

# DEVELOPING A FLIGHT DELAY MODEL USING MACHINE LEARNING

**TEAM ID: PNT2022TMID27775**

## **Train And Test The Model Using Decision Tree Classifier.**

- There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have it may be Classification algorithms are Regression algorithms.

Example: 1. Linear Regression.

2. Logistic Regression.

3. Random Forest Regression / Classification.

4. Decision Tree Regression / Classification.

- You will need to train the datasets to run smoothly and see an incremental improvement in the prediction flight delay.
- Now we apply the Decision Tree Classifier algorithm on our dataset.
- A decision tree a tree like structure whereby an internal node represents an attribute, a branch represents a decision rule, and the leaf nodes represent an outcome. This works by splitting the data into separate partitions according to an attribute selection measure.
- Standard Scaler: Sklearn its main scaler, the StandardScaler, uses a strict definition of standardization to standardize data. It purely centers the data by using the following formula, where  $\mu$  is the mean and  $\sigma$  is the standard deviation.

Build the model with the Decision Tree Classifier.

- We're going to use `x_train` and `y_train` obtained above in `train_test_split` section to train our Decision Tree Classifier model. We're using the `fit` method and passing the parameters as shown below.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(random_state = 0)
classifier.fit(x_train,y_train)
```

```
DecisionTreeClassifier(random_state=0)
```

```
decisiontree = classifier.predict(x_test)
```

```
decisiontree
```

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

Predict the values

- Once the model is trained, it's ready to make predictions. We can use the predict method on the model and pass x\_test as a parameter to get the output as decisiontree.
- Notice that the prediction output is an array of real numbers corresponding to the input array.

## Predicting The Result

```
decisiontree = classifier.predict(x_test)
```

```
decisiontree
```

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