**ASSESMENT**

**IXIGO-RESEARCH ENGINEER JOB POSITION**

**NAME: PRIYA GARG**

The data given has the following features for training and testing data. The features that are common in both the files have been marked in green and the features uncommon are marked in Red.

|  |  |  |  |
| --- | --- | --- | --- |
| Cols in Train Data | Data Type | Cols in Test Data | Data Type |
| runDate | DateTime | runDate | DateTime |
| stations | String | stations | String |
| trainCode | Integer | trainId | DateTime |
| trainStationId | Integer | trainCode | Integer |
| scheduledArrival | DateTime | index | DateTime |
| scheduledDeparture | DateTime | trainStationId | Integer |
| actualArrival | DateTime | scheduledArrival | DateTime |
| actualDeparture | DateTime | scheduledDeparture | DateTime |
| distance | Integer | distance | Integer |
| dayCount | Integer | dayCount | Integer |
| ArrivalDelay | Integer | day | String |
| DepartureDelay | Integer |  |  |

Table 1: Variables in the Dataset

**Data Analysis and pre-processing:**

The data in its current form is not compatible to be fed into Machine Learning Models. Therefore, various data pre-processing steps needed to be taken care of.

1. **Converting Datetime:** The datetime format given is in the form of “%d-%m-%Y %h:%M:%S”. The date has been converted into 6 different columns accordingly. The Code in python is shown in code snippet 1.



CS 1: Pre-processing Code Snippet

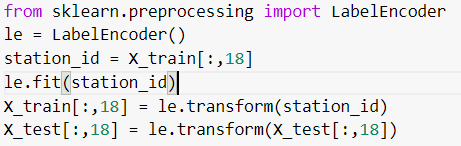
The final Columns that have been kept in the final Data are:

'Aday', 'Ahour', 'Aminute', 'Amonth', 'Asecond', 'Ayear', 'Dday', 'Dhour', 'Dminute', 'Dmonth', 'Dsecond', 'Dyear', 'day', 'dayCount', 'distance', 'runday', 'runmonth', 'runyear', 'stations', 'trainCode', 'trainStationId'

The column Day has been in test data so, in the training data the column “day” has been added.

1. **Converting String:**

The column Stations has a string data type and has been converted to datatype int using Label Encoding as shown in code snippet 2.

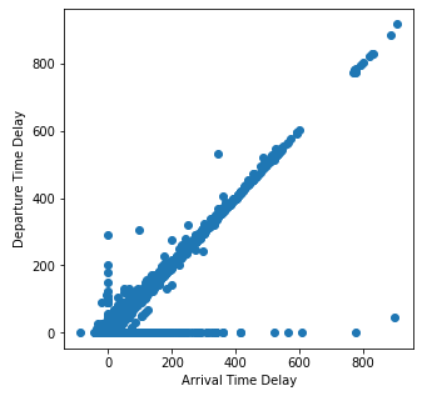


CS2 - Label Encoding

**Variables:**

Dependent Variables: ArrivalDelay and DepartureDelay

Independent Variables: 'Aday', 'Ahour', 'Aminute', 'Amonth', 'Asecond', 'Ayear', 'Dday', 'Dhour', 'Dminute', 'Dmonth', 'Dsecond', 'Dyear', 'day', 'dayCount', 'distance', 'runday', 'runmonth', 'runyear', 'stations', 'trainCode', 'trainStationId'



The relation between the dependent variables themselves i.e., arrival delay and departure delay is shown in Figure 1. The plot between arrival and departure delay indicates that for most of the samples, the difference between them is negligible and hence they are highly dependent on each other.

Therefore, in this study we focus on one variable as the output i.e., arrival delay and then use the predicted arrival delay for predicting departure delay.

Figure 1: Plot between arrivalDelay and departureDelay

To understand the range of variability of the dependent variables with their respective expected times are shown in Figure 2 and 3. It can be observed from the plots that the arrival delay is much more significant from 0 to 200 minutes and similar is the case with the departure delay. Also the plots indicate that there is a dependency of arrival and departure delay with the other factors as no clear result can be derived between the expected time and the delay.

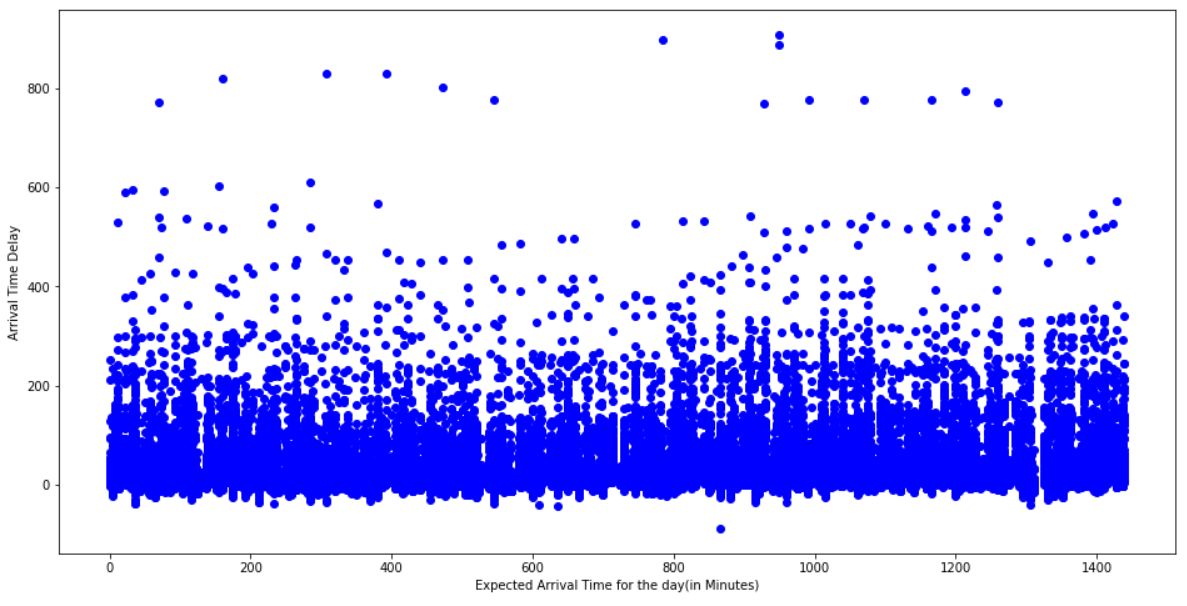


Figure 2: Plot between expected arrival time and arrival delay

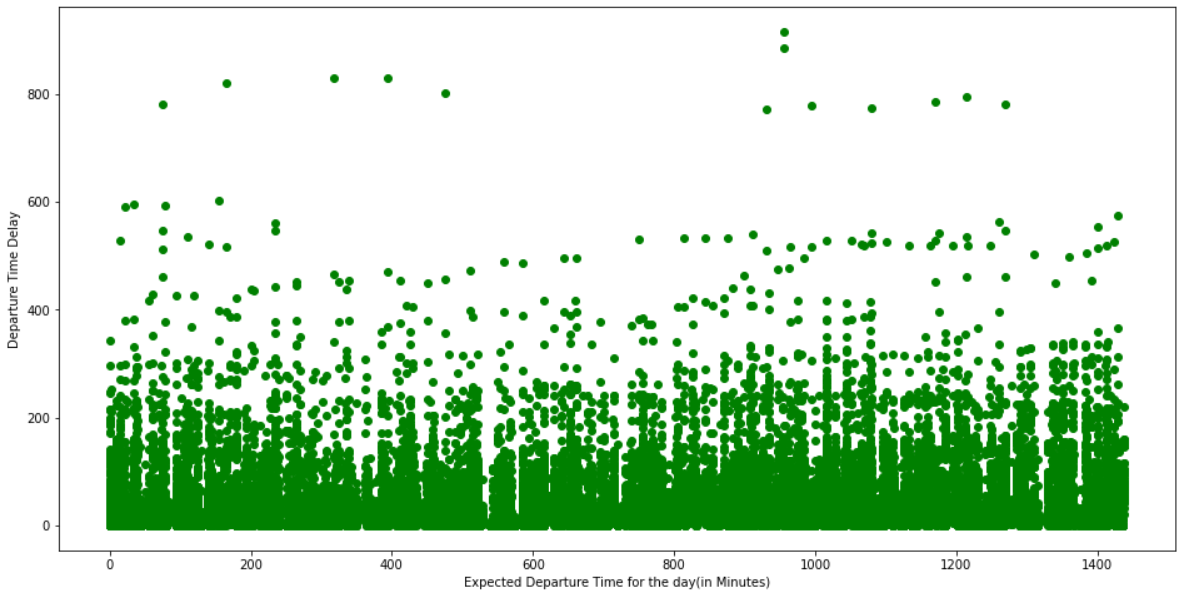


Figure 3: Plot between expected departure time and departure delay

**Machine Learning Techniques Applied:**

Various Machine Learning Models have already been applied for this research problem. Though various techniques have been applied, a clear view on the best techniques out of various learning techniques has not been given. In this report, various standalone learning techniques have been applied which are based on different learning strategies like rule-based learning, cost-minimization strategies, nearest-neighbour learning etc.

Standalone Machine Learning Algorithms that will be tested in this report:

1. Linear Regression
2. Ridge Regression
3. Polynomial Regression
4. Support Vector Regression
5. K-Nearest Neighbor
6. Decision Trees
7. Neural Network = MLP

Ensemble Machine Learning Algorithms tested in this report:

1. AdaBoost
2. Random Forest
3. Bagging
4. ExtraTree Regression

**Performance Measure:**

Since, this is a regression problem, the performance measures that have been taken in this study are:

1. Mean square Error (MSE):  MSE measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value.
2. R2 Score: In statistics, the coefficient of determination, denoted R² or r² and pronounced "R squared", is the proportion of the variance in the dependent variable that is predictable from the independent variable.

**Validation Technique:**

Hold-out validation technique or train and test split technique has been used in this study for the results estimation and validation. A test-train split of 0.2 is taken where 20% of samples are hold out for testing and 80% for training.

**Parameter Tuning:**

1- Decision Trees

After a deep analysis, it was seen that max\_depth parameter was the most effective parameters that was affecting the results.

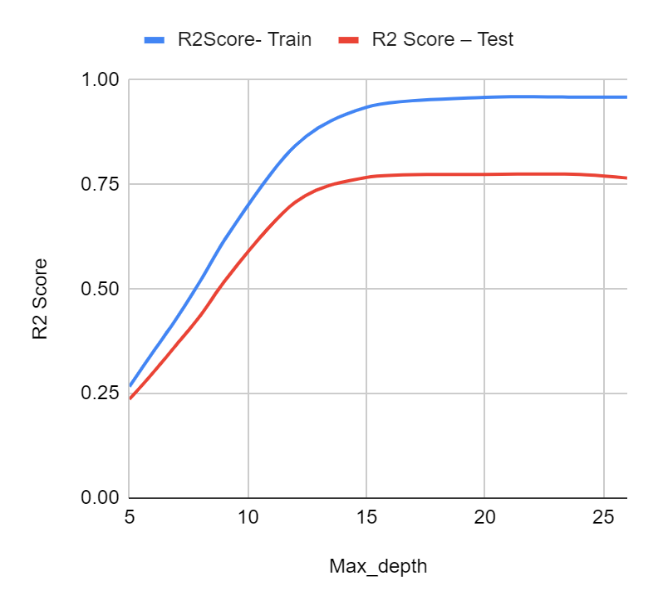
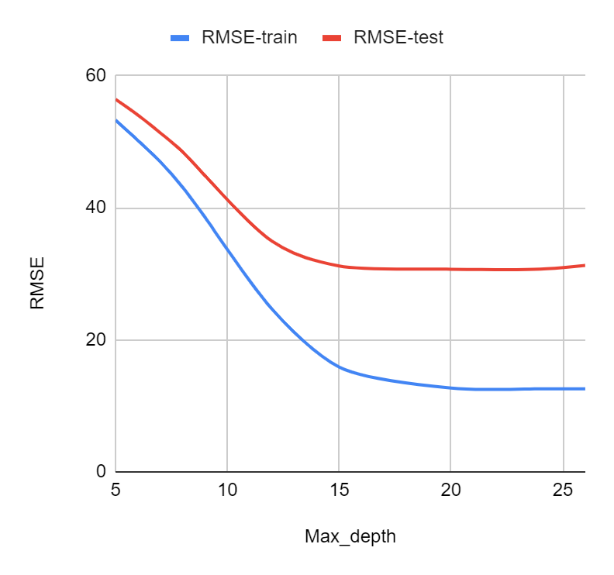


Figure 4: RMSE and R2-Score for various parameter Settings

2-Polynomial Regression

The degree of polynomial is the only effective parameter, The results are shown in Table 2.

Table 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Degree | Number of Features | Time Taken | Score | Mean Square Error |
| 1 | 20 | 0.531 | 0.09501 | 61.62017 |
| 2 | 231 | 1.210 | 0.1646 | 59.24233 |
| 3 | 2024 | 3.234 | **0.1943** | 53.09446 |

3- Multi-Layer Perceptron

The parameter mostly afafceting the results are number of hidden layers and number of hidden layer neurons. Table 3 indicates the RMSE and R2-Score for hidden layer and neurons.

Table 3: Variation of R2 score and RMSE for training and testing data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of hidden layers | Number of neurons in hidden layer | R2Score- Train | R2Score – Test | RMSE - Train | RMSE - Test |
| 1 | 200 | -3.32783e-06 | -9.57886e-05 | 62.59377 | 64.77728 |
| 1 | 300 | -3.35943e-06 | -3.91021e-05 | 62.59377 | 64.77544 |
| 3 | 300 | -1.14454e-05 | -0.00012 | 62.59402 | 64.77831 |
| 3 | 200 | -4.03152e-05 | -3.56704e-06 | 62.59493 | 64.77429 |
| 7 | 200 | -8.82752e-05 | -0.00029 | 62.5964 | 64.78365 |

The higher results of decision trees which are based on rule-based learning motivated the further use of Dtree based ensembles for the given problem statement. The parameter tuning results of various ensembles are shown below.

1-Random Forest

After a deep analysis, it was seen that number of base classifiers parameter was the most effective parameters that was affecting the results.

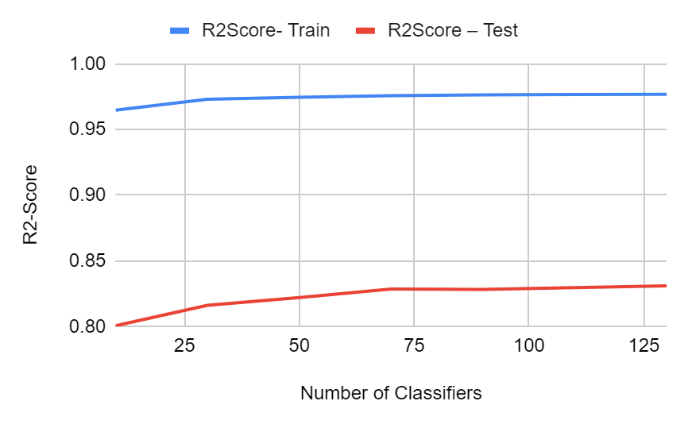
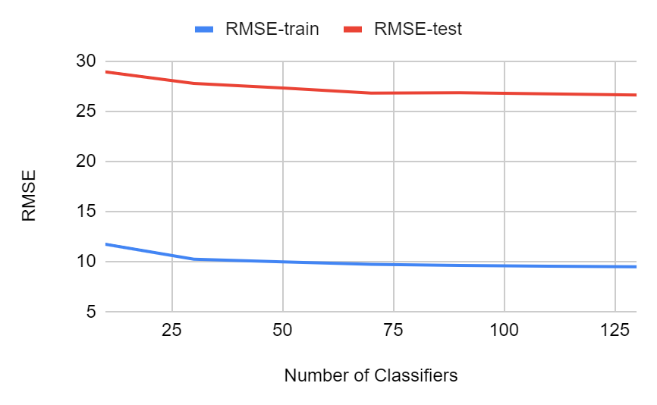


Figure 5: RMSE and R2-Score for various parameter Settings

2-Bagging

Two parameters that were really changing the results were – Decision Trees, SVM, and Random Forest (which is itself an ensemble technique). The results are shown in Table 4.

Table 4: RMSE and R2-Score variation with classification technique, maximum samples and maximum features

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Classification Technique** | **Max\_samples** | **Max\_features** | **RMSE-train** | **R2Score- Train** | **RMSE-test** | **R2Score – Test** |
| DTree | 0.5 | 0.5 | 27.16349 | 0.81167 | 36.94478 | 0.67468 |
| **DTree** | **0.7** | **0.7** | **20.60465** | **0.89163** | **31.27434** | **0.76688** |
| DTree | 0.8 | 0.8 | 19.263 | 0.90529 | 31.86283 | 0.75802 |
| SVR | 0.5 | 0.5 | 63.4871 | -0.02875 | 65.724 | -0.02954 |
| SVR | 0.7 | 0.7 | 66.85092 | -0.14065 | 69.07565 | -0.13722 |
| RF | 0.7 | 0.7 | 23.74243 | 0.85612 | 36.50683 | 0.68235 |
| RF | 0.5 | 0.5 | 31.78612 | 0.74212 | 42.81142 | 0.56316 |
| RF | 0.8 | 0.8 | 24.66015 | 0.84478 | 37.73764 | 0.66057 |
| RF | 0.8 | 0.8 | 24.51836 | 0.84656 | 39.60895 | 0.62607 |

3-AdaBoost

Parameter n\_estimators was affecting the results i.e., why the variation of RMSE and R2 score for test and train data are shown in figure 6.

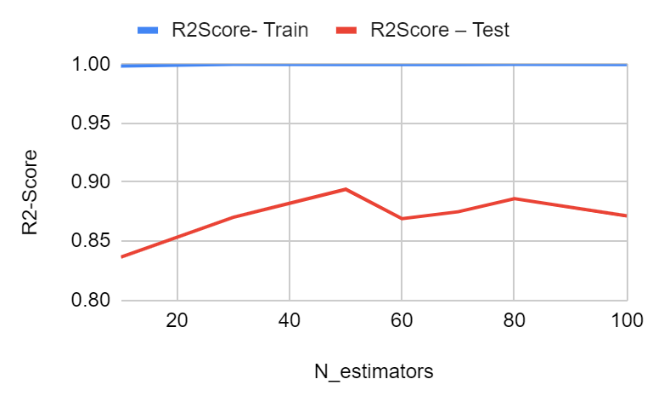
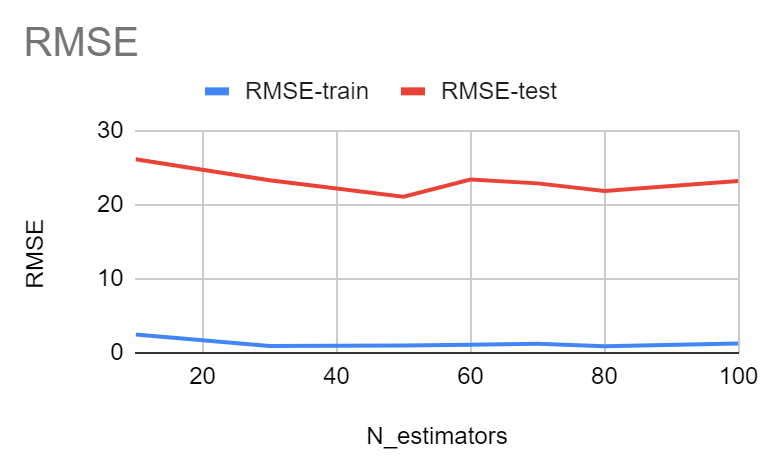


Figure 6: RMSE and R2-Score for various parameter Settings

**Results:**

For each technique, a range of combinations have been compared to get the best-case scenario for each ML technique. The results of each of the technique has been given in Table 5. Adaboost with Dtree as its base classifier has shown the highest results hence it is chosen to predict the arrival delay for the trains. After predicting the arrival delay, the model is trained again on the dataset with arrival delay as the independent variable and departure delay is calculated.

Table 5: Final Results for various techniques

|  |  |  |
| --- | --- | --- |
| **ML Technique** | **Mean Square Error** | **R2 Score** |
| Linear Regression | 63.51202 | 0.03859 |
| Ridge Regression | 63.49802 | 0.03902 |
| Polynomial Regression | 53.09446 | 0.1943 |
| Decision Trees | 30.70165 | 0.77419 |
| Support Vector Machine | 87.76743 | -0.83596 |
| Multi-Layer Perceptron | 64.77429 | -3.56704e-06 |
| K-Nearest Neighbors | 57.18634 | 0.22056 |
| Random Forest | 26.62538 | 0.83104 |
| AdaBoost | **21.86481** | **0.88605** |
| Extra Trees Regression | 28.55826 | 0.80562 |
| Bagging | 31.27435 | 0.76688 |