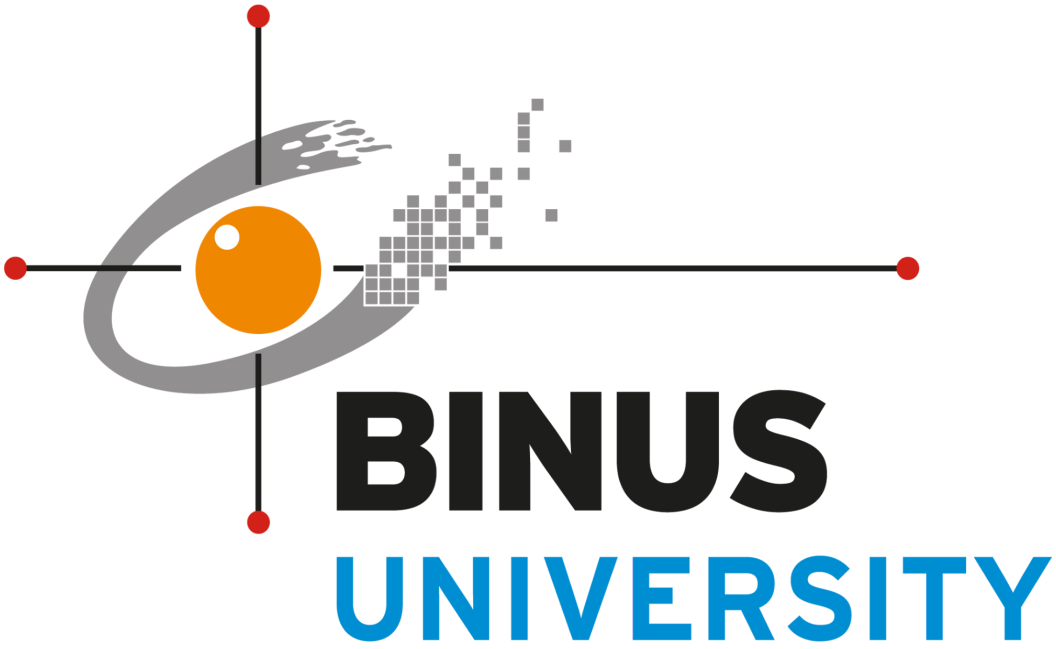
Laporan Akhir Big Data Analysis

Big Data Processing



Ditulis Oleh :
Imanuel Tio - 2401959194
Felix Tio - 2440004880
Albert Silva - 2440012352
Elvis Susanto - 2440011532

Pendahuluan

Salam sejahtera untuk kalian semua pembaca. Sebelum melanjutkan, kami sebagai penulis ingin menyampaikan terima kasih sebesar-besarnya telah diberikan kesempatan yang indah ini untuk menyelesaikan laporan dan segala eksperimen di dalamnya, kepada pihak Universitas Bina Nusantara, dan kepada Bapak Dosen Fepri Putra Panghurian, S.Kom, M.T.I. sebagai penyalur tugas ini.

Sebagai sebuah pembuka, paper ini dapat diambil garis besar bahwa berisi dokumentasi pengerjaan analisis dari sebuah dataset berisikan tiket-tiket dan jadwal penerbangan perusahaan-perusahaan aviasi di India. Analisis dapat dilihat terbagi menjadi dua jenis dimana yang pertama menunjukkan sisi deskriptif yang memaparkan isi data dari segi statistika, dan sebuah analisis yang mencakup perihal seperti hitungan prediktif.

Kami berharap bacaan ini dapat membantu memperluas pandangan kalian para pembaca untuk topik Big Data Analysis, atau bahkan dapat menggapai mereka yang lebih mengerti untuk memberikan saran dan masukan dalam lebih merapikan hasil kerja kami. Sekali lagi kami sampaikan terima kasih yang sebesar-besarnya dan permohonan maaf bilamana terdapat apapun yang terasa dapat menyinggung pihak manapun. Selamat membaca.

Isi

(Sebelum membaca dokumentasi ini, bilamana anda ingin sambil mengakses dan melihat langsung source code dan dataset dapat mengakses tautan berikut) :
Dataset Clean CSV :
https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction

I. Persiapan Workspace

Dalam tugas kali ini, kami menentukan bahwa analisis akan dilakukan diatas dataset berisikan dokumentasi tiket penerbangan perusahaan-perusahaan aviasi di India, dengan menggunakan bahasa pemrograman Python di atas compiler yang adalah Google Colaboratory.

I. i. Persiapan Workspace - Import Library

Untuk dapat membentuk analisis, dibutuhkan keberadaan beberapa fungsi yang terdapat di library-library seperti pandas, NumPy, seaborn, dan lainnya. Maka hal pertama yang dilakukan adalah mengimport library-library tersebut.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import xgboost as xgb
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn.metrics import mean_squared_error,r2_score
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import StratifiedKFold
```

I. ii. Persiapan Workspace - Import Dataset

Untuk memasukkan file yang adalah dataset dengan format .CSV dari website kaggle, data mentah dari website tersebut dipindahkan ke dalam repository GitHub dengan tujuan mempermudah proses pengambilan data.

```
!wget https://raw.githubusercontent.com/Tio6536/BigData/main/Clean_Dataset.csv

--2022-06-16 05:41:37--  https://raw.githubusercontent.com/Tio6536/BigData/main/Clean_Dataset.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24683279 (24M) [text/plain]
Saving to: ‘Clean_Dataset.csv’

Clean_Dataset.csv   100%[===================>]  23.54M  --.-KB/s    in 0.1s

2022-06-16 05:41:38 (164 MB/s) - ‘Clean_Dataset.csv’ saved [24683279/24683279]
```

I. iii. Persiapan Workspace - Pembersihan Dataset

Untuk menggunakan dataset, file .CSV mentah ditampung di sebuah variabel dataframe.

```
df = pd.read_csv("Clean_Dataset.csv")
```

```
df.describe()
```

	Unnamed: 0	duration	days_left	price
count	300153.000000	300153.000000	300153.000000	300153.000000
mean	150076.000000	12.221021	26.004751	20889.660523
std	86646.852011	7.191997	13.561004	22697.767366
min	0.000000	0.830000	1.000000	1105.000000
25%	75038.000000	6.830000	15.000000	4783.000000
50%	150076.000000	11.250000	26.000000	7425.000000
75%	225114.000000	16.170000	38.000000	42521.000000
max	300152.000000	49.830000	49.000000	123071.000000

Dengan keberadaan beberapa data redundant yang tidak memiliki nilai relevan untuk analisis, data-data tersebut disama ratakan nasibnya untuk dihilangkan dari dataset.

```
df.isnull().sum()
```

```
Unnamed: 0          0
airline             0
flight              0
source_city         0
departure_time      0
stops               0
arrival_time        0
destination_city    0
class               0
duration            0
days_left           0
price               0
dtype: int64
```

```
df.drop("Unnamed: 0", axis=1, inplace=True)
```

Setelah mencapai titik ini, data sudah siap untuk diberikan perubahan, dimanipulasi dan dilakukan analisis di atasnya. Namun sebelum itu dengan guna kerapihan kerja, ada baiknya yang diganti dan dicobai adalah sebuah duplikat dari data tersebut, sehingga akan lebih singkat prosesnya bilamana datang waktunya untuk membandingkan dengan sumber awal data.

```
vari = df.copy(deep=True)
```

Sekarang kita memegang 2 dataset yang adalah dataframe awal sekaligus sebuah duplikat untuk dataframe tersebut. Sebagai pembuktian untuk apa saja variabel yang relevan untuk analisis, penggunaan fungsi .describe() hanya akan menunjukkan 3 kolom dengan masing-masing kolom tersebut mencakup nilai numerik.

```
vari.describe()
```

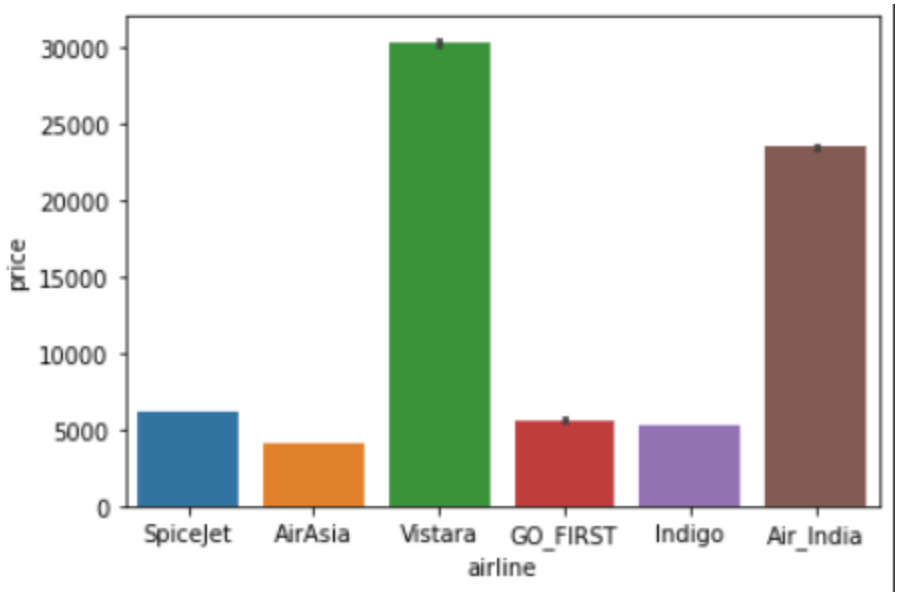
	duration	days_left	price
count	300153.000000	300153.000000	300153.000000
mean	12.221021	26.004751	20889.660523
std	7.191997	13.561004	22697.767366
min	0.830000	1.000000	1105.000000
25%	6.830000	15.000000	4783.000000
50%	11.250000	26.000000	7425.000000
75%	16.170000	38.000000	42521.000000
max	49.830000	49.000000	123071.000000

II. Visualisasi Data

Bagian pertama dari dua tahap analisis kami ialah memvisualisasikan data yang sudah didapatkan. Kami mencoba memvisualisasikan data dengan 6 kasus tertentu yang dibungkus dalam bentuk pertanyaan.

II. i. Visualisasi Data - Perbedaan Harga Antar Maskapai

Dari hasil visualisasi dapat disimpulkan bahwa setiap maskapai memiliki rata-rata harga tiket pesawat yang berbeda-beda dimana tiket pesawat dengan rata-rata harga terendah dimiliki oleh maskapai airasia sedangkan yang tertinggi dimiliki oleh maskapai vistara



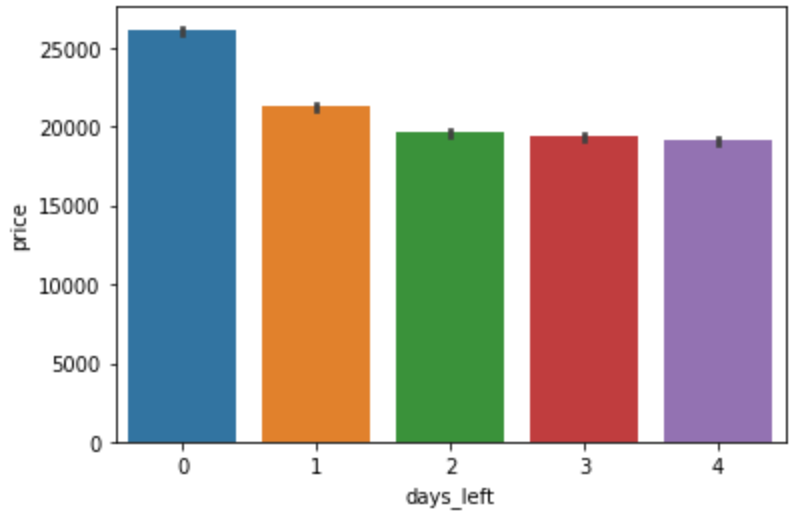
Maskapai Vistara dan Air_India memiliki rata-rata harga yang cukup tinggi dibanding dengan kompetitornya dikarenakan adanya kelas Business pada maskapai tersebut sedangkan maskapai lain tidak memiliki kelas Business. Kelas business ini sendiri memiliki harga yang cukup jauh jika dibandingkan dengan kelas Economy maskapai tersebut.

II. ii. Visualisasi Data - Pengaruh Booking Terhadap Harga Tiket

Untuk memudahkan pengelompokan data, maka akan dilakukan pengelompokan dengan jangka 10 hari dari satu kelompok ke kelompok data lainnya

```
0 = 1-10 hari jarak antara pemesanan dengan penerbangan
1 = 11 - 20 hari jarak antara pemesanan dengan penerbangan
2 = 21 - 30 hari jarak antara pemesanan dengan penerbangan
3 = 31 - 40 hari jarak antara pemesanan dengan penerbangan
4 = 41 - 49 hari jarak antara pemesanan dengan penerbangan
```

Maka didapatlah data sebagai berikut dimana dapat dilihat bahwa semakin jauh jarak antara pemesanan dengan penerbangan maka harga tiket dari suatu maskapai akan jauh lebih murah. Pemesanan tiket pada kelompok 1 memiliki rata-rata harga tertinggi jika dibanding kelompok kelompok lain, sedangkan kelompok 5 memiliki rata-rata harga termurah.



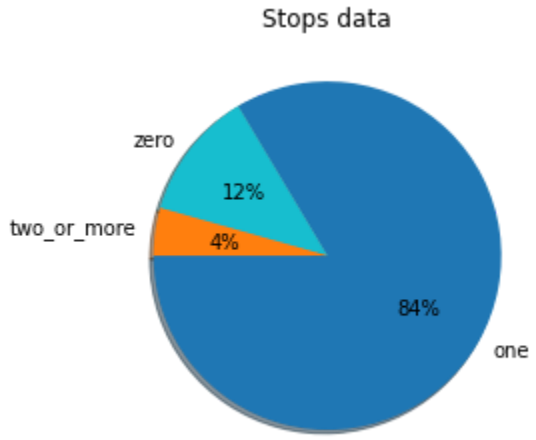
Lalu rata-rata user pun memesan tiket pesawat 15 hari keatas sebelum penerbangan hal ini dipengaruhi beberapa faktor seperti harga dari tiket pesawat yang jauh lebih murah jika dibanding kelompok 0 dan harus menyiapkan jadwal dari jauh jauh hari.

	days_left	Occurence
0	25	6633
1	18	6602
2	39	6593
3	32	6585
4	26	6573
5	24	6542
6	19	6537
7	31	6534
8	33	6532
9	40	6531

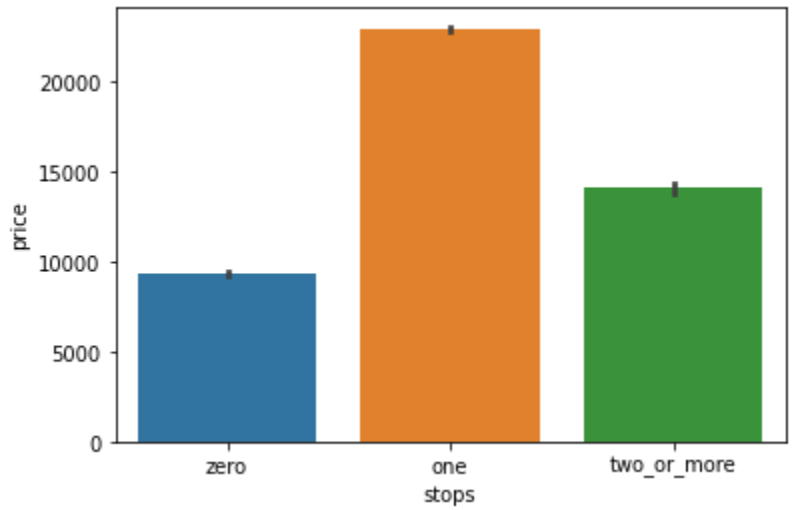
II. iii. Visualisasi Data - Pengaruh Pemberhentian Penerbangan Terhadap Harga Tiket

Pertama kita akan melakukan pengecekan terhadap persentase seberapa banyak suatu penerbangan melakukan pemberhentian

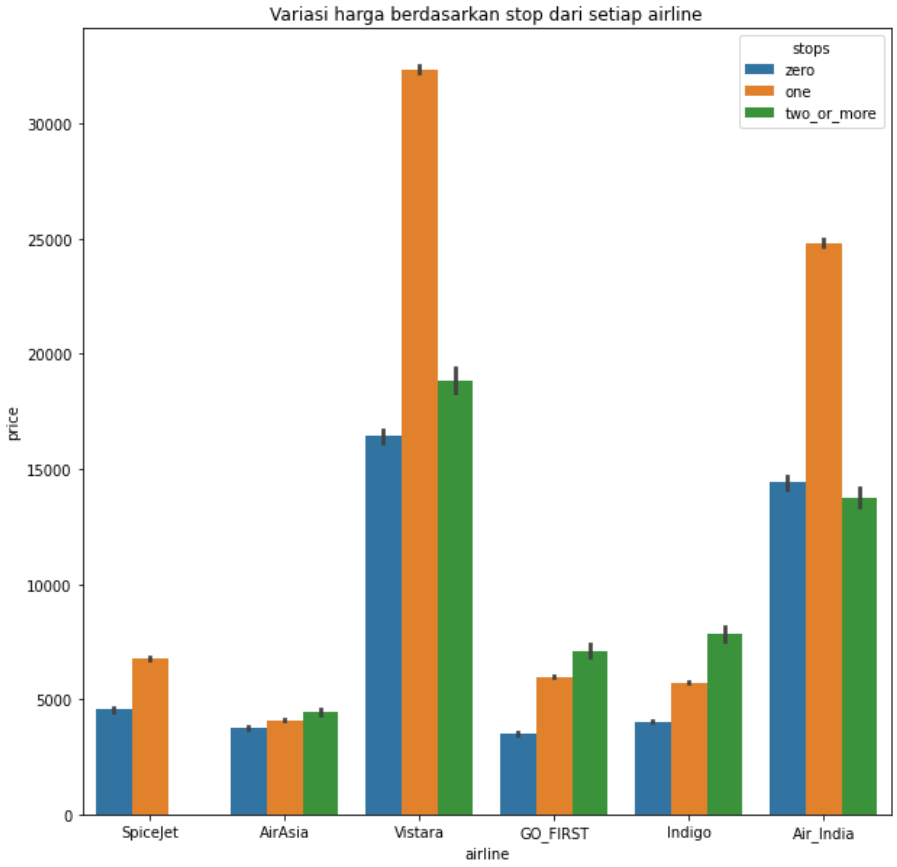
```
fig, ax = plt.subplots()
ax.pie(stops['Occurence'], labels = stops['Number of Stops'],
          colors = ['tab:blue', 'tab:cyan','tab:orange'],
          autopct='%.0f%%',
          shadow = True,
          startangle = 180)

ax.set_title('Stops data')
plt.show()
```



Dapat dilihat bahwa penerbangan yang langsung ke tujuan memiliki rata-rata harga termurah dibanding penerbangan dengan pemberhentian. Secara logika semakin banyak pemberhentian maka harga dari suatu tiket pesawat akan semakin tinggi namun nyatanya penerbangan dengan 1 pemberhentian memiliki rata-rata harga lebih tinggi jika dibanding penerbangan dengan 2 atau lebih pemberhentian. Hal ini tentunya memiliki faktor faktor lain yang mendasari alasan tersebut.



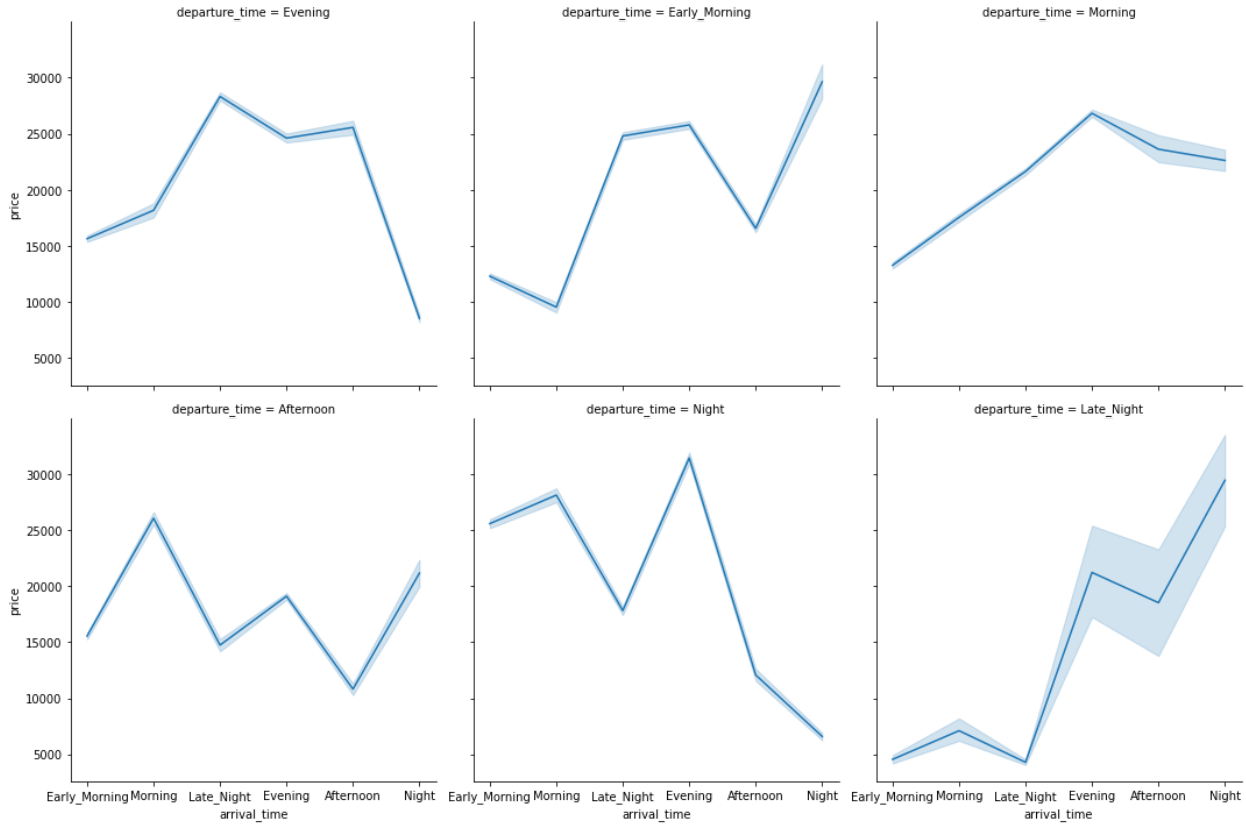
Pada grafik tersebut dapat dilihat bahwa vistara dan Air_india memiliki 1 pemberhentian yang lebih mahal jika dibanding 2 pemberhentian hal ini disebabkan oleh cost efficiency dari sebuah bahan bakar pesawat. Dimana jika semakin banyak pemberhentian maka harga tiketnya kemungkinan akan lebih murah serta ditentukan pula oleh jarak tempuh dari pesawat tersebut.



II. iv. Visualisasi Data - Pengaruh Jadwal Penerbangan Terhadap Harga Tiket

Disini kami ingin mencari tahu bilamana jadwal keberangkatan dan waktu sampai dapat memberi pengaruh terhadap harga tiket pesawat. Proses ini dilakukan dengan menggunakan relplot dari library seaborn. Untuk kemudahan, jangka waktu disimpulkan menjadi 6 kategori yang adalah Evening, Early_Morning, Morning, Afternoon, Night, dan Late_Night. Maka dibentuklah 6 diagram berdasarkan 6 jangka waktu keberangkatan tersebut dibandingkan dengan sisi waktu kedatangannya, sekaligus harganya.

```
sns.relplot(data=df,kind="line",col="departure_time",x="arrival_time",y="price",col_wrap=3)
```



Dari sini dapat disajikan sebuah urutan peringkat berdasarkan harga termurah sebagai berikut :
[Peringkat]. [Keberangkatan] => [Kedatangan]

1. Evening => Night
2. Early_Morning => Morning
3. Morning => Early_Morning
4. Afternoon => Afternoon
5. Night => Night
6. Late_Night => Late_Night

II. v. Visualisasi Data - Pengaruh Durasi Penerbangan Terhadap Harga Tiket

Dalam menghitung durasi masing-masing data penerbangan, kami mencoba mengelompokkan data berdasarkan jangka durasi penerbangannya dengan panjang 10 jam.

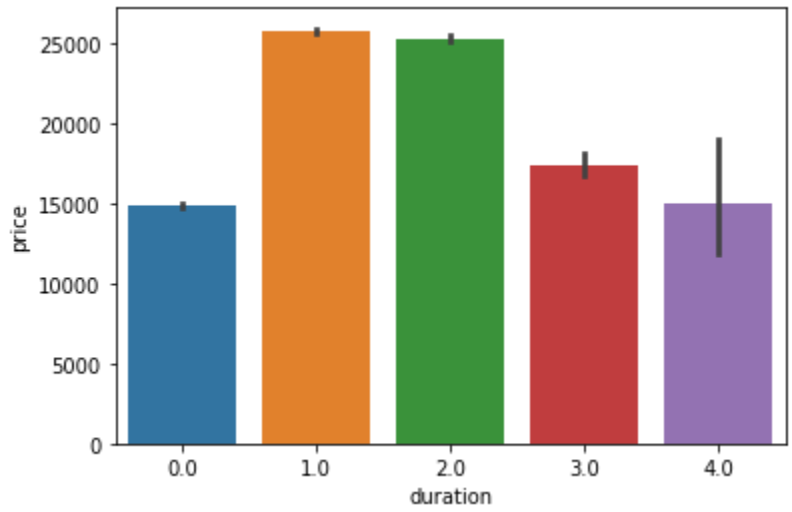
```
timeMin = df.copy(deep=True)
```

```
timeMin.loc[(timeMin['duration']>=0) & (timeMin['duration']<=10),'duration']=0
timeMin.loc[(timeMin['duration']>=10) & (timeMin['duration']<=20),'duration']=1
timeMin.loc[(timeMin['duration']>=20) & (timeMin['duration']<=30),'duration']=2
timeMin.loc[(timeMin['duration']>=30) & (timeMin['duration']<=40),'duration']=3
timeMin.loc[(timeMin['duration']>=40) & (timeMin['duration']<=50),'duration']=4
```

Pengelompokkan ini menghasilkan urutan berikut :

0 => durasi <= 10 jam di index 0 (pertama)
10 => durasi <= 20 jam di index 1
20 => durasi <= 30 jam di index 2
30 => durasi <= 40 jam di index 3
40 => durasi <= 50 jam di index 4
Maka dengan ke lima index durasi ini, dapat dibentuk perbandingan terhadap harga tiket.

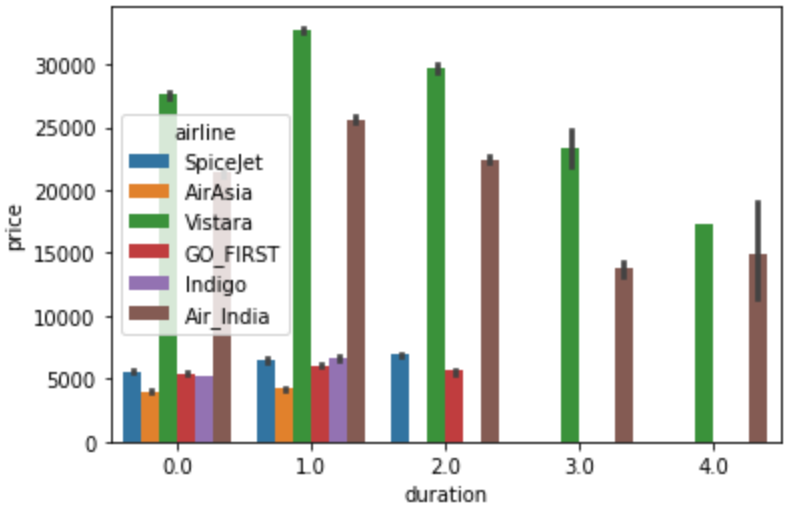
```
sns.barplot(x="duration",y="price",data=timeMin)
```



Dari sini dapat disajikan sebuah urutan peringkat berdasarkan harga termurah sebagai berikut :
[Urutan Kemurahan]. [Data yang Mana]
1. Index 0
2. Index 4
3. Index 3
4. Index 2
5. Index 1

Mengetahui ini, kita juga bisa mencari tahu durasi-durasi penerbangan ini di distribusikan oleh maskapai mana.

```
sns.barplot(x="duration",y="price",data=timeMin,hue="airline")
```



Terlihat secara rata bahwa kedua Vistara dan Air_India merupakan para pemegang harga-harga termahal tersebut.

II. vi. Visualisasi Data - Pengaruh Rute Penerbangan Terhadap Harga Tiket

Dengan beberapa variasi kota keberangkatan dan kedatangan penerbangan, kami juga mencoba membandingkan beberapa penerbangan ini diatas 6 diagram yang membagi sekian rute menjadi relplot antar kota tujuan dan harga.

III. Analisis Prediktif

Sebagai bentuk mendatangkan advanced analysis ke dalam tugas ini, kami memilih rute prediktif untuk diberlakukan pada dataset maskapai ini.

III. i. Analisis Prediktif - Persiapan Dataset

Untuk melanjutkan ke arah predictive analysis ini, kami menyiapkan sebuah duplikasi lagi terhadap dataset pertama. Sebagai tindak lanjut, dibutuhkan juga sebuah cara untuk menstandarisasi data menggunakan fit_transform() untuk masing-masing kolom yang berada pada data.

```
Label=LabelEncoder()

vari2 =df.copy(deep=True)

vari2['source_city']=Label.fit_transform(vari2['source_city'])
vari2['destination_city']=Label.fit_transform(vari2['destination_city'])
vari2['departure_time']=Label.fit_transform(vari2['departure_time'])
vari2['arrival_time']=Label.fit_transform(vari2['arrival_time'])
vari2['class']=Label.fit_transform(vari2['class'])
vari2['stops']=Label.fit_transform(vari2['stops'])
vari2['airline']=Label.fit_transform(vari['airline'])
```

Pada bagian ini pula data akan menjadi scaling menjadi bentuk numerik yang akan membantu kita dalam melakukan analisis terhadap variabel-variabel yang awalnya berbentuk kata-kata.

```
vari2.drop("flight", axis=1, inplace=True)
```

Bagian flight tidak relevan, maka disingkirkan.

III. ii. Analisis Prediktif - Dataset Splitting

Untuk membentuk sebuah skenario dimana dataset dipersiapkan untuk kebutuhan prediksi terhadap harga, kita mengeluarkan terlebih dahulu yang adalah variabel price.

```
targetvar2 = vari2.drop('price',axis=1)
```

Dilakukan pemisahan terhadap data yang ada dengan persentase 80% untuk training dan 20% untuk testing.

```
X_train_2, X_test_2, y_train_2, y_test_2 = train_test_split(targetvar2, vari2['price'], test_size=0.2, random_state=42)
```

III. iii. Analisis Prediktif - Dataset Training & Testing

Dengan bantuan library XGBoost, dan fungsi fit() untuk training data maka didapatkan sebuah nilai untuk training dan testing.

```
modelxgb.fit(X_train_2,y_train_2)

XGBRegressor(learning_rate=0.01, max_depth=5, n_estimators=150,
             objective='reg:squarederror')

resultxgbtrain=modelxgb.score(X_train_2,y_train_2)

resultxgbtrain

0.8605474365369867

resultxgbtest=modelxgb.score(X_test_2,y_test_2)

resultxgbtest

0.8603815070891747
```

Didapatkan nilai yang lumayan tinggi :
0.8605 - Data Training
0.8603 - Data Testing

III. iv. Analisis Prediktif - Prediction Result

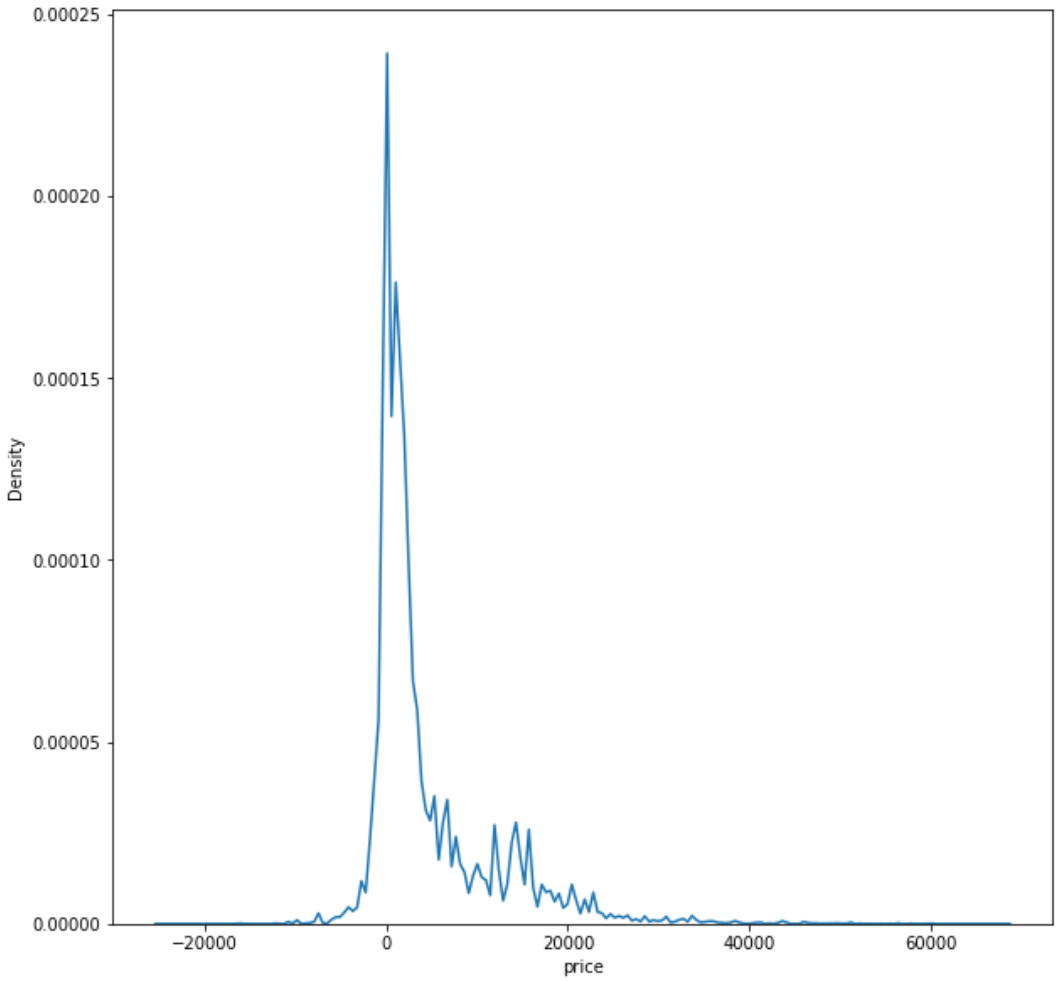
Memulai prediksi di bagian testing untuk kolom harga, kami lakukan seperti berikut :

```
y_prediction_test_2=modelxgb.predict(X_test_2)
```

Setelah mendapatkan sebuah hasil, ada baiknya untuk dibandingkan lagi dengan data yang sebelumnya sudah ada.

```
dataBanding2 = pd.DataFrame()
dataBanding2['Real'] = y_test_2
dataBanding2['Predicted'] = y_prediction_test_2

plt.figure(figsize = (10,10))
sns.kdeplot(y_test_2-y_prediction_test_2,bw=0.01)
plt.show()
```

Berikut hasil plt.show() dalam bentuk grafik (perbandingan antara data faktual dengan hasil prediksi).



Diatas merupakan grafik KDE (Kernel Density Estimation) dari hasil pengurangan antara hasil prediksi dengan hasil yang ada di dalam dataset. KDE sendiri menunjukkan semakin banyaknya data yang ada dalam sebuah kurva maka kurva tersebut akan meninggi. Dengan adanya KDE kita dapat melihat bahwa titik tertinggi pada kurva diatas memiliki ada di daerah dekat angka 0. Dari hal tersebut kita dapat menyimpulkan bahwa data paling banyak terdapat pada Hasil Data Awal - Hasil Data Prediksi memiliki hasil yang tidak jauh berbeda. Dengan begitu, kita dapat menyimpulkan bahwa model diatas dapat dibilang cukup akurat untuk menentukan harga tiket pesawat dengan rata-rata perbedaan antara hasil prediksi dengan data yang ada cukup minim.

III. v. Analisis Prediktif - Perbandingan Untuk Selisih

Dengan kedua keberadaan data faktual dengan data prediksi, kami mencoba untuk membandingkan sejauh apa selisih yang berada diantara kedua data tersebut.

```
dataLiatHasil = pd.DataFrame()
dataLiatHasil['Diff'] = y_test_2-y_prediction_test_2

dataLiatHasil.describe()
```

	Diff
count	60031.000000
mean	4609.996063
std	7121.772205
min	-25281.871094
25%	289.897217
50%	1773.419434
75%	6639.968750
max	68523.089844

III. vi. Analisis Prediktif - Visualisasi Prediksi Data

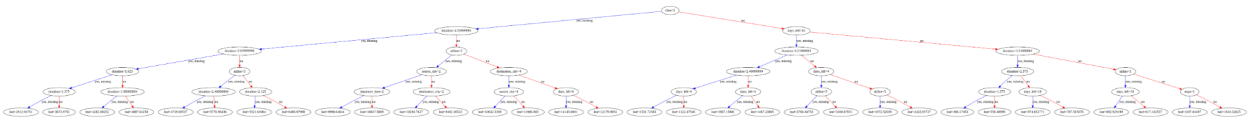
Untuk bagian ini, terdapat library SHAP yang dapat membantu membentuk visualisasi hasil dari regresi yang sudah dibuat.

```
pip install shap
```

```
import shap
```

Untuk mendapatkan visualisasi dengan ukuran yang diberikan, kita menerima bantuan dari library Matplotlib.

```
xgb.plot_tree(modelxgb)
plt.gcf().set_size_inches(100,100)
plt.show()
```



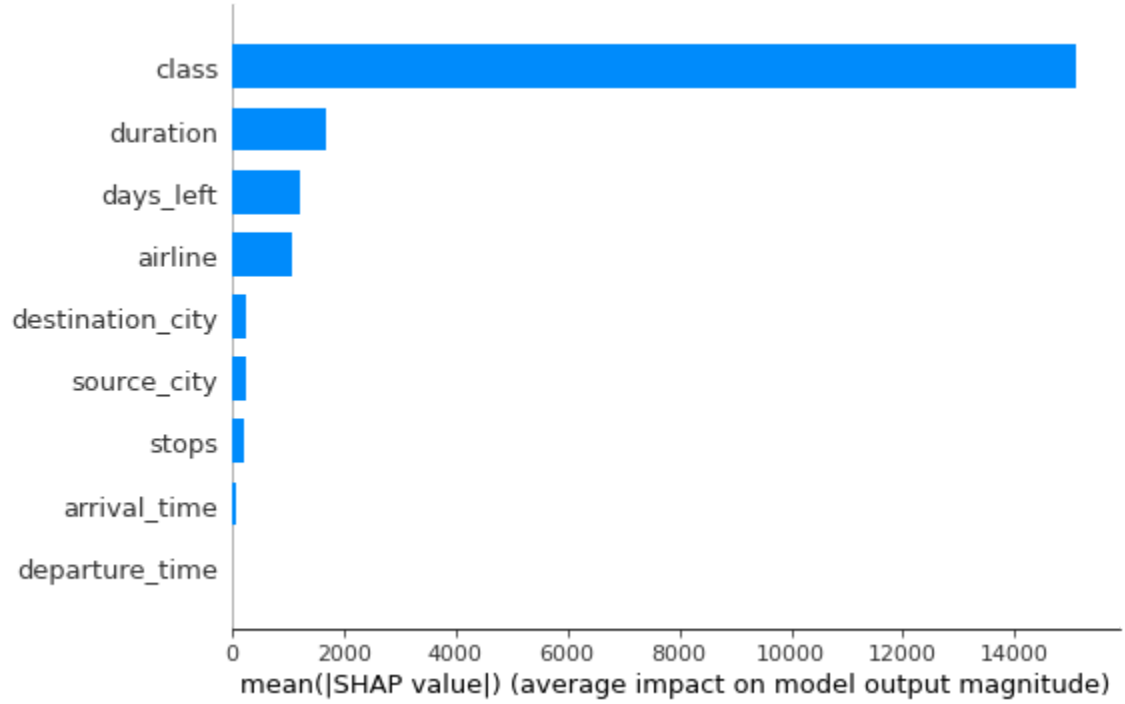
(Untuk mengakses atau mengunduh gambar yang lebih jelas, silahkan cek secara langsung dari .IPYNB yang tertera)

III. vii. Analisis Prediktif - Kesimpulan Analisis Prediktif

Sebagai satu penarikan kesimpulan terhadap bagian analisis ini, kami membentuk juga sebuah visualisasi yang membandingkan antara masing-masing kolom dan seberapa berpengaruh

mereka terhadap perubahan harga tiket.

```
explainingmodel=shap.Explainer(modelxgb)
valueshap=explainingmodel(X_test_2)

shap.summary_plot(valueshap,X_test_2,plot_type="bar")
```



Dapat dilihat bahwa kelas (antara ekonomi atau bisnis) suatu penerbangan memiliki pengaruh paling tinggi dalam menentukan harga dari sebuah tiket penerbangan.

IV. Hasil dan Kesimpulan

Eksperimen kali ini cukup terlihat jelas bahwa kami bertujuan awal memutari sekitar variabel yang adalah harga tiket. Kami sendiri sudah secara perlahan berusaha memperjelas hubungan variabel-variabel yang lain dengan harga tiket tersebut. Dapat ditarik kesimpulan dari observasi yang kami lakukan bahwa class (ekonomi dan bisnis) memiliki pengaruh yang paling besar dalam penentuan harga tiket. Akan tetapi, fakta ini tidak menutupi kemungkinan variabel lain dapat memberikan pengaruh pada penentuan harga tiket pesawat juga.

Penutup

Kami beropini bahwa pengerjaan projek ini merupakan hal yang esensial dalam membantu seseorang untuk memasuki dunia analisis data. Bahkan sekumpulan data yang nampak sederhana secara kasat mata, seperti dataset tiket pesawat ini, dapat memberikan begitu banyak wawasan terhadap pengolahan, perbandingan, dan visualisasi.

Kami berharap pula untuk siapapun yang di kemudian waktu berkesempatan untuk membaca isi laporan ini, bahwa telah lebih terbuka wawasannya untuk bidang ini. Kami juga berharap untuk laporan ini dapat mendatangkan kritik dan saran agar dapat dilakukan pengembangan yang lebih baik.

Terima kasih banyak atas kesempatan dan perhatiannya, kami memohon maaf jika terdapat kesalahan kata yang menyinggung pihak manapun.

Referensi:

[1]
Yilmazkuday, D., & Yilmazkuday, H. (2014). The Role of Direct Flights in Trade Costs. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.2432355

[2]
Bashir, D., Montanez, G. D., Sehra, S., Segura, P. S., & Lauw, J. (2020). An Information-Theoretic Perspective on Overfitting and Underfitting. ArXiv.org. https://doi.org/10.48550/arXiv.2010.06076