan ningún valor decimal se convertirán en enteros. Normalización y homologación de los valores en campos categóricos con el fin de agrupar y estandarizar, obteniendo así los siguientes tipos de datos:

Tabla 7: Tipos de datos dataset covid-19.

Variable	Tipo de datos
PROVINCIA_ISO	object
SEXO	object
GRUPO_EDAD	object
FECHA	datetime64[ns]
NUM_CASOS	object
NUM_HOSP	object
NUM_UCI	object
NUM_DEFU	object

Fuente: Elaboración Propia

- **Datos meteorológicos por provincia**

  Se realizó la transformación del nombre de todas las columnas a mayúscula, la conversión de la variable de tiempo *fecha* a tipo "datetime", las variables que estén tipo float y no tengan ningún valor decimal se convertirán en enteros, así como las variables que son de tipo string y que en realidad todos son datos son numéricos decimales se realizará su correspondiente transformación a tipo float. A la variable prec (Precipitación diaria de 07 a 07) se transformó el valor 'Ip' (significa precipitación inapreciable, es decir, cantidad inferior a 0.1 mm) por 0,0.

  Para todas las variables de horas se transformo el valor "24" por "00" ya que hacen referencia a la misma hora, más adelante se explicará porque estas variables de horas no serán tomadas en cuenta para la etapa de minería de datos (creación de modelos). Inicialmente solo dos campos eran numéricos, tras realizar el proceso de transformación obtenemos los siguientes tipos de datos:

Tabla 8: Tipos de datos dataset meteorológico.

Variable	Porcentaje de nulos
FECHA	datetime64[ns]
PROVINCIA	object
ALTITUD	int64
TEMP_MED	float64
PREC	float64
TEMP_MIN	float64
HORA_TEMP_MIN	object
TEMP_MAX	float64
HORA_TEMP_MAX	object
DIR	float64
VEL_MEDIA	float64
RACHA	float64
HORA_RACHA	object
PRES_MAX	float64
HORA_PRES_MAX	object
PRES_MIN	float64
HORA_PRES_MIN	object
SOL	float64

Fuente: Elaboración Propia

- **Datos población por provincia**

  Se transforma la variable *Provincias* y se hace la homologación con los mismos nombres de las provincias como en los demás conjuntos de datos, es decir, se reemplazan algunos nombres; la variable *Total* se transforma a entero ya que todos sus datos tienen decimales con ceros, además de que este debe ser un valor entero; una vez calculada la población mensual en el preprocesamiento se eliminan las variables

intermedias y que no son de utilidad como *YEAR, YEAR_ACUM, TOTAL_POB, TASA_MENSUAL*

- **Dataset total**

  Se realiza la transformación de la variable *FECHA* cambiando su granularidad de diaria a mensual, de esta forma se procede a hacer la agrupación de los datos por *FECHA* y *PROVINCIA* y todas las demás medidas se le hace un promedio excepto la variable *NUM_CASOS* que será la suma de todos los días para el mes correspondiente. Se eliminan las variables *NUM_CASOS, POB_MEN* tras el calculo de la tasa de incidencia de covid-19 en el ítem anterior, obteniendo así el dataset final con los siguientes tipos de datos:

Tabla 9: Tipos de datos dataset total.

Variable	Porcentaje de nulos
FECHA	period[M]
PROVINCIA	object
ALTITUD	float64
TEMP_MED	float64
PREC	float64
DIR	float64
VEL_MEDIA	float64
RACHA	float64
PRES_MIN	float64
SOL	float64
TASA_INCIDENCIA	float64

Fuente: Elaboración Propia

Este dataset final será exportado como archivo data_refined.csv y será tomado como partida de referencia para la construcción de los modelos en nuestra etapa de minería de datos.

4.3.4. MINERÍA DE DATOS

En esta etapa de nuestro proceso de KDD partiremos de nuestro dataset final (data_refined) en donde descubriremos patrones y conocimiento mediante técnicas y algoritmos de machine learning, se seleccionó un modelo de regresión lineal ya que es fácilmente interpretable y comprensible, lo que lo hace útil para comunicar los resultados a diferentes audiencias, incluidos los responsables de la toma de decisiones y el público en general. Además, al ser un modelo más simple en comparación con métodos más complejos, puede ser más fácil de implementar y aplicar en entornos donde los recursos computacionales y técnicos pueden ser limitados. Un modelo de regresión lineal puede servir como un punto de partida inicial para evaluar la relación entre las condiciones meteorológicas y la propagación del virus. Puede proporcionar una idea preliminar de la influencia de las variables meteorológicas y ayudar a identificar aquellas que tienen un mayor impacto en la predicción de casos de COVID-19. A partir de ahí, se pueden explorar modelos más avanzados y sofisticados si es necesario. (Hao y cols., 2022)

Nuestro segundo modelo seleccionado fue Random Forest Regression que pueden capturar relaciones no lineales y complejas entre las variables predictoras (como las condiciones meteorológicas) y la variable objetivo (los casos de COVID-19). Esto es especialmente útil cuando las relaciones entre las variables no son lineales o cuando existen interacciones complejas entre múltiples variables, son menos susceptibles a los efectos de los datos ruidosos y los valores atípicos lo que permite obtener predicciones más precisas. Los modelos de Random Forest tienden a tener una buena capacidad de generalización, lo que significa que pueden ofrecer buenos resultados de predicción en conjuntos de datos nuevos o no vistos previamente. Esto es particularmente valioso en el contexto de la predicción de COVID-19, donde es importante tener modelos que puedan adaptarse a cambios en las condiciones meteorológicas y en la dinámica de la pandemia. (Ciria Mayordomo, 2021)

Nuestro tercer modelo seleccionado es ForecasterAutoreg, nos brinda un enfoque autorregresivo que tiene en cuenta la dependencia temporal de los datos. En el contexto de la predicción de COVID-19, esto implica que el modelo puede capturar la evolución de los casos a lo largo del tiempo, teniendo en cuenta patrones y tendencias anteriores. Al combinar esto con las condiciones meteorológicas, el modelo puede capturar posibles efectos a largo plazo de las variables meteorológicas en la propagación del virus. Este modelo es relativamente interpretable, lo que significa que puede proporcionar información sobre la contribución relativa de las variables predictoras, incluidas las condiciones meteorológicas, en la predicción de casos de COVID-19. Esto permite una mejor comprensión de cómo las variables meteorológicas pueden afectar la propagación del virus y ayuda en la toma de decisiones basadas en evidencia. puede adaptarse a la disponibilidad de nuevos datos en tiempo real, lo que es esencial para la predicción de casos de COVID-19 en constante evolución. Esto permite que el modelo se actualice y ajuste a medida que se disponga de más información sobre las condiciones meteorológicas y la propagación del virus. (Hernandez-Matamoros, Fujita, Hayashi, y Perez-Meana, 2020)

A continuación, se describen su proceso.

4.3.4.1. REGRESIÓN LINEAL MÚLTIPLE

Para este modelo la variable dependiente será la TASA_INCIDENCIA que será la variable que queremos predecir las demás variables de nuestro dataset final serán las variables independientes. En este caso también tenemos una variable categórica *PROVINCIA*. Esta variable es transformada mediante el método de codificación hash. Se realizaron pruebas con la codificación One-Hot encoding obteniendo resultados muy similares o un poco menores con respecto a las métricas de medida, esto debido a que la principal debilidad de One-Hot encoding es que las características que produce son equivalentes al cardinal categórico, lo que causa problemas de dimensionalidad cuando la cardinalidad es demasiado alta, por el contrario la codificación hash representa los datos categóricos en valor numérico mediante la función hash.

La principal ventaja de usar Hash Encoding es que puede controlar la cantidad de columnas numéricas producidas por el proceso, debemos tener en cuenta que Hashing transforma los datos en dimensiones menores, puede provocar la pérdida de información y una gran cantidad de características que se representan en dimensiones menores pueden representar múltiples valores con el mismo valor hash, esto se conoce como colisión. Es por esto que

se debe escoger un valor optimo de columnas numéricas que representaran la variable. En general es ideal cuando el conjunto de datos tiene características de alta cardinalidad, para nuestro caso la variable *PROVINCIA* tiene 52 valores. A su vez se crea la variable *MONTH* a partir de la fecha, se tendrán en cuenta todas las variables excepto la variable *FECHA*. La ecuación del modelo quedaría de la siguiente forma:

$$TASA\_INCIDENCIA = b_o + b_1 MONTH + b_2 ALTITUD + b_3 TEMP_M ED$$
$$+b_4 PREC + b_5 DIR + b_6 VEL_M EDIA + b_7 RACHA + b_8 PRES_M IN + b_9 SOL$$
$$+b_{10} HASH_0 + b_{11} HASH_1 + b_{12} HASH_2 + b_{13} HASH_3 + b_{14} HASH_4 + b_{15} HASH_5$$

Dividimos nuestro dataset total en train y test en una relación de 70 % a 30 % respectivamente. Creamos el modelo a partir de linear_model.LinearRegression(), al realizar el entrenamiento, obtenemos los siguientes coeficientes y estadísticos para la ecuación del modelo:

OLS Regression Results

Dep. Variable:	y	R-squared:	0.152
Model:	OLS	Adj. R-squared:	0.143
Method:	Least Squares	F-statistic:	16.33
Date:	Wed, 03 May 2023	Prob (F-statistic):	8.94e-40
Time:	00:06:36	Log-Likelihood:	-11891.
No. Observations:	1383	AIC:	2.381e+04
Df Residuals:	1367	BIC:	2.390e+04
Df Model:	15		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-1.02e+05	1.29e+04	-7.904	0.000	-1.27e+05	-7.67e+04
MONTH	18.8856	12.267	1.540	0.124	-5.179	42.950
ALTITUD	11.2185	1.428	7.858	0.000	8.418	14.019
TEMP_MED	-87.2365	11.045	-7.898	0.000	-108.904	-65.569
PREC	26.5415	24.342	1.090	0.276	-21.210	74.293
DIR	0.1807	3.276	0.055	0.956	-6.246	6.607
VEL_MEDIA	49.0528	67.507	0.727	0.468	-83.376	181.482
RACHA	-50.3266	42.701	-1.179	0.239	-134.094	33.440
PRES_MIN	101.8353	12.560	8.108	0.000	77.196	126.475
SOL	131.1854	24.936	5.261	0.000	82.269	180.102
HASH_0	-2.1629	50.759	-0.043	0.966	-101.737	97.411
HASH_1	0.9132	65.880	0.014	0.989	-128.324	130.151
HASH_2	6.9595	44.026	0.158	0.874	-79.407	93.326
HASH_3	6.5977	24.957	0.264	0.792	-42.361	55.557
HASH_4	10.9428	25.665	0.426	0.670	-39.404	61.290
HASH_5	-26.8198	51.486	-0.521	0.603	-127.819	74.180

Omnibus:	1043.603	Durbin-Watson:	1.964
Prob(Omnibus):	0.000	Jarque-Bera (JB):	19859.351
Skew:	3.389	Prob(JB):	0.00
Kurtosis:	20.283	Cond. No.	3.84e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.84e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

Figura 8: Estadisticos modelo regresión lineal múltiple.

Fuente: Elaboración propia

Tomando los coeficientes de cada una de las variables, nuestra ecuación final queda de la siguiente forma:

$$\begin{aligned} TASA\_INCIDENCIA = -102000 + 18{,}88 \cdot MONTH + 11{,}21 \cdot ALTITUD \\ -87{,}23 \cdot TEMP\_MED + 26{,}54 \cdot PREC + 0{,}18 \cdot DIR + 49{,}05 \cdot VEL\_MEDIA \\ -50{,}32 \cdot RACHA + 101{,}83 \cdot PRES\_MIN + 131{,}18 \cdot SOL - 2{,}16 \cdot HASH\_0 \\ +0{,}91 \cdot HASH\_1 + 6{,}95 \cdot HASH\_2 + 6{,}59 \cdot HASH\_3 + 10{,}94 \cdot HASH\_4 \\ -26{,}81 \cdot HASH\_5 \end{aligned}$$

La constante b_o es -10200 es decir nuestra intercepto, los demás coeficientes b_i son los valores numéricos que acompañan a cada una de las variables independientes de nuestro modelo, la interpretación del coeficiente b_1 es que por cada unidad que se incrementa el *MES* el incremento de la tasa de incidencia será de 18,88 unidades; el coeficiente b_2 por cada unidad que se incremente la *ALTITUD* el incremento de la tasa de incidencia será de 11,21 unidades; el coeficiente b_3 por cada unidad que se incremente la *TEMP_MED* hay un decrecimiento de la tasa de incidencia de $-87{,}23$ unidades; de igual forma sucesivamente es la interpretación de los demás coeficientes.

De la **figura 8** concluimos que todas las variables introducidas como predictores tienen un R^2 bajo (0.152) es decir es capaz de explicar en un 15.2 % la variabilidad observada de la tasa de incidencia, acorde al p-value obtenido para el coeficiente parcial de regresión para algunas variables como HASH, VEL_MEDIA, DIR y PREC, estas no contribuyen de forma significativa el modelo. Se realizó el entrenamiento sin estas variables (HASH, VEL_MEDIA, DIR y PREC) y se obtuvieron resultados en las métricas muy similares, por tal motivo se dejaron en el entrenamiento de este modelo para evidenciar el impacto de estas variables en los resultados. Los intervalos de confianza se obtuvieron del modelo entrenado mediante el comando de python *modelo.conf_int(alpha=0.05)*, en donde el nivel de significancia utilizado es del 0.05, los resultados se muestran en la siguiente tabla **10**:

Tabla 10: Intervalos de confianza.

	2.5 %	97.5 %
CONST	-127300.37	-76677.83
MONTH	-5.179000	42.950290
ALTITUD	8.417721	14.019315
TEMP_MED	-108.903632	-65.569304
PREC	-21.210277	74.293363
DIR	-6.245870	6.607292
VEL_MEDIA	-83.375974	181.481665
RACHA	-134.093622	33.440448
PRES_MIN	77.195649	126.475020
SOL	82.268584	180.102275
HASH_0	-101.736920	97.411155
HASH_1	-128.324197	130.150506
HASH_2	-79.407085	93.326155
HASH_3	-42.361196	55.556558
HASH_4	-39.404144	61.289839
HASH_5	-127.819261	74.179561

Fuente: Elaboración Propia

Los residuos del modelo muestra que hay una alta acumulación sobre la línea recta del cero, es decir el promedio de los errores es cero; los residuos Q-Q con respecto se acerquen más los puntos a la recta mayor normalidad de los residuos, pero hay una gran cantidad de valores por debajo de -2, la pendiente de la recta está muy arriba y no logran estar todos los valores sobre la diagonal, porque hay unos datos que están por debajo de la distribución, esto se evidencia en el gráfico de distribución que esta sesgado a la derecha, el p-valor es menor al valor de significancia (5 %), por tanto no se satisface la hipótesis, es decir, los residuos no satisfacen una distribución normal.
Los diagnósticos de los residuos fueron los siguientes:

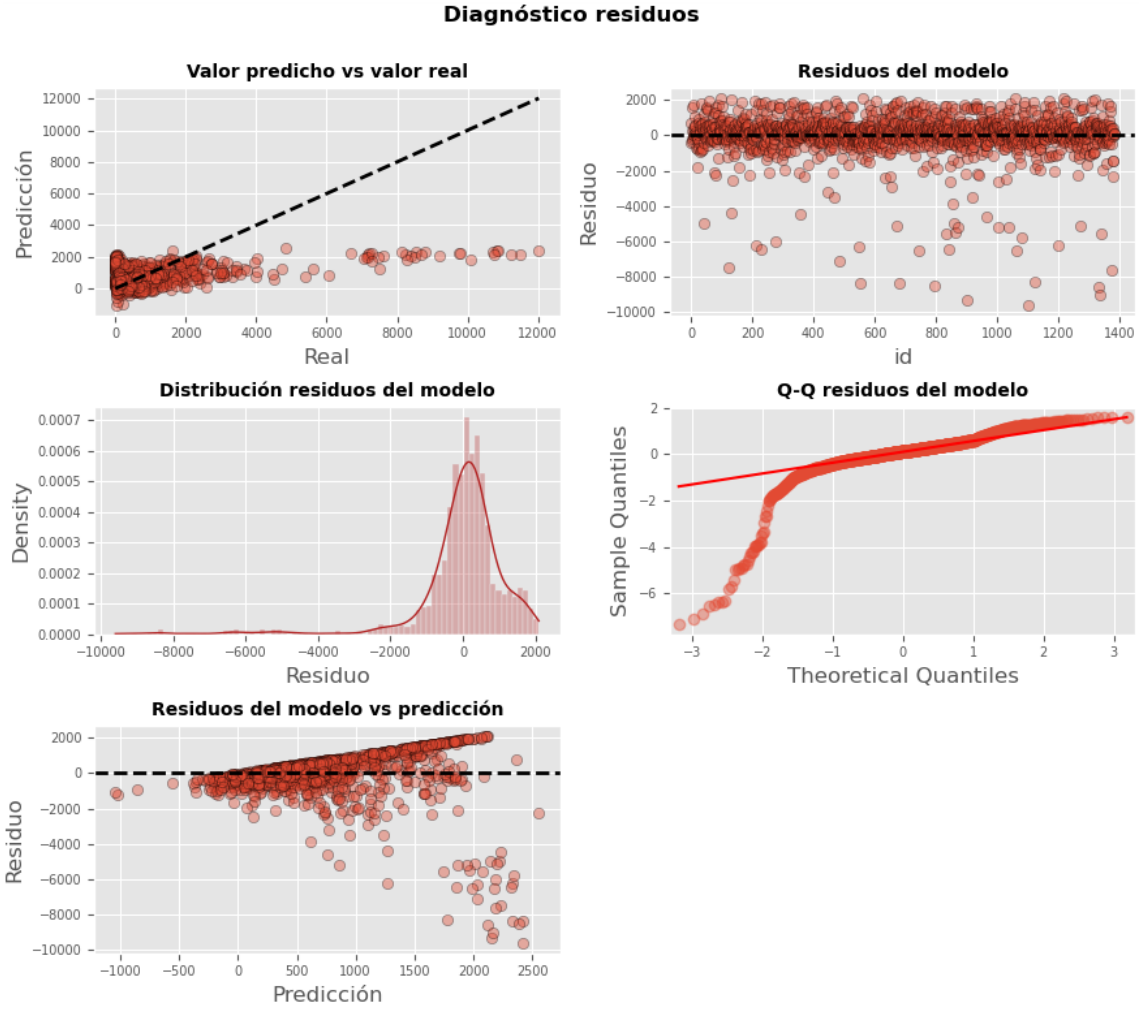

Figura 9: Diagnóstico residuos modelo regresión lineal múltiple.

Fuente: Elaboración propia

Para el test de normalidad se obtuvo:

Tabla 11: Test estadísticos para normalidad.

Test Estadístico	Estadístico	p-value
Shapiro-Wilk	0.7023	4.6242e-44
D'Agostino's K-squared	1043.6025	2.4243e-227

Fuente: Elaboración Propia

Es decir, se comprueba si los residuos siguen una distribución normal empleando dos test estadísticos: Shapiro-Wilk test y D'Agostino's K-squared test. Este último es el que incluye el summary de statsmodels bajo el nombre de Omnibus de la gráfica summary. En ambos test, la hipótesis nula considera que los datos siguen una distribución normal, por lo tanto, si el p-value no es inferior al nivel de referencia alpha seleccionado, no hay evidencias para descartar que los datos se distribuyen de forma normal y Ambos test muestran claras evidencias para rechazar la hipótesis de que los datos se distribuyen de forma normal (p-

value $\ll 0.01$). Las métricas para la evaluación del modelo se mostrara en el apartado de resultados.

4.3.4.2. RANDOM FOREST REGRESIÓN

De igual forma que en el modelo de regresión Lineal múltiple la variable dependiente será la *TASA_INCIDENCIA* que será la variable que queremos predecir las demás variables de nuestro dataset final serán las variables independientes. La variable categórica *PROVINCIA*, fue transformada mediante el método de codificación hash. Se realizaron pruebas con la codificación One-Hot encoding obteniendo resultados muy similares o un poco menores con respecto a las métricas de medida. A su vez se crea la variable *MONTH* a partir de la fecha, se tendrán en cuenta todas las variables excepto la variable *FECHA*. Se divide el dataset total en train y test de forma aleatoria en una relación de 70 % a 30 % respectivamente, esta división se realiza de acuerdo a la explicación del modelo de regresión lineal.

Para la creación de este modelo se utilizó RandomForestRegressor de la librería sklearn ensemble, este modelo cuenta con 16 hiperparámetros, de los cuales se utilizaron los siguientes:

- **n_estimadores:** El número de árboles en el bosque. Int, por defecto = 100.
- **criterion:** La función para medir la calidad de una división. Los criterios admitidos son "squared_error" para el error cuadrático medio, que es igual a la reducción de la varianza como criterio de selección de características y minimiza la pérdida de L2 utilizando la media de cada nodo terminal, "friedman_mse", que utiliza el error cuadrático medio con la puntuación de mejora de Friedman para el potencial splits, "absolute_error" para el error absoluto medio, que minimiza la pérdida L1 usando la mediana de cada nodo terminal, y "poisson" que usa la reducción en la desviación de Poisson para encontrar divisiones. El entrenamiento con "absolute_error" es significativamente más lento que cuando se usa "squared_error", default = "squared_error".
- **random_state:** Controla tanto la aleatoriedad del arranque de las muestras utilizadas al construir árboles (si bootstrap = True) como el muestreo de las características a considerar cuando se busca la mejor división en cada nodo. Int, instancia de RandomState o None, predeterminado = None.

El modelo creado utilizó todos los hiperparámetros por defecto a excepción de los mencionados anteriormente *criterion*, *n_estimators* y *random_state*; los cuales se utilizaron los valores de absolute_error para *criterion*, un rango de números enteros (2, 4, 8, 16, 32, 64, 128, 256) para los *n_estimators* y 0 para *random_state*. No se realiza el gráfico del árbol de decisión ya que esta es una propiedad de los clasificadores y no para los regresores. A partir de nuestro dataset de train se evaluó el MAE para cada uno de los valores de los *n_estimators*, esta representación gráfica se muestra a continuación:

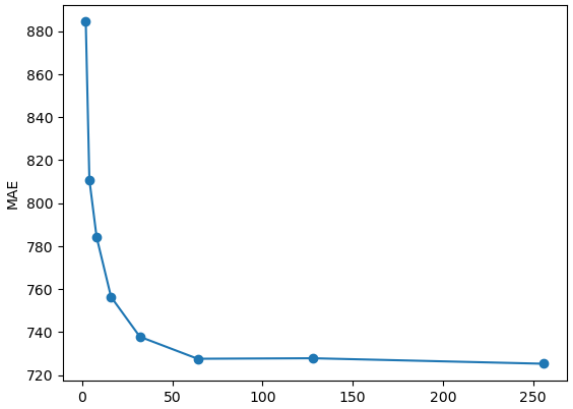

Figura 10: MAE para cada estimador del modelo Random Forest Regresión.

Fuente: Elaboración propia

Se evidencia que el MAE decrece conforme avanza el valor de n_estimators, pero a partir de 64 casi que el valor del MAE permanece constante, al realizar la búsqueda del mínimo MAE se encuentra con n_estimators = 256, este valor será el utilizado para entrenar y ajustar el modelo.

4.3.4.3. SERIE TEMPORAL FORECASTERAUTOREG

Nuestra variable dependiente será la *TASA_INCIDENCIA* que será la variable que queremos predecir las demás variables de nuestro dataset final serán las variables independientes. En comparación a los dos modelos anteriores se divide nuestro dataset de train y test en un rango de fecha tomando el set de train desde 2020-01 hasta 2022-02-01, y el dataset de test será desde 2022-03 hasta 2023-02. Se creo un modelo de serie temporal a nivel nacional (España). Nuestra partición de datos tiene la siguiente forma como se muestra a continuación:

```
Fechas train : 2020-01-01 00:00:00 --- 2022-02-01 00:00:00  (n=26)
Fechas test  : 2022-03-01 00:00:00 --- 2023-02-01 00:00:00  (n=12)
```

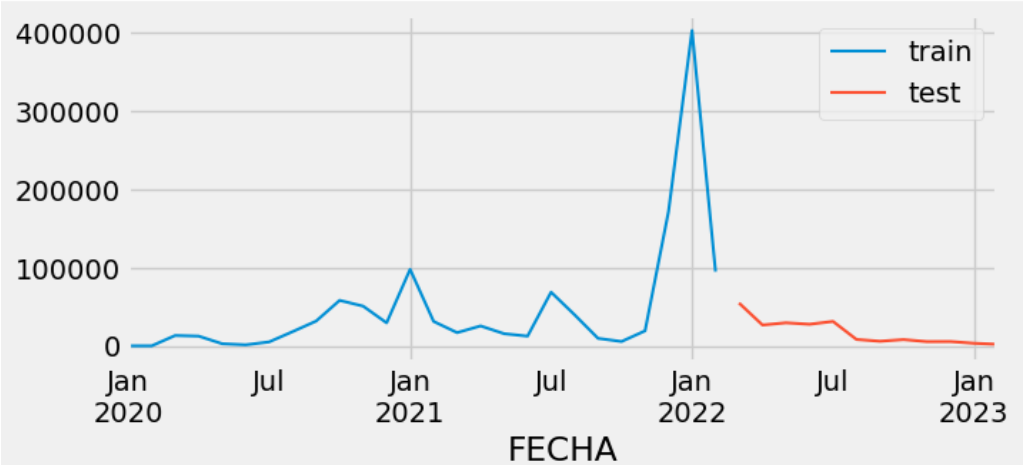

Figura 11: División train y test ForecasterAutoreg.

Fuente: Elaboración propia

Para la creación de este modelo se utilizó ForecasterAutoreg de la librería skforecast, este modelo convierte cualquier regresor compatible con la API scikit-learn en un pronosticador recursivo autorregresivo (de varios pasos) y cuenta con los siguientes hiperparámetros:

- **regressor:** Una instancia de un regresor compatible con la API de scikit-learn.
- **lags(int, list, 1D np.array, range):** retrasos utilizados como predictores. El índice comienza en 1, por lo que el retraso 1 es igual a t-1. int: incluye retardos de 1 a lags (incluido). List o np.array: incluir solo los retrasos presentes en lags.

El modelo creado utilizó como regressor un RandomForestRegressor (con los mismos parámetros descritos en el modelo anterior esto con el fin de contrastar ambos modelos bajo las mismas condiciones) y lags = 10.

La descomposición de nuestra serie temporal nos muestra que a lo largo del tiempo no hay una tendencia clara de la tasa de incidencia de covid-19; existe una estacionalidad con picos en enero en todos los años, esto en parte es cierta debido a fenómenos sociales de fin de año y que el incremento de casos de covid se vea reflejado en el mes de enero, pero estos picos no son de la misma magnitud en los eneros de cada año, esto no se evidencia en la estacionalidad de la serie temporal; el residuo o componente aleatorio (que no pudo ser explicado por la tendencia o estacionalidad) es bastante atípico o invariante, esto quiere decir que muy posiblemente nuestro modelo no pueda predecir de forma correcta las tasas de incidencia futuras y que nuestras métricas de evaluación no van a ser las mejores, en el apartado de resultados se observaran estas métricas. Los componentes descritos se muestran a continuación:

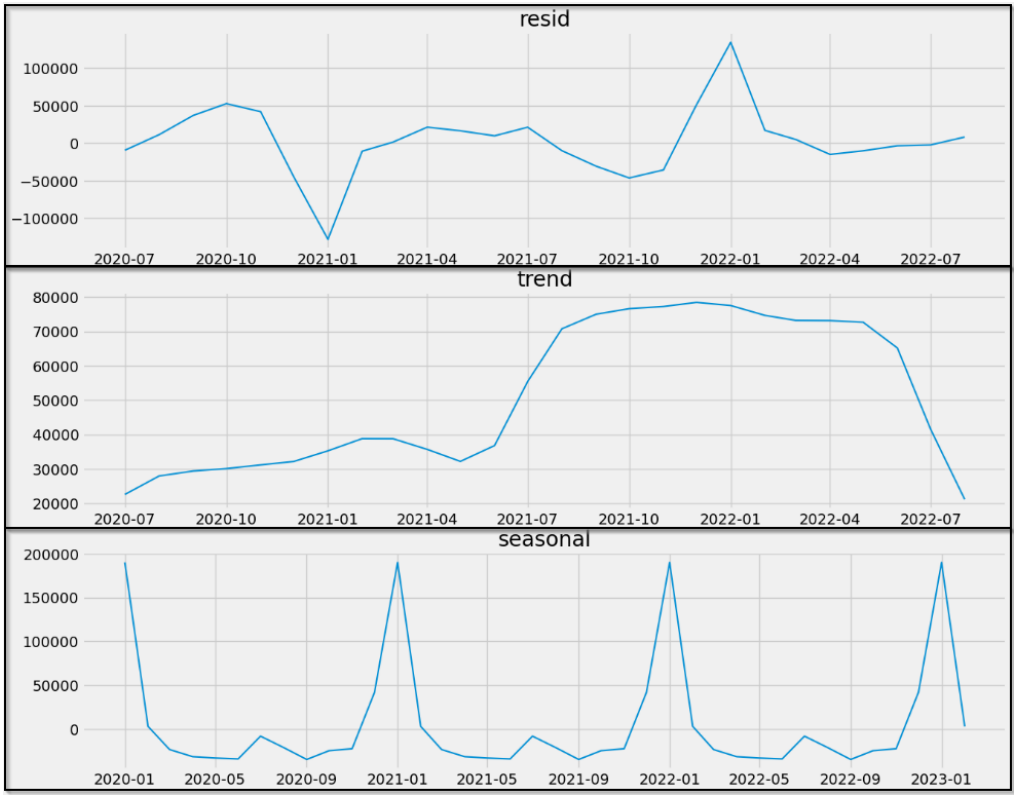

Figura 12: Descomposición serie temporal ForecasterAutoreg.

Fuente: Elaboración propia

4.3.4.4. K-MEANS

Este es un tipo de modelo no supervisado, es decir, no vamos a tener una etiqueta para predecir y aunque la tenemos no la vamos a utilizar, es decir, la tasa de incidencia no será una etiqueta a predecir, por el contrario, será una variable más en nuestro modelo, esto para generar clúster o una clasificación de los datos, con el fin de encontrar patrones de comportamiento o características que diferencien cada uno de los cluster, el k-means se pudo haber utilizado en un comienzo para agrupar los datos en grupos más específicos y realizar el entrenamiento de nuestros modelos anteriores, es decir agrupar las 52 provincias en grupos con características similares. No se optó por este camino ya que se quiso evidenciar el comportamiento de los factores climáticos por provincia, comenzando por una granularidad más baja, se planteará un trabajo futuro comenzando con una clusterización de los datos. La variable *FECHA* será eliminada ya que como patrón no nos interesa, y la variable *PROVINCIA* será transformada como index en nuestro dataset.

Para la creación de este modelo se utilizó KMeans de la librería sklearn clúster, este modelo cuenta con 9 hiperparámetros, de los cuales se utilizaron los siguientes:

- **n_clusters:** El número de clústeres a formar, así como el número de centroides a generar. Int, por defecto = 8.
- **random_state:** Determina la generación de números aleatorios para la inicialización del centroide. Use un int para hacer que la aleatoriedad sea determinista. Int, RandomState instance o None, por defecto = None.

El modelo creado utilizó todos los hiperparámetros por defecto a excepción de los mencionados anteriormente, para n_cluster se utilizó un rango de valores de 1 a 10 para realizar la gráfica del codo y validar su inercia respecto al modelo, con respecto a random_state se tomo un valor de 42, se utilizó un valor int para que la aleatoriedad sea determinista y evitar distintos valores en cada iteración de nuestro valor de n_cluster. La representación de la gráfica del codo se muestra a continuación:

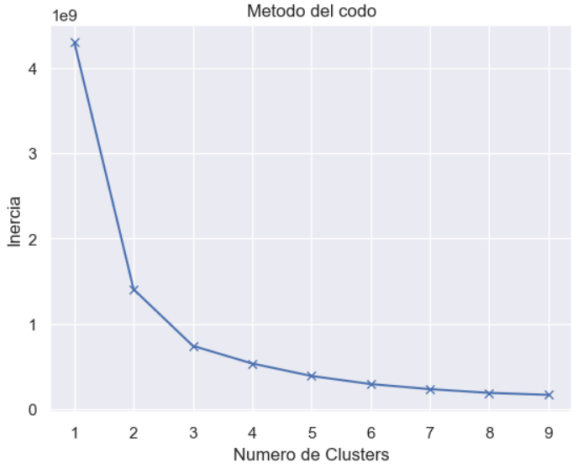

Figura 13: Método del codo K-means.

Fuente: Elaboración propia

Basado en la figura anterior, se debe tomar un n_cluster igual a 3 o 4, que es donde comienza la curvatura del codo, se realizaron pruebas con 4 clúster y se evidencio que

dos de estos clúster tenían características muy similares, por tal razón el valor optimo de n_cluster será tomado como 3. Nuestro modelo se entrena y se ajusta bajo el parámetro anterior y se le asigna el clúster correspondiente a cada uno de los registros de nuestro dataset.

4.4. RESULTADOS

Para cada uno de los modelos descritos anteriormente se mostrará su gráfica de predicción versus los valores reales, así como las mismas métricas de evaluación (R^2, MAE y RMSE) para cada modelo y al final mediante un apartado de comparación y contraste se evaluará cual es el modelo que mejor se ajusta a nuestros datos y los beneficios que este aporta.

- **REGRESIÓN LINEAL MÚLTIPLE**

  Realizando una comparación de los valores reales con los predichos por nuestro modelo, notamos la recta que creo el modelo se encuentra desfasada y desajustada al valor real de los datos, por la naturaleza de los datos se puede crear una mejor recta que explique el comportamiento de nuestra variable dependiente (TASA_INCIDENCIA) según la siguiente imagen, pero bajo nuestras variables de estudio (meteorológicas) no se puede generar una mejor recta que explique la TASA_INCIDENCIA, se debería añadir más variables con una mayor importancia que aporten a nuestro modelo.

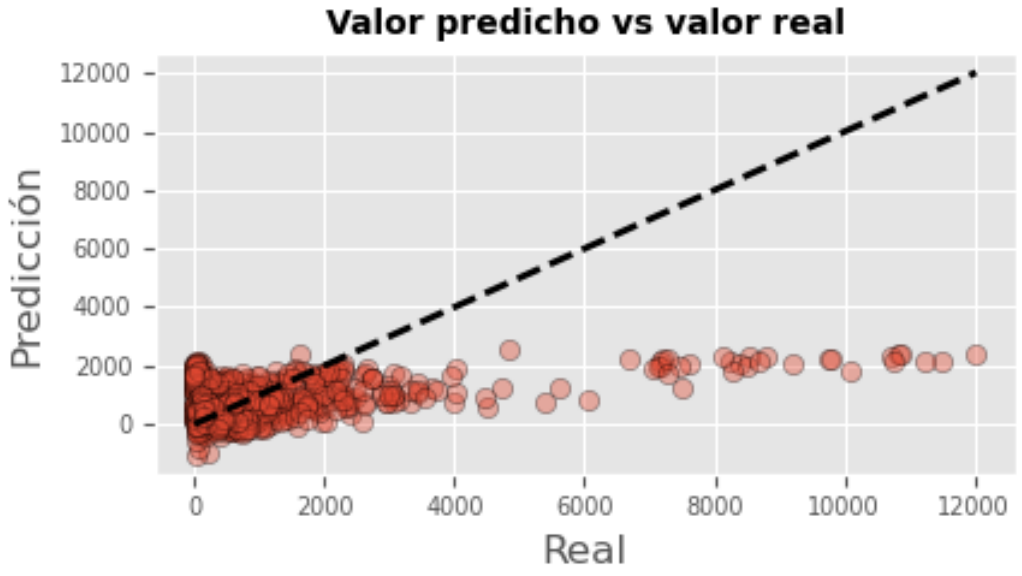

Figura 14: Valor predicho vs Valor real Modelo Regresión Lineal Múltiple.
Fuente: Elaboración propia

Las métricas calculadas para la evaluación del modelo son:

- $R^2 = 0{,}15$
- $MAE = 756{,}23$
- $RMSE = 1337{,}30$

Se concluye que todas las variables introducidas como predictores tienen un R^2 bajo (0.15) es decir es capaz de explicar en un 15.2 % la variabilidad observada de la tasa

de incidencia. Las predicciones del modelo se alejan en promedio 756,23 unidades de nuestro valor real esto es explicado por el MAE ya que calcula el error en la misma escala de los datos. Por el contrario, el valor RMSE de 13373,30 que mide la diferencia promedio al cuadrado entre los valores estimados y el valor real, penaliza valores extremos o valores atípicos es por esta razón que es un valor más alto que el MAE, para ambos casos lo ideal es lo más cercano a cero y saber la naturaleza de los datos, para nuestro caso la variable dependiente TASA_INCIDENCIA, está entre el rango de 0,0 hasta 12025,3, teniendo un promedio de 728,86.

- **RANDOM FOREST REGRESIÓN**

Realizando una comparación de los valores reales con los predichos por nuestro modelo de RandomForestRegressor, notamos que existen unos picos los cuales el modelo no puede predecir, esto es debido a la naturaleza de los datos ya que durante la pandemia hubo picos de infección los cuales hacen que sea distintos a sus homólogos en los mismos meses de diferente año. Estos picos se deben a factores sociales (eventos, movilidad, época decembrina, apertura de establecimientos, etc) los cuales tienen muy poca correlación con los factores climáticos. La comparación se muestra en el siguiente gráfico junto con el n_estimators utilizado y el resultado del MAE al evaluar el modelo.

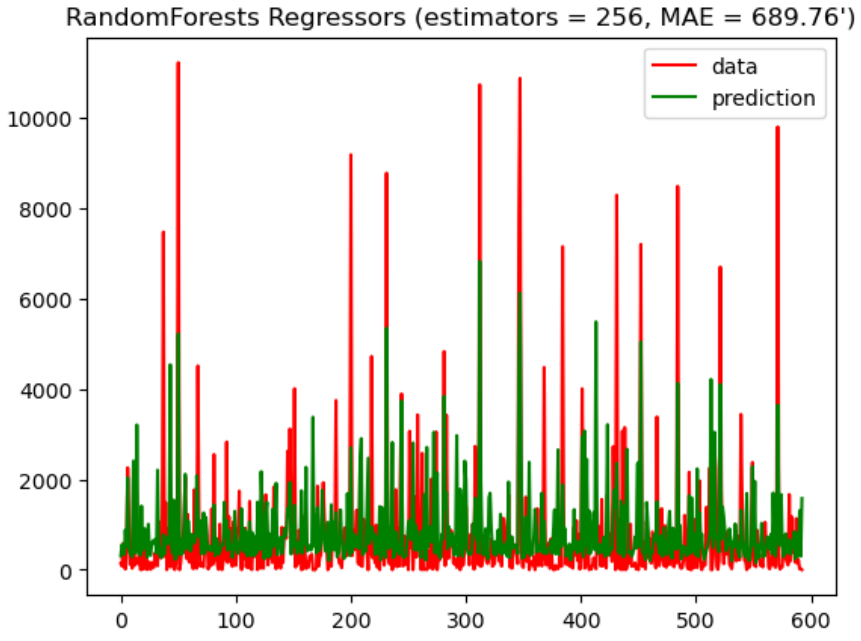

Figura 15: Valor predicho vs Valor real Modelo Random Forest regresión.
Fuente: Elaboración propia

Las métricas calculadas para la evaluación del modelo son:

- $R^2 = 0{,}338$
- $MAE = 689{,}75$
- $RMSE = 1169{,}12$

Se concluye que todas las variables introducidas como predictores tienen un R^2 bajo (0.338) es decir es capaz de explicar en un 33.8 % la variabilidad observada de la tasa de incidencia. Las predicciones del modelo se alejan en promedio 689,75 unidades de nuestro valor real esto es explicado por el MAE ya que calcula el error en la misma escala de los datos. Por el contrario, el valor RMSE de 1169,12.

La importancia de las variables que utilizamos en nuestro modelo se muestra en la siguiente tabla:

Tabla 12: Importancia de las variables en el modelo.

Variable	Importancia
TEMP_MED	0,138512
PRES_MIN	0,116705
SOL	0,116503
PREC	0,112269
RACHA	0,102015
DIR	0,096510
VEL_MEDIA	0,083530
MONTH	0,076722
ALTITUD	0,037044
HASH_3	0,029064
HASH_4	0,024272
HASH_2	0,018971
HASH_0	0,018602
HASH_5	0,017449
HASH_1	0,011834

Fuente: Elaboración Propia

Se evidencia que la variable *TEMP_MED, SOL, PREC, PRES_MIN* toman mayor importancia a nuestro modelo, es decir a portan mayor información para predecir la tasa de incidencia de covid-19, las variables HASH no toman mayor importancia ya que el conjunto de estas seis toman la representación de cada uno de los valores que tenía la variable *PROVINCIA* anteriormente, por otro lado, las provincias son las encargadas de segmentar el dataset ya que cada una tiene sus propias condiciones meteorológicas.

- **SERIE TEMPORAL FORECASTERAUTOREG**

Luego de entrenar y ajustar nuestro modelo se realiza la comparación de los valores reales con los predichos por nuestro modelo ForecasterAutoreg, notamos que las predicciones están más alejadas con respecto a la real, es decir, esta prediciendo una tasa de incidencia más alta para el periodo 2022-03 en adelante, pero tiene una pequeña similitud con respecto a la tendencia de la gráfica. Las razones de las diferencias son las mismas expuestas en los modelos anteriores.

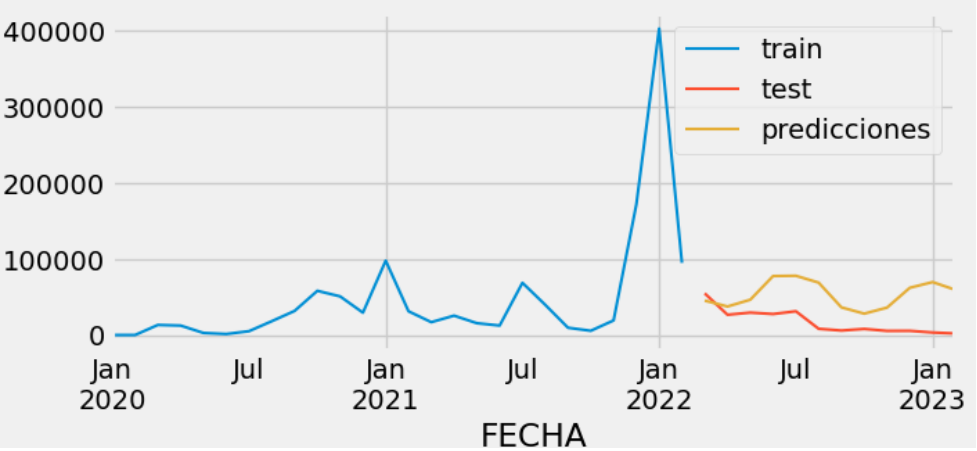

Figura 16: Valor predicho vs Valor real Modelo ForecasterAutoreg.

Fuente: Elaboración propia

Las métricas calculadas para la evaluación del modelo son:

- $R^2 = -6{,}46$
- $MAE = 37937{,}62$
- $RMSE = 42756{,}51$

El modelo tiene un R^2 negativo de -6,46 es decir nuestro modelo no es capaz de explicar la variabilidad observada de la tasa de incidencia. Las predicciones del modelo se alejan en promedio 37937,62 unidades de nuestro valor real esto es explicado por el MAE ya que calcula el error en la misma escala de los datos. Por el contrario, el valor RMSE de 42756,51.

Mediante el método **get_feature_importances()** nos devuelve la importancia de las características basadas en impurezas del modelo almacenado en el pronosticador. Sólo es válido cuando el pronosticador ha sido entrenado usando como regresor GradientBoostingRegressor o RandomForestRegressor, para nuestro caso este último.

Tabla 13: Importancia de las variables ForecasterAutoreg.

Variable	Importancia
lag_1	0,097890
lag_2	0,092794
lag_3	0,377607
lag_4	0,069648
lag_5	0,047422
lag_6	0,113457
lag_7	0,077389
lag_8	0,059880
lag_9	0,031993
lag_10	0,031921

Fuente: Elaboración Propia

Se evidencian en la tabla 8 que los 10 lags que se utilizaron para entrenar el modelo en donde para cada lag devolvió la importancia de todas las características, para el lag_3 tomo más importancia las características o variables para explicar nuestra variable dependiente, seguido por el lag_6, lag_1 y lag_2 respectivamente; los lags que tomaron menor importancia fueron los lag_9 y lag_10.

- **K-MEANS**

Luego de entrenar y ajustar nuestro modelo y de asignar el clúster correspondiente a cada registro, se agrupa el datataset por medio del clúster y las demás variables como medida se tomo la media, el resultado fue el siguiente:

cluster	ALTITUD_MED	TEMP_MED_TO	PREC_MED	DIR_MED	VEL_MEDIA	RACHA_MED	PRES_MIN_MED	SOL_MED	TASA_INCIDENCIA_MED
0	391.701261	16.432753	1.555208	32.199383	3.105878	9.874806	970.835240	7.427960	348.876726
1	446.272727	7.266569	1.227126	31.866569	2.430792	8.177273	972.132185	6.170455	8685.637500
2	439.271277	11.360588	1.381100	33.481717	3.028772	9.668964	967.373605	6.083078	2391.619149

Figura 17: Características meteorológicasológicas por clúster.

Fuente: Elaboración propia

Se evidencia que el clúster 0 tiene una tasa de incidencia más baja, seguido por el clúster 2 y el clúster 1 que tiene una tasa de incidencia más alta. El clúster 0 tiene una temperatura media superior, seguido por el clúster 2 y con una temperatura más baja el clúster 3; de igual forma en el mismo orden tiene más horas de sol y una racha mayor el clúster 0; la altitud juega un papel importante en la determinación de factores climáticos de una región, ya que a menor altura es más probable que tenga una temperatura mayor, es por esta razón que el clúster 1 a pesar de tener un poco más horas de sol con respecto al clúster 2, tiene una temperatura menor, ya que está a una altitud mayor que el clúster 2. Estas son las variables que toman más importancia, por experiencia en los modelos anteriores y se evidencia un patrón claro en cada clúster creado.

Para evidenciar este fenómeno gráficamente, se realiza el análisis PCA para transformar nuestras variables y dejar dos componentes (X, Y), estos dos componentes se transforman en un dataframe y se les asignan unas nuevas columnas para tener más información al momento de graficar X y Y. Las columnas agregadas a nuestro PCA fueron el clúster correspondiente la temperatura promedio, la tasa de incidencia y un color correspondiente a cada clúster, esto para evidenciar mejor el fenómeno de segmentación.

La representación gráfica del PCA con los clúster se muestra a continuación, en donde se evidencia claramente la segmentación y separación de los tres clústers definidos, en donde nuestro clúster 0 (azul) tiene unas tasas de incidencia menor; el clúster 1 (amarillo) tiene unas tasas de incidencias más altas y el clúster 2 (verde) tiene unas tasas de incidencia intermedias, las características meteorológicasológicas de cada clúster se explicaron en la figura **17**.

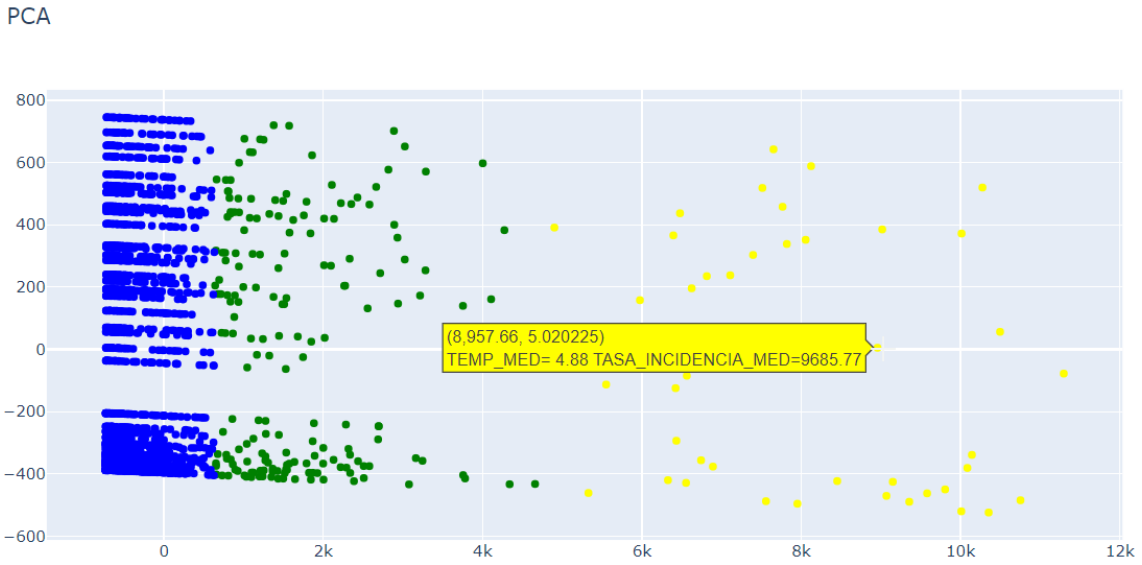

Figura 18: PCA K-means.

Fuente: Elaboración propia

En la figura anterior se muestra en total 1976 puntos, que equivalen a los 38 meses de cada uno de las 52 provincias. Se realizó el estudio agrupando estos datos por clúster y por provincia, llegando a una cantidad de 144 registros, y aquí el fenómeno claro e interesante es que por lo general casi todas las provincias hacen parte de los tres clúster ya que sus condiciones meteorológicas varían a lo largo del año, pero por regla general cuando cada provincia perteneció al clúster 0 tenía menor tasa de incidencia, por el contrario, cuando estaba en el clúster 1 tenía mayor tasa de incidencia.

4.4.1. COMPARACIÓN ENTRE MODELOS

Realizando la comparación entre modelos basado en las métricas seleccionadas, la siguiente tabla muestra los resultados obtenidos:

Tabla 14: Métricas de error de los modelos.

Modelo	R^2	MAE	RMSE
Regresión Lineal Múltiple	$0,15$	$756,23$	$1337,30$
Random Forest Regresión	$0,33$	$689,75$	$1169,12$
Serie Temporal Forecasterautoreg	$-6,46$	$37937,62$	$42756,51$

Fuente: Elaboración Propia

En general el modelo que tiene una mayor explicabilidad sobre nuestra variable dependiente es el modelo Random Forest Regresión, este modelo explica el 33 % de la variabilidad, seguido por la regresión Lineal múltiple con un 15 % y de últimas Forecasterautoreg con un -6 %, es posible que el valor de R^2 sea negativo en algunos casos, como en modelos de ajuste no lineal o en modelos con un número insuficiente de observaciones, en el contexto

de un modelo Forecasterautoreg, un valor de R^2 negativo sugiere que el modelo no es adecuado para predecir los valores futuros de la variable dependiente a partir de sus valores pasados. En este caso, se debe considerar la revisión del modelo o la inclusión de otras variables predictoras que puedan mejorar su capacidad de predicción.

Así mismo para el modelo Random Forest las predicciones se alejan en promedio $689,75$ unidades de nuestro valor real, seguido por la regresión lineal con $756,23$ y para Forecasterautoreg con $37937,62$. Debemos tener en cuenta que el MAE calcula el error en la misma escala de los datos es decir un MAE alto no necesariamente significa un modelo impreciso, ya que dependerá de la naturaleza y la escala de nuestros datos para nuestro caso la variable dependiente TASA_INCIDENCIA, está entre el rango de $0,0$ hasta $12025,3$, teniendo un promedio de $728,86$; de igual forma el modelo que presento un menor valor de RMSE fue Random Forest con $1169,12$, seguido por la regresión lineal con $1337,30$ y por último $42756,51$. El RMSE penaliza valores extremos o valores atípicos es por esta razón es un valor más alto que el MAE.

Con respecto a muestro modelo de regresión lineal múltiple las variables que más tuvieron importancia fueron SOL, PRES_MIN , TEMP_MED y PREC respectivamente, cada una con una gran influencia en el modelo; con respecto a muestro modelo de Random Forest regresión las variables que más tuvieron importancia fueron TEMP_MED, PRES_MIN , SOL y PREC respectivamente, es decir a portan mayor información para predecir la tasa de incidencia de covid-19; con respecto a muestro modelo Forecasterautoreg al ser una serie temporal no toma como importancia las variables de forma independiente, por el contrario toma los lags, es decir los retrasos utilizados en el predictor $(t-1)$, de los 10 lags que tomamos para entrenar el modelo los que tuvieron una mayor importancia o aportaron mayor información fueron lag_3, lag_6, lag_1 y lag_2 respectivamente

4.4.2. FLUJO DE AUTOMATIZACIÓN PIPELINE

Se creó un proceso de automatización, el cual será un pipeline que tendrá cada una de las etapas del proceso KDD, esto con el fin de tener un CI/CD que nos permite tener mayor rapidez y eficiencia en la entrega de software mediante la automatización del proceso; mayor calidad del software al automatizar permite a los equipos de desarrollo detectar y corregir errores más rápidamente, lo que reduce la posibilidad de que se introduzcan errores en el software; mayor colaboración y comunicación; mayor flexibilidad y escalabilidad, puede manejar grandes cantidades de código y usuarios sin problemas; mayor seguridad se puede implementar fácilmente un conjunto de políticas de seguridad para garantizar que el software se entregue de manera segura y confiable.
Nuestro pipeline se construyó todo sobre GitHub y la arquitectura se muestran en las siguientes dos figuras:

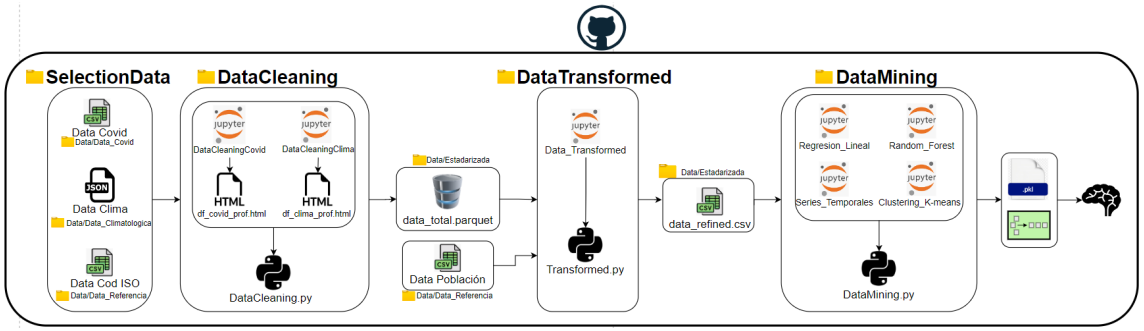

Figura 19: Arquitectura creación modelos ML en GitHub.

Fuente: Elaboración propia

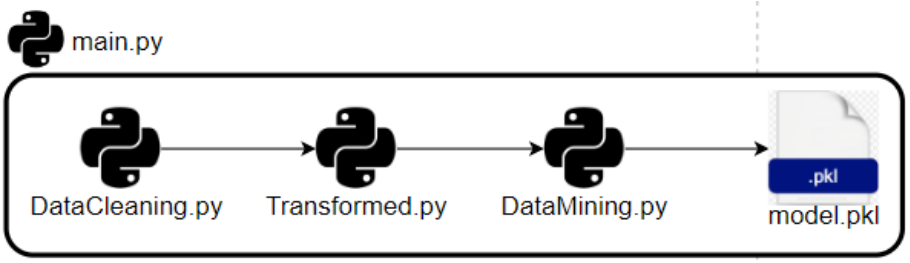

Figura 20: Pipeline de automatización en GitHub.

Fuente: Elaboración propia

En la figura **19** se muestra cada una de las etapas de nuestro proceso KDD con su ruta correspondiente en GitHub, junto con sus archivos correspondientes que servirán de insumos en la siguiente etapa del proceso. Algunas etapas cuentan con unos archivos de jupyter notebook que sirvieron para el análisis exploratorio para la toma de decisiones, basados en los resultados de los notebooks se construyeron unos archivos Python para realizar la ejecución del pipeline de forma automática, como se muestra en la figura **20**.
La finalidad de la automatización en este trabajo es que cualquier persona con tan solo clonar el repositorio puede realizar la ejecución de todo el proceso por medio de un archivo que orquesta las etapas (main.py), así como la ejecución de los notebooks de análisis exploratorios y poder obtener valor y conocimiento a partir de los resultados. El pipeline está adaptado para una futura actualización en la base de datos (Covid, clima, etc) y obtener nuevos resultados de forma automatiza, cumpliendo así la esencia del CI/CD. Hay que tener en cuenta la limitante de almacenamiento de GitHub, para este caso se alcanzó a almacenar todos los insumos y archivos intermedios, razón por la cual algunos están en formato parquet ya que tiene un formato de compresión mucho mayor.

5. CONCLUSIÓN

Desde el comienzo del proyecto se planteo identificar la posible influencia de las condiciones meteorológicas en la propagación del virus de COVID-19 en España, para ello se crearon unos modelos los cuales pueda predecir o explicar nuestra variable objetivo, es decir la tasa de incidencia, que se creó a partir de la población de cada provincia y el número de casos de cada mes. Para ello se han utilizado diferentes modelos de predicción como regresión lineal múltiple, random forest regresión, series temporales Forecasterautoreg y k-means.

En un principio se realizó un análisis de la calidad de los datos de las diferentes fuentes, para validar la utilidad en el cumplimiento del objetivo del proyecto, tras otros análisis estadísticos, cantidad de nulos, correlación entre las variables y refinamiento de los datos, se logró la unión de las diferentes fuentes de datos en set de datos final el cuál no se tuvieron en cuenta muchas variables iniciales y pasar de una granularidad diaria a mensual, debido a que los resultados fueron muy similares y el procesamiento de los modelos computacionalmente eran más exigentes.

Tras la realización y evaluación de nuestros modelos supervisados bajo nuestras métricas seleccionadas ninguno de nuestros modelos ha sido capaz de predecir una variabilidad de los datos superior al 35 % (siendo el mejor random forest, seguido de la región lineal y por último la serie temporal), y la distancia entre los datos reales con los predichos no son tan bajos, de igual forma de todas nuestras variables seleccionadas de nuestro data set final solo cobraron importancia en los modelos SOL, PRES_MIN, TEMP_MED y PREC; lo que significa que deberíamos incluir a nuestros modelos otras variables que logren explicar de mejor manera nuestra variable objetivo, por tal razón se puede concluir que los factores climáticos tienen una influencia en la propagación del Covid en España, pero no son el factor principal en la propagación de este. De ahí a que las principales recomendaciones de la OMS sea evitar la interacción social con personas infectadas.

Para confirmar nuestro objetivo principal se creo un modelo de clasificación k-means en donde se entreno con todas las variables de nuestro data set final incluyendo la tasa de incidencia de Covid, mediante el análisis del codo y pruebas, se llegó a un óptimo de 3 clústeres, cada uno con sus propias condiciones meteorológicas específicas, pero el clúster que tenía una tasa de incidencia mayor era aquel que tiene unas condiciones meteorológicas menores en algunas variables de alta importancia (SOL, TEMP_MED) y el clúster con menor tasa de incidencia era aquel que tiene unas condiciones meteorológicas mayores en algunas variables de alta importancia, al graficar cada uno de los datos de cada clúster se ve claramente la segmentación y la agrupación de estos.

Todo este proceso ha sido desarrollado utilizando la metodología KDD y se automatizó cada una de las fases de la metodología mediante el lenguaje de programación Python haciendo uso de GitHub como repositorio para almacenar las fuentes y cada script de ejecución de cada de etapa, como resultado cada persona que clone el repositorio podrá hacer una ejecución inmediata para ver los resultados, también se pensó para realizar una CI/CD conforme se vayan actualizando las fuentes de los casos reportados para obtener resultados más actuales e ir contrastando.

5.1. TRABAJOS FUTUROS

Algunas acciones de mejora y trabajos futuros para este tipo de proyectos es integrar otro tipo de variables o características como lo son atributos de interacción, movilidad, eventos sociales y deportivos, para tener una mejor explicabilidad a nuestros modelos, este tipo de características. Realizar otros métodos de predicción o clasificación, pero no iniciando desde datos clasificados por provincia, sino por el contrario acotar el análisis, a datos clasificados por grupos o clúster, es decir, hacer como primera medida una clusterización sobre todas las provincias y extraer los grupos con características similares y aplicar los modelos para poder obtener métricas más optimas.

Se propone la creación de otro tipo de modelos como lo son XGBoost, que puede realizar una búsqueda exhaustiva de hiperparámetros utilizando técnicas como la validación cruzada y la optimización bayesiana para encontrar la combinación óptima que maximice la precisión del modelo de predicción además de tener un enfoque que puede capturar relaciones complejas y no lineales entre las variables; así como un modelo Forecasting autorregresivo recursivo con variables exógenas (ARIMAX) es decir, cuando se desea tener en cuenta y aprovechar información adicional que puede influir en la variable de interés. Permite una mayor flexibilidad y precisión en la predicción al incorporar variables exógenas en el modelo autorregresivo. Sin embargo, es importante tener en cuenta que la selección adecuada de las variables exógenas y la validación del modelo son aspectos críticos para obtener resultados precisos y confiables.

6. REFERENCIAS

Araujo, M. B., y Naimi, B. (2020). Spread of sars-cov-2 coronavirus likely to be constrained by climate. *MedRxiv*, 2020–03.

Awad, M., Khanna, R., Awad, M., y Khanna, R. (2015). Support vector regression. *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*, 67–80.

Benítez Iglesias, R., Escudero Bakx, G., Kanaan Izquierdo, S., Masip Rodó, D., y Cencerrado Barraqué, A. (2018). Inteligencia artificial, febrero 2018.

Bobadilla, J. (2021). *Machine learning y deep learning: usando python, scikit y keras*. Ediciones de la U.

Chen, S., Yang, J., Yang, W., Wang, C., y Bärnighausen, T. (2020). Covid-19 control in china during mass population movements at new year. *The Lancet*, *395*(10226), 764–766.

Chimmula, V. K. R., y Zhang, L. (2020). Time series forecasting of covid-19 transmission in canada using lstm networks. *Chaos, Solitons & Fractals*, *135*, 109864.

Ciotti, M., Ciccozzi, M., Terrinoni, A., Jiang, W.-C., Wang, C.-B., y Bernardini, S. (2020). The covid-19 pandemic. *Critical reviews in clinical laboratory sciences*, *57*(6), 365–388.

Ciria Mayordomo, D. (2021). Covid sent: Identificación de sentimientos en titulares de noticias del covid-19.

cnecovid. (2023). *Datos oficiales covid-19, tipo @ONLINE*. Descargado 2022-12-11, de `https://cnecovid.isciii.es/covid19/#documentaci%C3%B3n-y-datos`

de la Fuente Fernández, S. (2013). *Series temporales*. Descargado 2023-03-27, de `chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.estadistica.net/PAU2/series-temporales.pdf`

Fayyad, U., Piatetsky-Shapiro, G., y Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, *17*(3), 37–37.

Gu, Y. (2021a). *Covid-19 projections using machine learning, tipo @ONLINE*. Descargado 2021-03-24, de `https://covid19-projections.com/`

Gu, Y. (2021b). *Modelo seir, tipo @ONLINE*. Descargado 2021-03-24, de `https://`

github.com/youyanggu/yyg-seir-simulator

Gu, Y. (2021c). *proyecciones, tipo @ONLINE.* Descargado 2021-03-24, de https://github.com/youyanggu/covid19_projections/tree/master/projections

Hao, B., Hu, Y., Sotudian, S., Zad, Z., Adams, W. G., Assoumou, S. A., ... Paschalidis, I. C. (2022). Development and validation of predictive models for covid-19 outcomes in a safety-net hospital population. *Journal of the American Medical Informatics Association*, *29*(7), 1253–1262.

He, S., Peng, Y., y Sun, K. (2020). Seir modeling of the covid-19 and its dynamics. *Nonlinear dynamics*, *101*, 1667–1680.

Hernandez-Matamoros, A., Fujita, H., Hayashi, T., y Perez-Meana, H. (2020). Forecasting of covid19 per regions using arima models and polynomial functions. *Applied soft computing*, *96*, 106610.

Holmäng, A., y von Grothusen, A. (2021). *Intäktsestimering med hjälp av maskininlärning.*

INE. (2023). *Población por provincias, tipo @ONLINE.* Descargado 2023-02-15, de https://www.ine.es/jaxi/Datos.htm?path=/t20/e245/p08/l0/&file=03003.px#!tabs-tabla

Joaquín Amat Rodrigo, J. E. O. (2021). *Skforecast: forecasting series temporales con python y scikit-learn.* Descargado 2023-03-10, de https://www.cienciadedatos.net/documentos/25_regresion_lineal_multiple#Bibliograf%C3%ADa

oficial del estado, B. (2015). *Ley 18/2015, de 9 de julio, tipo @ONLINE.* Descargado 2022-12-13, de https://www.boe.es/boe/dias/2015/07/10/pdfs/BOE-A-2015-7731.pdf

opendata. (2023). *Datos climaticos por provincia, tipo @ONLINE.* Descargado 2022-12-14, de https://opendata.aemet.es/centrodedescargas/productosAEMET

Plazas, D. (2023). *Perfilamiento data covid-19.* Descargado 2023-01-07, de https://htmlpreview.github.io/?https://github.com/Danilo0221/TFM/blob/main/DataCleaning/df_covid_prof.html

R, S. E. (2023). *Understand random forest algorithms with examples.* Descargado 2023-03-20, de https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

Rativa, I. N. S. (2020). Técnicas de aprendizaje automático aplicadas en los sistemas de predicción. *Tecnología Investigación y Academia*, *8*(1), 37–53.

Rodrigo, J. A. (2016). *Introducción a la regresión lineal múltiple.* Descargado 2023-03-02, de `https://www.cienciadedatos.net/documentos/25_regresion_lineal_multiple#Bibliograf%C3%ADa`

Sakly, H., Al-Sayed, A. A., Said, M., Loussaief, C., Seekins, J., y Sakly, R. (2023). Artificial intelligence and big data for covid-19 diagnosis. En *Trends of artificial intelligence and big data for e-health* (pp. 83–119). Springer.

Toğa, G., Atalay, B., y Toksari, M. D. (2021). Covid-19 prevalence forecasting using autoregressive integrated moving average (arima) and artificial neural networks (ann): case of turkey. *Journal of infection and public health*, *14*(7), 811–816.

Torres-Degró, A. (2011). Tasas de crecimiento poblacional (r): Una mirada desde el modelo matemático lineal, geométrico y exponencial. *CIDE digital*, 143–162.

Valenzuela González, G., y cols. (2022). Aprendizaje supervisado: Métodos, propiedades y aplicaciones.

Vandermeer, J. H., y Goldberg, D. E. (2013). *Population ecology: first principles*. Princeton University Press.

Vannieuwenhuyze, A. (2020). Inteligencia artificial fácil. *Machine learning y Deep learning prácticos, traducido por GOYANES ARNEDO, B., ediciones ENI, Cornellà de Llobregat*.

Wang, J., Tang, K., Feng, K., Lv, W., y cols. (2020). High temperature and high humidity reduce the transmission of covid-19. *Available at SSRN*, *3551767*, 2020b.

ANEXOS I

Tabla 15: Variables y descripción del set de datos de COVID-19 por provincias.

Variable	Descripción
provincia_iso	Código ISO de la provincia de residencia. NC (no consta)
sexo	Sexo de los casos: H (hombre), M (mujer), NC (no consta)
grupo_edad	Grupo de edad al que pertenece el caso: 0-9, 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, $\geq$ 80 años. NC: no consta. Después del 28 de Marzo solo grupos de más de 60 años.
fecha	**Casos:** En los casos anteriores al 11 de mayo, se utiliza la fecha de diagnóstico, en su ausencia la fecha de declaración a la comunidad y, en su ausencia, la fecha clave (fecha usada para estadística por las CCAA). En los casos posteriores al 10 de mayo, en ausencia de fecha de diagnóstico se utiliza la fecha clave1. **Hospitalizaciones, ingresos en UCI, defunciones:** los casos hospitalizados están representados por fecha de hospitalización (en su defecto, la fecha de diagnóstico, y en su defecto la fecha clave, los casos UCI por fecha de admisión en UCI (en su defecto, la fecha de diagnóstico, y en su defecto la fecha clave) y las defunciones por fecha de defunción (en su defecto, la fecha de diagnóstico, y en su defecto la fecha clave).
num_casos	Número de casos notificados confirmados con una prueba diagnóstica positiva de infección activa (PDIA) tal como se establece en la Estrategia de detección precoz, vigilancia y control de COVID-19 y además los casos notificados antes del 11 de mayo que requirieron hospitalización, ingreso en UCI o fallecieron con diagnóstico clínico de COVID19, de acuerdo a las definiciones de caso vigentes en cada momento.
num_hosp	Número de casos hospitalizados
num_uci	Número de casos ingresados en UCI
num_def	Número de defunciones

Fuente: RNVD