Data Pre-Processing & Model Building

Date	20 November 2022
Team ID	PNT2022TMD52214
Project Name	Developing a Flight Delay Prediction using Machine Learning

In this milestone, we will be preprocessing the dataset that is collected. Preprocessing includes:

1. Processing the dataset.
2. Handling the null values.
3. Handling the categorical values if any.
4. Normalize the data if required.
5. Identify the dependent and independent variables.
6. Split the dataset into train and test sets.

Import Required Libraries:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib as mpl
import matplotlib.patches as patches
from matplotlib.patches import ConnectionPatch
from collections import OrderedDict
from matplotlib.gridspec import GridSpec
%matplotlib inline
```

Importing The Dataset:

```
df=pd.read_csv('Downloads/flightdata.csv')
df.head()
```

Analyze The Data:

```
# Before type casting of 'dep_time', 'dep_delay', 'arr_time', 'arr_delay'
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 327346 entries, 0 to 336769
Data columns (total 19 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   year            327346 non-null  int64  
 1   month           327346 non-null  int64  
 2   day             327346 non-null  int64  
 3   dep_time        327346 non-null  float64
 4   sched_dep_time  327346 non-null  int64  
 5   dep_delay       327346 non-null  float64
 6   arr_time        327346 non-null  float64
 7   sched_arr_time  327346 non-null  int64  
 8   arr_delay       327346 non-null  float64
 9   carrier         327346 non-null  object 
 10  flight          327346 non-null  int64  
 11  tailnum         327346 non-null  object 
 12  origin          327346 non-null  object 
 13  dest            327346 non-null  object 
 14  air_time        327346 non-null  float64
 15  distance        327346 non-null  int64  
 16  hour            327346 non-null  int64  
 17  minute          327346 non-null  int64  
 18  time_hour       327346 non-null  object 
dtypes: float64(5), int64(9), object(5)
memory usage: 49.9+ MB
```

Handling Missing Values:

```
# Now again checking whether the dataset till contains any NULL values
df.isnull().sum()
```

```
year              0
month             0
day               0
dep_time          0
sched_dep_time    0
dep_delay         0
arr_time          0
sched_arr_time    0
arr_delay         0
carrier           0
flight            0
tailnum           0
origin            0
dest              0
air_time          0
distance          0
hour              0
minute            0
time_hour         0
dtype: int64
```

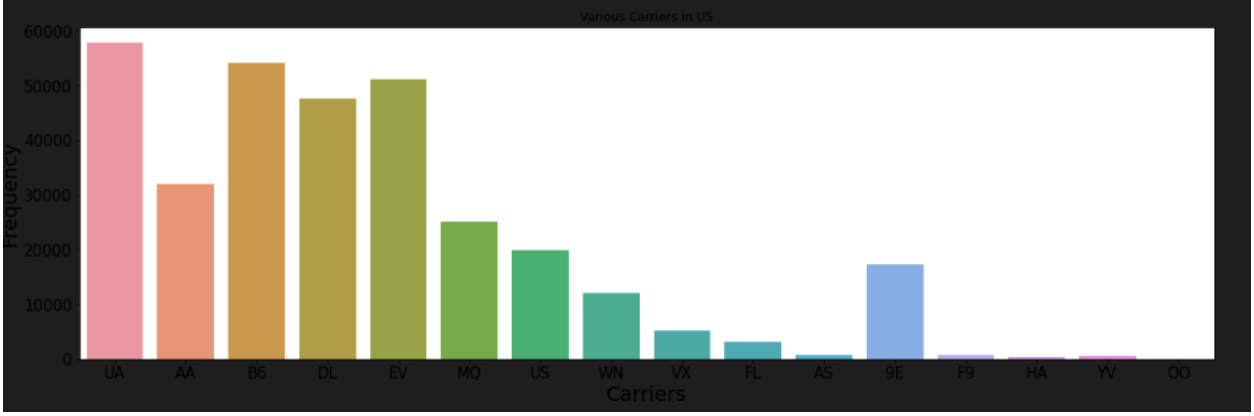
```
# Now checking whther the dataset contains the NULL va
df.isnull().sum()
```

```
year                 0
month                0
day                  0
dep_time          8255
sched_dep_time       0
dep_delay         8255
arr_time          8713
sched_arr_time       0
arr_delay         9430
carrier              0
flight               0
tailnum           2512
origin               0
dest                 0
air_time          9430
distance             0
hour                 0
minute               0
time_hour            0
dtype: int64
```

Data Visualization:

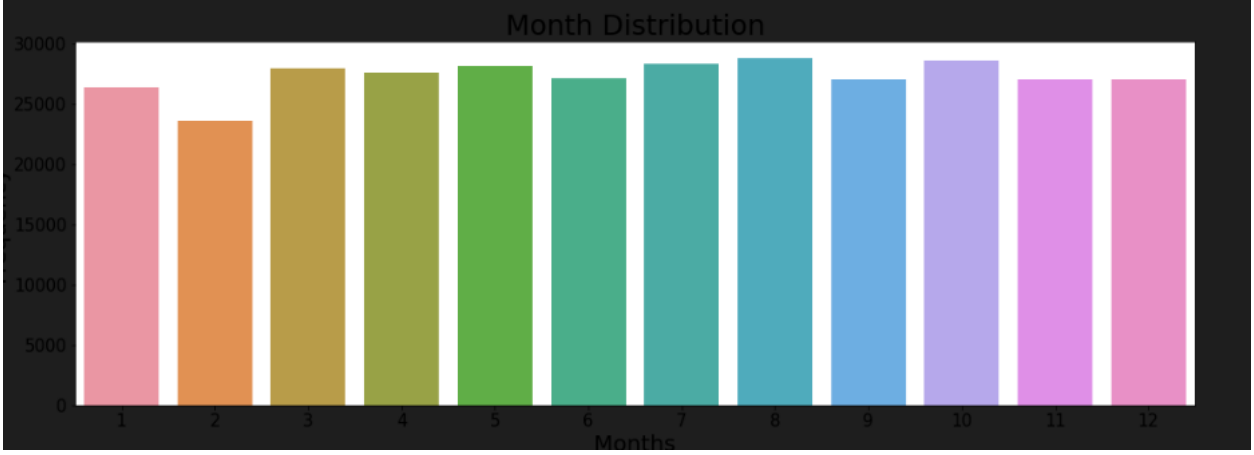
```
plt.title('Various Carriers in US')
plt.xticks(size = 15)
plt.yticks(size = 15)
plt.xlabel("Carriers", size = 20)
plt.ylabel("Frequency", size = 20)
plt.show()
```

```
opt/conda/envs/Python-3.9/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.
rgument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
 warnings.warn(
```




```
plt.figure(figsize = (18, 6))
sns.countplot(df['month'])
plt.title('Month Distribution', size = 25)
plt.xticks(size = 15)
plt.yticks(size = 15)
plt.xlabel("Months", size = 20)
plt.ylabel("Frequency", size = 20)
plt.show()
```

```
t/conda/envs/Python-3.9/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. Fr
gument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
arnings.warn(
```



```
sns.heatmap(df.corr(),annot=True)
```

```
<AxesSubplot:>
```

Dropping Un Necessary Columns:

```
df=df[["MONTH","DAY_OF_MONTH","DAY_OF_WEEK","ORIGIN","DEP_DEL15","CRS_ARR_TIME","ARR_DEL15"]]
df.isnull().sum()
```

```
MONTH             0
DAY_OF_MONTH      0
DAY_OF_WEEK       0
ORIGIN            0
DEP_DEL15       107
CRS_ARR_TIME      0
ARR_DEL15       188
dtype: int64
```

Splitting The Dataset into Dependent and Independent Variables:

```
# dependent variable

Y = df.iloc[:, -1]
print(Y)
```

```
0    d

1    b

2    a

3    c

4    f

5    e

Name: Age, dtype: category

Categories (6, object): ['a', 'b', 'c', 'd', 'e', 'f']
```

Split The Dataset Into Train Set And Test Set:

```
X = np.arange(1,25).reshape(12,2)
Y = np.array([0,1,1,0,1,0,0,1,1,0,1,0])
X
```

```
array([[ 1,  2],
       [ 3,  4],
       [ 5,  6],
       [ 7,  8],
       [ 9, 10],
       [11, 12],
       [13, 14],
       [15, 16],
       [17, 18],
       [19, 20],
       [21, 22],
       [23, 24]])
```

```
Y
```

```
array([0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0])
```

```
X_train,X_test,Y_train,y_test = train_test_split(X,Y)
X_train
```

```
array([[ 7,  8],
       [23, 24],
       [19, 20],
       [ 3,  4],
       [ 9, 10],
       [15, 16],
       [21, 22],
       [11, 12],
       [ 5,  6]])
```

Train And Test the Model Using Decision Tree Classifier:

Train the Model

```
from sklearn.mode_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0,3,random_state=0)
```

```
X_train.shape
```

```
(8, 2)
```

```
X_test.shape
```

```
(4, 2)
```

```
pred=model.predict(y_train)
```

Build the Model

```
from sklearn.linear_model import LinearRegression
```

```
model=LinearRegression() # initialzing the model
```

```
model.fit(X_train,Y_train) # fitting the model on training data
```

```
LinearRegression()
```

```
Y_test
```

```
array([0, 0, 1, 1])
```

```
pred_test= model.predict(X_test)
pred_test
```

```
array([0.5, 0.5, 0.5, 0.5])
```

```
pred_train = model.predict(X_train)
pred_train
```

```
array([0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5])
```

```
pred=model.predict(X_test)
```

```
pred
```

```
ray([0.5, 0.5, 0.5, 0.5])
```

```
# predict on random values

X_p= model.predict([[2:2]])
X_p
```

```
Input In [189]
```

Model Evaluation:

```
# accuracy score

from sklearn.metrics import accuracy_score,classification_report,confusion_matrix,roc_auc_score,roc_curve
```

```
# Confusion matrix

pd.crosstab(Y_test,pred_test)
```

col_0	0.5
row_0	
0	2
1	2

```
from sklearn import metrics
```

Saving The Model:

```
import pickle
pickle.dump(classifier,open("flight.pkl","wb"))
```