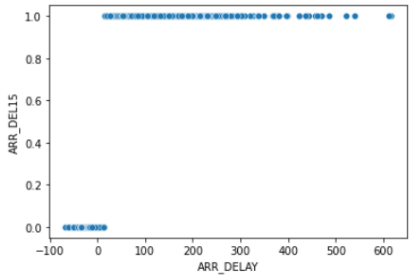
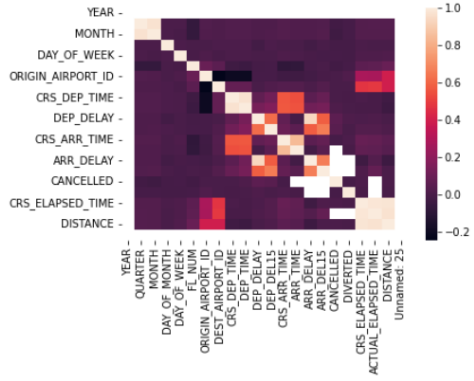


Project Development Phase Model Performance Test

Team ID	PNT2022TMID18451
Project Name	Developing a Flight Delay Prediction Model using Machine Learning
Maximum Marks	10 Marks

ML Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	Classification Model: Confusion Matrix, Accuracy Score & Classification Report	<pre>In [8]: sns.scatterplot(x='ARR_DELAY', y='ARR_DEL15', data=data)</pre> <pre>Out[8]:</pre>  <pre>In [9]: sns.heatmap(data.corr())</pre> <pre>Out[9]:</pre> 

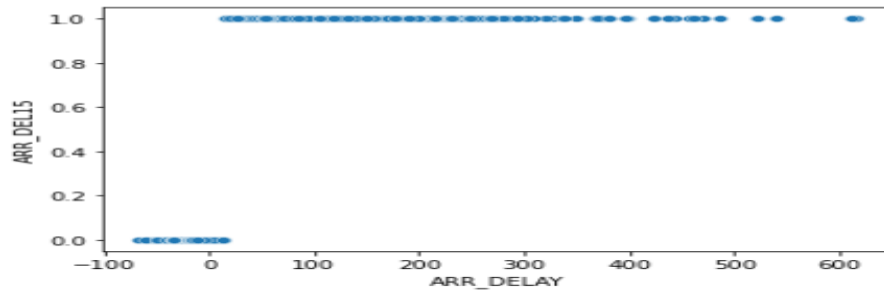
			<pre> In [34]: decisiontree = classifier.predict(x_test) In [47]: from sklearn.metrics import accuracy_score acc = accuracy_score(y_test,decisiontree) acc Out[47]: 1.0 In [48]: from sklearn.metrics import confusion_matrix cm = confusion_matrix(y_test,decisiontree) In [49]: cm Out[49]: array([[1938, 0], [0, 309]], dtype=int64) </pre>
2.	Tune the Model	Hyperparameter Tuning - StandardScaler	<pre> In [30]: from sklearn.preprocessing import StandardScaler sc = StandardScaler() x_train = sc.fit_transform(x_train) x_test = sc.transform(x_test) In [31]: from imblearn.over_sampling import SMOTE smote = SMOTE() In [32]: x_train_smote,y_train_smote = smote.fit_resample(x_train,y_train) In [33]: from sklearn.tree import DecisionTreeClassifier classifier = DecisionTreeClassifier(random_state=0) classifier.fit(x_train_smote,y_train_smote) Out[33]: DecisionTreeClassifier(random_state=0) In a Jupyter environment, please rerun this cell to show the HTML representation of the DecisionTreeClassifier object. On GitHub, the HTML representation is unable to render, please try loading the object into a notebook on your local machine. In [34]: decisiontree = classifier.predict(x_test) In [47]: from sklearn.metrics import accuracy_score acc = accuracy_score(y_test,decisiontree) acc Out[47]: 1.0 </pre>

BRIEF DETAILED SCREENSHOTS:

METRICS:

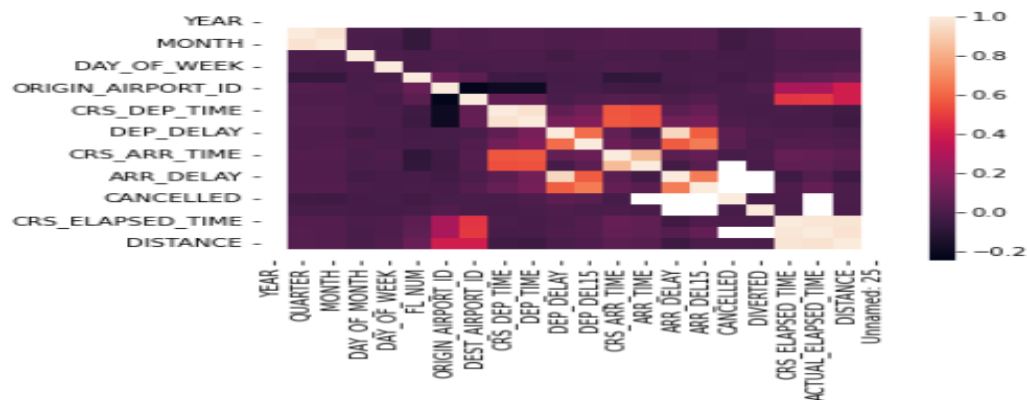
```
In [8]: sns.scatterplot(x='ARR_DELAY',y='ARR_DEL15',data=data)
```

Out[8]:



```
In [9]: sns.heatmap(data.corr())
```

Out[9]:



```
In [34]: decisiontree = classifier.predict(x_test)
```

```
In [47]: from sklearn.metrics import accuracy_score
acc = accuracy_score(y_test,decisiontree)
acc
```

Out[47]: 1.0

```
In [48]: from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,decisiontree)
```

```
In [49]: cm
```

```
Out[49]: array([[1938,    0],
               [    0,   309]], dtype=int64)
```

TUNE THE MODEL:

```
In [30]: from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

```
In [31]: from imblearn.over_sampling import SMOTE
smote = SMOTE()
```

```
In [32]: x_train_smote, y_train_smote = smote.fit_resample(x_train, y_train)
```

```
In [33]: from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(random_state=0)
classifier.fit(x_train_smote, y_train_smote)
```

```
Out[33]: DecisionTreeClassifier(random_state=0)
```

In a Jupyter environment, please rerun this cell to show the HTML representation of the output. On GitHub, the HTML representation is unable to render, please try loading the output in the notebook.

```
In [34]: decisiontree = classifier.predict(x_test)
```

```
In [47]: from sklearn.metrics import accuracy_score
acc = accuracy_score(y_test, decisiontree)
acc
```

```
Out[47]: 1.0
```