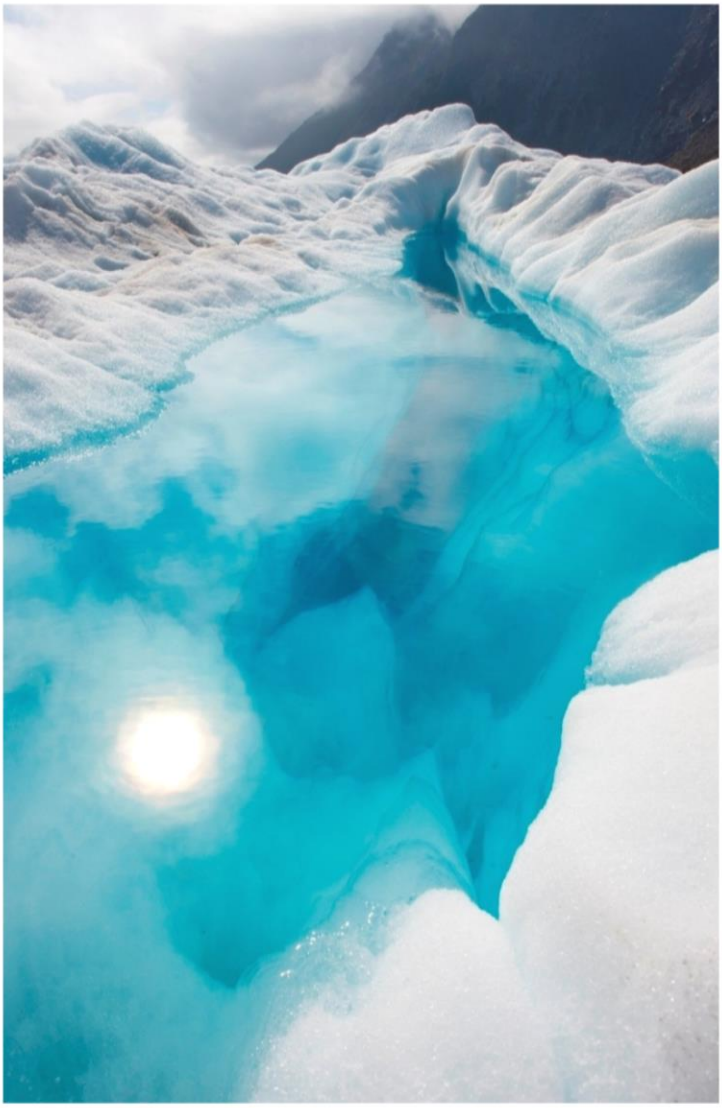


Cab fare prediction

Ajay Darshan | Data scientist course | March 16, 2019

Chapter 1 Introduction

PROBLEM STATEMENT

The objective of this project is to analyze historical data on cab fares and predict it. For this purpose, we will design a system that predicts the fare amount for a cab ride in the city.

DATA

Sample data is shown in below figure

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
0	4.5	2009-06-15 17:26:21 UTC	-73.844311	40.721319	-73.841610	40.712278	1.0
1	16.9	2010-01-05 16:52:16 UTC	-74.016048	40.711303	-73.979268	40.782004	1.0
2	5.7	2011-08-18 00:35:00 UTC	-73.982738	40.761270	-73.991242	40.750562	2.0
3	7.7	2012-04-21 04:30:42 UTC	-73.987130	40.733143	-73.991567	40.758092	1.0
4	5.3	2010-03-09 07:51:00 UTC	-73.968095	40.768008	-73.956655	40.783762	1.0

This data contains date and time of pickup, number of passengers and pickup and drop coordinates of latitude and longitude as independent variables. Using these variables fare amount is determined. As this involves prediction of fare amount, this is a regression problem.

Chapter 2 Methodology

PRE-PROCESSING

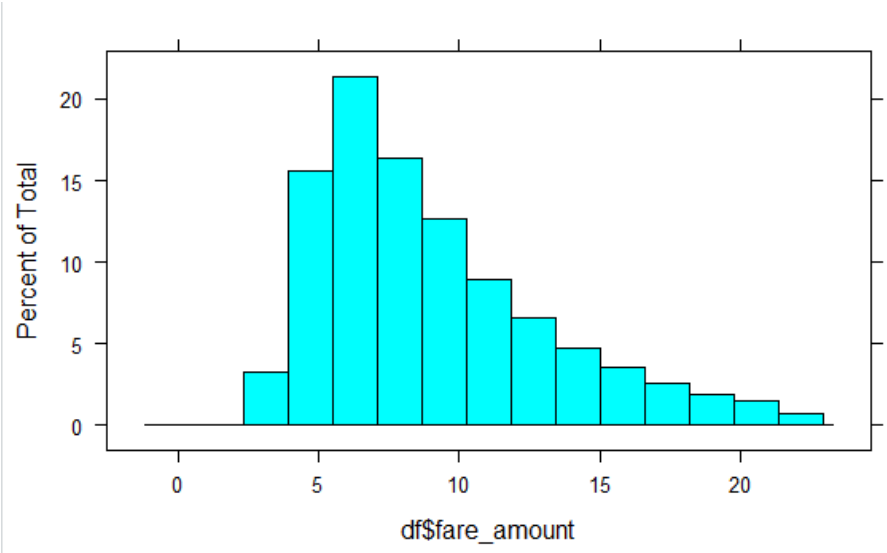
The first step in predictive analysis is pre-processing our data. This step is important as the well-prepared data gives better prediction on simpler model than bad data with complex model. It involves exploring data, cleaning data and visualizing it.

As a first step, all variables are converted into proper data types. Summary of numeric variables are shown in below figure.

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
count	16042.000000	16067.000000	16067.000000	16067.000000	16067.000000	16012.000000
mean	15.015004	-72.462787	39.914725	-72.462328	39.897906	2.625070
std	430.460945	10.578384	6.826587	10.575062	6.187087	60.844122
min	-3.000000	-74.438233	-74.006893	-74.429332	-74.006377	0.000000
25%	6.000000	-73.992156	40.734927	-73.991182	40.734651	1.000000
50%	8.500000	-73.981698	40.752603	-73.980172	40.753567	1.000000
75%	12.500000	-73.966838	40.767381	-73.963643	40.768013	2.000000
max	54343.000000	40.766125	401.083332	40.802437	41.366138	5345.000000

As highlighted in above figure, we can see that fare amount is having negative values and maximum value is too high. Also, passenger count of zero and very high value doesn't make sense. So as part of data cleaning we will remove rows having fare amount negative and passenger count having zero, fraction value and greater than 10. Here maximum passenger count of 10 is selected as maximum seating capacity of cabs which runs daily is 10-seater.

Below figure shows distribution of fare amount after cleaning data.



After this pickup hour, month, weekday and year are derived from pickup_datetime variable. This is a part of feature engineering.

MISSING VALUE ANALYSIS

Next step in pre-processing is missing value analysis. Below figure shows number of missing values in data.

```
fare_amount          24
pickup_datetime       0
pickup_longitude      0
pickup_latitude       0
dropoff_longitude     0
dropoff_latitude      0
passenger_count      55
```

Here two variables have missing values fare_amount and passenger_count. Two different methods are employed for these two variables.

For passenger_count, we cannot impute this value based on other independent variable, as these values cannot be influenced by other variables in real world. So, these rows having values missing are removed.

Below figure shows summary after performing above step.

	Variables	Missing_values
0	fare_amount	24
1	pickup_longitude	0
2	pickup_latitude	0
3	dropoff_longitude	0
4	dropoff_latitude	0
5	passenger_count	0
6	pickup_hour	0
7	pickup_month	0
8	pickup_weekday	0
9	pickup_year	0

For fare_amount, this value depends on other independent values. So, it is imputed using KNN imputation having k = 2.

FEATURE ENGINEERING

In this part, total distance travelled is derived using pickup and drop co-ordinates of latitude and longitude. Distance is calculated using haversine method. Using distance, it is easy to explain

relationship b/w distance and fare amount. Below figure shows, with increase in distance, fare amount also increases.

dist	fare_amount
1.030764	4.5
8.450134	16.9
1.389525	5.7
2.799270	7.7
1.999157	5.3

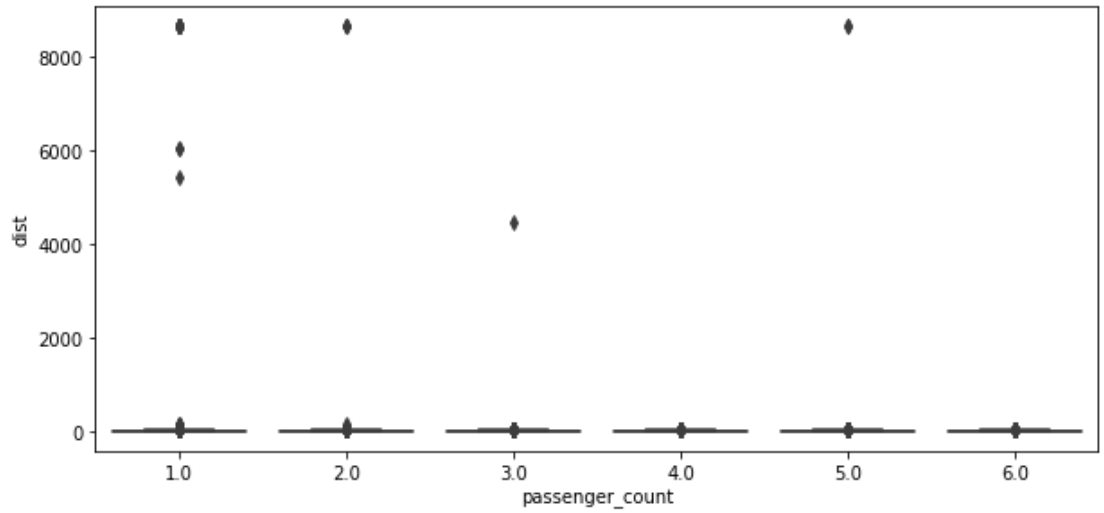
Summary of distance values are given below

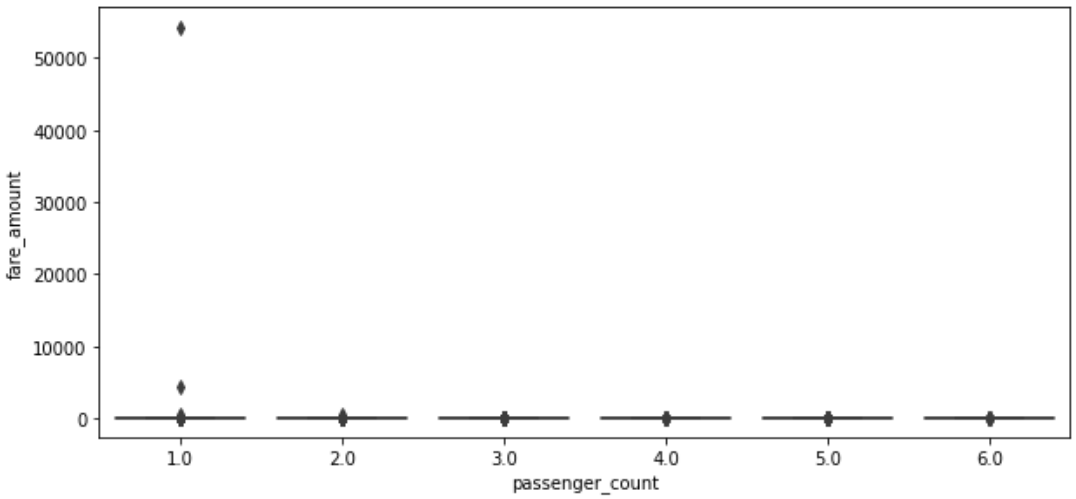
	passenger_count	pickup_hour	pickup_month	pickup_weekday	pickup_year	dist	fare_amount
count	15929.000000	15929.000000	15929.000000	15929.000000	15929.000000	15929.000000	15929.000000
mean	1.649633	13.493608	6.263634	4.033870	2011.731842	15.052597	15.057689
std	1.265923	6.522473	3.446568	1.968566	1.867333	311.468509	431.984417
min	1.000000	0.000000	1.000000	1.000000	2009.000000	0.000000	0.010000
25%	1.000000	9.000000	3.000000	2.000000	2010.000000	1.214832	6.000000
50%	1.000000	14.000000	6.000000	4.000000	2012.000000	2.125955	8.500000
75%	2.000000	19.000000	9.000000	6.000000	2013.000000	3.855249	12.500000
max	6.000000	23.000000	12.000000	7.000000	2015.000000	8667.542104	54343.000000

As seen from figure, there are values having distance zero. We will remove all rows having distance zero, as this will not contribute any information to model. Higher values will be considered during outlier analysis. Here co-ordinate variables are dropped as both distance and co-ordinates carry same information.

OUTLIER ANALYSIS

It is performed on two numeric variables distance and fare amount. Below figures show outliers in data.



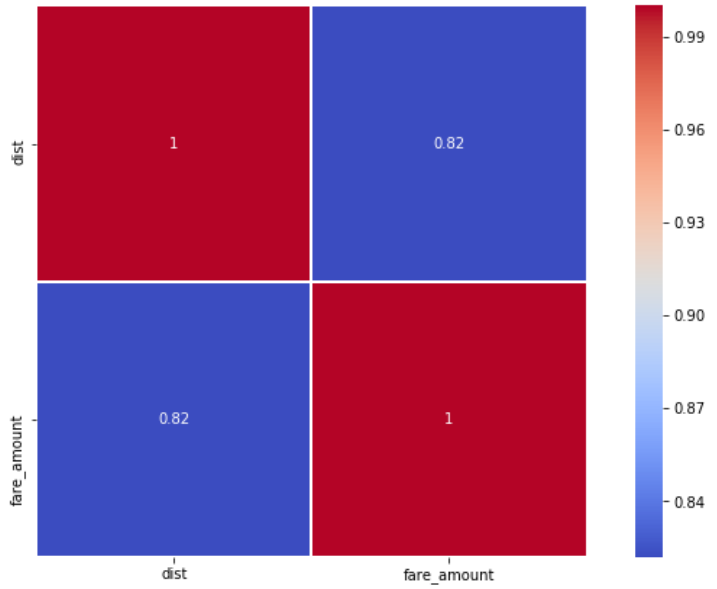


These outliers are imputed using KNN imputation with k=2.

FEATURE SELECTION

Feature selection is the process of selecting a subset of relevant features (predictors) for use in model construction. There is a possibility that many variables in our analysis are not important at all or may cause multicollinearity problem. Here, we use correlation analysis for numeric variables and ANOVA for categorical variables.

In correlation analysis, we remove variables, if they are highly correlated.



As seen from figure there is a high correlation between distance (independent variable) and fare amount (dependent variable). So, we will keep this variable.

ANOVA is applied on categorical variables v/s numeric fare amount.

Below table shows the summary of ANOVA

```
> summary(aov_results)
                Df Sum Sq Mean Sq F value   Pr(>F)
passenger_count  5    465    93.0   5.631 3.43e-05 ***
pickup_hour     23   1828    79.5   4.810 2.21e-13 ***
pickup_weekday   6     94    15.7   0.949    0.458
pickup_mnth     11    652    59.3   3.589 4.47e-05 ***
pickup_yr        6   6964  1160.6  70.265  < 2e-16 ***
Residuals    15421 254717    16.5
```

As seen from above figure, p value is greater than 0.05(95% confidence) for pickup weekday. We fail to reject NULL hypothesis. So pickup weekday and fare amount variables are independent

We will remove pickup weekday. Other independent variables are dependent on dependent variable.

MODELING

As this is regression problem, we will start from simple regression model and move towards complex ensemble models.

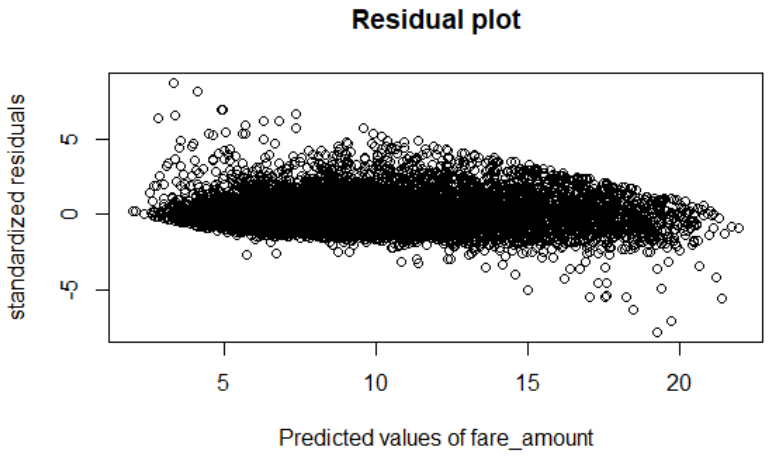
Linear regression

Below figure shows the summary of model applied on data

```
Call:
lm(formula = fare_amount ~ ., data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-16.7633  -1.2847  -0.3342   0.8685  18.7341

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       2.490309   0.132849  18.745  < 2e-16 ***
passenger_count2  0.036455   0.056182   0.649 0.516429
passenger_count3  0.218044   0.096806   2.252 0.024315 *
passenger_count4  0.326910   0.135590   2.411 0.015923 *
passenger_count5  0.106920   0.080014   1.336 0.181488
passenger_count6  0.320618   0.144854   2.213 0.026889 *
pickup_mnth02     0.009125   0.093537   0.098 0.922284
pickup_mnth03     0.121946   0.090534   1.347 0.178017
pickup_mnth04     0.154653   0.091808   1.685 0.092105 .
pickup_mnth05     0.334946   0.090938   3.683 0.000231 ***
pickup_mnth06     0.195445   0.090779   2.153 0.031340 *
pickup_mnth07     0.183842   0.097817   1.879 0.060206 .
pickup_mnth08     0.185581   0.098810   1.878 0.060382 .
pickup_mnth09     0.647210   0.096154   6.731 1.76e-11 ***
pickup_mnth10     0.563758   0.095248   5.919 3.33e-09 ***
pickup_mnth11     0.638605   0.095972   6.654 2.97e-11 ***
pickup_mnth12     0.527669   0.096045   5.494 4.01e-08 ***
pickup_yr2010     0.002961   0.070500   0.042 0.966497
pickup_yr2011    -0.026528   0.069873  -0.380 0.704206
pickup_yr2012     0.507229   0.069776   7.269 3.83e-13 ***
pickup_yr2013     1.323078   0.070067  18.883  < 2e-16 ***
pickup_yr2014     1.501499   0.070941  21.165  < 2e-16 ***
pickup_yr2015     1.838373   0.089728  20.488  < 2e-16 ***
pickup_hour01    -0.163709   0.152206  -1.076 0.282137
pickup_hour02    -0.145643   0.169015  -0.862 0.388860
pickup_hour03     0.132180   0.175692   0.752 0.451861
pickup_hour04    -0.005433   0.201742  -0.027 0.978517
pickup_hour05    -0.613028   0.215249  -2.848 0.004407 **
pickup_hour06    -0.693035   0.167381  -4.140 3.49e-05 ***
pickup_hour07    -0.126450   0.139725  -0.905 0.365488
pickup_hour08     0.425572   0.137432   3.097 0.001962 **
pickup_hour09     0.793624   0.134401   5.905 3.62e-09 ***
pickup_hour10     0.452160   0.139774   3.235 0.001220 **
pickup_hour11     0.615168   0.135169   4.551 5.39e-06 ***
pickup_hour12     0.897731   0.133078   6.746 1.59e-11 ***
pickup_hour13     0.897424   0.134566   6.669 2.69e-11 ***
pickup_hour14     0.731764   0.134184   5.453 5.04e-08 ***
pickup_hour15     0.653249   0.135426   4.824 1.43e-06 ***
pickup_hour16     0.610695   0.138535   4.408 1.05e-05 ***
pickup_hour17     0.684350   0.134477   5.089 3.65e-07 ***
pickup_hour18     0.634674   0.127908   4.962 7.07e-07 ***
pickup_hour19     0.338597   0.127795   2.650 0.008071 **
pickup_hour20     0.124967   0.128770   0.970 0.331836
pickup_hour21     0.176038   0.129772   1.357 0.174961
pickup_hour22     0.019598   0.129709   0.151 0.879903
pickup_hour23    -0.227245   0.133456  -1.703 0.088637 .
dist              2.091574   0.011879 176.074  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.168 on 12333 degrees of freedom
Multiple R-squared:  0.7265,    Adjusted R-squared:  0.7254
F-statistic:   712 on 46 and 12333 DF,  p-value: < 2.2e-16
```

We can see R squared value is 0.7265, this model can explain 72.65% of data. Also, from F statistics we can conclude that full model is adequate.

Below figure shows residual plot



So, we can consider this model for evaluating with test data. After applying on test data, model performance is evaluated with RMSE, MAPE. Evaluation results are shown below,

```
In [203]: mse = metrics.mean_squared_error(y_test,pred_lm)
          rms_lm = np.sqrt(mse)
          rms_lm

Out[203]: 2.2449881997022745

In [204]: mape_lm = mape(y_test,pred_lm)
          mape_lm

Out[204]: 0.18643852852394138
```

RMSE is 2.24 and MAPE is 0.1864, accuracy of model is 81.6%. This model has performed fairly good on test data. But still we can consider other model and evaluate the results.

Random forest

This model is applied on data and evaluated results are shown below

```
In [210]: mse = metrics.mean_squared_error(y_test,pred_rf)
          rms_rf = np.sqrt(mse)
          rms_rf

Out[210]: 2.2561621480216534

In [211]: mape_rf = mape(y_test,pred_rf)
          mape_rf

Out[211]: 0.19133435858894846
```

As seen from results above, RMSE is 2.25 and accuracy of model is 80.8%. This model has not performed better than Linear regression. So, we will check other models like GBM and XGBoost.

Gradient boosting

Results of this model on test data are given below

```
In [218]: mse = metrics.mean_squared_error(y_test,pred_gbm)
          rms_gbm = np.sqrt(mse)
          rms_gbm

Out[218]: 2.1777411099892805

In [219]: mape_gbm = mape(y_test,pred_gbm)
          mape_gbm

Out[219]: 0.1786119886496532
```

As seen from figure, this model gave better results than Linear regression and random forest. RMSE is 2.1774 and accuracy is 82.2%.

XGBoost

Results of model is shown in below figure.

```
rms_xgb = rms(y_test,pred_xgb)
rms_xgb

2.1734134326784598

mape_xgb = mape(y_test,pred_xgb)
mape_xgb

0.17839489762020352
```

This model also outperformed Linear regression model and Random forest. It has RMSE of 2.1734 and accuracy of 82.1%

Chapter 3 Conclusion

MODEL EVALUATION

Model evaluation on test data is necessary in selection of model. Here we will use error metrics RMS as primary measure for selecting the model. It is used because it provides measure of spread of the true y values about the predicted values. It will have same unit as that of value predicted, which helps us to understand the model easily. Here we can expect 68% of y values to be within one RMSE and 95% to be within two RMSE.

RMSE is calculated using below formula

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

MODEL SELECTION

By looking at selection criteria i.e. RMSE of all models, XGBoost and GBM has performed better than linear regression and random forest. Below figures shows distribution of errors for different models.

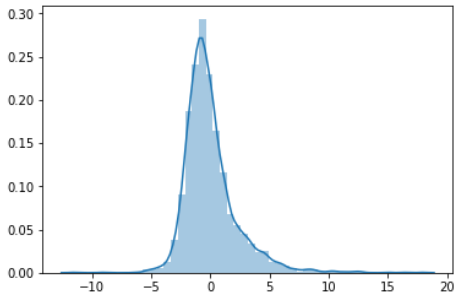


Figure: Distribution plot of (y_test - pred) for Linear regression

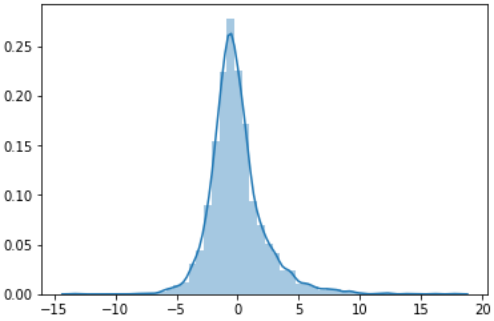


Figure: Distribution plot of (y_test - pred) for Random forest

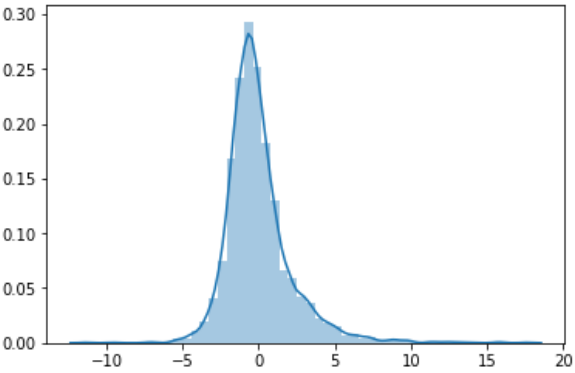


Figure: Distribution plot of (y_test - pred) for Gradient boosting

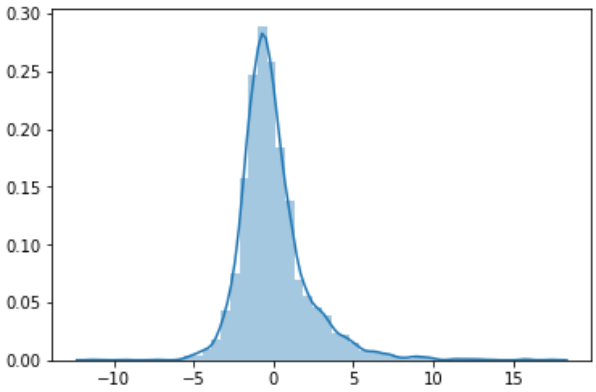


Figure: Distribution plot of (y_test - pred) for XGBoost

As seen from figures above, errors are normally distributed. So, 68% of y values will be within one error distance and 95% within two error distance.

After considering all above factors, both Gradient boosting and XGBoost can be selected for prediction. In this project, prediction values of both models are submitted.