****

Realizado: **Juan Javier Hidalgo Gómez.**

Tutor: **José María Sallan Leyes**

Modelos Predictivos y Análisis de Sentimientos en Airbnb.

**Índice General**

Introducción.

Objetivo.

Metodología.

1. Dataset Airbnb.

1.1 Descripción del Dataset.

1.1.1 Descripción tipos de Atributos.

1.2 Lectura y visualización de los datos.

1.2.1 Limpieza de los datos.

1.3 Análisis de los datos.

1.4 Análisis descriptivo de los datos.

1.4.1 Análisis Descriptivo de Datos Categóricos.

1.4.2 Análisis Descriptivo de Datos Categóricos en función del target.

1.4.3 Análisis Descriptivo de Datos discretos y continuos.

2. Machine Learning Regresión.

2.1 Matrix de correlación.

2.2 Análisis de los Componentes Principales (PCA).

2.3 Preprocesamiento de los datos.

2.4 Construimos nuestro modelo

2.5 Comparamos los modelos.

2.6 Creamos el modelo.

2.7 Ajustar los Hiperparámetros.

2.8 Visualiza la gráfica de residuos.

2.9 Obtiene las características más importantes.

2.10 Gráfico de error de predicción.

2.11 Curva de aprendizaje.

2.12 Predicción.

2.13 Finalizar el modelo para su despliegue.

2.14 Predicción datos no vistos.

3. Análisis de Sentimiento.

3.1 Reducción del Tamaño del Conjunto de Datos.

3.2 Preprocesamiento de texto.

3.3 Tokenize.

3.4 StopWords.

3.5 Stemming y Lematización.

3.6 Análisis automatizado de los comentarios utilizando VADER para la detección de sentimientos.

Conclusiones.

Bibliografía.

**Introducción.**[**¶**](http://localhost:8888/notebooks/Proyecto%20UPC%20Business%20Analytics/Proyecto%20Business%20Analytics_EDA.ipynb#Introducci%C3%B3n.)

En este trabajo fin de curso se pretende poner en práctica los conocimientos y técnicas aprendidas en el transcurso del curso Posgrado Business Analytics mediante su aplicación práctica en un conjunto de datos públicos.

Me planteo dos objetivos principales: por una parte, consolidar el uso de las técnicas y la interpretación de los resultados y por otro desarrollar las habilidades necesarias que permiten la aplicación de estas herramientas en el mundo real y en casos reales de negocio.

Entiendo este segundo objetivo como la constatación efectiva del aprovechamiento de los conocimientos adquiridos. Más allá la amplitud de conocimientos y destreza en el uso de técnicas y métodos en el ámbito del Business Analytics, éstos no son más que el medio para permitirnos conocer mejor el escenario en el que desarrollamos nuestra actividad y ofrecer una toma de decisiones adecuada a los objetivos marcados en el negocio.

Dividiremos este documento en varias partes. En el apartado de metodología comentaremos cuál ha sido el origen de datos, una breve descripción sobre el conjunto de datos escogido y las preparaciones básicas realizadas para favorecer su posterior tratamiento y análisis que haremos.

La parte de preparación de datos realizará un recorrido exhaustivo por los diferentes atributos analizando su contenido y distribución y haciendo, en su caso, adecuaciones para evitar influencias no deseadas en los análisis posteriores.

**Objetivo.**

Los principales objetivos de este proyecto son los siguientes:

* **Desarrollar un modelo de Machine Learning** con la técnica de aprendizaje supervisado, que pueda predecir a partir de una serie de datos de entrada el precio del alojamiento de Airbnb.
* **Estudiar y analizar** los datos de nuestro Dataset. La creación de nuestro modelo de ML, así como las herramientas necesarias para su elaborar y optimizar el mismo.
* **Definir** una serie de pruebas con diferentes modelos de aprendizaje supervisado, para encontrar el más idóneo.
* **Estudio de Análisis de Sentimiento**.

**Metodología.**

Este capítulo incluye el estudio de los atributos de nuestro Dataset, lectura, visualización y descripción.

**1. Dataset Airbnb.**

**1.1 Descripción del Dataset.**

Airbnb es una plataforma en línea que permite a las personas alquilar alojamiento a corto plazo. Esto abarca desde personas comunes con una habitación disponible hasta empresas de gestión de propiedades que alquilan múltiples inmuebles. Por un lado, Airbnb permite a los propietarios listar su espacio y ganar dinero por el alquiler. Por otro lado, proporciona a los viajeros un acceso fácil al alquiler de hogares privados.

Airbnb recibe comisiones de dos fuentes por cada reserva, a saber, de los anfitriones y los huéspedes. Por cada reserva, Airbnb cobra al huésped un 6-12% del costo de la reserva. Además, Airbnb cobra al anfitrión un 3% por cada transacción exitosa.

El ejercicio que acometemos es desarrollar un modelo de precios que pueda predecir de manera efectiva el alquiler de un alojamiento y que pueda ayudar a los anfitriones, viajeros y también a la empresa a diseñar estrategias rentables.

**Fichero Airbnb\_Data.csv:**

**id:** Identidad del alojamiento.

**log\_price:** Precio del alojamiento, el precio viene transformado a logaritmo. Partimos de la idea que el precio es por día.

**property\_type:** Tipo de propiedad seleccionada, es un array

['Apartment', 'House', 'Condominium', 'Loft', 'Townhouse', 'Hostel', 'Guest suite' ,'Bed & Breakfast', 'Bungalow', 'Guesthouse', 'Dorm', 'Other', 'Camper/RV', 'Villa', 'Boutique hotel', 'Timeshare', 'In-law', 'Boat', 'Serviced apartment', 'Castle', 'Cabin', 'Treehouse', 'Tipi', 'Vacation home', 'Tent', 'Hut', 'Casa particular', 'Chalet', 'Yurt', 'Earth House', 'Parking Space', 'Train', 'Cave', 'Lighthouse','Island']

**room\_type:** Tipo de Habitación.

['Entire home/apt', 'Private room', 'Shared room']

**amenities:** Servicios.

Json []

**accommodates:** Capacidad máxima del alojamiento.

**bathrooms:** El número de baños en el alojamiento.

**bed\_type:** Tipo de cama(s).

['Real Bed', 'Futon', 'Pull-out Sofa', 'Couch', 'Airbed']

**cancellacion\_policy:** Política de cancelación.

['strict', 'moderate', 'flexible', 'super\_strict\_30', 'super\_strict\_60']

**cleaning\_fee:** Tarifa de limpieza.

**city:** Ciudad.

['NYC', 'SF', 'DC', 'LA', 'Chicago', 'Boston']

**descripcion:** Descripción detallada del alojamiento.

**first\_review:** Fecha de la primera opinión.

**host\_has\_profile\_pic:** Foto.

**host\_identity\_verified:** Identidad Verificada.

**host\_response\_rate:** Tasa de respuesta del anfitrión.

**host\_since:** Fecha en que se creó el anfitrión/usuario.

**instant\_bookable:** Si el huésped puede reservar automáticamente el alojamiento sin que el anfitrión necesite aceptar su solicitud de reserva. Un indicador de un alojamiento comercial.

**last\_review:** Fecha de la última opinión.

**latitude:** Utiliza el sistema geodésico mundial (WGS84) para la proyección de latitud y longitud.

**longitude:** Utiliza el sistema geodésico mundial (WGS84) para la proyección de latitud y longitud.

**name:** Nombre del anuncio.

**neighbourhood:** Barrio.

**number\_of\_reviews:** El número de reseñas que tiene el anuncio.

**review\_scores:** Valoración.

**thumbnail\_url:** URL de la miniatura de la imagen alojada en Airbnb para el anuncio.

**zipcode:** Código postal.

**bedrooms:** Número de habitaciones.

**beds:** Número de camas.

* + 1. **Descripción tipos de Atributos.**

Para abordar el análisis y la limpieza de datos de manera técnica y sistemática, primero debemos inspeccionar las características del dataset, incluyendo los tipos de atributos y el número de valores nulos presentes en cada uno de ellos. Luego, procederemos con la limpieza de los diferentes atributos para preparar los datos adecuadamente antes de aplicar cualquier modelo de aprendizaje automático.

Tabla

Descripción generada automáticamente

**1.2 Lectura y visualización de los datos.**

Para realizar está investigación, analizamos un dataset. Este dataset contiene la información de un total de 74111 registros de alojamientos de diferentes ciudades de EEUU.

Tabla

Descripción generada automáticamente

Configuramos los atributos:

Realizamos una serie de transformaciones en el conjunto de datos para convertir diferentes tipos de variables a formatos específicos. Aquí hay un resumen de las transformaciones que se están aplicando:

Enteros: convertimos el atributo 'accommodates' a tipo entero mediante el método astype('int').

Float: Eliminamos el símbolo '%' del atributo 'host\_response\_rate' y convertimos los valores a tipo float mediante astype('float').

Booleanos: Modificamos varios atributos booleanos ('cleaning\_fee', 'host\_has\_profile\_pic', 'host\_identity\_verified', 'instant\_bookable') utilizando un diccionario de mapeo para asignar los valores 't' y 'f' a 1 y 0, respectivamente.

Fechas: Modificamos los atributos 'host\_since', 'first\_review' y 'last\_review' a tipo datetime utilizando pd.to\_datetime() con un formato específico '%Y-%m-%d'.

Por último, renombramos los atributos del conjunto de datos utilizando un diccionario de mapeo.

**1.2.1 Limpieza de los datos.**

La limpieza y manipulación de datos suelen ser pasos críticos y consumen una gran parte del tiempo en proyectos de este tipo. Es fundamental para garantizar la calidad y la coherencia de los datos antes de pasar a etapas más avanzadas como el desarrollo de modelos, la evaluación y la optimización.

* **Eliminar duplicados del conjunto de datos.**

No se encontraron filas duplicadas en dataset.

* **Vamos a verificar la cantidad de valores NaN (Not a Number) en nuestro conjunto de datos.**

Analizaremos cada atributo distinto de cero en nuestro dataset para determinar cómo abordar esos valores. Evaluaremos diferentes estrategias, como la eliminación, la sustitución por la media, la mediana u otras técnicas apropiadas.

* Texto

  Descripción generada automáticamente con confianza media **Estrategia:**

Para la imputación de valores faltantes en el atributo 'valoración' los pondremos a cero y 'tasarespInquilino' se utilizará la mediana en lugar de optar por la eliminación de registros, con el fin de preservar la integridad del conjunto de datos.

Para abordar los valores faltantes en la columna 'numLavabos', se aplicará un método de imputación mediante el reemplazo de los valores NaN con ceros.

Se normalizan los datos en el atributo 'barrio' mediante la eliminación de espacios en blanco al final de las cadenas y la supresión de ciertas palabras para mejorar el agrupamiento del atributo.

Una limpieza precisa de los códigos postales (CP), eliminando caracteres no deseados y normalizando el formato a los primeros 5 dígitos. Posteriormente, identifica registros donde el valor del barrio es 'NaN' y busca otros registros con el mismo CP. Cuando se encuentra un CP correspondiente con un barrio válido, este se asocia al registro sin barrio. Esta estrategia proporciona una estructura robusta para manejar casos especiales y errores, garantizando la integridad de los datos y la precisión en la asignación de barrios en los registros sin esta información.

Para fortalecer la robustez de los datos, se procede de la siguiente manera: a las entradas con valor NaN en el atributo 'fechaAlta', se les asigna la fecha de 'primeraOpinion' o 'ultimaOpinion', según esté disponible. Además, a aquellos NaN en 'primeraOpinion' o 'ultimaOpinion', se les asigna una marca temporal ficticia del 31 de diciembre de 2500, indicando que no han recibido opiniones de clientes en la plataforma.

En el atributo 'url', se agrega el texto 'No hay URL' para asegurar que, al eliminar los registros restantes con valores NaN, la muestra resultante sea lo suficientemente representativa.

Para interpretar o usar los valores en su escala original del atributo 'logPrice', necesitamos revertir la transformación logarítmica. La función inversa del logaritmo natural es la exponenciación.

El objetivo es modificar el atributo descripcion de un DataFrame para eliminar cualquier carácter que no esté en el rango ASCII estándar (0-127). Esto es necesario para asegurar que los datos estén limpios y preparados para un análisis de sentimiento posterior.

**1.3 Análisis de los datos.**

Los 29 atributos están compuestos tanto por datos categórico, discretos y continuos. En estos apartados vamos a responder a una serie de preguntas:  
  
        Análisis descriptivo:

* ¿Cuál es el precio promedio de los anuncios de Airbnb en el conjunto de datos?
* ¿Cuál es el tipo de habitación más común entre los anuncios de Airbnb?
* ¿Etc..?  
    
  Análisis Geográfico:
* ¿Cuáles son las principales ciudades?
* ¿Cuáles son los principales barrios por número de listados?
* ¿Etc..?

Estos son sólo algunos ejemplos del análisis que puede realizar en su conjunto de datos de Airbnb. Dependiendo de sus objetivos de investigación específicos, puede explorar preguntas y análisis más avanzados, como modelos de aprendizaje automático para predicción de precios, análisis de sentimiento de reseñas o análisis de agrupación para agrupar listados similares.

**1.4 Análisis descriptivo de los datos.**

El análisis descriptivo de datos es una etapa fundamental en la estadística que tiene como objetivo resumir y describir las características principales de un conjunto de datos. Esto se realiza mediante diversas técnicas y herramientas que permiten entender mejor la distribución y las tendencias de los datos. En este apartado, vamos a estudiar las características de tipo categórico, utilizando la representación gráfica de histogramas, para obtener la distribución de frecuencia de los datos.

**1.4.1 Análisis Descriptivo de Datos Categóricos.**

Los datos categóricos son aquellos que se pueden clasificar en diferentes categorías o grupos.

En la siguiente gráfica se muestra la distribución de frecuencias para el atributo 'tipoPropiedad'. Adjuntamos también una tabla que complementa la gráfica, proporcionando los valores de frecuencia y porcentaje. Esta tabla se incluye debido a que la representación visual en la gráfica no resultó clara.

Los alojamientos más demandados son los apartamentos y las casas.

Gráfico, Histograma

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

En la siguiente gráfica se muestra la distribución de frecuencias para el atributo 'tipoHabitacion'.

Gráfico, Gráfico de barras

Descripción generada automáticamente

La siguiente gráfica presenta la distribución de frecuencias para el atributo 'servicios'. Este atributo estaba en formato JSON, y se ha decodificado para contabilizar la frecuencia de cada valor asociado a cada clave. De esta manera, obtenemos todos los servicios ofrecidos por los alojamientos. Asimismo, se presenta una tabla con la frecuencia y el porcentaje correspondiente.

Gráfico

Descripción generada automáticamente

Porcentaje de Servicios por alojamientos:Tabla

Descripción generada automáticamente

La siguiente gráfica presenta la distribución de frecuencias para el atributo 'tipoHabitacion'.Gráfico, Gráfico de barras

Descripción generada automáticamente

La siguiente gráfica presenta la distribución por ciudad:

Gráfico, Gráfico de barras

Descripción generada automáticamente

La siguiente gráfica muestra la distribución de frecuencias para el atributo 'neighbourhood\_city'. En este caso, hemos realizado una concatenación de los atributos ciudad y barrio para identificar a qué ciudad pertenecen los barrios con mayor ranking de anuncios en Airbnb. Además, se incluye una tabla con los valores de frecuencia y porcentaje correspondientes.

Gráfico

Descripción generada automáticamente

**1.4.2 Análisis Descriptivo de Datos Categóricos en función del target.**

Los datos categóricos son aquellos que se pueden clasificar en diferentes categorías o grupos.

En la siguiente gráfica se presenta la distribución del log\_Precio promedio por tipo de propiedad. Se destaca que, dentro de cada tipo de alojamiento, los Timesshare, Lighthouse y Vacation home tienden a tener una media de precios más alta con relación al log\_Precio.

Gráfico, Gráfico de barras

Descripción generada automáticamente

La misma que la anterior, pero por precio real.

Gráfico, Gráfico de barras

Descripción generada automáticamente

Podríamos mirar cada atributo categórico respecto al target, lo presentaremos en el PBI.

**1.4.3 Análisis Descriptivo de Datos discretos y continuos.**

Tabla

Descripción generada automáticamente

**‘precioOriginal’**

La variable precioOriginal representa los precios originales de los alojamientos en Airbnb. Al observar su distribución, es común encontrar que los datos de precios suelen estar sesgados a la derecha. Esto significa que hay una gran cantidad de precios bajos y medianos, pero también hay algunos precios muy altos que estiran la cola derecha de la distribución.

**‘logPrecio’**

Para manejar la asimetría en los datos de precios y aproximar la distribución a una forma más normal, se aplica una transformación logarítmica a la variable precioOriginal, creando así la variable logPrecio. La transformación logarítmica tiene el efecto de comprimir los valores altos y expandir los valores bajos, resultando en una distribución que se asemeja más a una curva normal.

Gráfico

Descripción generada automáticamente

Es fundamental tener en cuenta que los outliers, o valores atípicos, pueden proporcionar información valiosa sobre las características únicas de ciertos datos. En el contexto de Airbnb, por ejemplo, un alojamiento con un número de habitaciones o camas significativamente mayor que el promedio podría ser una villa de lujo o una casa de vacaciones para grupos grandes. Por tanto, es crucial evaluar estos valores en el contexto adecuado antes de decidir si deben ser tratados o eliminados.

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

**2. Machine Learning Regresión.**

**Preparación del DataSet.**

Vamos a analizar los tres enfoques mencionados a continuación para preparar los datos para un modelo de machine Learning:

**Primer Dataset: Eliminación de atributos basados en correlación**

En este enfoque, el proceso implica eliminar atributos que no contribuyen significativamente a la predicción del modelo. Esto se determina generalmente mediante el análisis de la matriz de correlación entre los atributos y la variable objetivo (o entre los atributos entre sí en ausencia de una variable objetivo-específica). Aquí están los pasos detallados:

**1. Análisis de la matriz de correlación:**

Calcula la correlación entre todos los pares de atributos en el conjunto de datos. La correlación puede ser tanto lineal (para variables numéricas) como de rango (para variables categóricas ordinales).

**2. Eliminación de atributos redundantes o irrelevantes:**

Identifica aquellos atributos que tienen una alta correlación entre sí. La alta correlación indica que ambos atributos pueden estar capturando información similar, por lo tanto, uno de ellos podría eliminarse para reducir la complejidad del modelo y mejorar la interpretabilidad.

**3. Simplificación del modelo:**

Al eliminar atributos redundantes o irrelevantes, el modelo resultante puede volverse más simple y menos propenso a sobre ajustarse a los datos de entrenamiento. Esto también puede mejorar el rendimiento del modelo en datos de prueba no vistos.

**Segundo Dataset: Conversión de atributos categóricos en variables dummy**

En este segundo enfoque, se manejan los atributos categóricos convirtiéndolos en variables dummy (también conocidas como variables indicadoras o variables binarias). Este método es fundamental para que los modelos de machine learning puedan interpretar y utilizar adecuadamente la información categórica. Aquí están los pasos relevantes:

**1. Identificación de atributos categóricos:**

Encuentra todos los atributos en el conjunto de datos que son de naturaleza categórica.

**2. Creación de variables dummy:**

Para cada atributo categórico, se crea un conjunto de variables binarias (dummy) que representan cada una de las categorías posibles del atributo original. Por ejemplo, si un atributo categórico tiene tres categorías (A, B, C), se crearán tres variables binarias.

**3. Exclusión de atributos de alta cardinalidad:**

En algunos casos, como mencionas con el atributo "barrio" que tiene más de 600 categorías, la conversión a variables dummy podría resultar en una expansión masiva del conjunto de datos. Esto puede sobrecargar el modelo y dificultar su entrenamiento. En tales situaciones, se puede optar por estrategias alternativas como la codificación de frecuencia o la agrupación de categorías menos frecuentes en una categoría común.

**Tercer enfoque: Utilización de PCA en el primer dataset**

Utiliza PCA en el primer dataset después de haber eliminado los atributos redundantes o irrelevantes según el análisis de correlación.

**1. Aplicación de PCA:**

Encuentra todos los atributos en el conjunto de datos que son de naturaleza categórica.

**2. Identificación de componentes principales:**

PCA transforma los atributos originales en un nuevo conjunto de atributos (llamados componentes principales) que son combinaciones lineales de los atributos originales. Estos componentes están ordenados por la cantidad de variabilidad que explican en los datos.

**3. Selección de atributos:**

Basado en la varianza explicada por cada componente principal, puedes decidir cuántos componentes retener para el modelo final. Esto puede ayudar a eliminar atributos que contribuyen poco a la variabilidad total de los datos, simplificando así el modelo.

**2.1 Matrix de correlación**

La **matriz de correlación** muestra el valor de correlación de Pearson, que mide el grado de relación lineal entre cada par de atributos. Esta matriz nos ayuda a realizar un análisis adicional de los datos ya que proporciona información relevante sobre las dependencias entre los diferentes atributos. El valor de correlación puede estar entre -1 y +1. Si estos dos elementos tienden a aumentar o disminuir al mismo tiempo, el valor de correlación es positivo porque indica una relación directa. Si el valor es negativo, esto indica una relación inversa. Un valor nulo o cercano a cero indica que no existe una tendencia entre ambos atributos.

Gráfico

Descripción generada automáticamente con confianza media

Las correlaciones mayores de 0.84 que se muestran son las correlaciones perfectas que se producen entre los atributos consigo mismos. Las demás correlaciones no son superiores o inferiores a +/- 0.84 por lo que ninguna correlación es lo suficientemente relevante.

Tabla

Descripción generada automáticamente con confianza media

**2.2 Análisis de los Componentes Principales (PCA)**

El algoritmo de Análisis de Componente Principal (PCA), es una técnica de aprendizaje no supervisado que transforma un gran conjunto de atributos en uno más pequeño que aun contiene la mayor parte de la información del conjunto original. Al ser un conjunto de datos más pequeño, facilita el análisis de estos y es más rápido para los algoritmos de Machine learning. En resumen, la idea de PCA es reducir la cantidad de atributos del conjunto de datos, mientras conserva la mayor cantidad de información.

**Estandarizamos los atributos.**

Nos aseguramos de que los atributos no se califiquen con más importancia debido a su diferencia de escala. Todo el conjunto de datos será estandarizado, transformando a una distribución normal con la media centrada a **0** y la desviación típica a **1**.

**Aplicamos PCA.**

Los componentes son nuevas variables que se construyen como combinaciones lineales o mezclas de las variables iniciales, evitando que estas nuevas variables no estén correlacionadas y la mayor parte de la información está dentro de las variables iniciales se comprime en los primeros componentes.

El objetivo es elegir los suficientes atributos para que expliquen una varianza superior a un **90%**.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

* Características seleccionadas para exclusión basándose en su contribución a los componentes redundantes.

features\_to\_exclude = ['numHabitaciones', 'numOpiniones', 'Essentials\_presente', 'politicaCancelacion', 'barrio', 'numCamas', 'tipoPropiedad', 'tarifaLimpieza']

**2.3 Preprocesamiento de los datos.**

Explicación de cómo vamos a preprocesar los datos:

Los datos continuos los sometemos a un proceso de estandarización o un proceso de escalado para limitarlos a un rango de valores que les permita ser comparados entre sí, independientemente de su naturaleza. Los datos discretos permanecerán igual.

**2.4 Construimos nuestro modelo**

Para comparar y seleccionar el modelo de mejor rendimiento entre varias opciones disponibles, utilizaremos la librería PyCaret. En este análisis, los predictores no siguen una distribución normal. Para abordar esto, utilizaremos el parámetro normalize=True al trabajar con el dataframe sin estandarización **(data\_codes)**. El siguiente sería **data\_scaled**, que contiene los datos estandarizados. Antes de aplicar PyCaret, hemos transformado los atributos categóricos de data\_codes a valores enteros.

Además, para una comparación más completa, también consideraremos:

**Dataset de Dummy:**

Un dataframe donde las variables categóricas se convierten en variables dummy (o one-hot encoding).

**Dataset de PCA con Atributos Sin Valor:**

Dataset de PCA con Atributos Sin Valor: Un conjunto de datos reducido mediante PCA, eliminando los atributos que no contribuyen significativamente.

Esto nos permitirá evaluar cómo afectan la estandarización, la reducción de dimensionalidad mediante PCA y la representación de variables categóricas en el rendimiento de los modelos de aprendizaje automático.

**División de los datos.**

En el contexto del aprendizaje automático (machine learning), cuando se menciona la división de los datos en conjuntos de entrenamiento (train) y prueba (test), generalmente se refiere al proceso de particionar los datos disponibles para evaluar el rendimiento de un modelo de manera adecuada. Sin embargo, además de estas dos divisiones, a veces también se mantiene un tercer conjunto llamado conjunto de validación (validation set) o conjunto de prueba final (hold-out set).

**Implementación con PyCaret**

A continuación, mostramos los resultados de utilizar la librería PyCaret para nuestros cuatro conjuntos de datos mencionados anteriormente:

* data\_codes (con normalización):

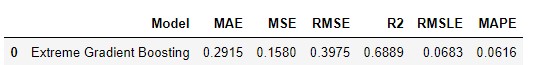
Predictores transformados a valores enteros. Normalización aplicada debido a la no normalidad de los predictores.

Imagen que contiene Texto

Descripción generada automáticamente

* data\_scaled (datos estandarizados):

Datos transformados para tener media cero y desviación estándar uno.



* Dataset de Dummy:

Variables categóricas convertidas en variables dummy utilizando one-hot encoding.

Tabla

Descripción generada automáticamente con confianza baja

* Dataset de PCA con Atributos Sin Valor:

Reducción de dimensionalidad mediante PCA, eliminando atributos sin contribución significativa.



**2.5 Comparamos los modelos.**

La función setup() en PyCaret inicializa el entorno y crea un pipeline para preparar datos para modelado y despliegue. Requiere dos parámetros obligatorios: un dataframe de pandas y el nombre de la columna objetivo. Otros parámetros son opcionales para personalizar el preprocesamiento.

Al ejecutar setup(), PyCaret infiere automáticamente los tipos de datos de las características. Luego muestra una tabla con los tipos inferidos. Verificar los tipos de datos es crucial porque PyCaret realiza tareas de preprocesamiento específicas según el tipo de dato, esenciales para el aprendizaje automático.

Tabla

Descripción generada automáticamente con confianza media

En el apartado anterior hemos guardado un dataset de datos no vistos.

Comparar todos los modelos es el punto de partida recomendado tras la configuración en PyCaret. Esta función entrena y puntúa todos los modelos usando validación cruzada k-fold. La salida es una tabla que muestra el promedio de MAE, MSE, RMSE, R2, RMSLE y MAPE a través de los pliegues (10 por defecto) para todos los modelos disponibles.

Tabla

Descripción generada automáticamente

**2.6 Creamos el modelo.**

Aunque compare\_models() es una función potente y un buen punto de partida, no devuelve ningún modelo entrenado. PyCaret recomienda usar compare\_models() después de la configuración para identificar los modelos con mejor rendimiento y seleccionar algunos candidatos. Para crear un modelo, se utiliza create\_model(), que entrena un modelo y lo puntúa con validación cruzada estratificada. Al igual que compare\_models(), la salida es una tabla de puntuación que muestra MAE, MSE, RMSE, R2, RMSLE y MAPE por pliegue.

Tabla

Descripción generada automáticamente

**2.7 Ajustar los Hiperparámetros.**

Al crear un modelo con la función create\_model(), se usan los hiperparámetros por defecto. Para ajustarlos, se utiliza la función tune\_model(), que optimiza automáticamente los hiperparámetros en un espacio de búsqueda predefinido y puntúa el modelo con validación cruzada k-fold. La salida es una tabla de puntuación que muestra MAE, MSE, RMSE, R2, RMSLE y MAPE por pliegue.

Tabla

Descripción generada automáticamente

**2.8 Visualiza la gráfica de residuos.**

Antes de finalizar el modelo, se puede usar la función plot\_model() para analizar el rendimiento en diversos aspectos, como el gráfico de residuos, el error de predicción y la importancia de las características. Esta función toma un modelo entrenado y devuelve gráficos basados en el conjunto de prueba o validación.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

**2.9 Obtiene las características más importantes.**

**Imagen que contiene Gráfico

Descripción generada automáticamente**

**2.10 Gráfico de error de predicción.**[**¶**](http://localhost:8888/notebooks/Proyecto%20UPC%20Business%20Analytics/Proyecto%20Business%20Analytics_ML.ipynb#2.10-Gr%C3%A1fico-de-error-de-predicci%C3%B3n.)

**Gráfico, Gráfico de dispersión

Descripción generada automáticamente**

**2.11 Curva de aprendizaje.**

**Gráfico, Gráfico de líneas

Descripción generada automáticamente**

Las dos líneas convergen a medida que aumenta el tamaño del conjunto de entrenamiento, es una señal positiva. Indica que el modelo no está sufriendo sobreajuste ni subajuste grave, tiene la capacidad de generalizar bien los datos no visto.Escenarios Posibles en la Curva de Aprendizaje:

1. Convergencia Rápida y Estable:

Ambas curvas (train y cross-validation) convergen a un valor cercano y estable a medida que se incrementa el número de muestras de entrenamiento. Esto indica que el modelo está aprendiendo bien de los datos y generaliza adecuadamente.

1. Sobreajuste:  
   La curva de entrenamiento muestra un rendimiento excelente (puntuación alta), pero la curva de validación cruzada se mantiene por debajo de la curva de entrenamiento y no converge. Esto indica sobreajuste, donde el modelo memoriza los datos de entrenamiento, pero no generaliza bien a nuevos datos.
2. Subajuste:  
   Ambas curvas convergen a un valor bajo y no mejoran significativamente incluso con más datos. Esto sugiere subajuste, donde el modelo es demasiado simple para capturar la complejidad de los datos.
3. Divergencia o No Convergencia:

A veces las curvas no convergen y pueden divergir o no mostrar una tendencia clara hacia un rendimiento estable. Esto puede indicar problemas en el modelo, como selección incorrecta de características, problemas en la configuración del modelo, o incluso errores en la implementación.

**2.12 Predicción.**

Antes de finalizar el modelo, es recomendable realizar una última comprobación prediciendo el conjunto de prueba/hold-out y revisando las métricas de evaluación.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Realizaremos el ejercicio con el estimador **XGboost Extreme Gradient Boosting**, es uno de los tres mejores en puntuación después de realizar la comparativa de modelos.

**2.13 Finalizar el modelo para su despliegue.**

La finalización del modelo es el último paso del experimento. Un flujo de trabajo típico en PyCaret comienza con setup(), seguido de la comparación de modelos con compare\_models() y la preselección de algunos candidatos según la métrica de interés. Luego, se aplican técnicas de modelado como ajuste de hiperparámetros, ensamblaje y apilamiento. Este flujo de trabajo lleva al mejor modelo para hacer predicciones sobre datos nuevos. La función finalize\_model() ajusta el modelo al conjunto de datos completo, incluida la muestra de prueba/retención (30% en este caso). El propósito es entrenar el modelo en todos los datos antes de desplegarlo en producción.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**2.14 Predicción datos no vistos.**

Finalmente, utilizaremos el dataframe data\_novistos, preparado al principio del ejercicio, para realizar predicciones. Este dataframe contiene el 10% de las muestras del conjunto de datos original que nunca se expusieron a PyCaret, lo que permite evaluar el rendimiento del modelo en datos completamente nuevos y no vistos anteriormente.

Imagen que contiene Tabla

Descripción generada automáticamente

**3. Análisis de Sentimiento.**

**3.1 Reducción del Tamaño del Conjunto de Datos.**

El conjunto de datos de comentarios consta de 71,898 reseñas, lo cual resulta en una cantidad excesiva de registros para el análisis deseado. Para optimizar el manejo y procesamiento de los datos, procederemos a reducir el tamaño del conjunto mediante la selección de un subconjunto representativo de comentarios.

La estrategia que implementaremos consiste en manejar un subconjunto de datos sin valoración asignada, los cuales se encuentran con valores de cero. Durante el proceso de Análisis Exploratorio de Datos (EDA), hemos sustituido los valores nulos por ceros. Esta decisión se tomó con el objetivo de mejorar la relevancia y la claridad de las visualizaciones gráficas generadas a partir de estos datos.

Antes de realizar el filtrado, es crucial analizar la distribución de los datos originales para asegurar que el subconjunto seleccionado mantenga la representatividad y características esenciales del conjunto completo.

Para determinar el idioma en el que están escritos los comentarios sobre los alojamientos, utilizamos la biblioteca langdetect en combinación con pandas para procesar y filtrar los datos.

Texto

Descripción generada automáticamente

**3.2 Preprocesamiento de texto.**

El preprocesamiento de texto es un paso crucial en la realización de análisis de sentimiento, ya que ayuda a limpiar y normalizar los datos de texto, lo que facilita el análisis. El paso de preprocesamiento implica una serie de técnicas que ayudan a transformar los datos de texto sin procesar en un formulario que puede utilizar para el análisis. Algunas técnicas comunes de preprocesamiento de texto incluyen tokenización, eliminación de palabras detenidas, derivación y lematización.

Diagrama, Texto

Descripción generada automáticamente

**NLTK**

La librería [Natural Languaje Toolkit](https://www.nltk.org/) es un *de facto* para NLP en Python para diversos idiomas. No solo contiene algoritmos para operaciones de NLP si no también corpus de textos y vocabulario organizado como [WordNet](https://wordnet.princeton.edu/" \t "_blank).

**3.3 Tokenize.**

La tokenización es un paso de preprocesamiento de texto en el análisis de sentimientos que implica descomponer el texto en palabras individuales o tokens. Este es un paso esencial en el análisis de datos de texto, ya que ayuda a separar palabras individuales del texto sin procesar, lo que facilita su análisis y comprensión. La tokenización generalmente se realiza utilizando la función "word\_tokenize" incorporada de NLTK, que puede dividir el texto en palabras individuales y signos de puntuación.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

**3.4 StopWords.**

La eliminación de palabras detenidas es un paso crucial de preprocesamiento de texto en el análisis de sentimientos que implica eliminar palabras comunes e irrelevantes que es poco probable que transmitan mucho sentimiento. Las palabras de parada son palabras que son muy comunes en un idioma y no tienen mucho significado, como "y", "el", "de" y "eso". Estas palabras pueden causar ruido y sesgar el análisis si no se eliminan.  
Al eliminar las palabras vacías, es más probable que las palabras restantes en el texto indiquen el sentimiento que se expresa. Esto puede ayudar a mejorar la precisión del análisis de sentimiento. NLTK proporciona una lista integrada de palabras de parada para varios idiomas, que se pueden utilizar para filtrar estas palabras de los datos de texto.

La presencia de signos de puntuación, números y caracteres especiales en el texto puede introducir ruido en el análisis de datos. Para mitigar este efecto, se procederá a la normalización del texto mediante la eliminación de dichos caracteres no alfabéticos. La siguiente función realiza esta tarea sustituyendo estas expresiones por espacios.

La personalización de la bolsa de stopwords es una práctica esencial para optimizar el análisis de texto en proyectos específicos. Identificar y añadir palabras que no aportan información relevante permite una mejor extracción de datos significativos y una mayor eficiencia en el procesamiento del lenguaje natural.

Texto, Gráfico de dispersión

Descripción generada automáticamente

Texto

Descripción generada automáticamente

**3.5 Stemming y Lematización.**

Stemming y lematización son técnicas utilizadas para reducir las palabras a sus formas raíz. Stemming implica eliminar los sufijos de palabras, como "ing" o "ed", para reducirlos a su forma base. Por ejemplo, la palabra "formed" se derivaría a "form".  
La lematización, sin embargo, implica reducir las palabras a su forma base en función de su parte del habla. Por ejemplo, la palabra "rising" se lematizaría a "rise".

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**3.6 Análisis automatizado de los comentarios utilizando VADER para la detección de sentimientos.**

En este análisis, utilizaremos el analizador de sentimientos VADER (Valence Aware Dictionary and sEntiment Reasoner) para evaluar los comentarios de los alojamientos. VADER es un analizador de sentimientos diseñado para trabajar específicamente con textos extraídos de redes sociales y noticias. Utiliza un enfoque basado en léxico para clasificar palabras, signos de puntuación, frases, emojis, etc., como positivos, negativos o neutrales.

Para ejecutar el análisis de sentimiento automatizado utilizando VADER y concatenar el resultado con el DataFrame original, nos enfocaremos en la puntuación compuesta ("compound"), que sintetiza las puntuaciones de sentimiento positivas, neutras y negativas. A continuación, se presenta un procedimiento detallado y técnico para realizar esta tarea.

La puntuación compuesta, una «puntuación compuesta normalizada y ponderada», se obtiene sumando las puntuaciones de valencia de cada palabra del léxico, ajustadas en función de reglas específicas y, a continuación, normalizadas para que se sitúen dentro del intervalo de -1 (negativo más extremo) a +1 (positivo más extremo). Esta métrica sirve como una valiosa medida unidimensional del sentimiento para una frase determinada. Los umbrales típicos son los siguientes

* Sentimiento positivo: Puntuación compuesta ≥ 0,05
* Sentimiento neutral: -0,05 < puntuación compuesta < 0,05
* Sentimiento negativo: Puntuación compuesta ≤ -0,05

Gráfico, Gráfico en cascada

Descripción generada automáticamente

**Conclusiones.**

* Análisis Exploratorio de Datos (EDA).

Durante el Análisis Exploratorio de Datos (EDA) se implementaron varias estrategias clave para tratar los datos faltantes y mejorar la calidad del dataset, asegurando así una base sólida para el análisis de sentimiento posterior. Las estrategias específicas incluyeron:

Imputación de Valores Faltantes:

* 'valoración': Se imputaron los valores faltantes con ceros, manteniendo la integridad de los datos y permitiendo la inclusión de todos los comentarios.
* 'tasarespInquilino': Se utilizó la mediana para reemplazar los valores faltantes, en lugar de eliminar registros, preservando así el conjunto de datos completo.
* 'numLavabos': Los valores NaN fueron reemplazados por ceros, simplificando el manejo de esta variable.

Normalización y Limpieza de Datos:

* 'barrio': Se normalizaron los datos eliminando espacios en blanco finales y suprimiendo ciertas palabras, mejorando el agrupamiento de este atributo.
* Códigos Postales (CP): Se limpiaron los códigos postales eliminando caracteres no deseados y normalizando el formato a los primeros 5 dígitos. Se imputaron los valores faltantes de 'barrio' usando el CP correspondiente.
* 'fechaAlta': Se asignaron fechas de 'primeraOpinion' o 'ultimaOpinion' a los valores NaN en 'fechaAlta'. Para valores NaN en 'primeraOpinion' o 'ultimaOpinion', se utilizó una fecha ficticia del 31 de diciembre de 2500.

Tratamiento de Valores Faltantes y Transformaciones:

* 'url': Se agregó el texto 'No hay URL' a los valores faltantes, asegurando una muestra representativa tras eliminar otros valores NaN.
* 'logPrice': Para interpretar los valores en su escala original, se creó un nuevo campo 'precioOriginal', aplicando la exponenciación a 'logPrecio'.

Limpieza de Textos para Análisis de Sentimiento:

* 'descripcion': Se eliminaron caracteres fuera del rango ASCII estándar (0-127), garantizando que los datos estén limpios y preparados para el análisis de sentimiento.

En resumen, las estrategias implementadas durante el EDA mejoraron la integridad y claridad del dataset. La imputación de valores faltantes, la normalización de datos, la creación del campo 'precioOriginal' y la limpieza precisa aseguraron una base robusta para el análisis posterior, permitiendo una comprensión más profunda y precisa de los sentimientos expresados en los comentarios.

* Machine Learning.

Después de trabajar con 4 dataframes diferentes, hemos seleccionado data\_scaled, que contiene los datos estandarizados. Además, hemos elegido el modelo 'XGBoost' con un coeficiente R^2 de 0.67. Luego, ajustamos los hiperparámetros del modelo y obtuvimos un resultado final en el dataframe no visto con un R^2 de 0.72.

Estas conclusiones sugieren que el modelo 'XGBoost', después de la optimización de hiperparámetros, ha mejorado su capacidad predictiva cuando se evalúa en datos completamente nuevos y no vistos anteriormente. El aumento en el R^2 del conjunto de datos no visto indica que el modelo generaliza bien y puede ser una elección sólida para hacer predicciones en nuevos casos.

* Análisis de Sentimientos.

En este estudio, implementamos una estrategia para manejar un subconjunto de datos que inicialmente no tenía una valoración asignada. Estos datos, con valores originales nulos, fueron sustituidos por ceros durante el proceso de Análisis Exploratorio de Datos (EDA). Esta decisión tuvo varios propósitos y resultados clave:

**Claridad en las Visualizaciones:**

La sustitución de valores nulos por ceros permitió generar visualizaciones gráficas más claras y relevantes. Los gráficos obtenidos reflejaron de manera más precisa la distribución y tendencia de los datos, facilitando así la interpretación y el análisis.

**Evaluación Completa de Comentarios:**

Gracias a esta estrategia, pudimos incluir y evaluar todos los comentarios que inicialmente no tenían una valoración asignada. Esto amplió el alcance del análisis, asegurando que no se excluyera ninguna opinión o feedback relevante por falta de valoración.

**Consistencia de Datos:**

La uniformidad en el tratamiento de datos faltantes mejoró la consistencia de nuestro dataset, permitiendo aplicar técnicas de análisis de manera más efectiva.

**Aplicación de VADER:**

Utilizamos el modelo VADER (Valence Aware Dictionary and sEntiment Reasoner) para realizar el análisis de sentimiento de los comentarios. VADER es una herramienta diseñada específicamente para el análisis de sentimientos en texto, y su aplicación nos permitió obtener una visión clara y precisa de las opiniones expresadas.

**Resultados del Análisis de Sentimiento:**

Al incluir los comentarios con valor cero, se obtuvo una visión más completa del espectro de opiniones. VADER proporcionó puntuaciones de sentimiento que reflejan tanto la valencia (positividad o negatividad) como la intensidad del sentimiento, permitiéndonos comprender mejor las emociones y opiniones de los usuarios.

En resumen, la decisión de reemplazar valores nulos por ceros durante el EDA, combinada con la utilización de VADER, no solo mejoró la claridad de las visualizaciones, sino que también permitió una evaluación más exhaustiva de los datos. Esto aseguró un análisis robusto y comprensivo del sentimiento expresado en los comentarios, ofreciendo una valiosa comprensión de las opiniones y emociones de los usuarios.

**Bibliografía.**

* [Ejemplos kaggle](https://www.kaggle.com/)
* [Data Dictionary (Airbnb)](https://docs.google.com/spreadsheets/d/1iWCNJcSutYqpULSQHlNyGInUvHg2BoUGoNRIGa6Szc4/edit?usp=sharing)
* [Logarithmic Price Scale vs. Linear Price Scale: What's the Difference?](https://www.investopedia.com/ask/answers/05/logvslinear.asp)
* [Librería Python Pycaret](https://www.pycaret.org/tutorials/html/REG101.html)
* [Learning curve](https://en.wikipedia.org/wiki/Learning_curve_(machine_learning))
* [Learning curve II](https://www.dataquest.io/blog/learning-curves-machine-learning/)
* [Análisis de Sentimiento](https://medium.com/@yogi.sarumaha/airbnb-sentiment-analysis-with-python-e81e66fee6a6)
* [Repositorio Github](https://github.com/Juanja67/PostGrado-Business-Analytics-)