

Machine Learning
Assignment 1
Flight Delay Forecasting

Walid Khaled Hussein Shaker

Motivation:

In this report, flight delay forecasting has been implemented using different machine learning models such as linear regression, polynomial regression, and finally regularizing based on lasso regression. Comparison between these three models have been conducted and train and test error have been calculated to evaluate each model performance using different metrics such as mean absolute error, mean squared error MSE, and coefficient of determination. In addition, data pre-processing has been deployed before training the model. Pre-processing stage included splitting the data into train and test sets based on the constrain provided, hence resetting index to overcome the index problem arose. One hot encoding is applied as well to encode the categorical features of the data into numerical representation. Moreover, missing values check function is created to check whether the data requires an imputation or not. After that, MinMaxScaler is applied as a feature scaling for data better uniformly distribution. Data visualization is performed by plotting flight duration as an independent variable/predictor versus flight delay as a dependent variable which is the target for our models' predictions. Finally, outliers have been removed using Local Outlier Factor to increase model accuracy.

Task Definition and Data Description:

In this task dataset is provided from a company that analyses flights delay. The data was collected over a period of 4 years. It contains 4 predictors and 1 target. The predictors are Departure Airport, Scheduled departure time, Destination Airport, and Scheduled arrival time, while the target is Flight Delay (in minutes). The task is to build different machine learning models for flight delay predictions.

Data Reading and Exploration:

Data is imported using Pandas and it explored as follows:

Columns of train data:

```
Index(['Depature Airport', 'Scheduled depature time', 'Destination Airport',  
      'Scheduled arrival time', 'Delay'],  
      dtype='object')
```

Size of train data:

```
(675513, 5)
```

Train data Display:

	Depature Airport	Scheduled depature time	Destination Airport	Scheduled arrival time	Delay
0	SVO	2015-10-27 07:40:00	HAV	2015-10-27 20:45:00	0.0
1	SVO	2015-10-27 09:50:00	JFK	2015-10-27 20:35:00	2.0
2	SVO	2015-10-27 10:45:00	MIA	2015-10-27 23:35:00	0.0
3	SVO	2015-10-27 12:30:00	LAX	2015-10-28 01:20:00	0.0
4	OTP	2015-10-27 14:15:00	SVO	2015-10-27 16:40:00	9.0

Calculating Flight Duration Column:

Flight duration is calculated in minutes through subtracting scheduled departure time from scheduled arrival time after converting them into datetime.

	Depature Airport	Scheduled_depature_time	Destination Airport	Scheduled arrival time	Delay	Flight Duration
0	SVO	2015-10-27 07:40:00	HAV	2015-10-27 20:45:00	0.0	785.0
1	SVO	2015-10-27 09:50:00	JFK	2015-10-27 20:35:00	2.0	645.0
2	SVO	2015-10-27 10:45:00	MIA	2015-10-27 23:35:00	0.0	770.0
3	SVO	2015-10-27 12:30:00	LAX	2015-10-28 01:20:00	0.0	770.0
4	OTP	2015-10-27 14:15:00	SVO	2015-10-27 16:40:00	9.0	145.0

Statistics about data, count, unique, top, and frequency:

	Delay	Flight Duration
count	675513.000000	675513.000000
mean	9.912939	196.035351
std	44.895875	121.853260
min	0.000000	45.000000
25%	0.000000	120.000000
50%	0.000000	160.000000
75%	5.000000	225.000000
max	1436.000000	1590.000000

Trainset splitting and Reset Index:

The data is split based on Scheduled departure time. The train data is all the data from year 2015 till 2017. The test data is all the data samples collected in year 2018.

However, when splitting data this way, an index problem arises as illustrated below:

Having a look on last 7 elements of the train set, we find that all data until 499059 are index and ordered, after that we have 3 data points with different order

	Depature Airport	Scheduled_depature_time	Destination Airport	Scheduled arrival time	Delay	Flight Duration
499055	SVO	2017-12-31 22:15:00	IKA	2018-01-01 02:05:00	0.0	230.0
499056	HAV	2017-12-31 22:40:00	SVO	2018-01-01 10:00:00	0.0	680.0
499057	SVO	2017-12-31 22:50:00	MLE	2018-01-01 07:35:00	0.0	525.0
499058	JFK	2017-12-31 22:20:00	SVO	2018-01-01 07:10:00	213.0	530.0
499164	SVO	2017-12-31 13:50:00	VOZ	2017-12-31 15:05:00	1242.0	75.0
499168	SVO	2017-12-31 15:10:00	EGO	2017-12-31 16:35:00	1142.0	85.0
499535	OVB	2017-12-31 14:30:00	SVO	2017-12-31 19:00:00	478.0	270.0

On the test set, having a look on last 5 elements, it does not start with zero index, and this affects the size of test set on the following steps.

	Depature Airport	Scheduled_depature_time	Destination Airport	Scheduled arrival time	Delay	Flight Duration
499059	ATH	2018-01-01 01:20:00	SVO	2018-01-01 05:30:00	0.0	250.0
499060	LHR	2018-01-01 01:30:00	SVO	2018-01-01 05:05:00	0.0	215.0
499061	DXB	2018-01-01 01:35:00	SVO	2018-01-01 07:15:00	0.0	340.0
499062	TLV	2018-01-01 02:00:00	SVO	2018-01-01 06:10:00	1.0	250.0
499063	BEY	2018-01-01 02:05:00	SVO	2018-01-01 06:00:00	0.0	235.0

To overcome this problem, a resetting index function is created and applied for train and test set after splitting.

Tail of train set after index reset:

	Depature Airport	Scheduled_depature_time	Destination Airport	Scheduled arrival time	Delay	Flight Duration
499055	SVO	2017-12-31 22:15:00	IKA	2018-01-01 02:05:00	0.0	230.0
499056	HAV	2017-12-31 22:40:00	SVO	2018-01-01 10:00:00	0.0	680.0
499057	SVO	2017-12-31 22:50:00	MLE	2018-01-01 07:35:00	0.0	525.0
499058	JFK	2017-12-31 22:20:00	SVO	2018-01-01 07:10:00	213.0	530.0
499059	SVO	2017-12-31 13:50:00	VOZ	2017-12-31 15:05:00	1242.0	75.0
499060	SVO	2017-12-31 15:10:00	EGO	2017-12-31 16:35:00	1142.0	85.0
499061	OVB	2017-12-31 14:30:00	SVO	2017-12-31 19:00:00	478.0	270.0

Head of test set after index reset:

	Depature Airport	Scheduled_depature_time	Destination Airport	Scheduled arrival time	Delay	Flight Duration
0	ATH	2018-01-01 01:20:00	SVO	2018-01-01 05:30:00	0.0	250.0
1	LHR	2018-01-01 01:30:00	SVO	2018-01-01 05:05:00	0.0	215.0
2	DXB	2018-01-01 01:35:00	SVO	2018-01-01 07:15:00	0.0	340.0
3	TLV	2018-01-01 02:00:00	SVO	2018-01-01 06:10:00	1.0	250.0
4	BEY	2018-01-01 02:05:00	SVO	2018-01-01 06:00:00	0.0	235.0

Splitting train set to x_train and y_train and test set into x_test and y_test:

For x_train and x_test, Scheduled departure time, and Scheduled arrival time are dropped as they are no longer needed after calculating flight duration. Delay will be assigned for y_train and y_test.

```
x_train.tail(5)
```

	Depature Airport	Destination Airport	Flight Duration
499057	SVO	MLE	525.0
499058	JFK	SVO	530.0
499059	SVO	VOZ	75.0
499060	SVO	EGO	85.0
499061	OVB	SVO	270.0

```
x_test.head(5)
```

	Depature Airport	Destination Airport	Flight Duration
0	ATH	SVO	250.0
1	LHR	SVO	215.0
2	DXB	SVO	340.0
3	TLV	SVO	250.0
4	BEY	SVO	235.0

Categorical Features Encoding:

Check types of x_train, x_test

```
Number categorical features: 2
Depature Airport           object
Destination Airport        object
Flight Duration            float64
dtype: object
```

Encoding:

Seeing that 2 categorical features exist, so they need to be converted into numerical values. One Hot Encoding is applied and result of data size is shown below.

It is noticeable that about 334 columns are added to original data size.

Encoding Categorical Features

```
before encoding: x_train size (499062, 3) and x_test size (176451, 3)
after encoding: x_train size (499062, 337) and x_test size (176451, 337)
```

Data Imputation:

A function is created to check if there are missing values by counting the NAN values in the data after being encoded. The function output is attached below:

```
Number of missing values in x_train, x_test: 0 , 0
there is no need for imputation
```

Feature Scaling:

MinMaxScaler is applied to transform all data to be located in range [0,1] using the following formula:

$$x_i = \frac{x_i - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}$$

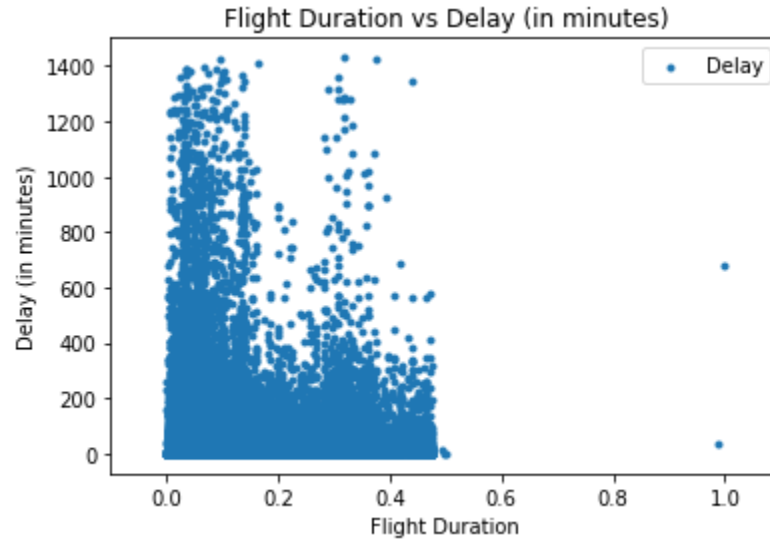
After scaling, types of x_train, and x_test changed from data frame to numpy array:

```
type(x_train), type(y_train), type(x_test), type(y_test)

(numpy.ndarray,
 pandas.core.series.Series,
 numpy.ndarray,
 pandas.core.series.Series)
```

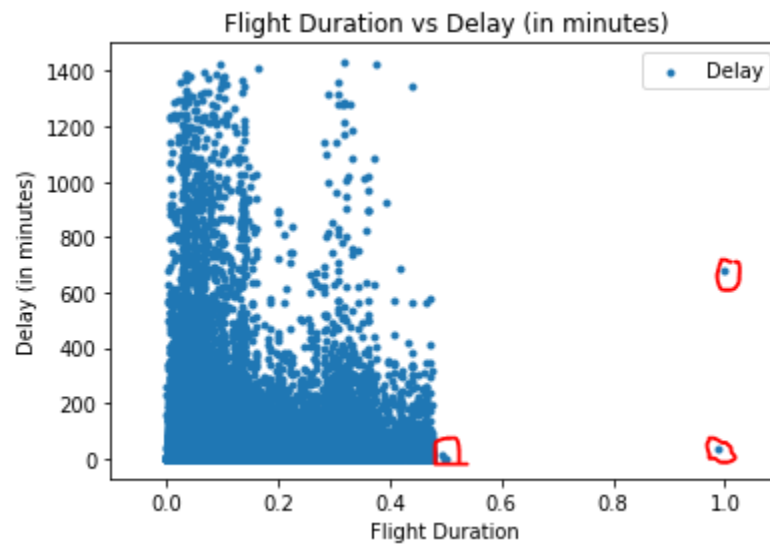
Data Visualization:

Flight duration, which is the time difference between departure and arrival, has been selected to be plotted against the flight delay.



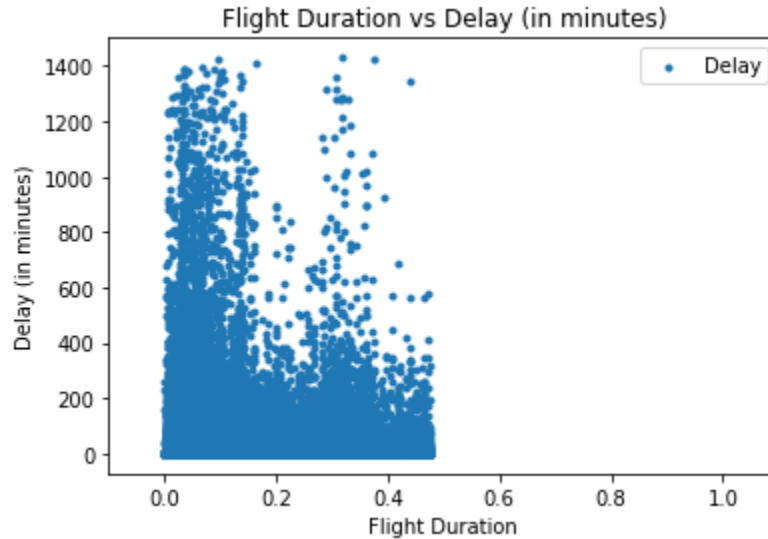
Outlier Detection & Removal:

Outlier is defined as an observation point that is distant from other observations. Having a look on data visualization section, we observe that there are outliers can affect model accuracy, so it is recommended to detect outliers and either to fix or remove them.



However, outlier detection cannot rely on observations, it has to be performed via automatic method. A simple approach to detect outliers is to locate those samples that are far from the other samples in the feature space. The local outlier factor, LOF is used to implement that approach. It is a technique that attempts to harness the idea of k-nearest neighbors (KNN) for outlier detection. It works as marking each row in the training dataset as normal (1) or an outlier (-1).

After Outlier Detection and Removal:



By counting the outliers detected and removed, it is noted that 26 data point have been removed. The output is following:

Outlier Detection & Removal

```
Before OutLier Removal : x_train size (499062, 1) and y_train size (499062,)
After OutLier Removal : x_train size (499036, 1) and y_train size (499036,)
26 data point have been identified as outliers
```

Before training our models:

An assumption has been made such that x_train will be flight duration as departure and arrival airport do not influence the target that much. So, in the three models presented below flight duration feature is considered as x_train set.

Linear Regression Model:

In this regression task, we will predict the flight delay in minutes based upon flight duration.

Simple Linear Regression Equation

$$y = \beta_0 + \beta_1 x_1$$

First, obtain the slope and the intercept as:

```
LR Model intercept : 8.804765154662833
LR Model coefficient : [25.13030736]
```

Second, apply the Linear Regression and show the prediction vs actual values:

	Actual	Predicted
0	0.0	11.997215
1	0.0	11.424211
2	0.0	13.470653
3	1.0	11.997215
4	0.0	11.751642
...
176446	0.0	10.196346
176447	0.0	9.214054
176448	0.0	9.295911
176449	0.0	11.178638
176450	379.0	13.470653

176451 rows × 2 columns

Third, calculate the train and the test errors to evaluate the model using some metrics as shown below:

```
Train error
Mean Absolute Error: 15.41618604929478
Mean Squared Error: 2150.5407163333775
Root Mean Squared Error: 46.37392280510004
Coefficient of Determination R score: 0.0018828529642611613

Test error
Mean Absolute Error: 14.400233690195993
Mean Squared Error: 1619.1917417258996
Root Mean Squared Error: 40.239181673164026
Coefficient of Determination R score: -0.010244818595628313
```

Does the model overfit/Underfit?

Compare the train error corresponding to the test error depicted above, we can conclude that the model exhibits underfitting as training error and test error are very high and they are close to each other. In a nutshell, this model is too simple for the task.

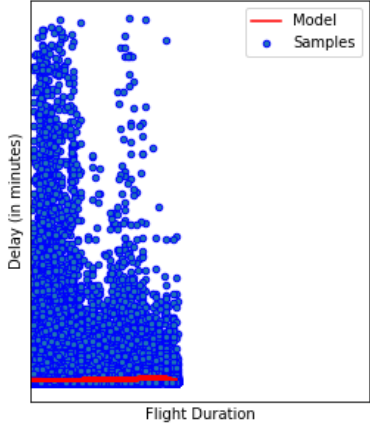
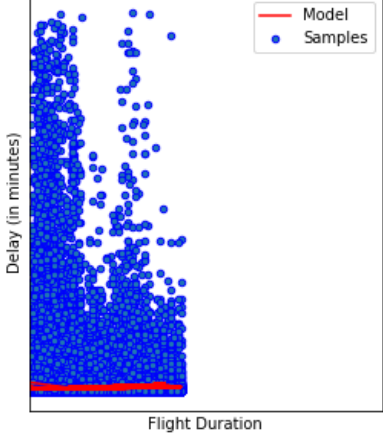
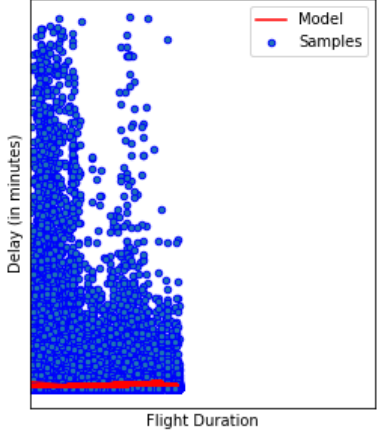
Polynomial Regression Model:

In this model, higher order function is estimated for the model because of non-linearity of the dataset. Different degrees are tested as shown:

```
degrees = [10,15,20]
```

The models are evaluated using cross validation mean squared error MSE and standard deviation std. Also, metrics such as absolute error and coefficient of determination have been conducted.

In the table below, comparison between the results of different higher order degree models will be presented.

	Degree 10	Degree 15	Degree 20
Cross Validation MSE and std	MSE:2150.9903507333256 std:481.11865416382335	MSE: 2150.830795357931 std: 481.19326370628335	MSE: 2150.820728504999 std: 481.2274391683091
Train Error			
Mean Absolute Error	15.42874919773324	15.42137402970501	15.420550986974623
Mean Squared Error	2148.2128344049916	2147.9128846665662	2147.8859576280897
Root Mean Squared Error	46.34881696877485	46.34558106946731	46.345290565796326
R score	0.002963278389992219	0.003102492204870244	0.0031149896845125147
Test Error			
Mean Absolute Error	14.401240910311982	14.39490296068346	14.391420893263648
Mean Squared Error	1618.2794325493533	1618.1770388531625	1618.1248522742596
Root Mean Squared Error	40.22784399578671	40.22657130371867	40.22592264043498
R score	-0.009675611382663485	-0.009611725989496112	-0.009579165780857268
	<p>Degree 10 MSE = 2.15e+03(+/- 4.81e+02)</p> 	<p>Degree 15 MSE = 2.15e+03(+/- 4.81e+02)</p> 	<p>Degree 20 MSE = 2.15e+03(+/- 4.81e+02)</p> 

Does the model overfit/Underfit?

Compare the train error corresponding to the test error depicted above, we can conclude that the model exhibits underfitting as training error and test error are very high and they are close to each other. However, the train our data with higher order polynomial function, the results are not satisfying yet, because there is incomplete information for the prediction. The flight delay cannot depend only on flight duration. That is why increasing the complexity for this polynomial regression for only one predictor (flight duration), will not have an impact to the prediction.

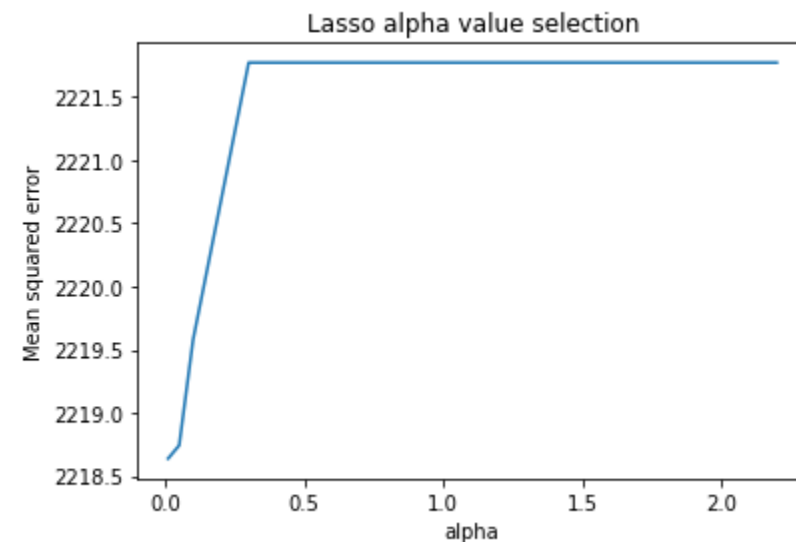
Regularization Using Lasso:

First splitting x_{train} to x_{train} and x_{val} . The idea of having a validation set is that validation set is used to tune hyperparameters.

For the hyperparameter alpha :

```
alphas = [2.2, 2, 1.5, 1.1, 1, 0.3, 0.1, 0.05, 0.01]
#random variables to select the one with the min error.
```

Calculating mean squared error for each alpha and plotting them to get the best alpha that has a minimum MSE.



Best value of alpha: 0.01

Calculate the train and the test errors to evaluate the model using some metrics as shown below:

```
Train error
Mean Absolute Error: 15.390032406530523
Mean Squared Error: 2142.9852078354625
Root Mean Squared Error: 46.29238822782275
Coefficient of Determination R score: 0.0019321771988336511
Test error
Mean Absolute Error: 14.393394770180794
Mean Squared Error: 1619.1305471497467
Root Mean Squared Error: 40.23842128053419
Coefficient of Determination R score: -0.010206638124537815
```

Does the model overfit/Underfit?

Compare the train error corresponding to the test error depicted above, we can conclude that the model exhibits underfitting as training error and test error are very high and they are close to each other. In a nutshell, this model is too simple for the task.

Comparison between 3 models:

Linear Regression Model	Polynomial Regression Model	Regularization Lasso
Underfit	Underfit	Underfit

Explanation:

We obtained a bad results as the data are incomplete. Flight delay as a target cannot rely on flight duration, there are many predictors have to be taken into consideration when looking for a flight delay. Adverse weather conditions are the most common predictor that can cause delay. There are some popular predictors such as Air Traffic Control (ATC) restrictions, waiting for connecting passengers, baggage loading, bird strikes, and late arrival of the aircraft (the familiar knock-on effect). So, disregarding all of these features and build the model upon flight duration only will result in underfitting as obtained above.

GitHub link:

<https://github.com/Walid-khaled/Walid-khaled-Machine-Learning-AI>