PREDICTION OF FLIGHT TICKET PRICE
IME672 – Data Mining
Indian Institute of Technology Kanpur

Supervised by: Dr. Faiz Hamid

20114261 Abdul Ahad Khan
20114264 Ayushi Mishra
20114016 Razi Haneef
20114271 Shubhendu Singh
20114021 Sk Raju

Problem Description

We have been given with two datasets having flight booking records between Beijing and Shanghai. The task is to predict flight price for a flight ticket considering day of booking, date difference between departure and booking date, cabin class, airline and other parameters.

Data Understanding

Our data had 14 attributes and 568917 tuples.

Sl.No	Attributes	Type	Distinct Values
1	ID	Nominal	568917
2	flightNumber	Nominal	100
3	craftTypeCode	Nominal	45
4	depAirport	Nominal	2
5	traAirport	Nominal	1
6	arrAirport	Nominal	2
7	departureDate	Numerical	12823
8	arrivalDate	Numerical	14704
9	cabinClass	Ordinal	3
10	priceClass	Nominal	32
11	Price	Numerical	681
12	Rate	Numerical	87
13	createDate	Numerical	1249
14	dateDifference	Numerical	15

Table 1: Data Attribute Summary

1. *ID*: Each of the tuple had distinct values for ID. Hence this should not have an impact on the flight price.
2. *flightNumber:* It is a code for an airline service with a combination of two character and 1-4 numbers. We had 100 different flight numbers indicating there are 100 different flight services (to and fro included) between Beijing and Shanghai.
3. *craftTypeCode:* These codes are defined by international bodies of aviation, ICAO and the IATA. It is used by air traffic control team in planning of airline operations. The dataset had around 45 different craftType codes.
4. *depAirport:* Airport from which flight departs. It is usually represented by some meaningful codes relating to cities of airport. We have only 2 distinct values for this attribute (as well as for arrAirport) as the dataset under consideration is only for flights between Beijing and Shanghai.
5. *traAirport:* It represents the transit airports between departure airport and destination airport if existed. The dataset had only one transit airport (if existed) as LYA (Luoyang) and no value if there is no transit airport.
6. *arrAirport:* It represents the destination airport represented with code. It had 2 distinct values.
7. *departureDate:* It contains the departure date and time of each booking. The data was over the span of 7 months (ranging from January 2019 to July 2019).
8. *arrivalDate:* It contains the arrival date and time of each booking. The range of data was same as that of departure date.
9. *cabinClass:* This is an ordinal attribute with three distinct values in order of price as F,C and Y. These are called travel classes named F (First Class), C (Business Class) and Y(Economy Class) with First Class having the highest flight price and Economy Class having the least flight price.
10. *priceClass*
11. *Price:* This is a numerical attribute representing the cost of each booking in the dataset. Even though we have 568917 bookings, there are only 681 distinct price values.

ID	flightNumber	craftTypeCode	depAirport	traAirport	arrAirport	departureDate	arrivalDate	cabinClass	priceClass	price	rate	createDate	dateDifference
14393	HO1252	320	PEK	NaN	SHA	2019-01-04 06:35:00	2019-01-04 08:55:00	C	C	1860	1.00	2019-01-03 14:26:15	1
14409	MU5138	33L	PEK	NaN	SHA	2019-01-04 07:00:00	2019-01-04 09:15:00	C	I	1640	0.31	2019-01-03 14:26:15	1
14415	MU5138	33L	PEK	NaN	SHA	2019-01-04 07:00:00	2019-01-04 09:15:00	C	J	5360	1.00	2019-01-03 14:26:15	1
14429	HU7605	350	PEK	NaN	SHA	2019-01-04 07:20:00	2019-01-04 09:35:00	C	I	1635	0.29	2019-01-03 14:26:15	1
14431	HU7605	350	PEK	NaN	SHA	2019-01-04 07:20:00	2019-01-04 09:35:00	C	I	1640	0.29	2019-01-03 14:26:15	1

Figure 1: Dataset Sample

12. *Rate:* It is also a numeric attribute with values ranging between 0 and 1 that let us know if there was any discount provided during the booking. 1 indicates no discount and 0 indicating 100% discount.
13. *createDate:* It is a numerical attribute containing date and time of flight booking.
14. *dateDifference:* This is the difference between departure date and booking date. There are only 15 distinct values in this attribute. This value is expected to be positive as date of booking cannot be after date of departure.

Data Integration

There are two datasets for data understanding, one of them having data of flights from Beijing to Shanghai and other having data of flights from Shanghai to Beijing. Both this data sets were having same set of attributes (with same attribute names) and similar data type. So both this datasets were integrated into one before proceeding into any further analysis.

Data Cleaning

1. *Missing Values*

   There were 580005 missing values for the attribute traAirport. This implies that 580005 out of the 598617 were booking of direct flights from Shanghai to Beijing or vice versa. only 18612 of the total booking were having one transit airport between source and destination airports.

   This nominal attribute was converted into a asymmetric binary attribute by using 0 for missing values in traAirport attribute (i.e,, direct flights) and 1 for LYA. This is a careful correction which will never the degrade the quality of data. Histogram of price vs traAirport (Figure 2) indicates the difference in value of flight prices for direct flights(marked as 1) and indirect flights(i.e. flights having transit airport during the journey, marked as 1).

   This correction helped us in the rectification of the problem of missing values.

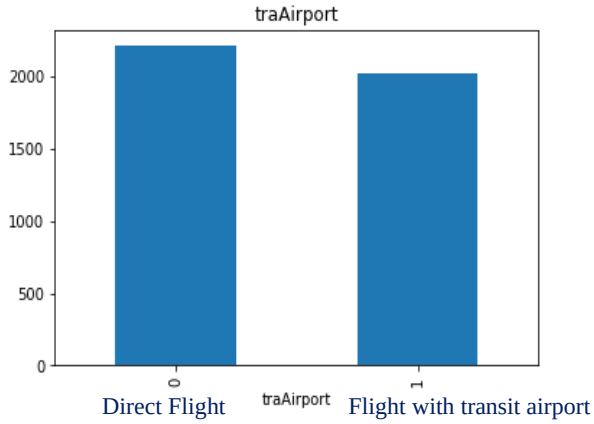

Figure 2: Price vs traAirport

2. *Noisy Data*

   It was found that there were data with booking date of flights after departure date of flights. It was observed from the dateDifferenece attribute showing negative values. This is never possible. This error might have occurred due to any technical issues that might have occurred in the booking website. It was also observed that booking time of all the flights were same, so this was also removed from further analysis.

   These data are noisy and are removed from the analysis. There were around 175 tuples in the data with negative dateDifference value.

3. *Attribute Addition and Deletion*

   To analyse the dataset effectively some new attributes were added, and some were removed without losing any quality to the existing data.

   a) *departureDate, createDate, arrivalDate:*

   All these attributes have information of year, month, date, day and time of departure date, booking date and arrival date of flights. We expect to have price variation in flight tickets depending on hour, day, month, and year. The dataset contains only the data for the year of 2019. This indicates we cannot study the variation of flight price with respect to year. So, each of the date attributes are split into date, day, month, and time forming a new attribute while the parent attribute is deleted.

   b) *flightDuration*

   We expect to have price dependency on flight duration. To analyse the validity of this hypothesis, we add a new attribute flightDuration that takes the difference of arrival date and departure date.

   c) *Airline*

   Airline can be extracted from the first two characters of the attribute flightNumber. We expect to have flight price variation with respect to different airline service providers. To include this into analysis, a new attribute Airline is added as attribute. There were seven different airlines providing service between Beijing and Shanghai

   d) *ID*

   All the booking ID to be different. Since this does not have any effect on the flight price, we removed this attribute before proceeding into any analysis.

4. *Duplicate Values*

Before proceeding into any further analysis, the data was checked for any duplicate tuples. It was found that there were around 250000 duplicate tuples. All the duplicate tuples were dropped from the data to avoid overfitting during training of tuples. Duplicate tuples was generated after removal of attribute ID.

Data Visualization

1. *Price Variation with airline*

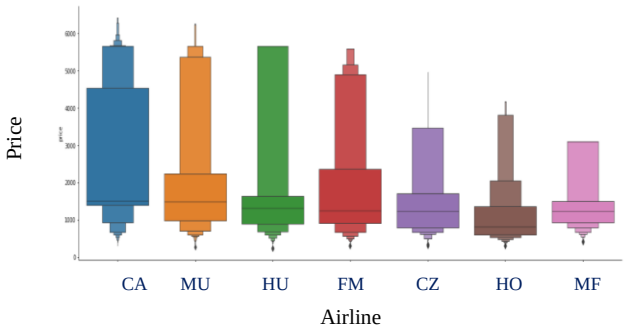

Figure 3: Boxplot of flight Price vs Airline

It is observed from figure 3 that flight CA has highest variation in prices identified from its 1st quartile and 3rd quartile values. Median values of flight prices are almost same for CA, MU, HU, FM, CZ and MF while HO has a slightly lesser value. From this it can be concluded that airline has very little effect on flight price.

2. *Price Variation with cabin class*

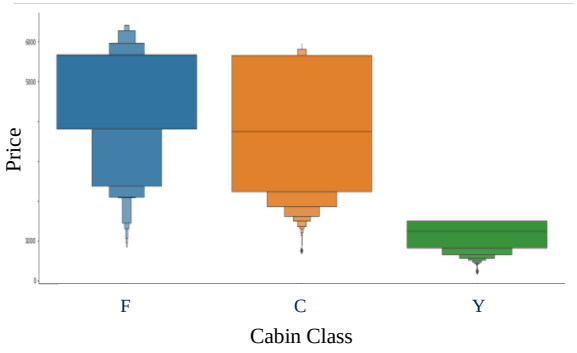

Figure 4: Boxplot of flight Price vs Cabin Class

Flight price is observed to be higher for F (First) class and least for Y (Economy) class. The variation of price is found to be higher in C (Business) class. Hence this attribute will have a stronger influence on the flight price.

3. *Price Variation with rate*

Flight price is found to have a linear relation with price as seen in figure 5. The scattering of data is expected due to the difference is cabin classes. The

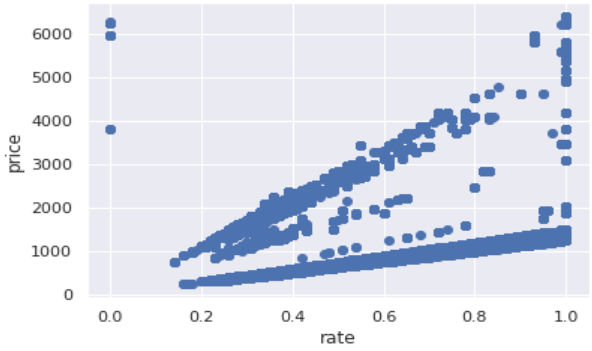

Figure 5: Flight price variation with rate

data lying together with least slope can be grouped as that of economy class. The datapoints with intermediate slope can be grouped as business class and the data with the largest slope can be grouped as first class. Hence a stronger relation with rate is expected.

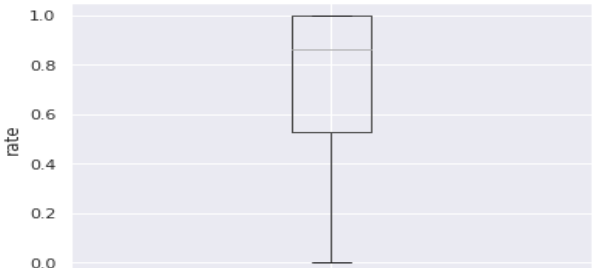

Figure 6: Boxplot of rate

To understand further on rate, we will study the box plot of rate as shown in figure 6. The value is mostly expected to range between 0.5 and 0.95 with a median value of 0.85.

4. *Price Variation with flight duration*

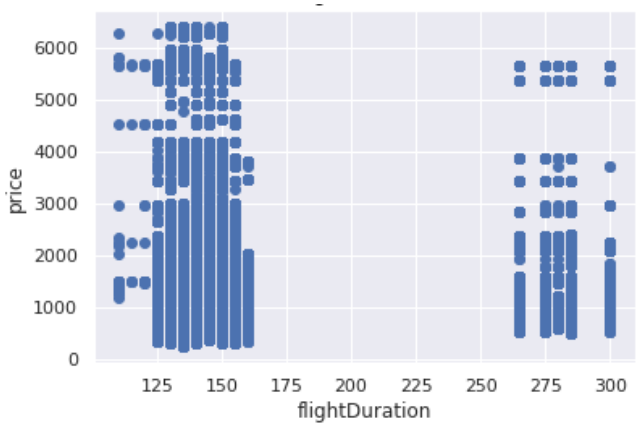

Figure 7: Flight price variation with flight duration

The flight price was expected to vary with flight duration. But there were no meaningful insights to support this hypothesis that could have taken from figure 7. Hence, we can conclude that flight duration does not have any significant effect on flight price.

5. *Price Variation with CraftTypeCode*
It is observed from figure 7 to that price have dependency on craftTypeCode. But there are some craftTypeCode that are too frequent while some others are less frequent. So, it is expected that the dependency on the model will be lesser.

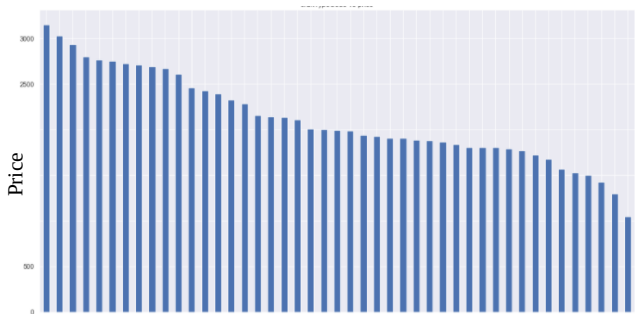

Figure 7: Flight price vs craftTypeCode

6. *Price Variation with dateDifference*
It is expected that price will generally decrease when it is booked well in advance of the departure date of flight and vice versa. This general trend can be seen from figure 8, where flight price variation with date difference between departure and booking dates.

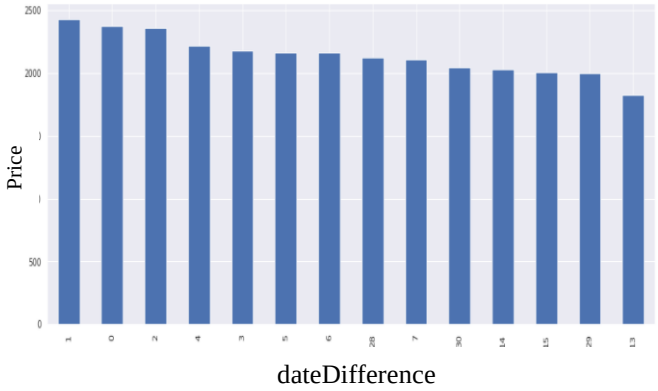

Figure 8: Flight price vs dateDifference

7. *Price Variation with day*
Flight price is found to be higher on Friday and the least value on Saturday. But this variation is not large. These values lie close to each other, The variation is depicted in figure 9.

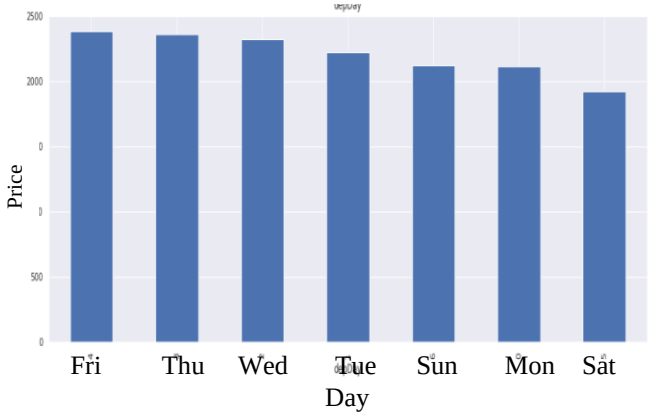

Figure 9: Flight price vs day

8. *Price Variation with Airline*
Flight price showed a large variation with respect to airline. This might be due to the fact that some airlines might provide large number of seats allotted to economy class while other airlines might have more number of first class seats. The variation is depicted in figure 10.

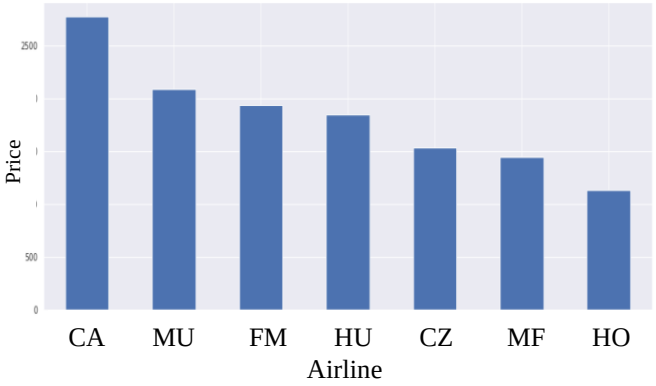

Figure 10: Flight price vs day

Correlation Heat Map

It was observed from the heat map, only the numerical attribute rate was having a significant impact on price apart from cabin class. Even though it was expected some other attributes could have influenced price, it was not the case.

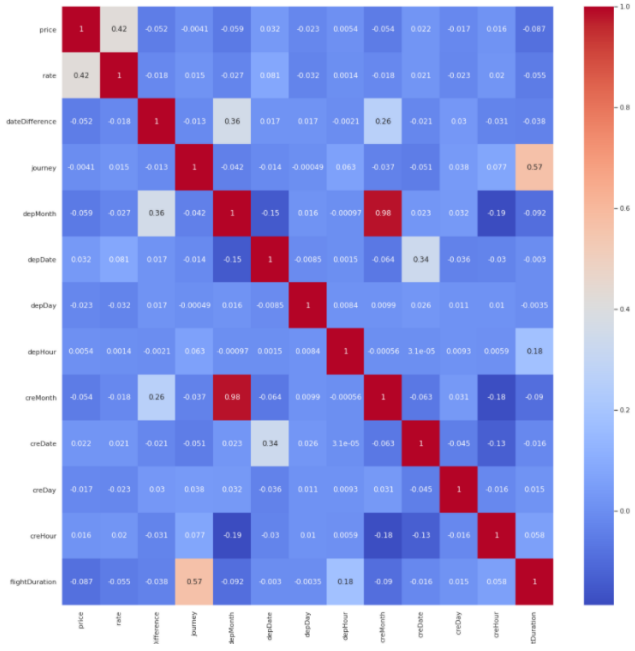
Figure 11: Correlation Heat Map

Data Modelling

Data Preparation for Modelling

1. *Splitting Data: Train & Test Data*
   Entire data should be split into two subsets namely training set and test set. Training dataset is used to train the model for the prediction, it is validated and tested against the test data set to check for the accuracy of the model.80% of the reduced dataset was taken as training dataset and remaining 20% was used as test data set.

2. *One hot encoding*
   Encoding is used to change the categorical value with a number. The most widely used encoding includes Level Encoding and One Hot Encoding. One hot encoding creates additional attributes corresponding to each different value in a categorical attribute with value as 1 in the original position and 0 in others. In our project cabinClass, arline, priceCategory are encoded as One Hot Encoding.

3. *Feature Selection by LASSO CV*
   The feature selection is the process that choose a reduced number of explanatory variables to describe a response variable. Feature selection is used to (i) make the model easier to interpret, removing variables that are redundant and do not add any information, (ii) reduce the size of the problem to enable algorithms to work faster, making it possible to handle with high-dimensional data and (iii) reduce overfitting.
   LASSO performs majorly two tasks: (i) regularization and (ii) feature selection. Regularization process is a method that penalizes the coefficient of the regression variables shrinking some of them to zero.
   During features selection process the variables that still have a non-zero coefficient after the shrinking process are selected to be part of the model. The goal of this process is to minimize the prediction error.
   In practice the tuning parameter λ, that controls the strength of the penalty, assume a great importance. Indeed, when λ is sufficiently large then coefficients are forced to be exactly equal to zero, this way dimensionality can be reduced. The larger is the parameter λ the greater number of coefficients are shrinked to zero. On the other hand, if $\lambda = 0$ we have an OLS (Ordinary Least Square) regression.
   In terms of the tuning parameter λ, bias increases and variance decreases when λ increases, indeed a trade-off between bias and variance has to be found. LASSO helps to increase the model interpretability by eliminating irrelevant variables that are not associated with the response variable, this way also overfitting is reduced.
   L1 regularization: Lasso Regularization adds a penalty to the error function. The penalty is the sum of the absolute values of weights.
   L2 regularization: L2 Regularization or Ridge Regularization also adds a penalty to the error function. But the penalty here is the sum of the squared values of weights.

4. *Removal of duplicate values after Feature Selection*
   There were around 140000 duplicate values in reduced dataset after feature selection was applied. All the duplicate tuples were removed. The final dataset for modelling contained around 120000 tuples and 18 attributes, out of which 95000 were training data and 25000 were test data,

KNN Regressor Model

KNN algorithm is popular for its application in classification, however it is also effective for regression. Feature similarity is used to predict the outcome of new tuple. The outcome is approximated based on its closeness to the data tuples in training set. The distance between each tuple in training set and the input tuple is calculated. K-closest points based on this distance are identified and the average of this k points will give the outcome of the input tuple. For a very low value of k, the data will be overfitted.

The range of data used for k was between 1 to 20 in our project. The model performance was evaluated using RMS and R-Squared Value as shown in table 2. The result was satisfactory.

Data Set	*RMS*	*R-Squared*
Training Set	336.70	0.9490
Test Set	447.21	0.9107

Table 2: KNN Regressor Model Performance

Ridge Regressor Model

Multicollinearity is the near linear relationship among the independent variables. When multicollinearity occurs in the dataset, it will affect the variance values drastically (compared to true values) even the least squares estimators are unbiased. Ridge regression adds

a degree of bias to the regression estimates reducing the variances (or standard errors). This is expected to give reliable datasets.

Ridge regressor model will take care of any dependencies among the assumed independent attributes. Ridge regressor gave a better result R-squared value compared to KNN Regressor model as shown in table 3.

Data Set	*RMS*	*R-Squared*
Training Set	491.80	0.8912
Test Set	488.16	0.8936

Table 3: Ridge Regressor Model Performance

Decision Tree Regressor Model

Decision tree builds regression or classification models in the forms of tree structure. It can be used for both nominal and numeric attributes. Decision tree has a node, a branch with a test and all the final leaf is result or classification.

The decision tree had a higher R-squared value compared to both KNN and Ridge regressor model as shown in Table 4.

Data Set	*RMS*	*R-Squared*
Training Set	5.64	0.9999856
Test Set	13.81	0.9999147

Table 4: Decision Tree Regressor Model

Random Forest Regressor Model

Random forest is a supervised learning algorithm which can be used for both classification and regression. This algorithm uses bagging concept were all small trees run parallelly. Interaction between any two trees during building is null. It uses meta estimator which is a combination of multiple trees. The number of features that can be split on each node is limited to ensure heavy dependency on attribute is reduced.

Data Set	*RMS*	*R-Squared*
Training Set	4.88	0.99998924
Test Set	13.73	0.9999157

Table 5: Random Forest Regressor Model

Stacking Regressor Model

Stacking Regressor is an ensemble machine learning algorithm that learns how to best merge the predictions from other other models.

We combined KNN Regressor model, Ridge regressor model and Random Forest Regressor Model in stacking regressor model.

Data Set	*RMS*	*R-Squared*
Training Set	1.53	0.99999894
Test Set	12.94	0.99992521

Table 6: Stacking Regressor Model Performance

Comparison of models

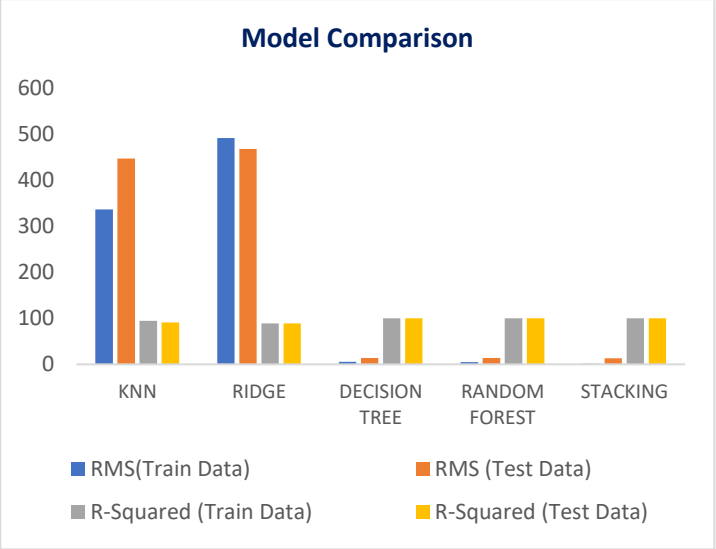

Figure 12: Comparison of all models

Conclusion

1. The dataset had around 600000 data tuples and 14 attributes. This dataset was reduced to 120000 tuples and 18 attributes before modelling.
2. The flight price showed a very strong correlation with the rate and cabin class attributes while other attributes never had any significant influence.
3. The reduced dataset was divided into two subsets: training set (80%) and test set (20%).
4. Training set were trained using KNN regressor, Ridge regressor, Decision tree regressor, Random Forest regressor and Stacking regressor models.
5. Decision tree regressor, Random Forest Regressor and Stacking regressor showed the highest R-squared values compared to KNN and Ridge regressor model.
6. All the models produced satisfactory results having R-squared values greater than 0.90.

Reference

1. https://www.kaggle.com/lpisallerl/air-tickets-between-shanghai-and-beijing
2. https://scikit-learn.org/
3. Towards DataScience