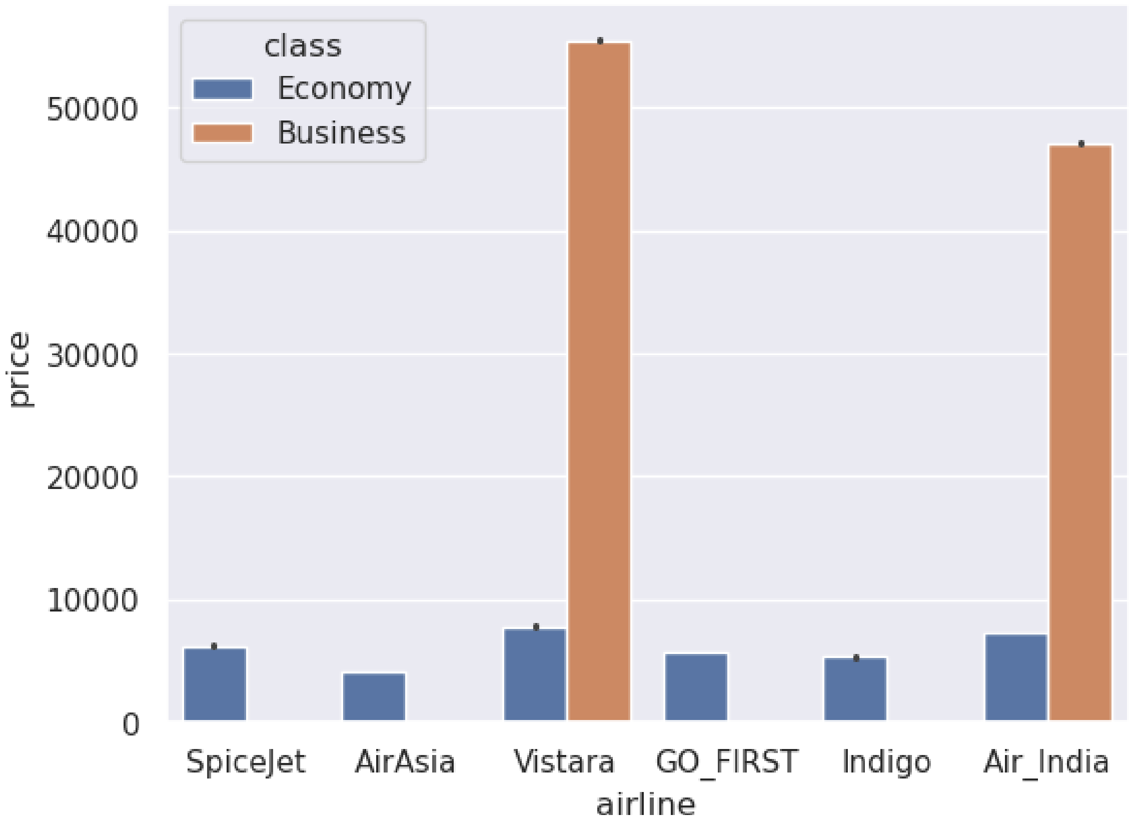
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

sns.set_theme(color_codes=True)

df = pd.read_csv('Dataset/flight.csv')
df
```

```
        Unnamed: 0   airline   flight source_city departure_time stops
\
0                0  SpiceJet  SG-8709       Delhi        Evening  zero

1                1  SpiceJet  SG-8157       Delhi  Early_Morning  zero

2                2   AirAsia   I5-764       Delhi  Early_Morning  zero

3                3   Vistara   UK-995       Delhi        Morning  zero

4                4   Vistara   UK-963       Delhi        Morning  zero

...            ...       ...      ...         ...            ...   ...

300148      300148   Vistara   UK-822     Chennai        Morning   one

300149      300149   Vistara   UK-826     Chennai      Afternoon   one

300150      300150   Vistara   UK-832     Chennai  Early_Morning   one

300151      300151   Vistara   UK-828     Chennai  Early_Morning   one

300152      300152   Vistara   UK-822     Chennai        Morning   one



         arrival_time destination_city     class  duration  days_left
price
0               Night           Mumbai   Economy      2.17          1
5953
1             Morning           Mumbai   Economy      2.33          1
5953
2       Early_Morning           Mumbai   Economy      2.17          1
5956
3           Afternoon           Mumbai   Economy      2.25          1
5955
4             Morning           Mumbai   Economy      2.33          1
5955
...               ...              ...       ...       ...        ...
...
300148        Evening        Hyderabad  Business     10.08         49
69265
```

```
300149          Night        Hyderabad  Business      10.42         49
77105
300150          Night        Hyderabad  Business      13.83         49
79099
300151        Evening        Hyderabad  Business      10.00         49
81585
300152        Evening        Hyderabad  Business      10.08         49
81585

[300153 rows x 12 columns]
```

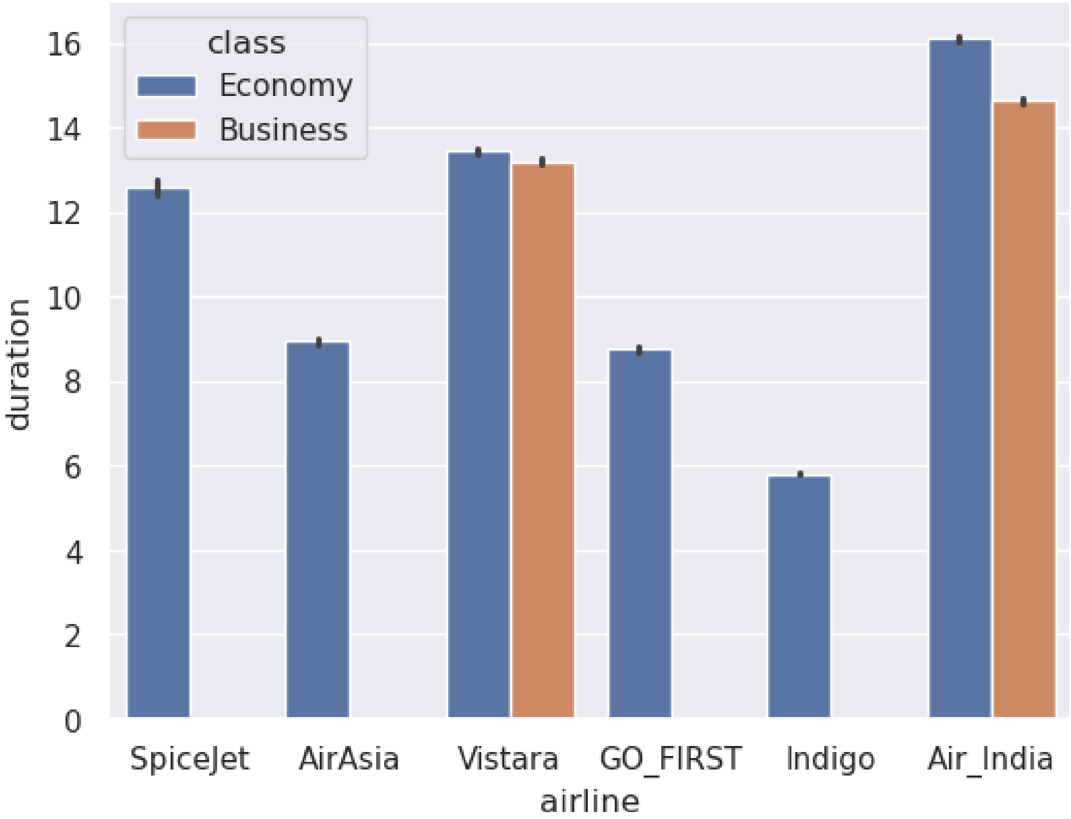
The dataset has 11 columns airline, flight, source_city, departure_time, stops, arrival_time, destination_city, class, duration days_left,price

```
df2=df.drop(columns=["Unnamed: 0","flight"])
df2

         airline source_city departure_time stops   arrival_time  \
0       SpiceJet       Delhi        Evening  zero          Night
1       SpiceJet       Delhi  Early_Morning  zero        Morning
2        AirAsia       Delhi  Early_Morning  zero  Early_Morning
3        Vistara       Delhi        Morning  zero      Afternoon
4        Vistara       Delhi        Morning  zero        Morning
...          ...         ...            ...   ...            ...
300148   Vistara     Chennai        Morning   one        Evening
300149   Vistara     Chennai      Afternoon   one          Night
300150   Vistara     Chennai  Early_Morning   one          Night
300151   Vistara     Chennai  Early_Morning   one        Evening
300152   Vistara     Chennai        Morning   one        Evening

       destination_city     class  duration  days_left  price
0                Mumbai   Economy      2.17          1   5953
1                Mumbai   Economy      2.33          1   5953
2                Mumbai   Economy      2.17          1   5956
3                Mumbai   Economy      2.25          1   5955
4                Mumbai   Economy      2.33          1   5955
...                 ...       ...       ...        ...    ...
300148        Hyderabad  Business     10.08         49  69265
300149        Hyderabad  Business     10.42         49  77105
300150        Hyderabad  Business     13.83         49  79099
300151        Hyderabad  Business     10.00         49  81585
300152        Hyderabad  Business     10.08         49  81585

[300153 rows x 10 columns]

sns.barplot(data=df,x="airline",y="price",hue="class")

<Axes: xlabel='airline', ylabel='price'>
```



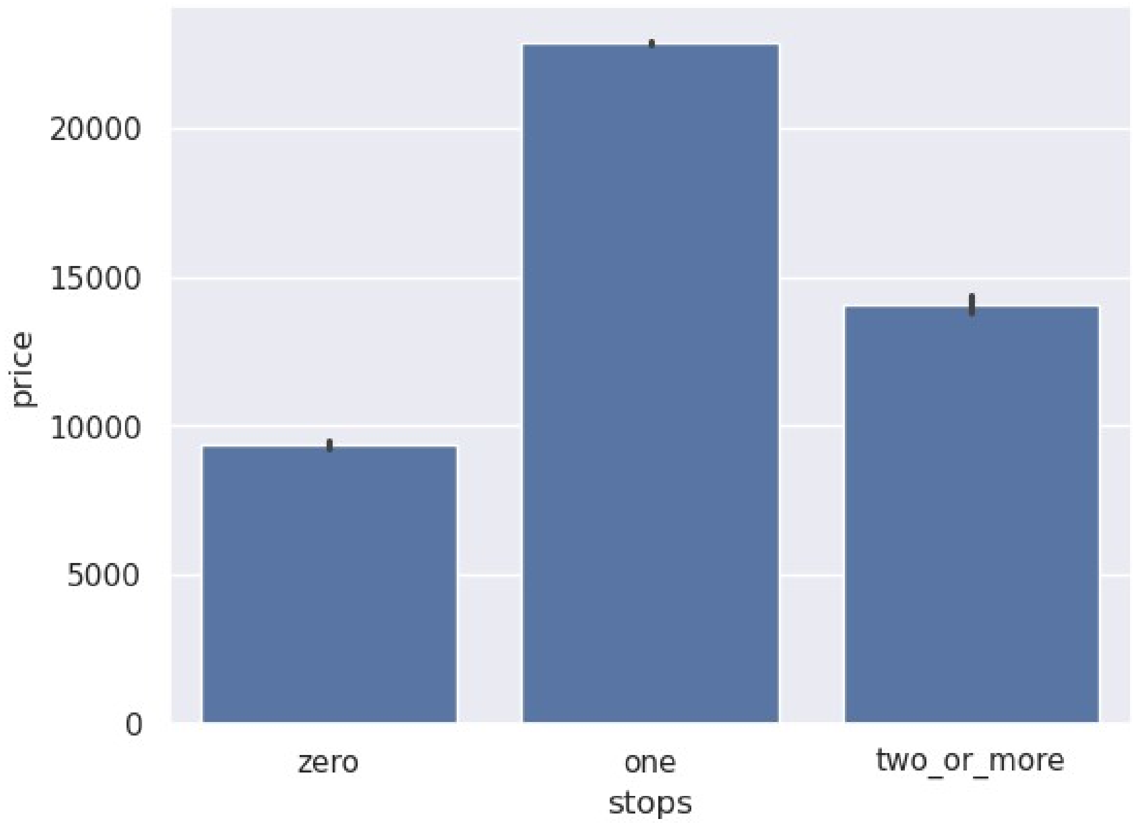
```
sns.barplot(data=df,x="airline",y="duration",hue="class")
```

```
<Axes: xlabel='airline', ylabel='duration'>
```



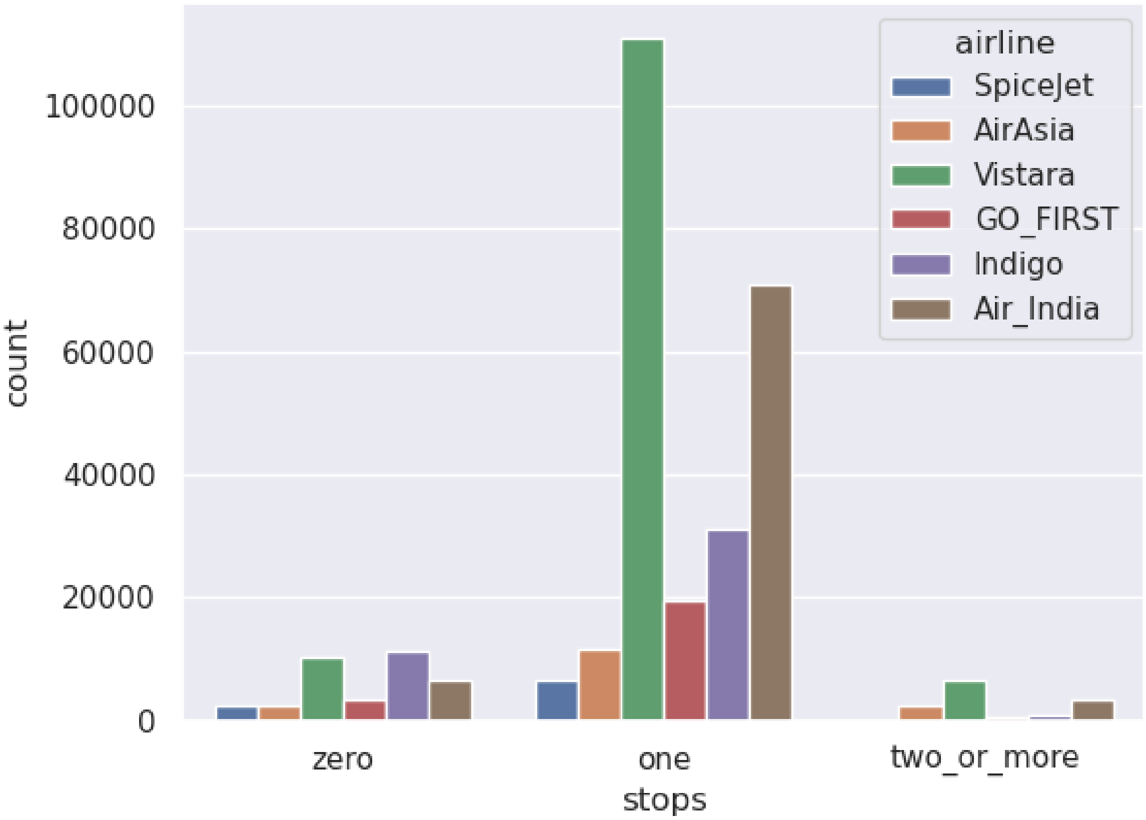
```
sns.barplot(data=df,x="stops",y="price")
```

```
<Axes: xlabel='stops', ylabel='price'>
```



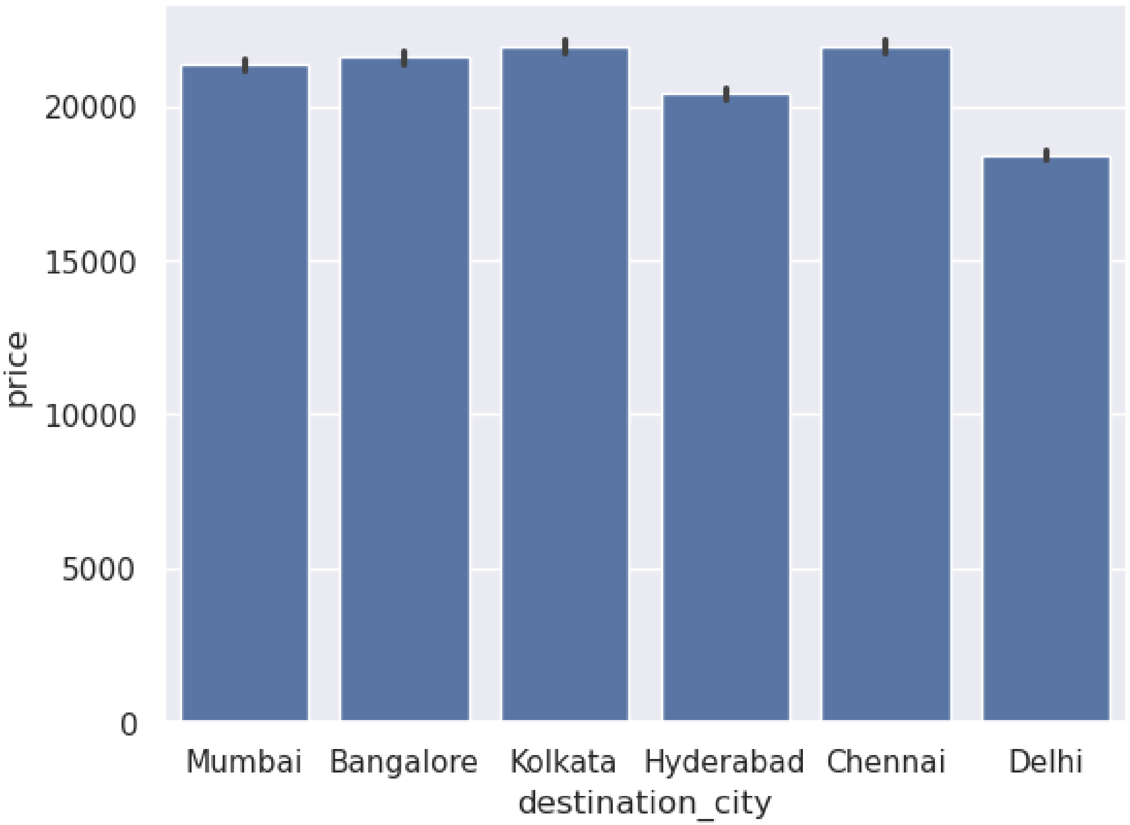
```
sns.countplot(data=df,x="stops",hue="airline")
```

```
<Axes: xlabel='stops', ylabel='count'>
```



```
sns.barplot(data=df,x="destination_city",y="price")
```

```
<Axes: xlabel='destination_city', ylabel='price'>
```



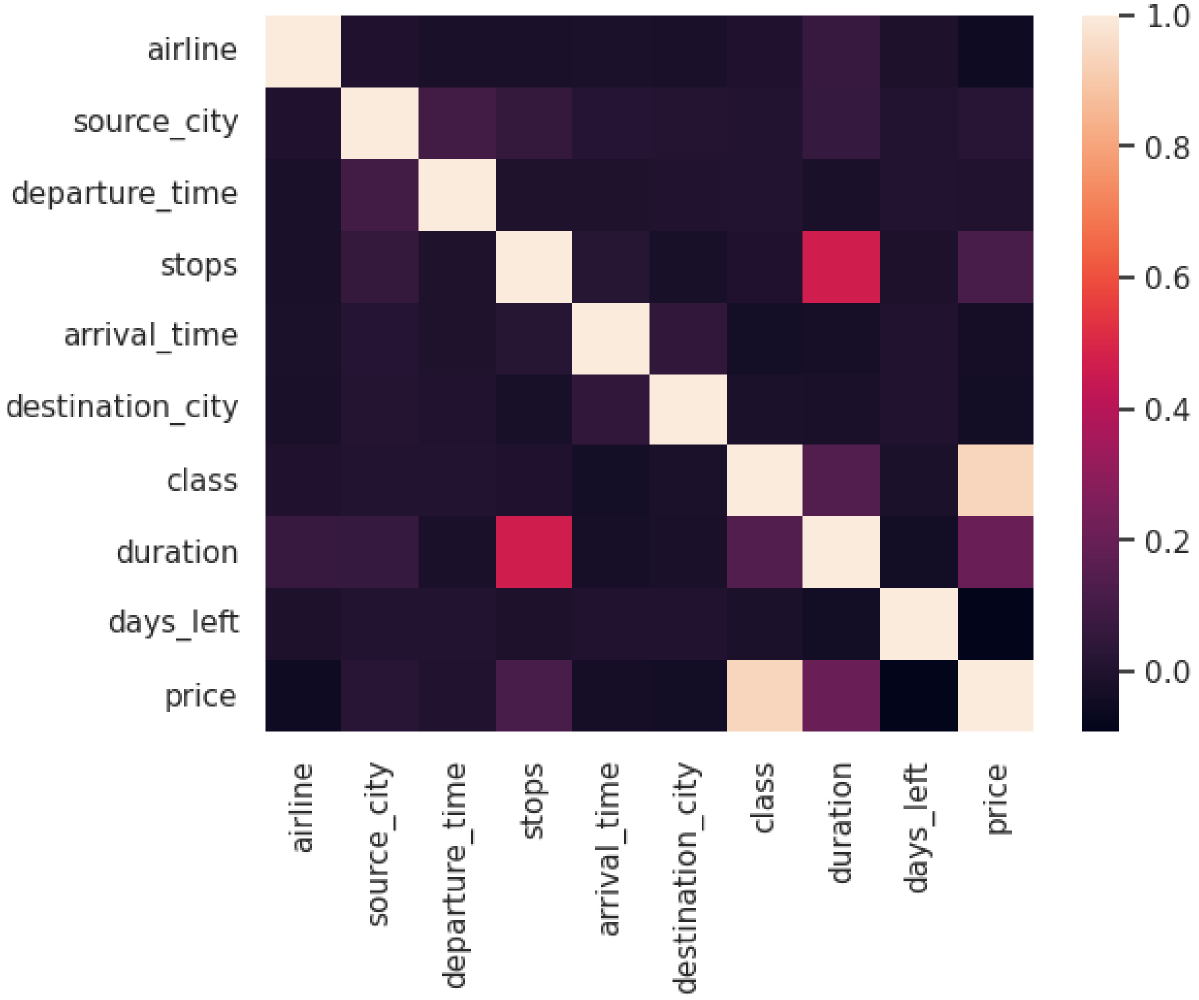
```
for col in
["airline","source_city","departure_time","arrival_time","stops","dest
ination_city","class"]:
    unique_vals = list(df2[col].unique())
    for idx in range(len(unique_vals)):
        df2[col]=df2[col].replace([unique_vals[idx]],idx)
df2.shape
```

```
/tmp/ipykernel_5275/3124727126.py:4: FutureWarning: Downcasting
behavior in `replace` is deprecated and will be removed in a future
version. To retain the old behavior, explicitly call
`result.infer_objects(copy=False)`. To opt-in to the future behavior,
set `pd.set_option('future.no_silent_downcasting', True)`
  df2[col]=df2[col].replace([unique_vals[idx]],idx)
```

```
(300153, 10)
```

```
sns.heatmap(df2.corr(),fmt=".2g")
```

```
<Axes: >
```



```
X = df2.drop('price',axis=1)
y = df2['price']

from sklearn.model_selection import train_test_split, cross_val_score,
cross_val_predict

X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.2,random_state=0)

from sklearn import metrics
import math
def ModelEval(model, Xt=X_train,Xtt=X_test,yt=y_train,ytt=y_test):
    model.fit(Xt,yt)

    yp= model.predict(Xtt)

    mae = metrics.mean_absolute_error(ytt,yp)
    mse = metrics.mean_squared_error(ytt,yp)
    r2 = metrics.r2_score(ytt,yp)
```

```python
    rmse = math.sqrt(mse)

    print(f"MAE score is {mae}")
    print(f"MSE score is {mse}")
    print(f"R2 score is {r2}")
    print(f"RMSE score is {rmse}")
```

```python
from sklearn.tree import DecisionTreeRegressor
ModelEval(model=DecisionTreeRegressor())
```

```
MAE score is 1152.0296763338943
MSE score is 11922878.05991025
R2 score is 0.976682089372084
RMSE score is 3452.9520790057672
```

```python
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(random_state=0)
ModelEval(rf)
```

```
MAE score is 1071.7713624440203
MSE score is 7423047.340866772
R2 score is 0.985482535876709
RMSE score is 2724.5269939691866
```

```python
from sklearn.ensemble import AdaBoostRegressor
rf = AdaBoostRegressor(random_state=0)
ModelEval(rf)
```

```
MAE score is 3636.786046750951
MSE score is 33527783.060335092
R2 score is 0.9344287641771843
RMSE score is 5790.318044834419
```