Maestría en Ciencia de Datos
Departamento de Matemática y Ciencias

Predicción de la tarifa de pasajes aéreos

Federico Libertun **2024**

Director: Daniel Fraiman

Universidad de
SanAndrés

Resumen

Easemytrip es un sitio web para reservar pasajes aéreos y por lo tanto, una plataforma que clientes potenciales utilizan para comprar pasajes. A partir de datos descargados desde ahí, tenemos como objetivo analizar cómo se relaciona el precio de las tarifas aéreas con otras variables, predecir cuál será el precio del pasaje y responder a las siguientes consultas:

¿Cómo varía el precio según la clase Economy y Business? ¿Cómo varía el pasaje según cada aerolínea? ¿En qué momento del día es más económico viajar? ¿Qué relación hay entre la cantidad de escalas realizadas o la duración del vuelo y el precio del ticket? ¿Cuándo es más conveniente sacar el pasaje? Estas y varias preguntas más serán respondidas en el presente caso de estudio.

Agradecimientos

Agradezco a mis compañeros de estudio Sol Guerreiro y Adaia Montaño por acompañarme en el trayecto de la maestría y a mis profesores que impartieron el conocimiento necesario para hacer esto posible. También agradezco a Matias Wolinsky por su buen sentido del humor durante la cursada.

Índice general

CAPÍTULO 1

Introducción

Easemytrip es un sitio web para reservar y comprar pasajes aéreos. El objetivo de este proyecto fue analizar cómo se relaciona el precio de las tarifas aéreas con otras variables, predecir cuál será el precio del pasaje y responder a las siguientes consultas:

- ¿Cómo varía el precio según la clase Economy y Business?
- ¿Cómo varía el pasaje según cada aerolínea?
- ¿En qué momento del día es más económico viajar?
- ¿Qué relación hay entre la cantidad de escalas realizadas o la duración del vuelo y el precio del ticket?
- ¿Cuándo es más conveniente sacar el pasaje?

Para responder estas y varias preguntas más, se descargaron más de 300.000 datos depurados sobre reservas de vuelo y utilizamos varios algoritmos de regresión para comparar los resultados. Los más importantes fueron:

- Random Forest
- Bagging
- Extra Trees Regressor

CAPÍTULO 2

Marco teórico

2.1. Acerca del dataset elegido

A partir de los datos descargados y depurados de la página web Easemytrip desde el 11/02/2022 hasta el 31/03/2022, obtuvimos 300.153 opciones diferentes de reserva de vuelos y contamos con la siguiente información:

- **Airline:** El nombre de la empresa áerea. Existen 6 compañías: SpiceJet, AirAsia, Vistara, GO-FIRST, Indigo y Air-India.
- **Flight:** El código de vuelo del avión.
- **Source city:** Ciudad de origen. Se analizaron 6 ciudades principales de la India: Delhi, Mumbai, Bangalore, Kolkata, Hyderabad y Chennai.
- **Departure time:** A partir del horario de partida, se categorizaron 6 momentos diferentes del día.
- **Stops:** Escalas que realiza el avión entre la ciudad de origen y destino: ninguna, 1 y 2 o más.
- **Arrival time:** A partir del horario de llegada, se clasificaron 6 momentos distintos del día.
- **Destination city**: La ciudad de destino: Delhi, Mumbai, Bangalore, Kolkata, Hyderabad y Chennai.
- **Class:** La clase de asiento del pasaje, puede ser ”Business” o ”Economy”.
- **Duration:** Es una variable continua que muestra la cantidad total de horas para viajar entre las ciudades de origen y destino, es decir, la duración del vuelo.
- **Days left:** Es la diferencia entre la fecha de salida y la fecha de reserva del pasaje.
- **Price:** El precio, es nuestra variable objetivo.

2.2. Planificación e Implementación técnica

En este proyecto se utilizaron las siguientes librerías de Python:

- pandas
- numpy
- matplotlib
- seaborn
- sklearn

Comenzamos realizando un análisis exploratorio de los datos en donde empleamos herramientas de visualización como gráficos de barras y boxplots para analizar características importantes de diferentes features.

Luego hicimos un preprocesamiento de los datos en donde transformamos variables categóricas y numéricas. Elegimos la técnica de Min-Max Scaler para escalar los datos. Realizamos un mapa de calor y el método de KBest Feature Selection para visualizar qué variables influyen más en el precio.

A continuación necesitamos separar los datos en train y test para entrenar varios modelos de machine learning. Los modelos que utilizamos fueron BaggingRegressor, RandomForestRegressor, XGBRegressor y ExtraTreesRegressor, entre otros.

Generamos una base de datos que resume las métricas para cada modelo. Interpretamos y comparamos los resultados entre un modelo y otro, y elejimos el que mejor métrica tuvo.

Por último, mencionamos las conclusiones que obtuvimos de este proyecto.

CAPÍTULO 3

Análisis Exploratorio de Datos

3.1. Pre-requisitos

Para comenzar lo primero que necesitamos es importar las librerías que utilizaremos en el proyecto:

Algoritmo 3.1: Librerías importadas

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
pd.set_option('display.max_columns',None)

from sklearn.preprocessing import MinMaxScaler  # Para escalar las variables num ricas
from sklearn.feature_selection import SelectKBest, f_classif

from sklearn.model_selection import train_test_split

# Importamos los modelos que vamos a usar:

from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn import linear_model
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
import xgboost as xgb
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.ensemble import BaggingRegressor

from sklearn import metrics
```

Luego importamos y guardamos el dataset con el nombre 'df'.

3.2. Analizando los datos

Luego de realizar los pre-requisitos, visualizamos las primeras 5 columnas del mismo. Comprobamos que consta de 300.153 observaciones y 11 features, mencionadas en el marco teórico. Utilizamos el código "describe(include='all')" para obtener características generales de cada feature, como por ejemplo la media en variables numéricas, o la cantidad de variables únicas por feature.

Utilizamos el código ”df.info()” y ”df.duplicated().sum()” y descubrimos que contamos con 8 variables categóricas que deberán ser convertidas a numéricas para poder utilizar modelos de machine learning. El dataset no contiene valores faltantes ni filas duplicadas.

3.3. Características del mercado

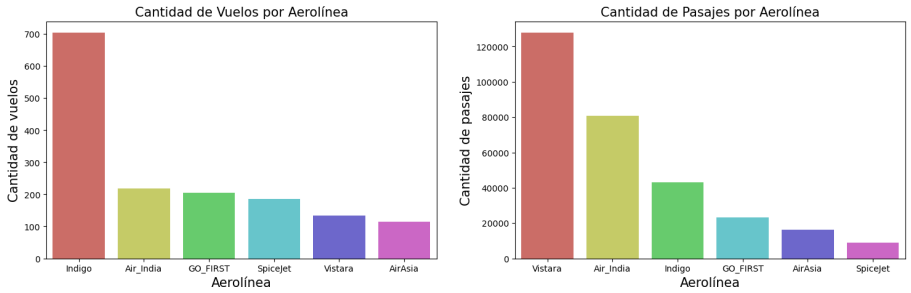


Figura 3.1: Cantidad de Vuelos y Pasajes por Aerolínea

Del gráfico anterior podemos ver que Indigo realizó más vuelos que el resto de las aerolineas. Vistara y Air India fueron las empresas que más pasajes vendieron. Sin embargo, Vistara fue la segunda empresa que menos viajes realizó en el período analizado. Esto nos puede dar un indicio de que la capacidad de pasajeros en cada vuelo de Vistara es mayor que para el resto de las compañías.

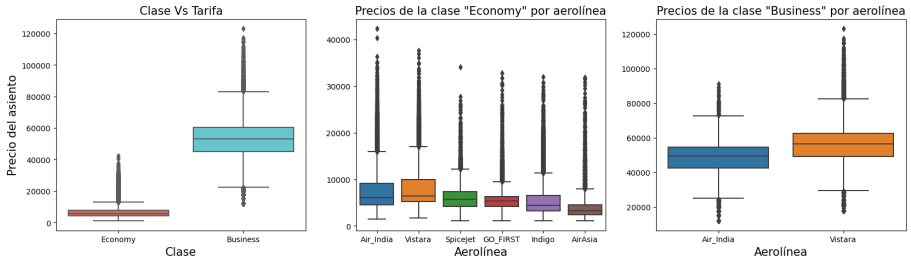


Figura 3.2: Precios de los pasajes por clases

La brecha entre el precio de asientos ”Business” y ”Economy” es grande. Si bien existen tarifas Economy atípicas iguales o más caras que la clase Business, la mediana de la clase Business es aproximadamente 5 veces superior que la clase Economy. Las únicas dos empresas que ofrecen asientos Business son Vistara y Air India. Vistara tiene precios más caros que Air India para la clase Business y similares para la clase Economy. Air India y Vistara brindan servicios Economy a un precio más elevado que el resto de las compañías. AirAsia es la aerolínea más económica.

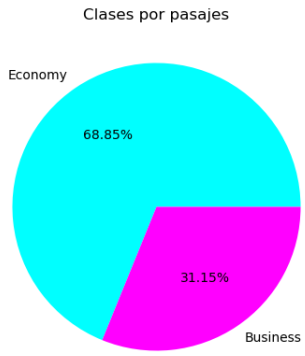


Figura 3.3: Clases por pasajes

El 69 % de los pasajeros viaja en clase Economy.

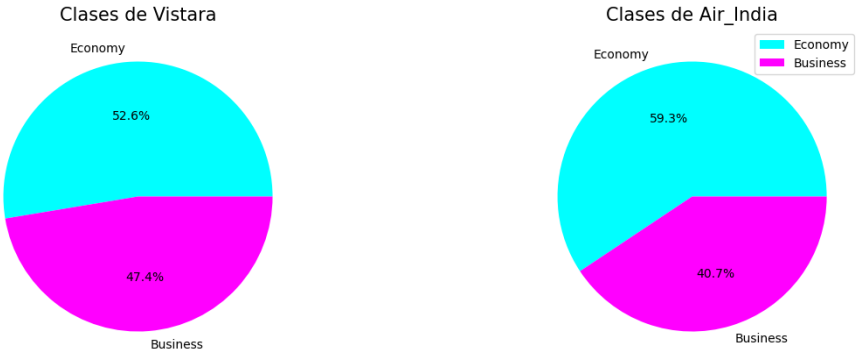


Figura 3.4: Composición de las clases de Vistara y Air India

El 47 % de los clientes de Vistara son Business y el 41 % de los pasajeros de Air India son Business.

Veamos el precio y la duración promedio para la clase Economy de un destino a otro:

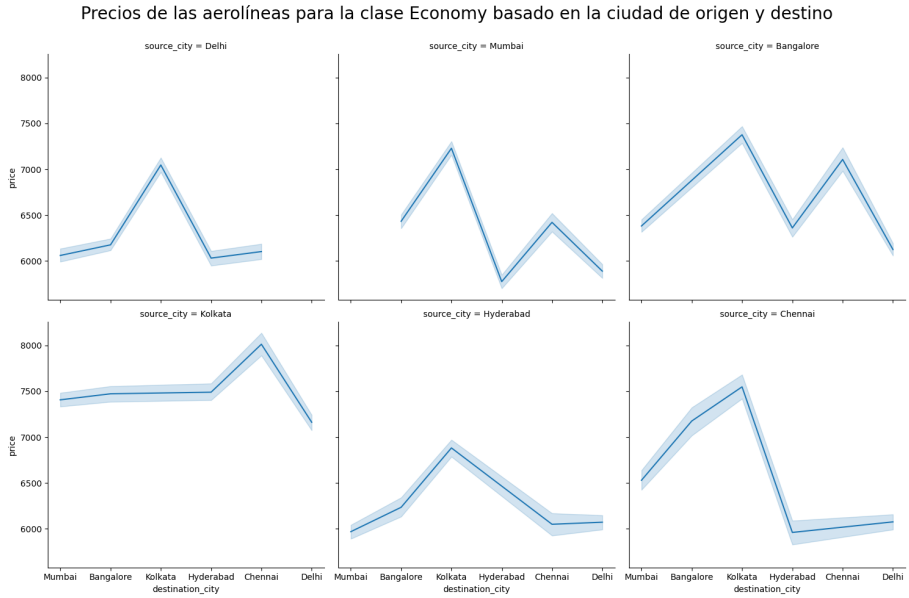


Figura 3.5: Precios de las aerolíneas para la clase Economy basado en la ciudad de origen y destino.

destination_city source_city	Bangalore	Chennai	Delhi	Hyderabad	Kolkata	Mumbai
Bangalore	-	7105.95	6124.9	6360.14	7375.64	6381.09
Chennai	7175.02	-	6075.96	5960.79	7547.3	6529.12
Delhi	6175.62	6102.32	-	6031.16	7045.62	6059.83
Hyderabad	6234.88	6049.88	6072.3	-	6881.68	5969.26
Kolkata	7471.62	8011.75	7161.4	7489.14	-	7405.79
Mumbai	6432.51	6420.92	5889.28	5774.89	7227.97	-

Figura 3.6: Precio promedio para la clase Economy de un destino a otro.

destination_city source_city	Bangalore	Chennai	Delhi	Hyderabad	Kolkata	Mumbai
Bangalore	-	13.86	9.38	13.12	11.66	10.28
Chennai	13.28	-	11.23	11.75	13.36	12.07
Delhi	9.96	11.88	-	12.08	12.0	9.54
Hyderabad	10.73	11.77	10.45	-	12.37	11.93
Kolkata	12.63	13.88	10.96	12.77	-	12.93
Mumbai	11.25	12.07	8.97	13.15	12.32	-

Figura 3.7: Duracion promedio para la clase Economy de un destino a otro.

En general, cuanto más dura el vuelo, más caro es el pasaje.

Los vuelos con origen en Kolkata suelen ser más caros que el resto de las ciudades.

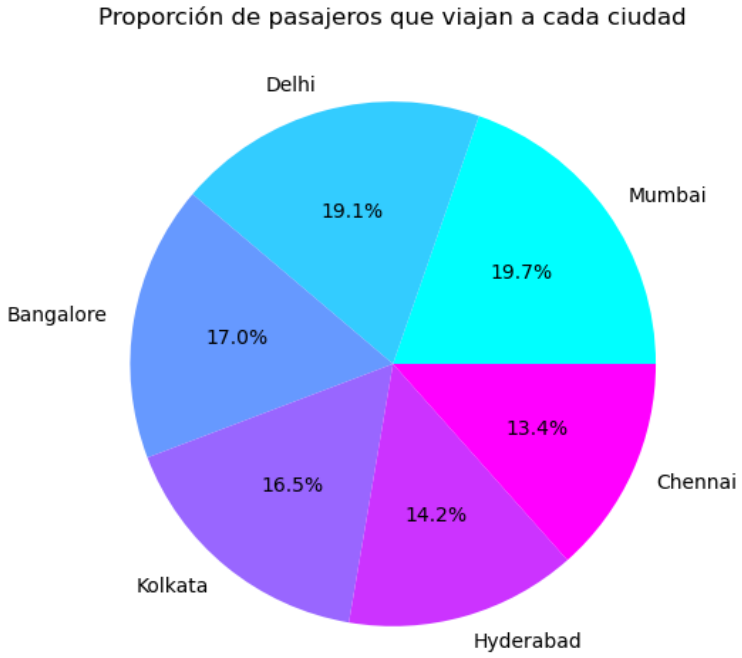


Figura 3.8: Proporción de pasajeros que viajan a cada ciudad.

Los destinos más populares son Delhi y Mumbai, y el menos concurrido por los pasajeros es Chennai.

3.4. ¿Cómo varía el precio con la duración del vuelo?

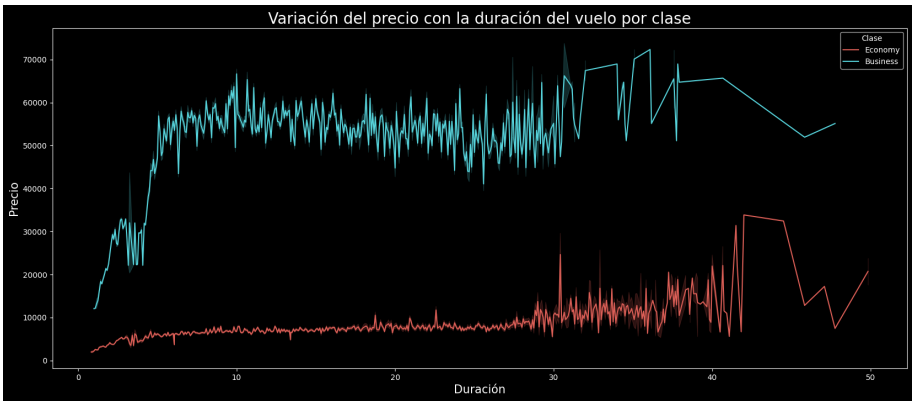


Figura 3.9: A mayor duración del vuelo, el precio se incrementa para ambas clases.

3.5. ¿Cómo varía el precio según las escalas realizadas entre la ciudad de origen y destino?

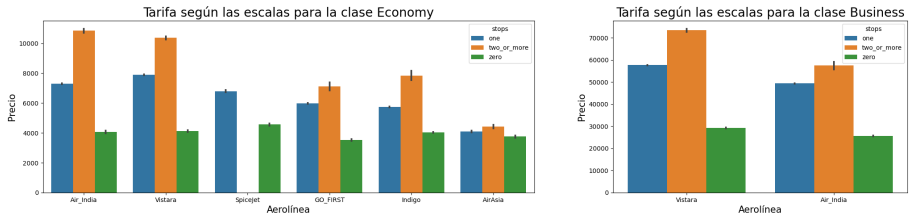


Figura 3.10: A mayor cantidad de escalas, mayor es el precio de cada pasaje y mayor es la diferencia de precios entre una compañia y otra.

Los diferentes análisis realizados tienden a mostrar que AirAsia se trata de una empresa low cost.

3.6. ¿En qué momento conviene sacar el pasaje?

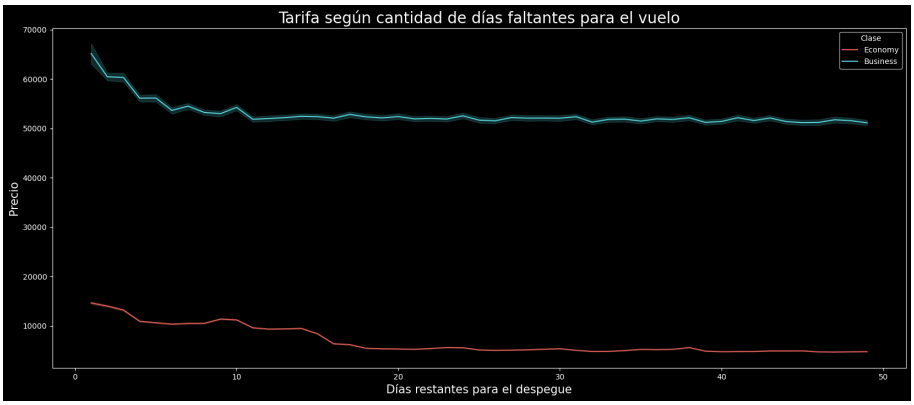


Figura 3.11: Tarifa según cantidad de días restantes para el despegue.

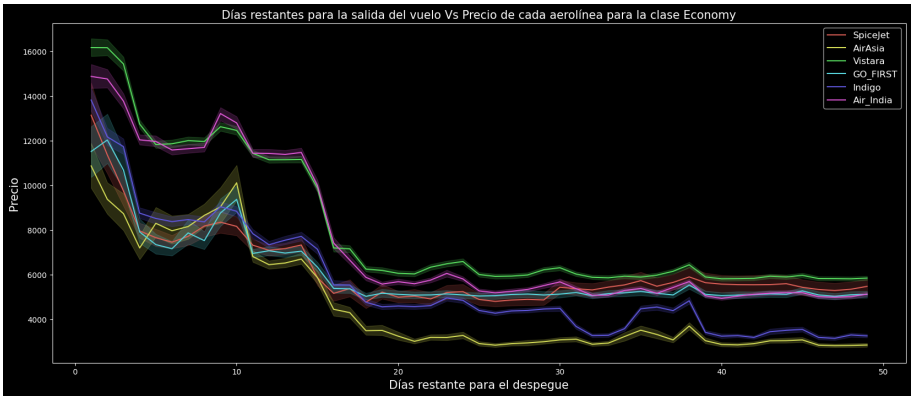


Figura 3.12: Días restantes para la salida del vuelo vs precio de cada aerolinea para la clase Economy.

Se puede ver un patrón en la forma en que evolucionan los precios según la cantidad de días restantes. Para la clase Economy, cuando quedan más de 20 días el precio es más estable. Cuando faltan menos de 20 días, los precios se incrementan drásticamente, llegando a su tope cuando faltan 2 días para el despegue.

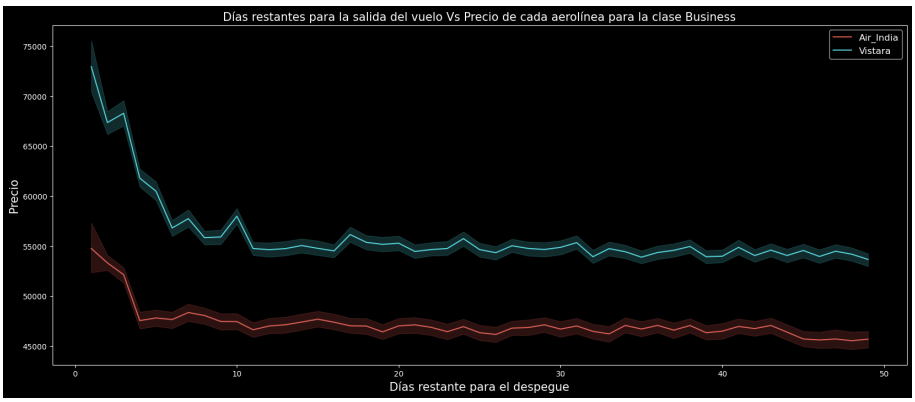
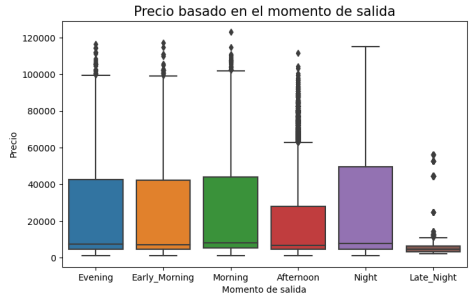
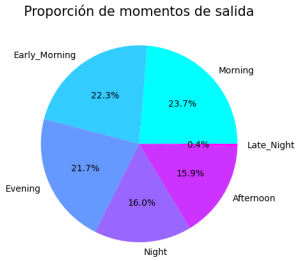
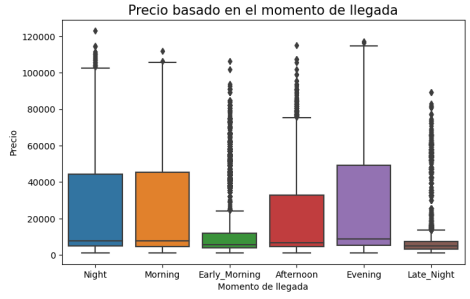


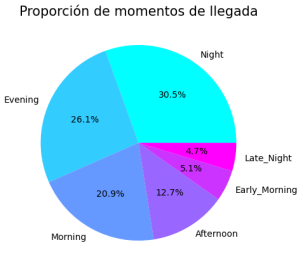
Figura 3.13: Días restantes para la salida del vuelo vs precio de cada aerolinea para la clase Business.

Para la clase Business también existe un patrón similar, que se da a partir de los 10 días. Como mencionamos antes, podemos ver que los precios de Vistara son más elevados que los de Air India en la clase Business y similares en la clase Economy.









La mayoría de los pasajeros eligen salir a la mañana o llegar a la noche a destino. Sin embargo, salir a la tarde o tarde por la noche suele ser más económico, que es cuando tiende a haber menos personas. El precio del pasaje

es similar para vuelos con horario de salida temprano en la mañana, mañana o tarde. Salir por la noche puede ser más caro en general.

En cuanto al momento de llegada, lo más conveniente es llegar temprano por la mañana o tarde por la noche ya que suele ser más económico. Es cuando suele haber menos gente. Llegar a la noche o por la mañana cuesta casi lo mismo, pero llegar por la tarde podría ser más costoso.

CAPÍTULO 4

Preprocesamiento de Datos

4.1. Transformación de Variables

Algoritmo 4.1: Identificación de variables categóricas

```
cat_cols = list(df.select_dtypes(include=['object']).columns)
print(f"Cantidad de columnas categ ricas: {len(cat_cols)}")
print(f"Columnas categ ricas:\n{cat_cols}")
```

Cantidad de columnas categóricas: 8

Columnas categóricas: ['airline', 'flight', 'source_city', 'departure_time', 'stops', 'arrival_time', 'destination_city', 'class']

Algoritmo 4.2: Identificación de variables numéricas

```
var_numericas = list(df.drop(columns=['price']).select_dtypes(exclude=['object'])
.columns)
print(f"Cantidad de columnas num ricas: {len(var_numericas)}")
print(f"Columnas num ricas:\n{var_numericas}")
```

Cantidad de columnas numéricas: 2

Columnas numéricas: ['duration', 'days_left']

Eliminamos la columna "Flight"que contiene 1.561 valores únicos.

Algoritmo 4.3: Necesitamos transformar las variables categóricas a numéricas.

```
df['stops'] = df['stops'].replace({'zero': 0, 'one': 1, 'two_or_more': 2})
.astype(int)

df['class'] = df['class'].replace({'Economy': 0, 'Business': 1}).astype(int)
```

Algoritmo 4.4: Generamos dummies para el resto de las variables categóricas con la técnica de get_dummies.

```
variables_dummies = ["airline","source_city","destination_city",
"departure_time","arrival_time"]

dummies = pd.get_dummies(df[variables_dummies], drop_first= True)

data_encoded = pd.concat([df,dummies],axis=1)

data_encoded = data_encoded.drop(variables_dummies,axis=1)
```

Ahora necesitamos escalar las variables numéricas. El escalamiento de datos es un paso de preprocesamiento en machine learning que apunta a estandarizar

el rango o la escala de las variables de entrada. El objetivo del escalado es garantizar que cada característica tenga una escala o rango similar, lo que puede facilitar que algunos modelos de machine learning converjan más rápido y mejoren su rendimiento.

La elección del método de escalamiento depende de la distribución y el rango de las variables de entrada, así como del modelo de machine learning específico que se utiliza. En general, es una buena práctica escalar los datos antes de entrenar un modelo de aprendizaje automático, a menos que se sepa que el modelo es insensible a la escala de las variables de entrada.

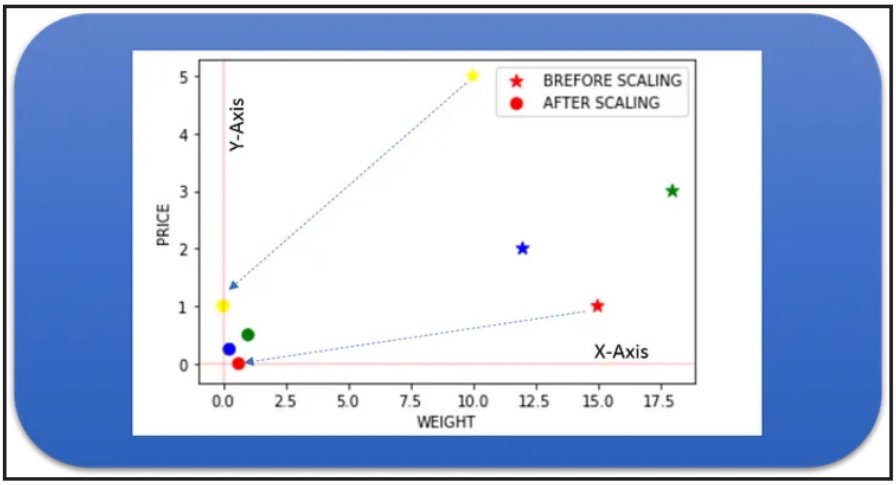
Existen muchas técnicas para realizar el proceso de escalamiento de los datos, las más utilizadas son:

- Logarítmica
- StandardScaler
- Min Max Scaler
- Quantile Transformer
- Robust Scaling
- Absolute Maximum Scaling

Para el presente caso de estudio utilizaremos MinMaxScaler, que escala los datos a un rango fijo de valores entre 0 y 1. Funciona restando el valor mínimo de cada característica y luego dividiendo por el rango (es decir, la diferencia entre los valores máximo y mínimo). La ventaja de este método es que conserva la forma de la distribución original y no cambia la posición relativa de los puntos de datos. También es relativamente fácil de usar y comprender. Sin embargo, es posible que MinMaxScaler no funcione bien si la distribución de los datos está muy sesgada o tiene valores atípicos, ya que puede magnificar los efectos de estos valores atípicos. En otras palabras, este scaler responde bien si la desviación estándar es pequeña y cuando una distribución es no Gausiana. Este scaler es sensible a outliers.

Min-Max scaler

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$



4.2. ¿Qué variables influyen más en el precio?

- Para visualizar la correlación lineal de cada variable con el precio, trazamos una matriz de correlación.

De acuerdo con la correlación de Pearson, las features más relevantes son: **la clase de asiento, la aerolínea, la duración del vuelo y la cantidad de escalas realizadas.** En especial, existe una fuerte correlación lineal entre la clase de asiento y el precio.

- Con el método "KBest Features"vemos si existen otras variables importantes que tengan o no correlación lineal con el precio.

KBest Features es una técnica en feature engineering que apunta a seleccionar las "kçaracterísticas más importantes de una base de datos basado en algunas métricas de estadística. La idea detrás de esta técnica es reducir la dimensionalidad del dataset al seleccionar solo las variables más informativas, que puedan mejorar el rendimiento de algunos modelos de machine learning y reducir overfitting.

KBest Feature selection funciona haciendo un ranking con métricas estadísticas tales como el test chi-cuadrado, la información mutua o el f-score, y selecciona las top "k"variables con los puntajes más altos. La métrica específica usada depende del tipo de datos y del problema en cuestión.

De acuerdo con esta técnica, las variables más importantes son **la clase, la aerolínea, la ciudad de origen y la ciudad de destino.**

- ¿Por qué usamos estos dos métodos?

KBest Feature Selection puede seleccionar características que no tengan correlación lineal alta con la variable objetivo pero que aún así sean informativas

para el modelo. El coeficiente de correlación de Pearson, en cambio, puede pasar por alto relaciones no lineales o no monótonas importantes.

En la práctica suele convenir usar múltiples técnicas de selección de atributos y evaluar su rendimiento en un set de validación para elegir el mejor conjunto de variables para el modelo de machine learning. Eso puede ayudar a garantizar que las variables seleccionadas sean relevantes, informativas y no redundantes.

- ¿Qué variables conviene seleccionar para este dataset?

Teniendo en cuenta que la mayoría de las variables están incluidas ya sea por la correlación de Pearson o por kbest feature selection, no eliminaremos más features de la base de datos.

CAPÍTULO 5

Modelos de Machine Learning

5.1. Entrenamiento y comparación de modelos

Ya separamos la variable objetivo "price"del dataset en x e y. Ahora necesitamos separar los datos en train y test.

Luego creamos objetos de los modelos de regresión con los hyper-parámetros que vienen por default. Los modelos que evaluamos fueron:

- LinearRegression
- DecisionTreeRegressor
- BaggingRegressor
- RandomForestRegressor
- SVR
- XGBRegressor
- KNeighborsRegressor(n_neighbors=5)
- ExtraTreesRegressor
- Ridge
- Linear_model.Lasso(alpha=0.1)

Generamos una base de datos que resume las métricas para cada modelo. Las métricas que vamos a evaluar son:

- Adj R Square
- Mean Absolute Error MAE
- Root Mean Squared Error RMSE
- Mean Absolute Percentage Error MAPE
- Mean Squared Error MSE
- Root Mean Squared Log Error RMSLE

- R2 score

Ordenamos el resultado por Adj_R_Square de mayor a menor y visualizamos los datos. De la tabla de resultados, los modelos top 3 comparando los errores por Adj_R_Square y R2_Score son:

1. Random Forest Regressor
2. Bagging Regressor
3. Extra Trees Regressor

5.2. Random Forest Regressor

Un modelo Random Forest está compuesto por un conjunto (ensemble) de árboles de decisión individuales. Cada uno de estos árboles es entrenado con una muestra aleatoria extraída de los datos de entrenamiento originales mediante bootstrapping. Esto implica que cada árbol se entrena con un conjunto de datos ligeramente diferente. En cada árbol individual, las observaciones se distribuyen a través de bifurcaciones (nodos), dando forma a la estructura del árbol hasta llegar a un nodo terminal. La predicción de una nueva observación se obtiene al agregar las predicciones de todos los árboles individuales que conforman el modelo.

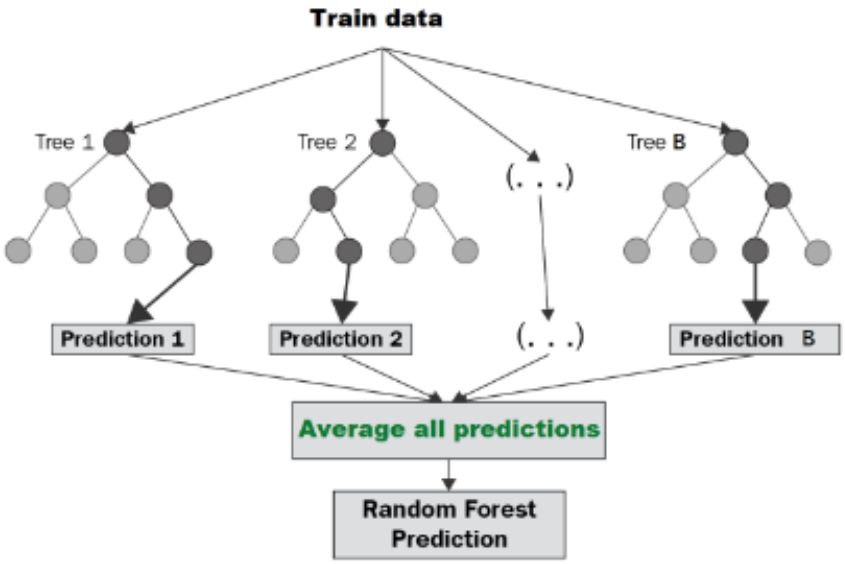


Figura 5.1: Random Forest Regressor.

Los algoritmos de Random Forest tienen tres hiperparámetros principales, que deben configurarse antes del entrenamiento:

- Tamaño del nodo
- Cantidad de árboles
- Cantidad de características muestreadas

Podríamos mejorar el rendimiento del modelo realizando un ajuste de hiperparámetros pero como el rendimiento es bueno, decidimos no hacerlo.

Entrenamos los datos con el modelo Random Forest Regressor, guardamos las predicciones y comparamos el resultado.

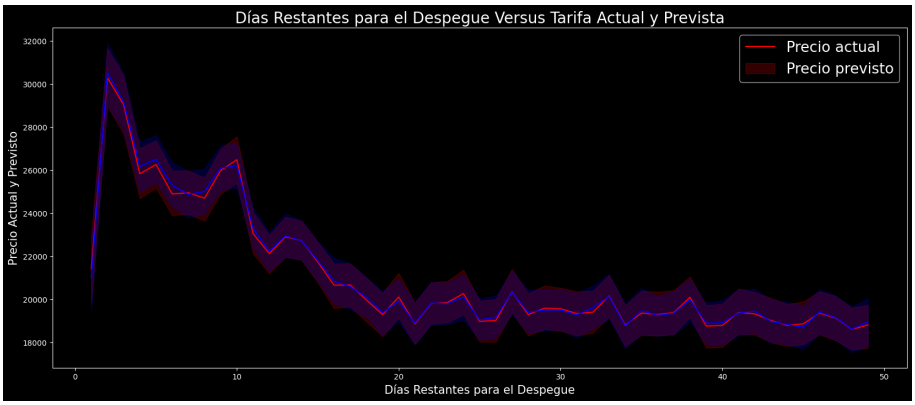


Figura 5.2: Dias restantes para el despegue vs tarifa actual y prevista.

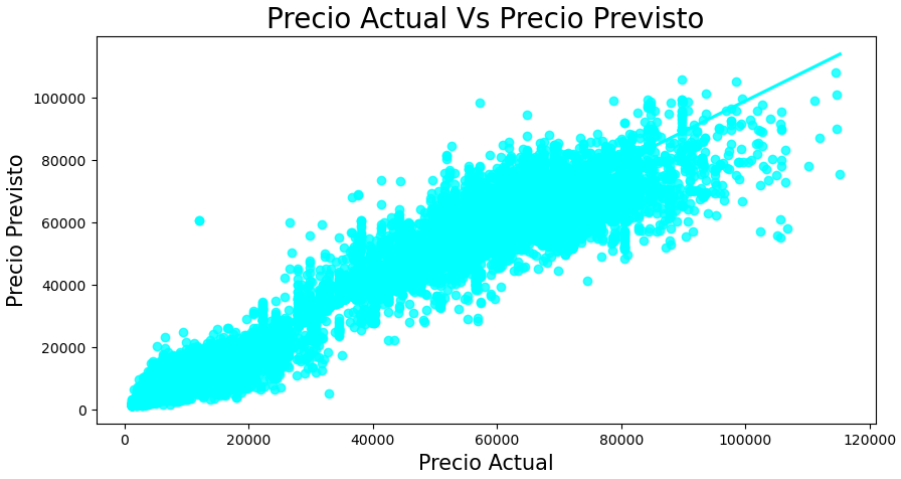


Figura 5.3: Comparación entre el precio actual y el precio previsto.

CAPÍTULO 6

Conclusiones

1. La brecha entre el precio de asientos Business y Economy es grande. Si bien existen tarifas Economy atípicas iguales o más caras que la clase Business, la mediana de la clase Business es aproximadamente 5 veces superior que la clase Economy.

2. Las únicas dos empresas que ofrecen asientos Business son Vistara y Air India. Vistara tiene precios más caros que Air India para la clase Business y similares para la clase .Ecconomy". Air India y Vistara brindan servicios .Ecconomy.a un precio más elevado que el resto de las compañias. Los diferentes análisis realizados tienden a mostrar que AirAsia se trata de una empresa low cost.

3. Salir a la tarde o tarde por la noche suele ser más económico. El precio del pasaje es similar para vuelos con horario de salida temprano en la mañana, mañana o tarde. Salir por la noche puede ser más caro en general. En cuanto al momento de llegada, lo más conveniente es llegar temprano por la mañana o tarde por la noche ya que suele ser más económico. Llegar a la noche o por la mañana cuesta casi lo mismo, pero llegar por la tarde podría ser más costoso.

4. A mayor cantidad de escalas realizadas, mayor es el precio de cada pasaje y mayor es la diferencia de precios entre una compañia y otra.

5. A mayor duración del vuelo, el precio se incrementa para ambas clases.

6. Se puede ver un patrón en la forma en que evolucionan los precios según la cantidad de días restantes para la salida del vuelo. Para la clase Economy, cuando quedan más de 20 días el precio es más estable. Cuando faltan menos de 20 días, los precios se incrementan drásticamente, llegando a su tope cuando faltan 2 días para el despegue. Para la clase Business también existe un patrón similar, que se incrementan cuando faltan 10 días o menos.

7. De los modelos de Machine Learning que probamos, elegimos Random Forest Regressor con un puntaje R2 de 0.984583 y un R2 ajustado de 0.984582, mejor que el resto de los modelos.

Bibliografía

[1] Krishna, V. 2009. Auction theory. *Academic press.*

[2] Gallego, G., Van Ryzin, G. 1994. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management science*, 40(8), 999-1020.