RWA: A Regression-based Scheme for Flight Price Prediction

by

Zhenbang Wang

Submitted in partial fulfilment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
April 2020

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Flying has become the primary transportation method for long-distance travel. Most of the travelers are intend to purchase the tickets with lowest cost. In practice, many travelers tend to purchase flight tickets as early as possible to avoid possible price hikes. However, this type of purchase behavior does not always lead to the most economical flight tickets.

In our research, we proposed a regression-based scheme, RWA, to improve the accuracy of flight price prediction. Specifically, we first collected a variety of different flight price data sets from publicly-available travel websites. After that, we devised a data splitting method to divide the training data set into two partitions because the price change patterns in these partitions are entirely different. Finally, RWA is applied to each of the partitions to arrive at the accurately-predicted flight price. To verify the effectiveness of RWA, extensive experiments were carried out in our research.

LIST OF ABBREVIATIONS AND SYMBOLS USED

RWA	Regression-based Weighted Average
LA	Los Angeles
MPP	Marked Point Process
KDD	K Nearest Neighbors
PA	Passive-Aggressive
DRS	Deep Regressor Stacking
SVM	Passive Reader Active Tag
RMSE	Root Mean Square Error
XGboost	eXtreme Gradient Boosting
CART	Classification And Regression Trees
LD	Less than key point Dataset
GD	Greater than key point Dataset
WD	Whole Dataset
M_1	Model of using WD as the dataset
M_2	Model of using GD as the dataset
M_3	Model of using LD as the dataset
P_1	Final result of GD part of dataset
P_2	Final result of LD part of dataset
w_1	Redundant feature set one
w_2	Redundant feature set two
f_1	Feature to be removed in feature set one
f_2	Feature to be removed in feature set two
e	Element in redundant feature set
y'	The predict value of RWA
g(x)	The dependent variable of Linear Regression
W_i	Weight of input of Linear Regression
MSE	Mean squared error

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Prof. Qiang Ye for the continuous support of my master study and research. I am truly thankful for his patience, motivation, enthusiasm, and immense knowledge. His guidance in all the time of my research. Whenever I stuck in a bottleneck, he can always point me a correct and clear way.

CHAPTER 1 INTRODUCTION

Flight price prediction is an important issue that has not been thoroughly studied. In our research, we attempt to apply machine learning techniques to this problem to improve prediction accuracy. In this chapter, the problem to be tackled is first described. Afterward, an overview of machine learning techniques and the proposed prediction scheme is presented. Finally, the outline of the thesis is included.

1.1 PROBLEM STATEMENT

Flying has become the primary transportation method for long-distance travel [1]. To maximize the profit, airlines employ a complicated price strategy called “yield management” to formulate the price of each flight [2][3][4][5][6]. With this technique, the price can be automatically adjusted according to many factors, such as the number of days before departure, seat availability, market competition, etc. [7]. The final goal is to secure the maximized profit from each flight. From the perspective of travelers, flight price has a serious impact on their travel choice and purchase behavior [8]. Comparing with other transportation such like train or cruises, flight ticket price is more changeable. When travelers choose airlines as their transportation method, most of them prefer to obtain air tickets at the lowest price. Since travelers tend to believe that flight price goes up when the purchase date is close to the departure date, they often purchase flight tickets on a date that is as far from the departure date as possible. However, this type of purchase behavior is not always correct. When it fails, travelers will spend more money event flight tickets are purchased in advance.

Actually, it is very difficult for travellers to predict when the best time to purchase fight tickets is thanks to the following reasons:

- Incomplete Information: Travellers can only access part of the airline's internal information. In fact, they do not have the access to the key data, such as the number of the remaining tickets and the agreement between different airline companies.
- Fragmented Information: The information that travellers can obtain is fragmented. For example, it is very difficult for an average traveller to deduce the relationship between flight price and flight characters, such as the number of stopovers, the departure time, etc.
- Irregular Change: Although travellers can collect historical flight price, the price change is not smooth. Actually, it seems to be highly irregular. Consequently, travellers cannot easily predict future flight price according to the historical values.

In this thesis, we attempt to use machine learning techniques to improve the accuracy of flight price prediction.

1.2 MECHINE LEARNING

Machine learning (ML) is an essential technique in artificial intelligence. Over the past year, it has been used in varied areas, such as cybersecurity, health care, finance, logistics, and manufacturing. With the assistance of machine learning, the efficiency or accuracy of many applications has been significantly improved. Theoretically, ML highly relies on models and logic [9].

Machine learning techniques can be divided into two main categories: supervised learning and unsupervised learning. In supervised learning, the algorithm can construct a model of a set of data that contains input labels and desired output [10]. This part of the data is known as training data. Each sample of training data contains one or more input and a desired output, and it is represented by an array or vector in machine learning models. Thus, the whole training data is represented by a matrix. There should also be an optimization function that can correct the model in each iteration. It can make the model more precise in each turn of training [11]. Supervised learning includes classification and regression. When the output is limited to a limited set of values, a classification algorithm will be used; when the output has any value within a certain range, a regression algorithm will be used [12].

The difference between unsupervised learning algorithms and supervised learning algorithms is that for unsupervised learning algorithms, the data only contains input value, and there is no desired output value provided. And it is always aiming to structure the data, for example, clustering the data into several groups. Therefore, the algorithm will learn from test data that has not been labeled or classified. Basing on the existence of commonalities in each sample of the data, unsupervised algorithms can identify similar data points without any feedback. Unsupervised learning technique is widely used in the field of density estimation and statistics [13].

Machine learning has become a standard technique which is focusing on prediction or classification based on the known properties [14]. In recent years, it has precise results in many aspects, such as medicine [15], cybersecurity [16] and insurance [17].

With the growth of the Internet, traveling agents provide every day's real-time price of different airlines. This makes it easier for consumers to get access to the high volume of the airline's data. As we all know, with the increase in data amount, machine learning can bring us a more precise result. Thus, it is possible to apply machine learning techniques to the airline flight price prediction problem.

1.3 SYSTEM OVERVIEW

To help travelers to purchase the cheapest flight ticket, over the past years, there have been many studies on the price prediction. Generally, the existing studies employ data mining and machine learning techniques to analyze the data and generate a recommendation of "buy" or "wait" [18] [19] [20] [21]. Some existing studies also use prediction methods estimate when flight price reaches the lowest point [22] or calculate the probability of going up or going down [23]. Few existing studies attempt to predict specific flight prices for travelers. To our knowledge, the studies that predict specific flight prices focus on either the Chinese or Russian market. None of the existing flight price prediction studies involve Canadian airlines. Besides, the accuracy of the existing flight price prediction studies is not satisfactory.

The Canadian airline industry is growing rapidly. For example, in the past year, there were 34.7 million domestic air passengers [24], and the year-over-year growth rate reached 4.3%. Air Canada, the largest airline in Canada, has an 83.4% load factor with a flight distance of 94.1 billion miles in total in the past year, which brings Air Canada 17.2 billion dollars of revenue. At Toronto Pearson, the busiest airport in Canada, there were over 17.8 million domestic passengers last year, and the number of domestic passengers is projected to reach over 51 million by 2035. Given the scale of the market, predicting flight prices for Canadian travelers has become more and more critical.

Although flight price prediction has been an important issue over the past years, it remains to be an open problem to be tackled thanks to the following characteristics of flight price [25] [26]:

- High Volatility: Flight price fluctuates seriously and flight price change is aperiodic.
- Diversity: It is tough to construct a model that works for the airlines in different countries because the pricing strategies adopted by airlines could be profoundly different.

- Nonstationary Pattern: The pattern of flight price change could change over time. A pattern that has been properly captured during a specific period may fail in the future. Consequently, the generated pattern needs to be updated once in a while.
- Context-awareness: Flight price is often closely related to the contextual information, such as whether the departure date is a holiday. However, it is challenging to model contextual features in advance.

In our research, we model flight price prediction as a time series problem and attempt to solve the problem with machine learning techniques. Because Air Canada occupies a large portion of the Canadian air market, our research focuses on the domestic flights of Air Canada. Technically, we propose a novel regress-based scheme, Regression-based Weight Average (RWA), to solve the flight price prediction problem. Specifically, the scheme involves four steps: data preprocessing, feature extraction and selection, price prediction, and regression-based weight average calculation. To improve the accuracy of flight price prediction, we propose a slope-based data split method, which can be used to divide the training data into multiple sections. Then RWA can be applied to each data section, ultimately arriving at high prediction accuracy.

The main contributions of this paper can be summarized as follows:

1. Slope-based Data Split: A novel data split method is proposed so that training data can be divided into multiple sections. The pattern of the data in one section is different from that in another section.
2. RWA: A regression-based prediction method, RWA, is devised to improve the accuracy of flight price prediction. Technically, this scheme employs linear regression to assign proper weight to several basic predictors. Then the weighted average of the results from the basic predictors is used as the final prediction result.

1.4 THESIS OUTLINE

The rest of this paper is organized as follows. Chapter 2 includes the related work. In Chapter 3, we present the 4-step process to predict flight price. The experimental results are described in Chapter 4. Our conclusions are included in Chapter 5. The limitations and future work are discussed in Chapter 6.

CHAPTER 2 RELATED WORK

The problem of flight price has been studied since 2003 [18]. And it also has different forms of application. In this chapter, we first present the various previous types of studies that are related to our study topic. Secondly, we will present the regression methods which have been used in previous studies and give a brief review of the disadvantages of current models in this problem. Thirdly, we will provide an introduction to the ensemble algorithms that we will use in our study.

2.1 FLIGHT PRICE PREDICTION

In 2003, Etzioni et al. proposed a model that aims to tell users this is the perfect timing to buy the ticket or not [18]. The model combines moving average, rule learning and Q-learning together. Their data contains two routes: Los Angeles to Boston and Seattle to Washington, D.C. And there are five features in their model, which are flight number, hours until the departure date, airline, price and route. They generate several rules, for example, when the hours before takeoff is greater or equals to 252 and the current price is greater or equals to 2223 and route is from LA to Boston, we should wait. After that, there should be several "buy" and "wait" suggestions. The final result is achieved by using an ensemble method doing the voting. By utilizing the sequence of buy or wait signal, the cost of each stimulating passenger was calculated. The total amount of money-saving by using their strategy can reach 61.8%. However, their method is not able to tell the user what is the specific price of one day and how when will the price drop down to the lowest point.

Similar to Etzioni et al. work, Bingchuan and Yudong use a Bayes classification method to tell the probability of price change in hours or days [21]. They selected the route between Shanghai and Tokyo and only focused on the flights which departure time is around 9-10 a.m. During a whole year data collection (from July 2015 to June 2016), they had over two million records, which have query days before the departure date is within 4 to 119 days. By using the probability, they build a decision system to smartly provide user buy-or-wait decisions.

Faker and Bejugum [20] provided an approach to design a decision-support system. It used the information of one specific airline, including general features and the percentage of discounts which are provided by users to calculate the probability of different situations. Then use the probability value to make the decision: it is the perfect timing to buy or not. Their model utilizes the interactive nature of the online environment providing pieces of advice to users and let users make their final decision.

In 2011, Groves and Gini proposed a regression model that is using the history diagram to predict the perfect timing for purchasing airline tickets [19]. They collected their data from Feb. 22 2011 until Jun. 23, 2011, and over 140 thousand records in total. There are two steps in their model. At first, they used a regression model to make predictions on the daily price. Secondly, after having a reliable threshold, they developed a reliable rule, which is if the price is lower than the value which is prediction price minus the threshold, travelers should buy the ticket. Otherwise, travelers should wait. Their results showed when the purchase date is over two months away from the departure date, their model can effectively lower the average cost. Their model also enables travelers to input their preference such like how many stops that travelers can accept. Wohlfarth et al. in the same year proposed a preprocess method naming MPP (Marked Point Process) [23]. It is focusing on predict the price will fall or drop at one specific point. They reduced the size of feature set and using a clustered method and a tree model to make predictions. Their data was collected from 9 flight tickets providers focusing on six roundtrips. To cover the most common stay length, they chose 3, 7, or 14 days as the staying time.

However, all these work above were trying to give suggestions to users on the trend of airline route prices. They did not provide any information about the whole period's price of the airline, which will leave a small space for users to make their choice.

In 2015, Yuwen et al. proposed an ensemble method, which is basing on learn++ [19]. They were focusing on five one-trip routes in China and chose KDD (K Nearest Neighbors) and PA (Passive-Aggressive) to finish their comparison task. Their strategy is that different routes should use different parameters to get the best result. Although for route CAN-SEL, they could get 2.85% error rate, the error rate of route BJS-HKG can even reach 17.87%.

Everton et al. in 2017 proposed a DRS (Deep Regressor Stacking) model basing on deep learning methodology and using Random Forest and SVM as their base method [26]. Their model is aiming to lower the RMSE (Root Mean Square Error). They adapted Spyromitros-Xioufis et al. [32] technique, which called Multi-Target Regressor Stacking. The key process is using the predicted value as the next turn's input and repeat.

In [30], Tao Liu et al. tried to use a Context-Aware Ensemble Regression model to predict the lowest price in a period of some specific airlines. They split features into groups and use each group of features to get results. They selected four feature groups in total. The first one is all the price information of the same itinerary, no matter the flight belongs to which airline company. The second one is the price of itineraries which departure is in recent days. The third one is the statistical values of the flight price, such like maximum and minimum price, mean price, the price rising and falling times and etc. The fourth one contains two features which are holiday or not and days before the departure date. And the last one is the searching time of airfare on three platforms, which can represent the demand of users. The final predict result is the method average of their different group's results.

In [28], Tziridis et al. proved the importance of different features by using nine mature models, making predictions without one different feature at each time. They selected eight features, which are departure time, arrival time, number of free luggage, days left until departure, number of intermediate stops, holiday or not, overnight flight or not, and day of the week or not. The flight that they collected is all from Thessaloniki to Stuttgart between December and July and 1814 in total. In their experiment, when dropping the feature days left until departure, the result became worst among all the experiment results. Besides, Bagging Regressor and Random Forest can always have good performance. They proved "Overnight or not" and "Holiday or not" have a slight influence on the prediction accuracy as well. The best accuracy they can have is 87.93%.

2.2 REGRESSION

Flight price prediction can be treated as a time series problem [24]. In this case, the regression method will be helpful. Then Janssen [29] focused on the route from San Francisco to New York and developed a Linear Quantile Mixed Regression method [30]. In his result, when the purchase date is far from the departure date, the prediction was precise, but for the days near the departure date, the prediction is not very effective.

Another regression model was proposed by Lantseva et al. in 2015, which was focusing on the Russian market. Their result showed for domestic flights, building one reliable and a precise model is hard [31].

In summary, currently the proposed models have several drawbacks which are

- Only can make decision for users
- The size of dataset is too small
- The final prediction is not precise enough

In [28], the authors have proved “days before departure days” is the most significant feature in the flight price prediction problem. In their result, they showed some features that have a negative influence on the final result. Moreover, from the result of [19] [30] [31], we can know when using one model to make the prediction, the method will be ineffective when the purchase date is near the departure date. Due to these two reasons, we first propose a data split strategy basing on days before departure. Besides, in [28], the authors also showed some features that have a negative influence on the final result. So we decide to use different feature set on different dataset. So our first hypothesis is when the data is split by the days before departure, different parts of the data can have different feature sets to get the best performance.

2.3 ENSEMBLE LEARNING

Ensemble learning has drawn much attention due to its extensibility and flexibility. In these years, ensemble learning has been proved that it can welly handle time series problems [32] [33]. The key point of ensemble learning is training a lot of learners by different methods. There are several traditional techniques, such as bagging, Random forest, Adaboost, and XGboost. Tziridis et al. results show that ensemble models can always have a low error rate and fast processing speed. In our approach, we choose XGboost, Random Forest, Bagging regressor, and Gradient Boosting trees as our base algorithms.

2.3.1 XGBoost

XGboost, at first, was proposed by Tianqi and Carlos [34]. XGboost is one of the boosting algorithms. The idea of the Boosting algorithm is to integrate many weak classifiers to form a robust classifier [35]. Because XGboost is a lifting tree model, it integrates many tree models to form a robust classifier. The tree model used is the CART regression tree model.

A CART regression tree is like a binary tree which will constantly split based on the value of features. Thus, the critical idea of the CART regression tree is by adjusting the segmentation feature to minimize the difference between the predicted value and real value [36].

The idea of the XGboost algorithm is to continuously add trees and continuously perform feature splitting to grow a tree. Each time when a tree is added, it is actually learning a new function to fit the residuals of the last prediction. When we get k trees after training, we need to predict the score of a sample. In fact, according to the characteristics of this sample, each tree will fall to a corresponding leaf node, and each leaf node corresponds to a score. In the end, adding up the scores corresponding to each tree, and that should be the predicted value of the sample.

In [37], Nielsen investigates the difference between XGboots and other traditional MART. Furthermore, he indicated the fact that XGboost can beat other methods in most cases, and this is because the algorithm is not relying on any distance metric. It learns the similarity between data points through adaptive adjustment of neighborhoods. This can make the model immune to the curse of dimensionality.

Mesut and Mustafa, in 2017, use XGboost on crude oil price prediction [38]. Their result showed that XGboost is useful in this problem. Huiting et al. [39] in the same year proposed a method which was focusing on short-term load forecasting. In their work, they chose using XGboost to evaluate the importance of different features.

2.3.2 Random Forest And Bagging Regressor

Random Forest is an old and effective machine learning approach which is proposed by Breiman in 2001 [40]. It is a product of the idea of ensemble learning. Many decision trees are integrated into a forest and used to predict the final result.

A decision tree is a tree structure which can be a binary tree or a non-binary tree [41]. Each non-leaf node represents a test on a feature attribute, each branch represents the output of this feature attribute on a certain range, and each leaf node stores a category. The process of making a decision using a decision tree is to start from the root node, test the corresponding feature attributes in the item to be classified, and select the output branch according to its value until it reaches the leaf node, using the category stored by the leaf node as the decision result.

However, the decision tree algorithm is easily overfitting because it uses the best strategy for attribute splitting. To address this problem, Random Forest algorithm trains hundreds of decision trees using different sample collections [42]. After training, the final output is the average of all models output or using the majority vote to decide.

Moreover, when each model is training, a random subset of the features is randomly selected. The purpose of this is to make the decision trees not too similar to each other. If several features are strongly related to output, then in many trees, these features will be finally selected.

Bagging regressor [43] [44] [45], fundamentally speaking, is using an ensemble of models where each model uses a bootstrapped dataset, and models' predictions are aggregated by getting the average or voting. This means, in bagging, there is no limitation of choosing models. However, in most of the case, the decision tree is chosen to perform the predictions.

The principle of the Bagging tree is similar to Random Forest. They all need a number of how many trees need to be generated. For each decision tree, only a part of the sample data is used. The difference between Bagging tree and Random Forest is:

1. Bagging tree contains fully grown decision trees. Moreover, at each node in the tree, there should be one search going over all features to find the feature that best splits the data at that node.

2. Random Forest, as mentioned before, only take a part of the features for each decision tree. Thus, during tree creation, a random number of features are chosen from all available features.

2.3.3 Gradient Boosting Trees

Gradient Boosting tree [46] is an iterative decision tree algorithm that has a strong generalization capability [47]. This algorithm consists of multiple regression trees, and it can be seen as an additional model of these trees.

For one single regression tree, a greedy algorithm is used to generate each node of the decision tree [36]. Starting from a tree with a depth of 0, enumerating all available features for each leaf node. Then, for each feature, arrange the training samples belonging to the node in ascending order of the feature value, determine the best split point of the feature by linear scanning, and record the maximum benefit of the feature. The feature with the most benefits should be chosen as the split feature. Then all these steps should be done recursively until the tree is built.

For the Gradient Boosting tree, the algorithm generates a new regression tree each iteration. Before the start of each iteration, the loss function at each training sample point will be calculated. Then a new regression tree is created to fit the residuals of the previous model by a greedy algorithm. Because this algorithm is an additional model, to simplify the complexity, only one basis function and its coefficients (structures) at each step need to be learnt. And is will gradually approximate the optimization objective function.

Comparing with XGboost, the differences are as follows:

1. Gradient Boosting tree has many non-linear transformations, strong expression ability, and it does not need to do complicated feature engineering and feature transformation.

2. Gradient Boosting tree is a continual process algorithm, which means it is not easy to parallelize. Besides, it has high computational complexity, which makes it not suitable for high-dimensional sparse features.

Yanyu and Ali [48] compared the Gradient Boosting model and Random Forest in traffic time prediction problem. They indicated the Gradient Boosting model could welly handle sharp discontinuities, and it can capture sudden changes of the feature value. Their results showed a Gradient Boosting model when facing a short-term prediction; it is a promising algorithm if the input has a complex nonlinear relationship.

CHAPTER 3 RWA: A REGRESSION-BASED SCHEME

As mentioned previously, the flight price has already become the main factor which affects people's decision on their journey. However, the current mature ticket price prediction system does not provide a precise way to predict unpublished tickets price, and the proposed methods are not accurate enough. Hence, in this chapter, we first introduced our data source, data processing, feature extraction, and selection. Then we introduced four XGboost, Random Forest, Bagging regressor, and Gradient Boosting tree algorithm. After that, the final approach RWA algorithm was proposed. Figure 3.1 shows the whole process of our method.

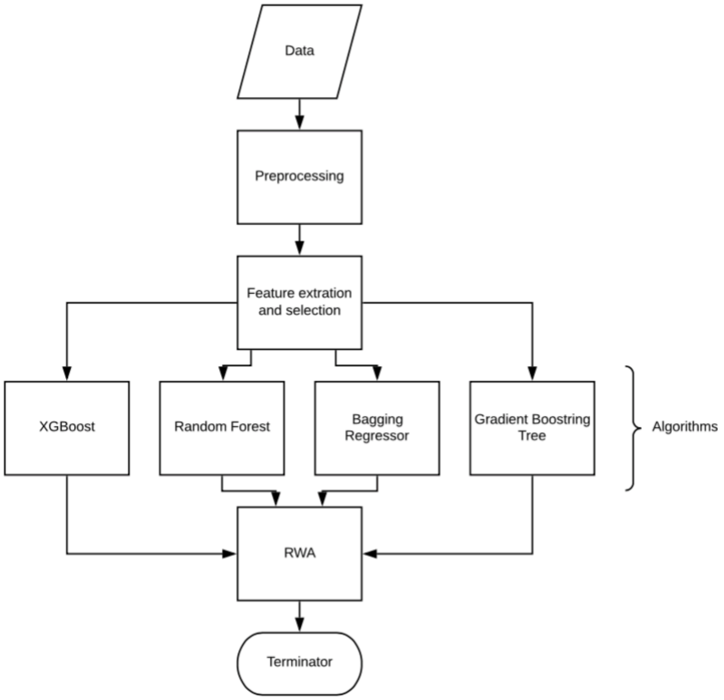

Figure 3. 1 The process of our method

3.1 DATA SOURCE AND COLLECTION

To collect the data, we use an auto script which can perform searching flight information on the Expedia [49] like a human being. After the searching result is shown, it can collect the information on the webpage and store it in a CSV file. During our whole data collection process, roundtrip tickets of Air Canada were collected. Five Canadian cities, which are Halifax, Montreal, Toronto, Calgary, and Vancouver, were chosen as departure and arrival cities.

A week's time was chosen as a sliding window for the departure date and arrival date. Thus, the date combination is forty-nine in total. The April dataset's departure date and arrival date is in the week of April 1st 2019, and May 1st 2019 (i.e., The departure date is between April 1st to April 7th 2019, and the arrival date is between May 1st and May 7th 2019.). This part data was collected from February 19th 2019, to March 31st 2019. The September dataset was collected from April 1st 2019 to August 31st 2019, and its departure date and arrival date is in the week of September 2nd 2019, and the week of September 16th 2019. In each data collection period, the auto script runs once a day to do the collection. As the result, we got 2,529,369 pieces of records in total.

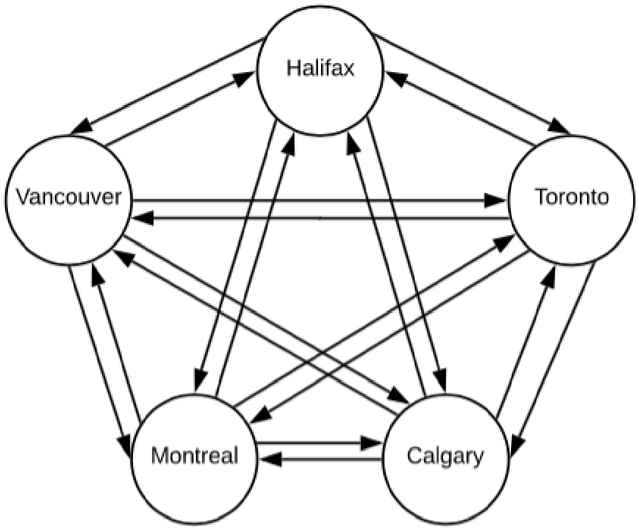

Figure 3. 2 The departure and arrival cities in data collection

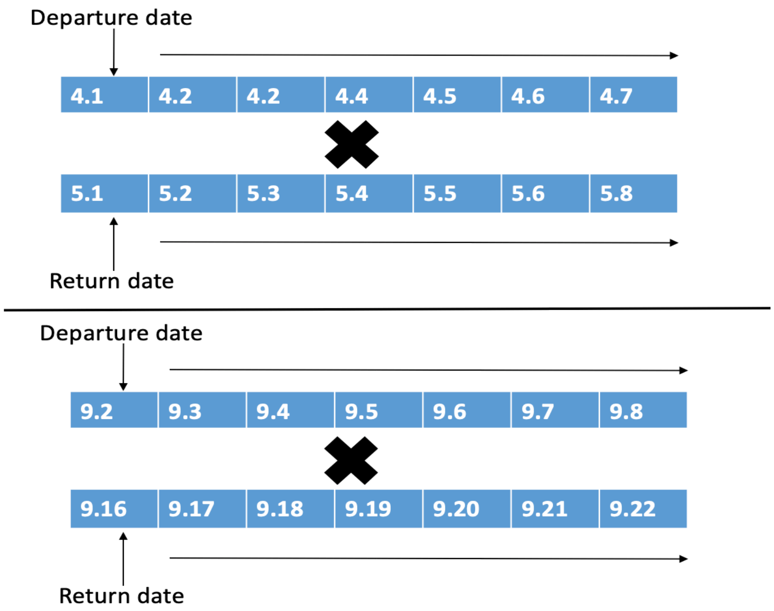

Figure 3. 3 The departure and arrival dates in data collection

3.2 DATA PREPROCESSING

The data we collected has ten features in total which are:

Collect date	The date between collect date and departure date which is in the form of string.
Departure place	The departure place of the airline ticket which is in format of string.
Destination	The arrival place of the airline ticket which is in format of string.
Departure date	The departure date of the airline ticket which is in format of string.
Return date	The return date of the airline ticket which is in format of string.
Departure time	The departure time of the departure day which is in format of "HH:MM"
Arrival time	The arrival time of the arrival day which is in format of "HH:MM"
Airline company	The airline company in charging of the ticket which is in format of string.
Duration	The duration of the total time cost on the first trip which is in format of "HH:MM".

Stops	The total number of stops that the first trip has which is in format of integer.
Layovers	The total stopping time which is in format of “HH:MM”.

Table 3. 1 The original collected features

One important thing is that Expedia has a character, which is when a user tries to buy a ticket on it; the very first price information is always the different first trips’ price plus the cheapest return trip’s price on the user’s selected return date. This means although we do not have any information on the return flight because all the tickets are coming with the same return ticket, it does not have any impact on our result.

Because the departure time and arrival time are in the format of HH: MM. First, we grouped up the departure and arrival time and used the group number as the feature instead of the specific time. We chose three hours as the interval, which means if the departure time or arrival time is between 00:00 to 03:00, we label it with 0 and so on.

Secondly, the duration and layovers are precise time, which means if we keep using it as the labels, there would be too many different labels in those two features. Thus, we transferred the duration and layovers into minutes. Then we took their approximate values based on multiples of ten by using the formula:

$$\mathrm{t} = (\mathrm{Minute} + \mathrm{hours} * 60)\ -\ (\mathrm{Minute} + \mathrm{hours} * 60)\ \%\ 10$$

where t is the assigned time label. For example, if the duration is 342 minutes, we labeled it as 340 minutes.

Thirdly, instead of assigning the null value in the records with the average value of that feature, we remove all the records containing null value because our dataset is large enough, and the records which contain null value only occupy a very small part. After that, we remove data records that contain one or more features that take a very small portion of the whole data set to balance the data. In this case, we observed that the total amount of records with the feature “stops over 2” is way too smaller than others, so these records are removed. The last thing is because the regressor cannot recognize string type value, and for each feature with string type value, the kinds of their labels are finite. To use these features for making prediction, integers need to be used to represent different string values.

From Pritscher and Feyen’s study, we can know that the days between purchase data (collect date) and the departure date have a significant impact on the ticket price [7]. Hence we consider the collect date feature as the most important feature. And the dataset which has:

$$\text{departure date} - \text{collect date} > \text{key days}$$

and the dataset which has:

$$\text{departure date} - \text{collect date} < \text{key days}$$

might have different characters. This lead to our solution which is splitting the dataset according to slope of the line of the average of the price in each day. By the Algorithm 3.1, the key value that we get is 27.

```
ALGORITHM 3.1
Initialize $\boldsymbol{p_{1\dots n}}$ as the average price of each day
Initialize $\boldsymbol{d_{1\dots n}}$ as each day
Initialize k as 1
Initialize result as 0
Initialize max as 0
FOR k FROM 1 to n
    let $\boldsymbol{slope_0}$ equals $(\boldsymbol{p_k} - \boldsymbol{p_1})/\ (\boldsymbol{d_k} - \boldsymbol{d_1})$
    let $\boldsymbol{slope_1}$ equals $(\boldsymbol{p_n} - \boldsymbol{p_k})/\ (\boldsymbol{d_n} - \boldsymbol{d_k})$
    IF $\boldsymbol{slope_1} - \boldsymbol{slope_0}$ is greater than max
        let max be the result of $\boldsymbol{slope_1}$ minus $\boldsymbol{slope_0}$
        let result be k
    END IF
END FOR
RETURN result
```

Figure 3.4 shows the relationship between days between collect date and departure date and the price. By observing the figure, it is easily to get the same result as we got from Algorithm 3.1.

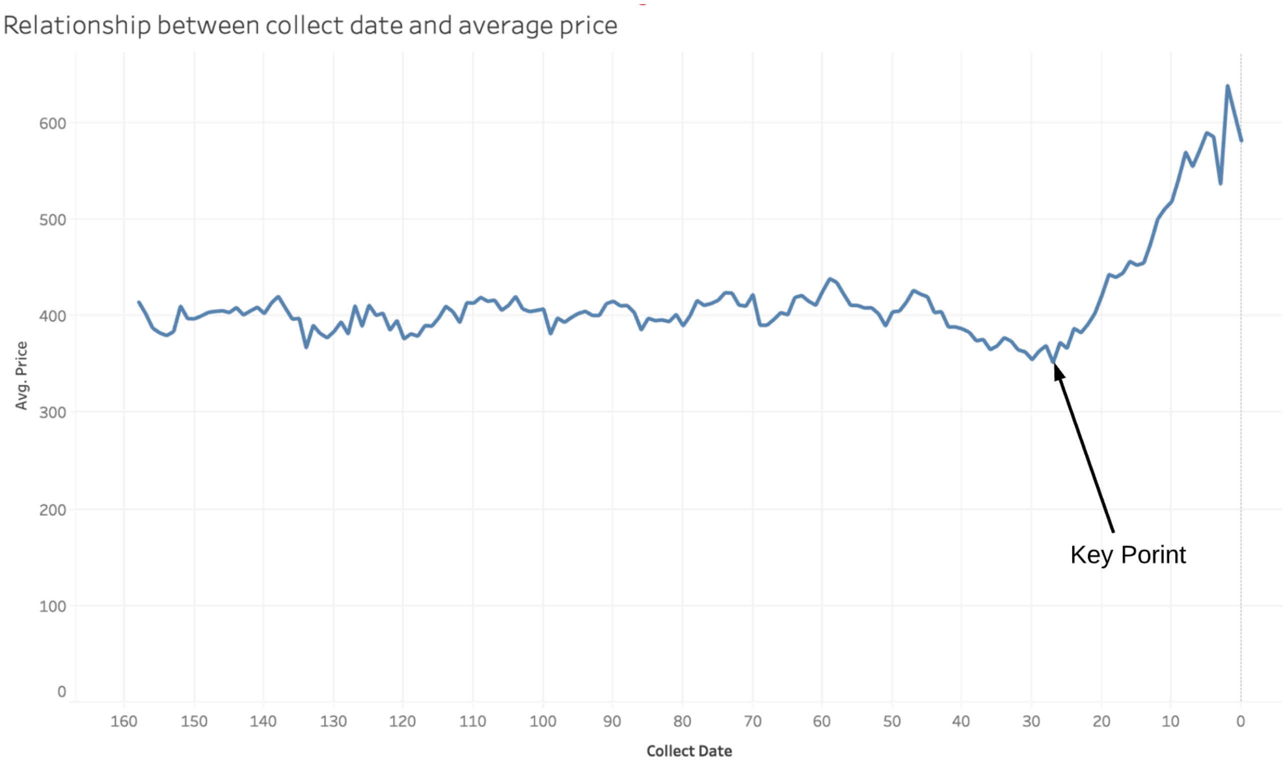

Figure 3. 4 The Relationship between collect date and Avg. price

Therefore, we manually split the dataset into two parts by the key point. So far, we have three datasets that are less than 27 days' dataset (LD), greater than 27 days' dataset (GD), and whole dataset (WD). Correspondingly, there should be three models, which are for WD, for GD, and for LD. Comparing with the old method, which in our case is using make the prediction, we consider:

$$P_1 = \text{the best result of } (M_1, M_2)$$

$$P_2 = \text{the best result of } (M_1, M_3)$$

where P_1 is the predict result of GD and P_2 is the predict result of LD.

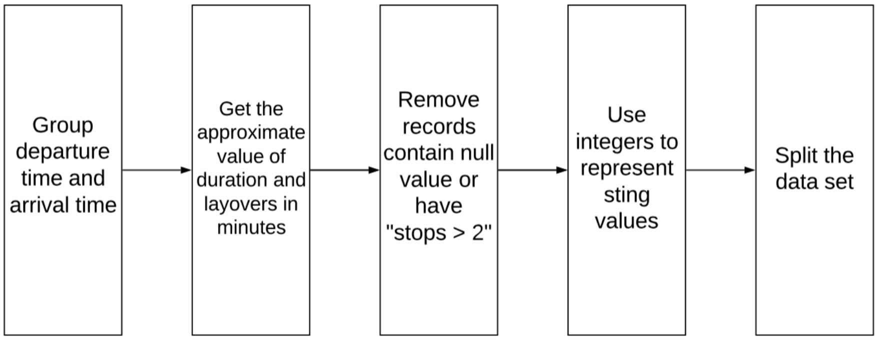
Group departure time and arrival time
Get the approximate value of duration and layovers in minutes
Remove records contain null value or have "stops > 2"
Use integers to represent sting values
Split the data set

Figure 3. 5 The process of data processing

3.3 FEATURE EXTRACTION AND SELECTION

3.3.1 Feature Extraction

As we all know from our daily experience of buying airline tickets, there are not only that eleven features affecting the price, but also some other factors such as holiday or not, rain or not, etc. However, Tziridis et al. study proved that weather and holiday have a slight influence on the final price. Moreover, he shows that the "days before departure" is the most important feature overall. In our case, we are focusing on round-trip case, which means we need not only the days before departure but also a feature to represent the days before the return date. So we added one more feature, which is the days between departure date and return date.

Secondly, it is common sense that the flight cost is highly related to the distance between departure place and arrival place. From [4] [5] [6], we can easily know that the airline companies will make their profit to cover the cost. In [49], the author indicated that gasoline is the main cost of airlines. Hence, we consider another feature, which is the geographical distance between two places.

Thirdly, because we collect our data in two separate periods, April and September, we add a month feature to represent the month of the journey.

Days	The date between departure date and return date which is in the form of integer.
Distance	The geography distance between departure place and arrival place which is in the form of integer.
Month	The month of when the departure date in which is in form of integer.

Table 3. 2 The extracted features

3.3.2 Feature Selection

Phase one of feature selection is about removing redundant features of our dataset. From all the features that we obtain, we can notice that there are two subsets $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$ of all the features:

$$\boldsymbol{w_1} = \{\boldsymbol{Departure\ date, Return\ date, Days}\}$$

$$\boldsymbol{w_2} = \{\boldsymbol{Departure\ place, Arraival\ Place, Distance}\}$$

Let:

$\boldsymbol{e_1}$ and $\boldsymbol{e_2}$ be any two elements in $\boldsymbol{w_i (i = 1, 2)}$

$$\boldsymbol{e_3 = w_i \sim \{e_1, e_2\}}$$

It is easy to observe that $\boldsymbol{e_3}$ could always be calculated by $\boldsymbol{e_1}$ and $\boldsymbol{e_2}$. That makes $\boldsymbol{e_3}$ become a redundant feature. So it means removing one feature from $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$ may lead to a better result. By the same logic, like Tziridis et al. work, we can test the impact on the final result of removing different features. However, we cannot tell the influences on the final result of removing one element from $\boldsymbol{w_1}$ and removing one element from $\boldsymbol{w_2}$ are independent. Hence, we add one more empty feature $\boldsymbol{\varepsilon}$ to each subset and do a brute force to find the feature to be dropped combination in $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$, which is showing in Algorithm 3.2.

$$\boldsymbol{w_1} = \{\boldsymbol{Departure\ date, Return\ date, Days, \varepsilon}\}$$

$$\boldsymbol{w_2} = \{\boldsymbol{Departure\ place, Arraival\ Place, Distance, \varepsilon}\}$$

The second phase is that after dropping $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$, we did an iteration over the remaining feature set $\boldsymbol{w_3}$. In each round, we test the one feature's influence on the result. Moreover, at the end the feature which has the worst negative impact on the result need to be removed. As mentioned before We cannot prove dropping one feature absolutely have no influence on the other one. So we cannot drop every single feature, which has a negative impact on the prediction result. However, phase one and phase two are independent of each other. This is showing in Algorithm 3.3.

```
ALGORITHM 3.2
Initialize $\boldsymbol{f_1}$ as null
Initialize $\boldsymbol{f_2}$ as null
Initialize $\boldsymbol{w_{1\dots n}}$ as total feature set
Initialize A as zero
Initialize $\boldsymbol{w_1}$ as [Departure date, Return date, Days, null]
Initialize $\boldsymbol{w_2}$ as [Departure place, Return place, Distance, null]
FOR EACH element e IN $\boldsymbol{w_1}$
    FOR EACH element d IN $\boldsymbol{w_2}$
        drop e from $\boldsymbol{w_{1\dots n}}$
        drop d from $\boldsymbol{w_{1\dots n}}$
        IF A is lower than current prediction accuracy
            let A equals to current prediction accuracy
            let $\boldsymbol{f_1}$ equals to e
            let $\boldsymbol{f_2}$ equals to d
        END IF
    END FOR
END FOR
RETURN [$\boldsymbol{f_1}, \boldsymbol{f_2}$]
```

```
ALGORITHM 3.3
Initialize $\boldsymbol{f}$ as null
Initialize $\boldsymbol{w_{1 \dots n}}$ as total feature set
Initialize $\mathbf{A}$ as zero
FOR $\boldsymbol{i}$ FROM 1 TO n:
    Drop $\boldsymbol{w_i}$ from $\boldsymbol{w}$
    IF $\mathbf{A}$ is lower than current prediction accuracy
        let $\mathbf{A}$ equals to current prediction accuracy
        LET $\boldsymbol{f}$ equals $\boldsymbol{w_i}$
    END IF
END FOR
RETURN $\boldsymbol{f}$
```

3.3.3 RWA: Regression-based Weighted Average

This algorithm was inspired by Xia et al. study. In their research, after getting the results by using different mature algorithms, they noticed some algorithms prediction values are always greater than the true values and some are always less than the true values [51]. So they simply did a math average:

$$\mathbf{y} = \frac{\sum_{i=1}^{n} \boldsymbol{f_i(x)}}{\boldsymbol{n}}$$

where y is average prediction value of each record and $\boldsymbol{f_i(x)}$ is the prediction value of each algorithm.

RWA is aiming to find the weighted average of the four based algorithms. The equation is as follow:

$$\boldsymbol{y'} = \boldsymbol{W_0 f_0(x)} + \boldsymbol{W_1 f_1(x)} + \cdots + \boldsymbol{W_n f_n(x)}$$

which can be write:

$$\boldsymbol{y'} = \sum_{i=1}^{n} \boldsymbol{W_i f_i(x)}$$

where $\boldsymbol{W_i}$ is the weight of the input, $\boldsymbol{f_i(x)}$ is the prediction value of each algorithm and $\boldsymbol{y'}$ is the final result.

Because when it comes to math average, we can have

$$\boldsymbol{W_1 = W_2 = \cdots = W_n = 1/n}$$

This means weighted average can always equal or better than math average as long as the best weights are found.

However, if by using brute force method to find the best weights for our model, the time cost would be too much. Suppose S is the total number of weights that need to be tried for $\boldsymbol{f_i(x)}$. We can have the time complexity is: $\mathbf{O}(\boldsymbol{s^{n-1}})$ which is $\mathbf{O}(\boldsymbol{s^{n}})$. That time complexity is unacceptable.

Linear regression is a common algorithm in prediction problems [52]. The principle of it is assigning parameters to the input values (i.e., feature), and its aim is to find the best set of parameters of the equation by using a greedy algorithm. The linear regression algorithm's mathematical expression is:

$$\mathbf{g(x)} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{W_i X_i}$$

where y is the prediction value and $\boldsymbol{X_i}$ is the feature value. In our case, we make the output value of the four algorithms as the input $\boldsymbol{X_i(i = 1 \ldots 4)}$ (i.e., The algorithms prediction values will be used as features of Linear regression). Then the final output would be our prediction result. In other words, we use linear regression to assign weights to base algorithms' result to find a curve which can fit the actual curve of the air ticket price most.

As mentioned before, ensemble methods can always be effective in flight price prediction problem. Among all the published ensemble algorithms, Bagging regressor and Random Forest, which have been proved that they have a good operation time and accuracy. By testing the performance of all the ensemble methods, XGboost and Gradient Boosting tree have an acceptable running time and accuracy. So in the final, XGboost, RF, BR, and GBT were chosen as our base algorithms.

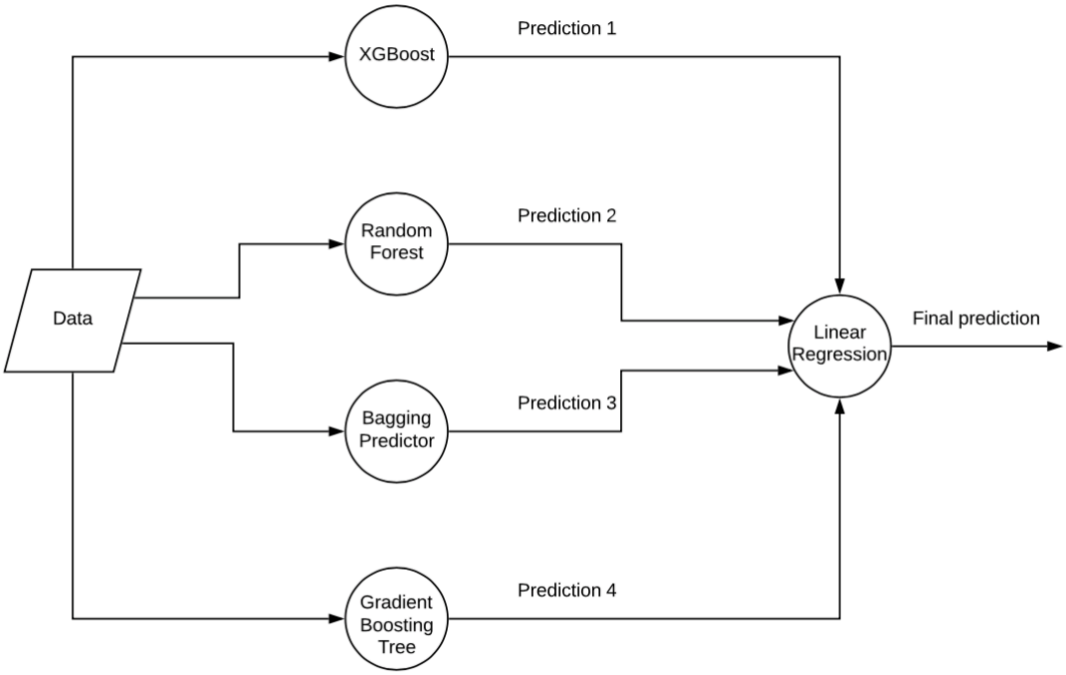

Figure 3. 6 The RWA model

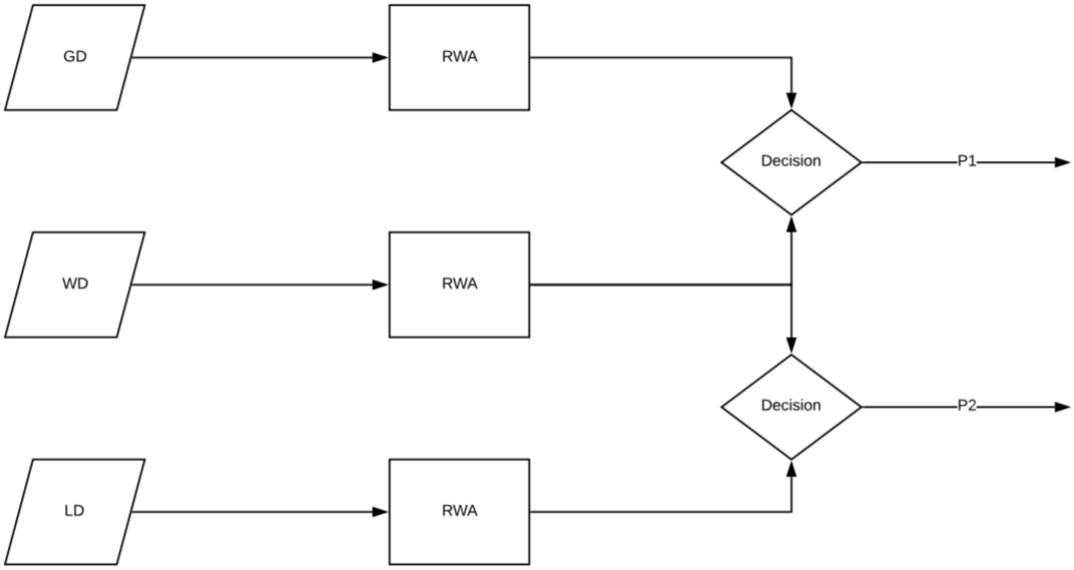

Figure 3. 7 The Slope-based RWA model

Combining with the previous data split method, our final Slope-based RWA model is showing in Figure 3.7.

CHAPTER 4 EXPERIMENTAL RESULTS

Since our strategy is clear, we split the experiment into two phases; phase one is about dropping redundant features in that two feature sets. Phase two is about once after removing a redundant feature. We dropped one more feature, which has a negative impact on the prediction result.

During the whole process, we recorded the performance of all five algorithms (i.e., four base algorithms and RWA). MSE (Mean squared error) was chosen as our metric. The expression is showing as follow:

$$\mathbf{y} = \frac{\sum_{i=1}^{n}(y_i' - y_i)^2}{n}$$

where y_i' is the prediction value for each single record and y_i is the true value. Comparing with mean absolute error which is:

$$\mathbf{y} = \frac{\sum_{i=1}^{n}|y_i' - y_i|}{n}$$

MSE can magnify the error result which can make it clearer to compare our RWA algorithm with other four.

4.1 EXPERIMENTAL RESULTS

4.1.1 Phase One

In phase one's experiment, we perform Algorithm 3.2, aiming to find the redundant features which have a negative impact on the result. In each step, all five algorithms' results are shown.

4.1.1.1 WD Results

Figure 4.1 to Figure 4.4 show the results of phase one which is feature selection for the WD. Because in algorithm 3.2, we use a brute force algorithm to find redundant features in feature subset w_1 and w_2. In each figure, there are four different results of dropping features in w_1 and

$\boldsymbol{w_2}$. In the whole process of the experiment, the result of RWA and other four base algorithms are recorded in order to make a comparison between our RWA and other four base algorithms. The number on the top of the bar chart is mean squared error.

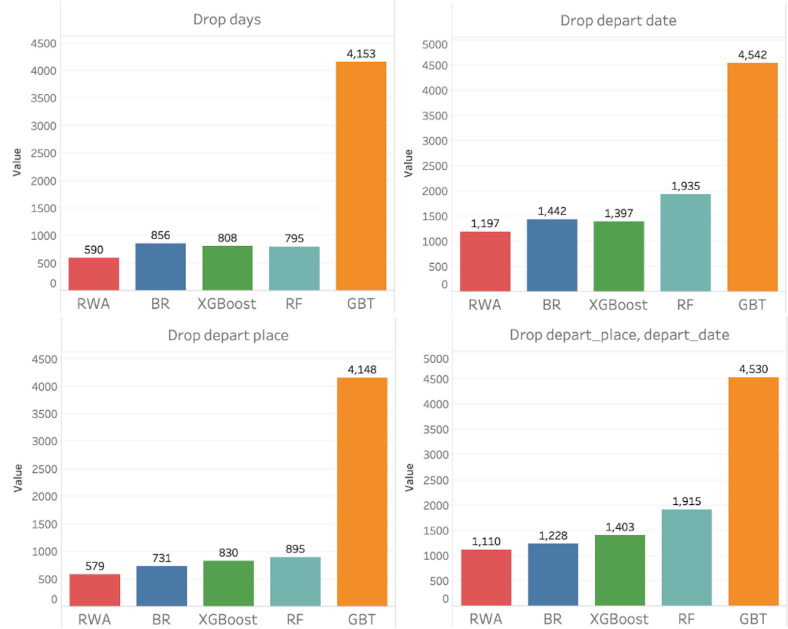

Figure 4. 1 Remove redundant feature result one of WD

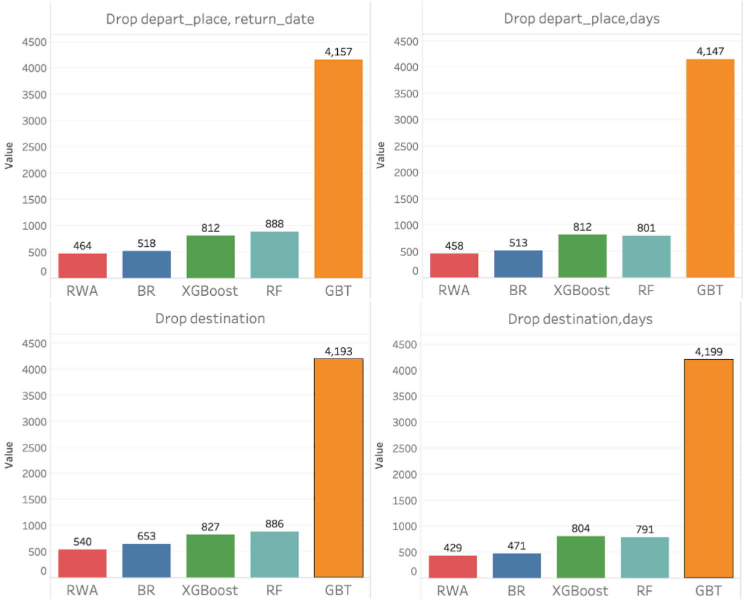

Figure 4. 2 Remove redundant feature result two of WD

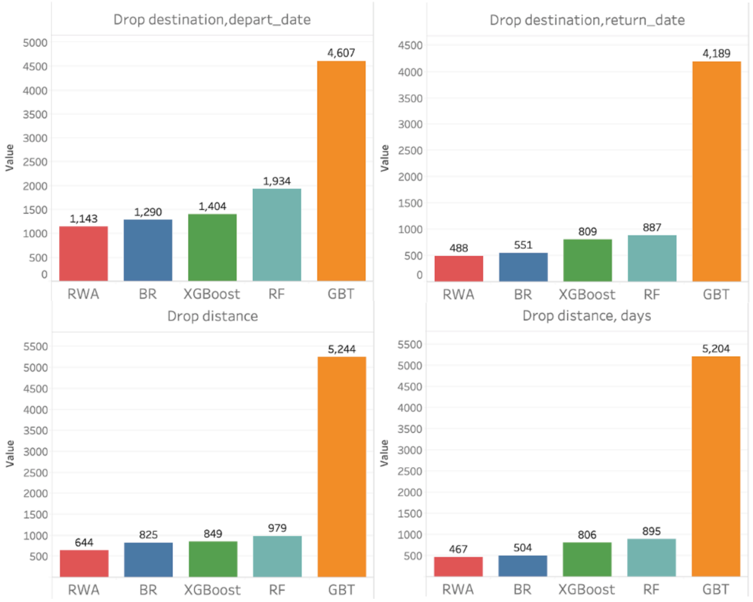

Figure 4. 3 Remove redundant feature result three of WD

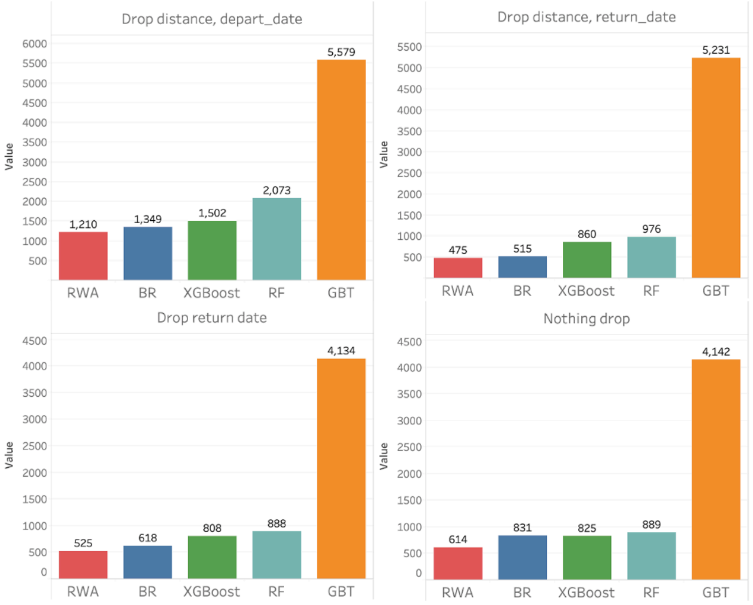

Figure 4. 4 Remove redundant feature result four of WD

4.1.1.2 GD Results

Same as WD part, we perform the same algorithm for GD and in each step, we record the result of the algorithms. Figure 4.5, 4.6, 4.7 and 4.8 shows the results of GD.

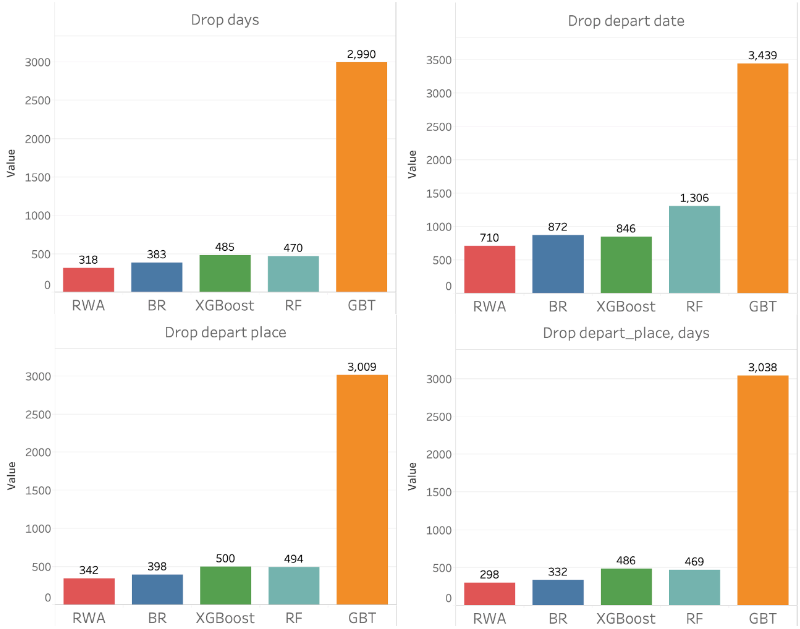

Figure 4. 5 Remove redundant feature result one of GD

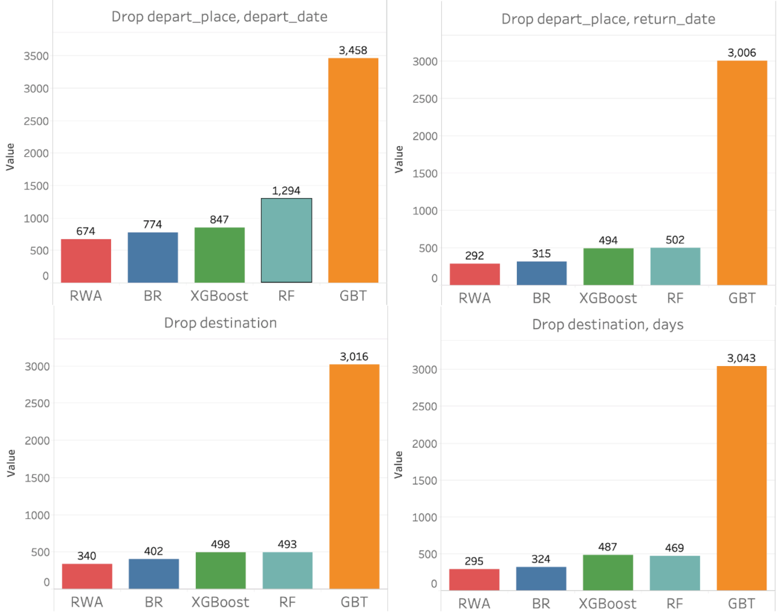

Figure 4. 6 Remove redundant feature result two of GD

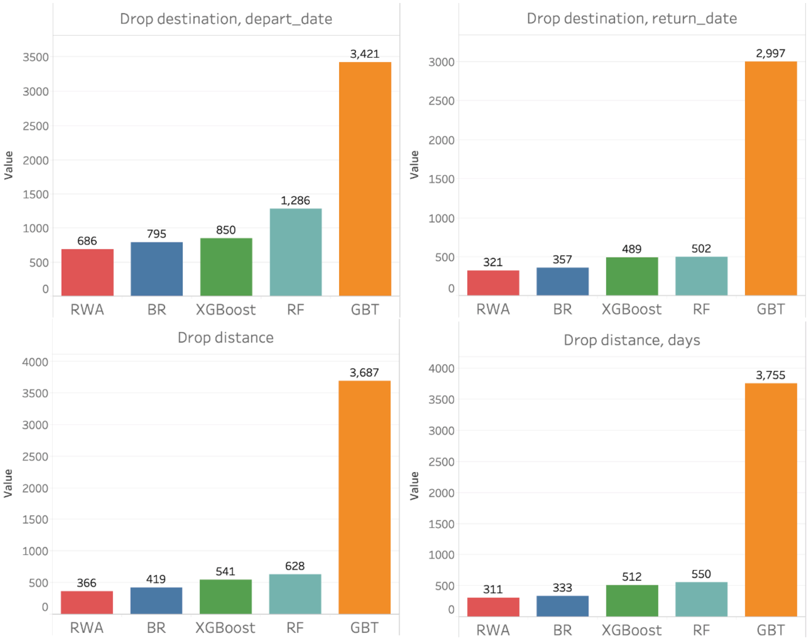

Figure 4. 7 Remove redundant feature result three of GD

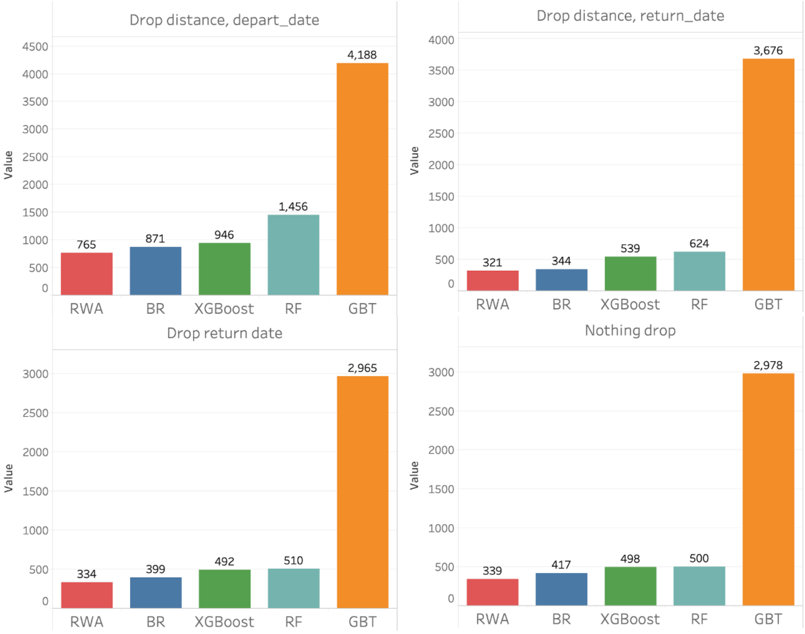

Figure 4. 8 Remove redundant feature result four of GD

4.1.1.3 LD Results

The following figures show the result of Algorithm 3.2 of LD.

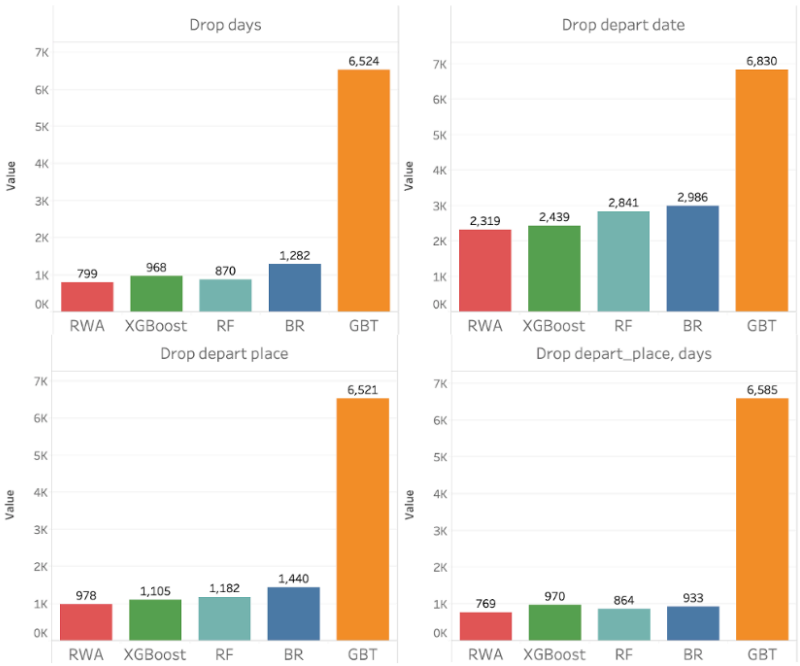

Figure 4. 9 Remove redundant feature result one of LD

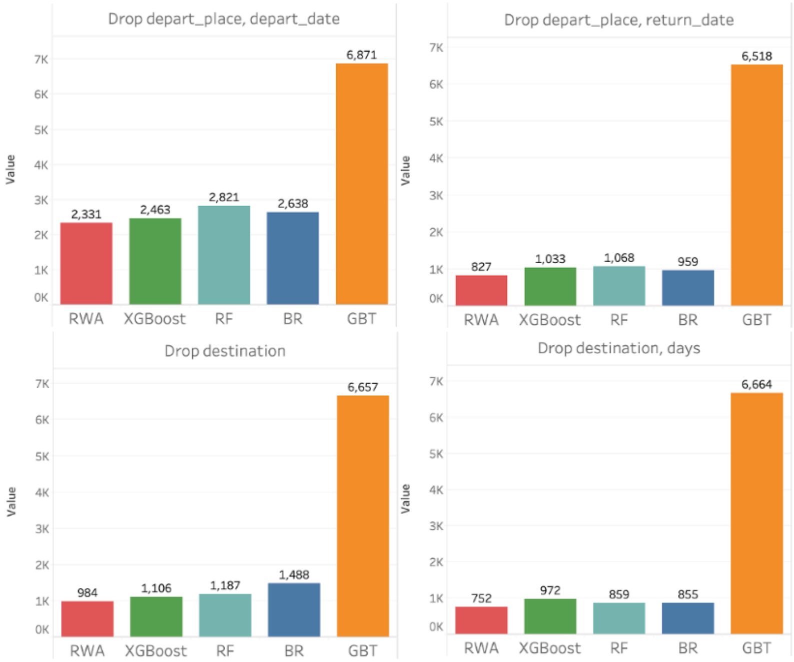

Figure 4. 10 Remove redundant feature result two of LD

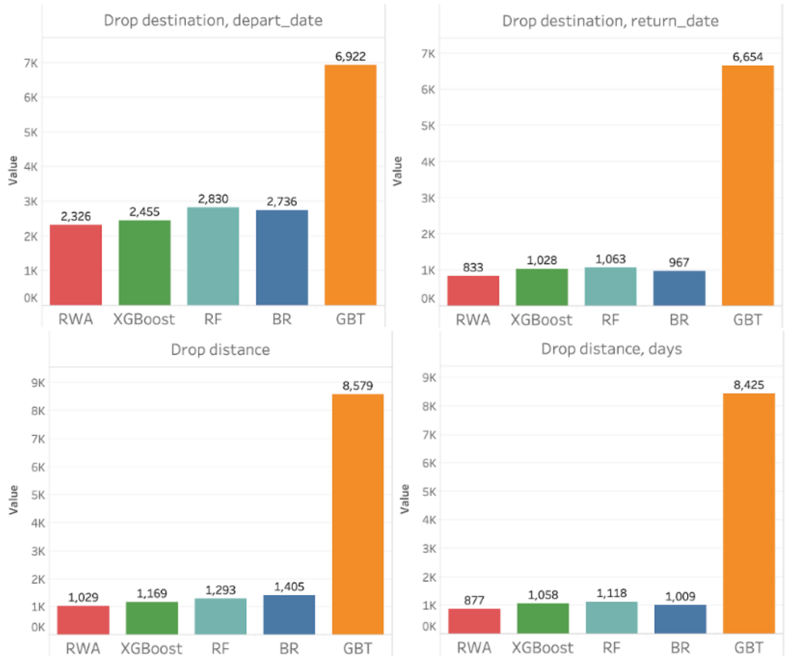

Figure 4. 11 Remove redundant feature result three of LD

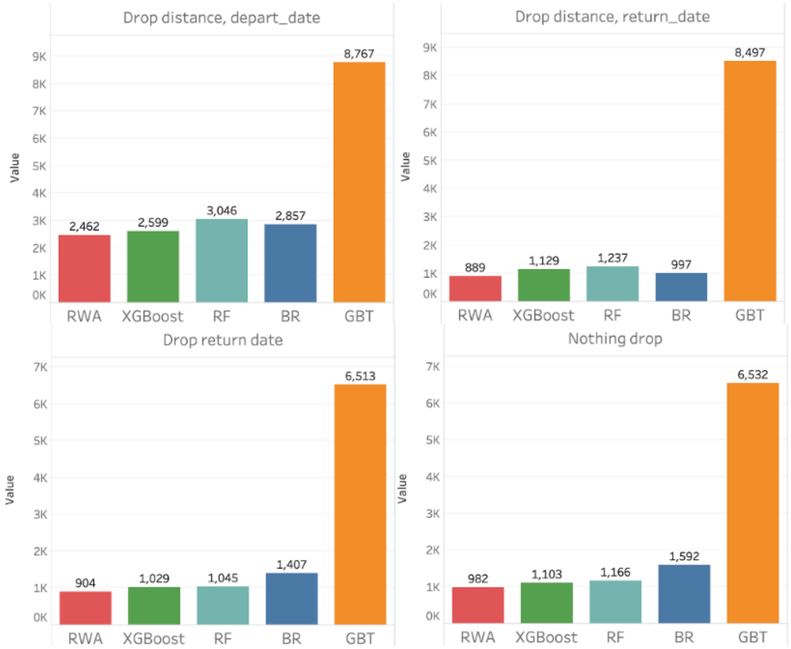

Figure 4. 12 Remove redundant feature result four of LD

The summarized result of these figures above is showing below:

The dataset	Features going to be removed
WD	Destination & Days
GD	Departure place & Return date
LD	Destination & Days

Table 4. 1 Redundant feature for each dataset

4.1.2 Phase Two

After removing redundant features above for each dataset, in phase two, we performed the Algorithm 3.3. In each figure, the result of one algorithm of dropping every features is showing.

4.1.2.1 WD Results

Figure 4.13, 4.14, 4.15, 4.16 and 4.17 shows this experiment result of WD. The results of dropping every features are sorted in ascending order.

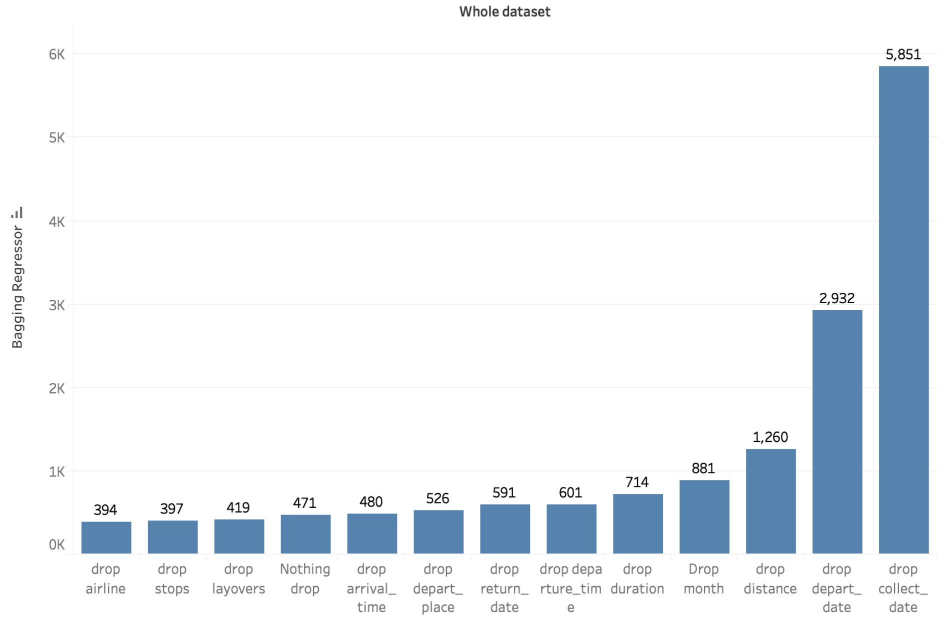

Figure 4. 13 Remove every feature result of Bagging Regressor for WD

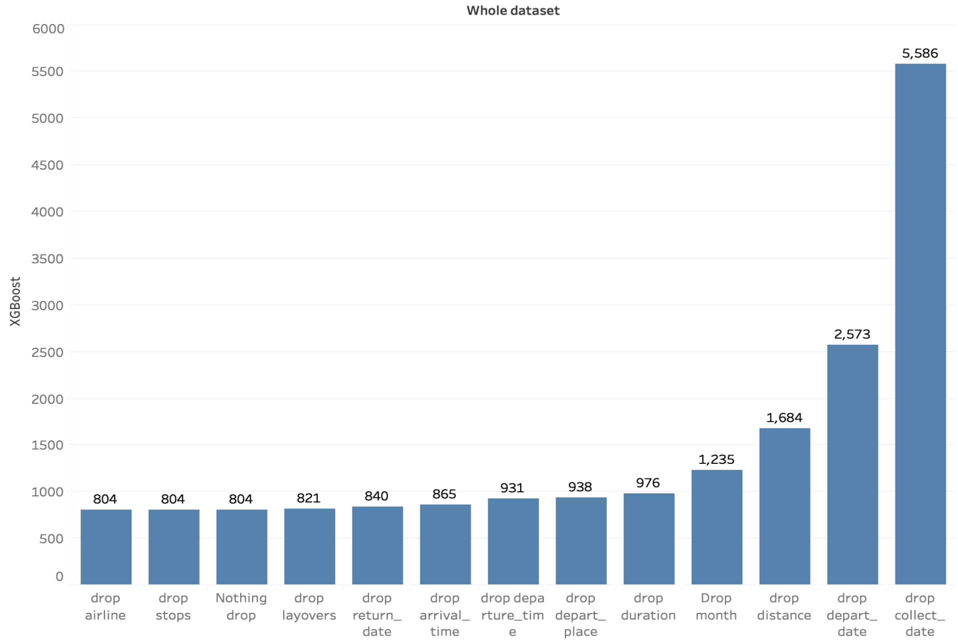

Figure 4. 14 Remove every feature result of XGboost for WD

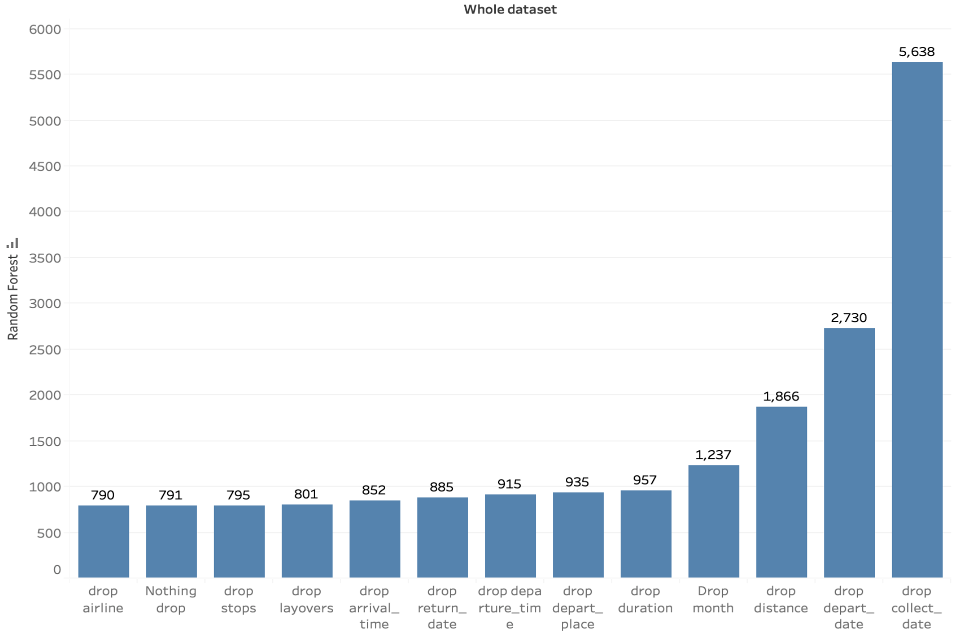

Figure 4. 15 Remove every feature result of Random Forest for WD

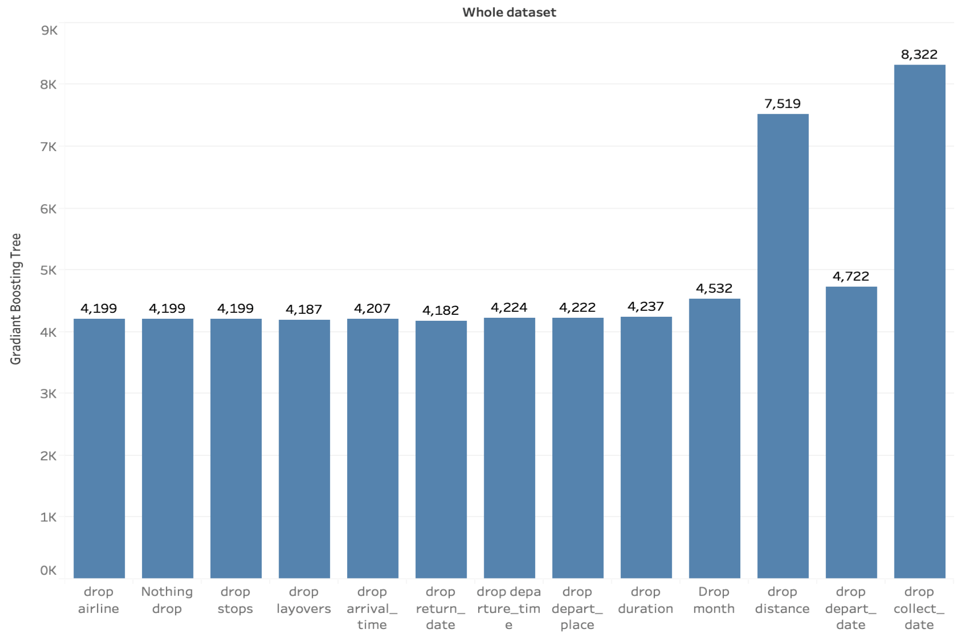

Figure 4. 16 Remove every feature result of Gradient Boosting Tree for WD

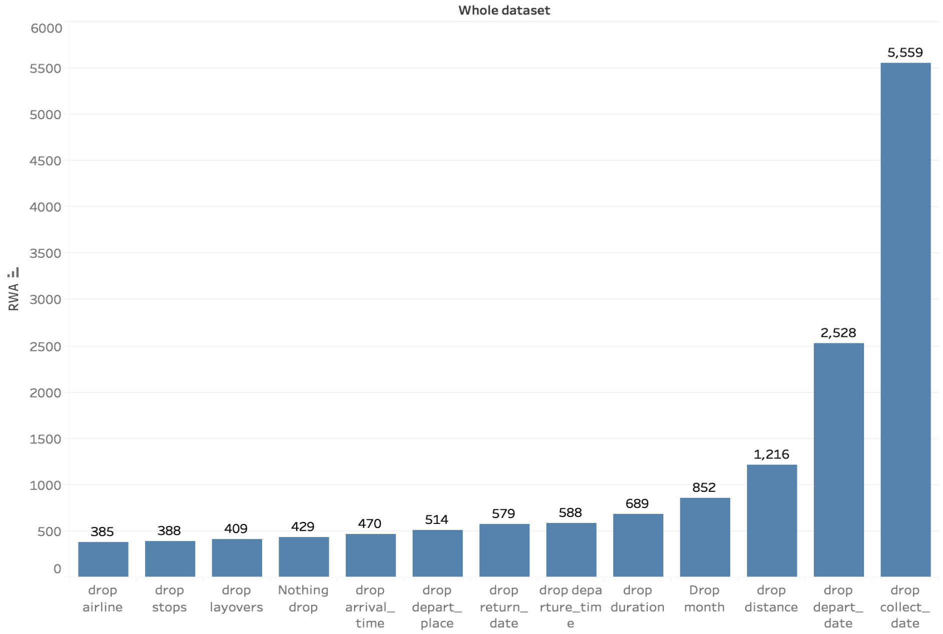

Figure 4. 17 Remove every feature result of RWA for WD

4.1.2.2 GD Results

Figure 4.18, 4.19, 4.20, 4.21 and 4.22 shows the same experiment results of GD.

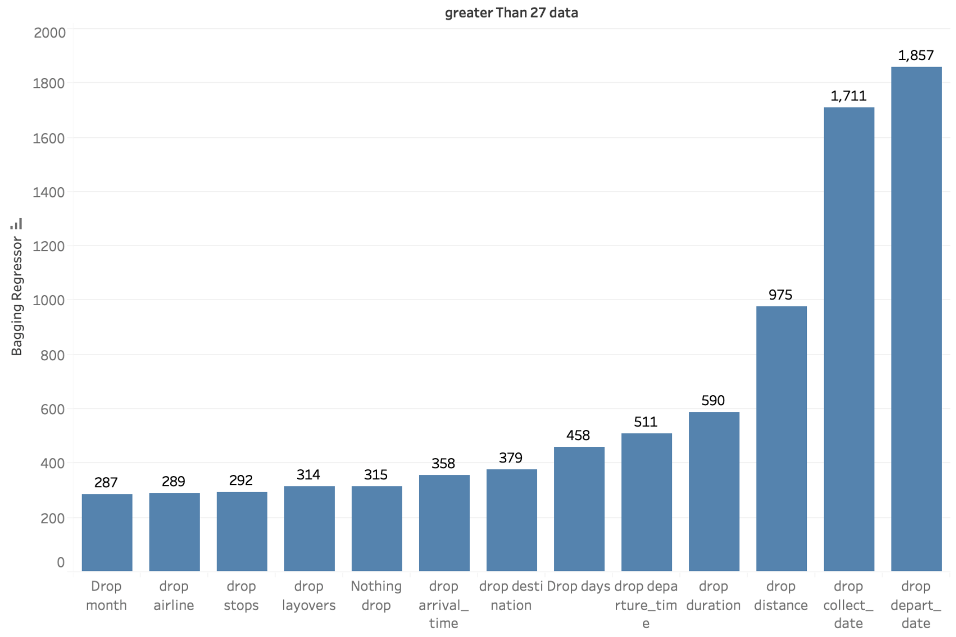

Figure 4. 18 Remove every feature result of Bagging Regressor for GD

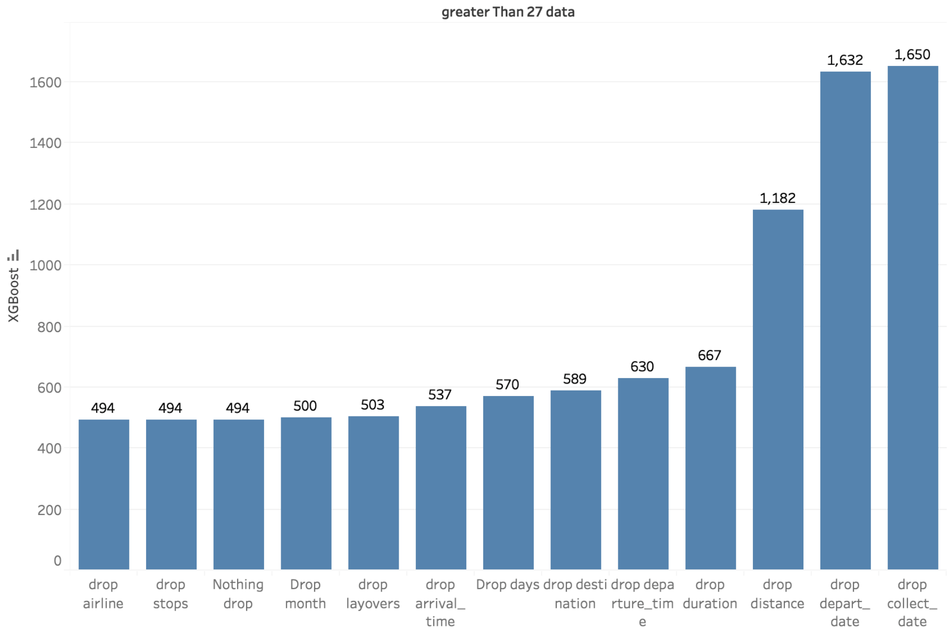

Figure 4. 19 Remove every feature result of XGboost for GD

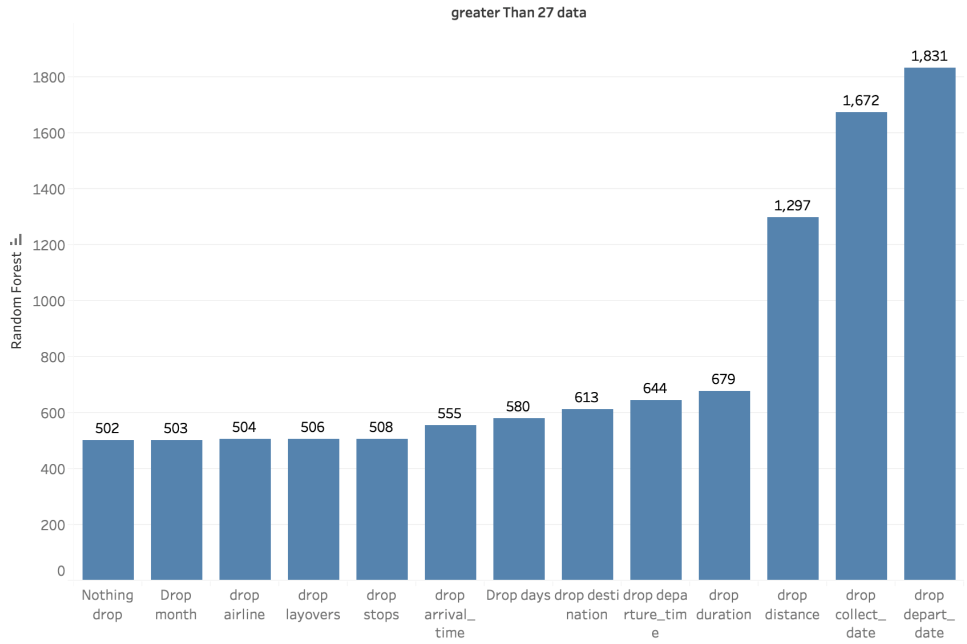

Figure 4. 20 Remove every feature result of Random Forest for GD

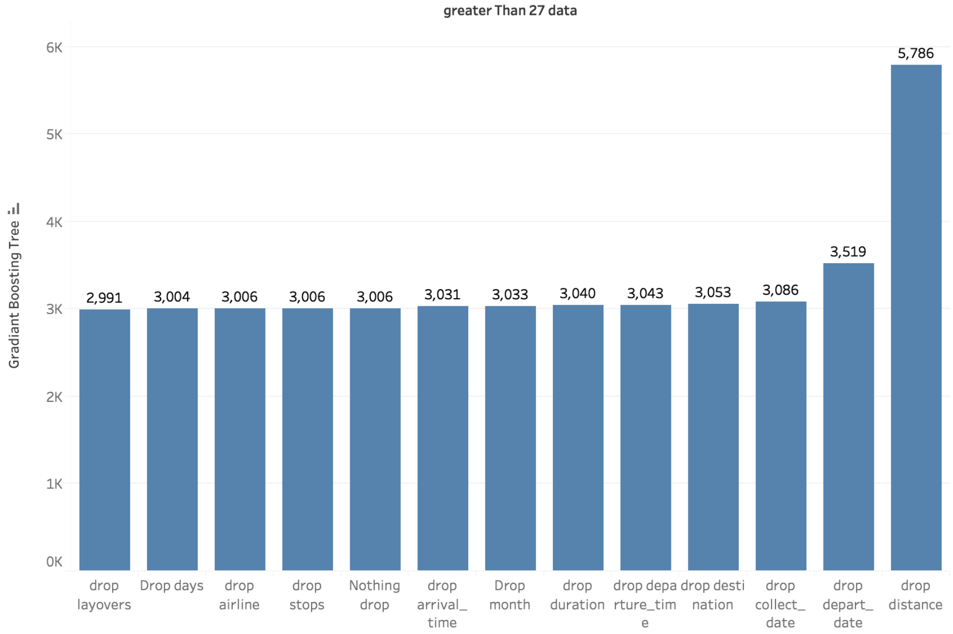

Figure 4. 21 Remove every feature result of Gradient Boosting Tree for GD

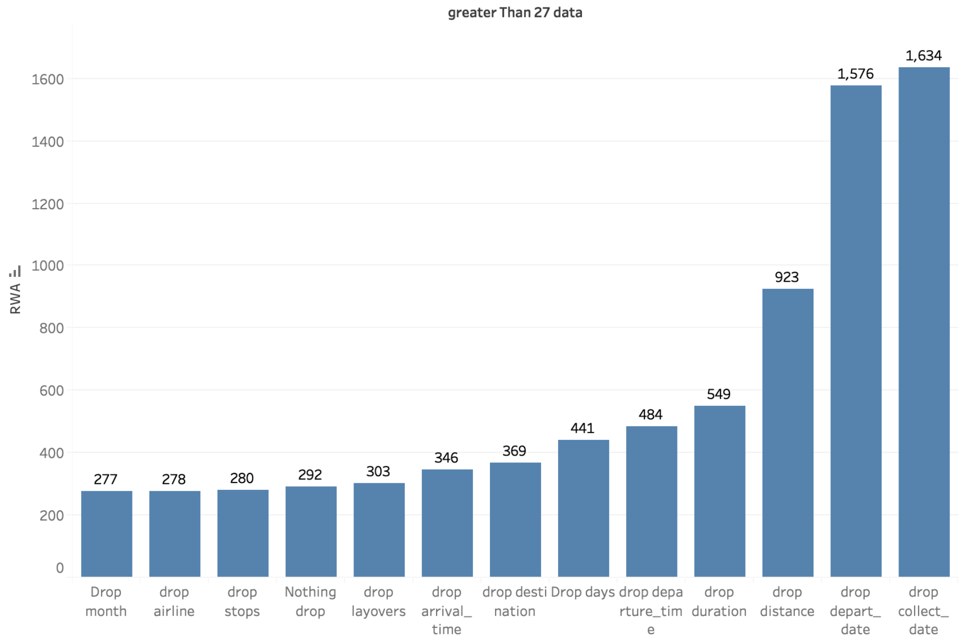

Figure 4. 22 Remove every feature result of RWA for GD

4.1.2.3 LD Results

Same as WD and LD, the figures below show the Algorithm 3.3 results of LD.

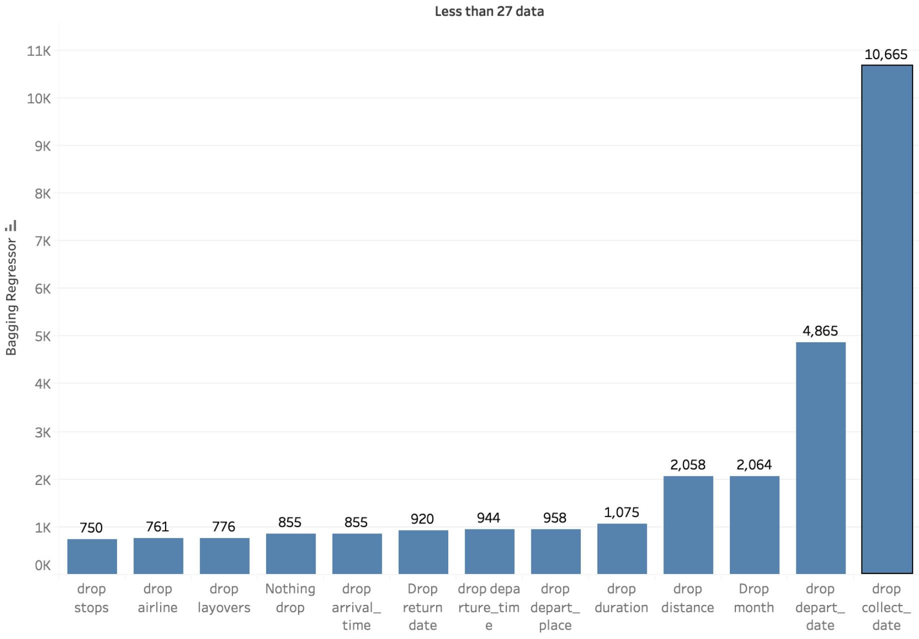

Figure 4. 23 Remove every feature result of Bagging Regressor for LD

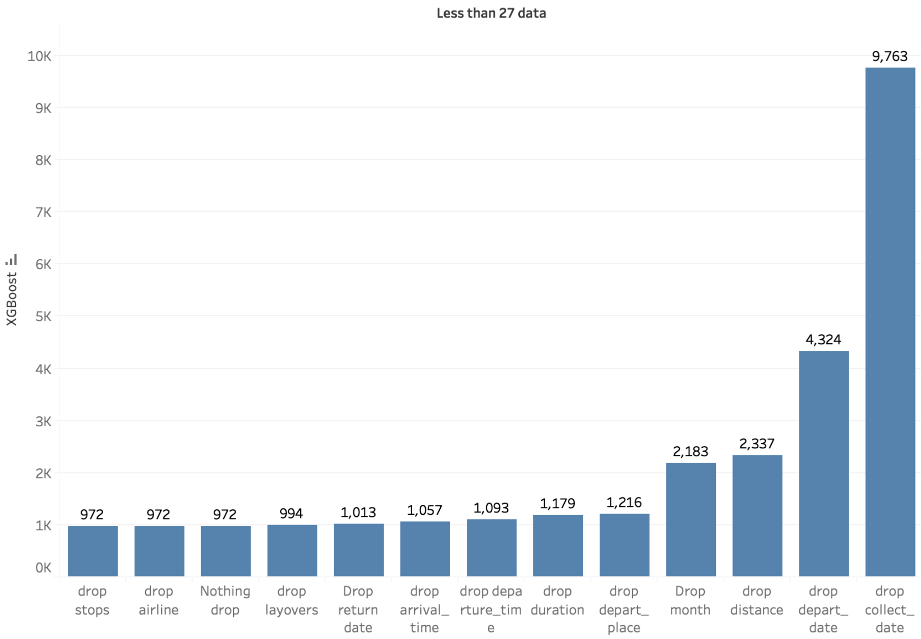

Figure 4. 24 Remove every feature result of XGboost for LD

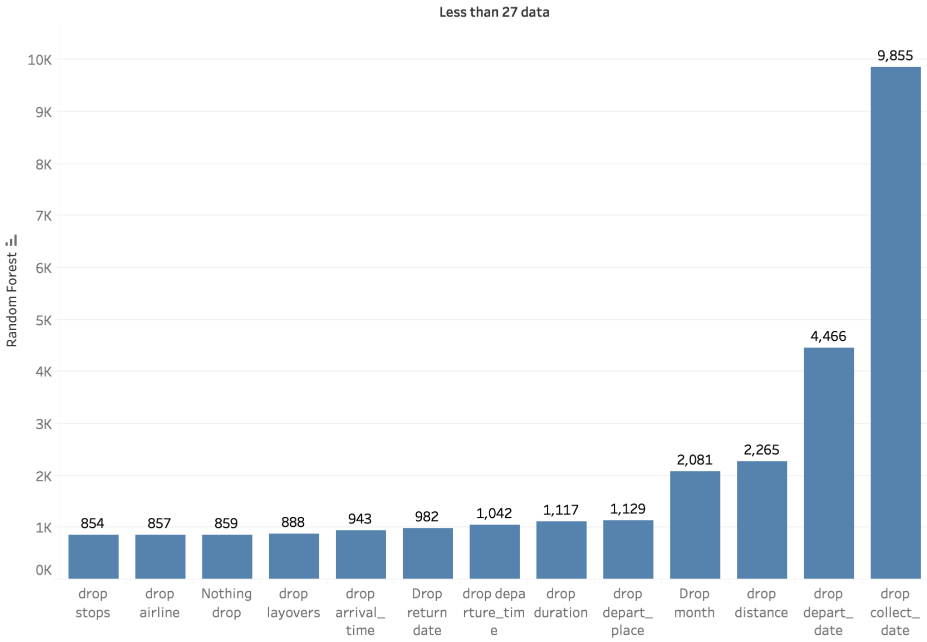

Figure 4. 25 Remove every feature result of Random Forest for LD

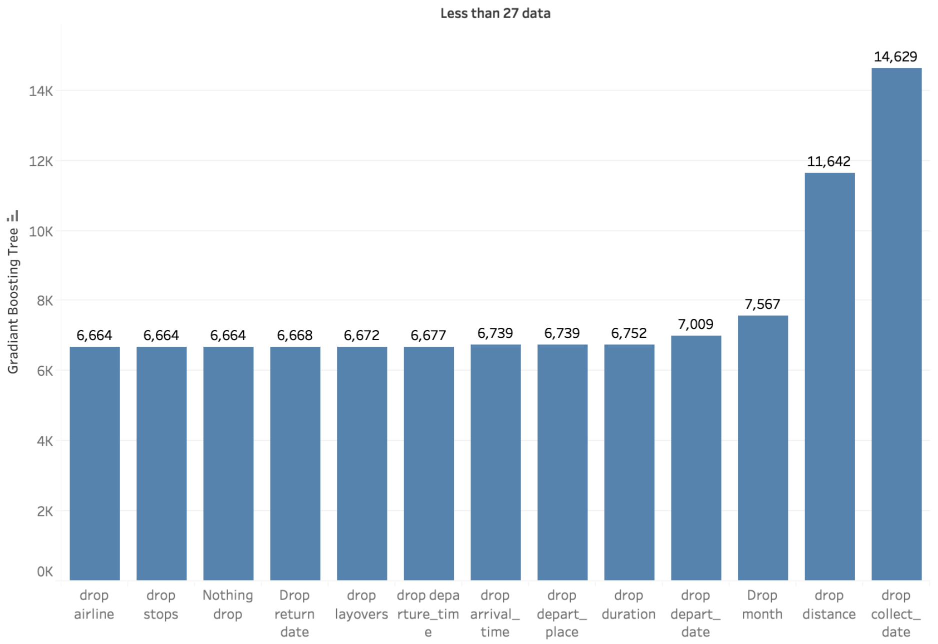

Figure 4. 26　Remove every feature result of Gradient Boosting Tree for LD

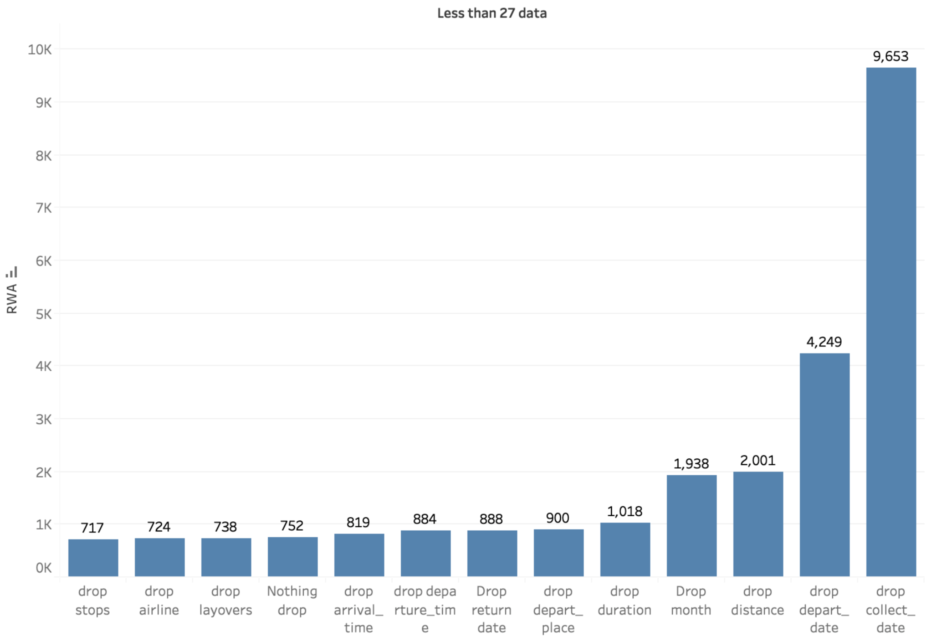

Figure 4. 27　Remove every feature result of RWA for LD

To the end of the experiments, the summarized result is showing below:

The dataset	Features going to be removed
WD	• Destination • Days • Airline
GD	• Departure place • Return date • Month
LD	• Destination • Days • Stops

Table 4. 2 Remove feature result for each dataset

After these two phase experiments, we have the best feature sets for each dataset. We make them go through our model again to get the predict price curve.

By using the best feature sets we got, we can have the final MSE result. After that, we calculate the average price of WD, GD and LD to get the percentage error rate. This is showing in table 4.3.

The dataset	Mean squared error	Percentage error rate
WD	385	4.57%
GD	277	4.19%
LD	717	5.77%

Table 4. 3 Result of three datasets

As mentioned before, we need to compare GD with the GD part in WD and compare LD with the LD part in WD. The result is showing in table 4.4 and 4.5:

The dataset	Mean squared error	Percentage error rate
GD	277	4.19%
GD part in WD	280	4.22%

Table 4. 4 Result of comparing GD and GD part in WD

The dataset	Mean squared error	Percentage error rate
LD	717	5.77%
LD part in WD	795	6.10%

Table 4. 5 Result of comparing GD and GD part in WD

4.2 RUNNING TIME

After getting the best feature set of each dataset, we collected the running time of WD, GD and LD for the prediction. And in the experiment, we monitor the running time of all five algorithms. The results are included in Table 4.6 and 4.7.

	XGboost	RF	BR	GBT	Regression	RWA
WD	317.61s	288.47s	386.72s	116.75s	0.18s	1109.73s
GD	291.58s	192.20s	253.17s	57.17s	0.08s	794.20s
LD	64.75s	48.52s	57.62s	16.52s	0.026s	187.44s

Table 4.6 Running Time: Training Phase

	XGboost	RF	BR	GBT	Regression	RWA
WD	15.48s	7.79s	19.95s	0.76s	0.02s	44.00s
GD	13.11s	6.42s	12.33s	0.40s	0.012s	32.27s
LD	2.97s	2.12s	2.99s	0.13s	0.004s	8.21s

Table 4.7 Running Time: Testing Phase

4.3 IN-DEPTH ANALYSIS

We can see from Figure 4.1 to Figure 4.19, our RWA model no matter on which dataset. XGboost, Random Forest, and Bagging regressor all have a similar performance. Although Gradient Boost Regressor shows a little worse result compared with the other four, we need it to be an input for our RWA to neutralize the prediction price curves of multiple algorithms. For example, if, in some days, XGboost, Random Forest, and Bagging regressor's results are all below the real price. In this case, no matter what weight we assign to them, they always cannot get approach to the real price. So we need a "bad performance" algorithm to act as the regulator.

Our phase one's result is showing a brute force search of the best combination of feature set one and feature set two. In GD and LD, distance is always an essential factor. Nevertheless, when it is far from the departure date, the price is highly related to the destination, and it is highly related to the departure place when it is near the departure date. The feature days (how many days between departure date and return date) have less impact on the GD, but it has a strong impact on LD. That is because, in GD, the departure date is far from the collect date and the return date is even further. So the days between collect date and departure date and the days between the return date and collect date are all greater than the key-days. In this case, the feature days would be less important. However, in LD, the days between departure date and collect date is less than the key-days, but the days between the return date and collect date might be greater than it. So it would be important to know the relationship between key-days and the distance between return and collect date. This will cause the feature return date to have more impact on the result than the feature days. The interesting thing is that, in phase one's result, LD and WD have the same feature set to get the best result. That is because the LD's price is more changeable with the different combinations of feature set one and feature set two. To make the result to be best, in the experiment of WD, the demand of LD will be satisfied first.

Our phase two result shows the importance of different features. Without the feature collect date (the days before departure), the prediction result will become way too inaccurate. However, this feature's impact on GD comparing with LD and WD is less. That is because GD's curve is more flat, which is shown in Figure 3.3. Among this flat curve, the collect date can have less influence on deciding how the price goes. That can also explain why the feature days have less influence on the final prediction result than the feature return date in GD. Also, from phase two results, we can notice that departure date and distance also have a strong influence on the prediction result. That can prove two things. First, the day of a week does have a strong impact on the airline price. Second, our hypothesis in section 3.3.1 is verified; that is, the airline company's price strategy is highly related to their cost, which no one has proved before.

On the final stage, the performance of LD has around 10% improvement compared with the LD part in WD. While the performance of GD and GD part in WD has a slight difference. This can verify when we use WD as our dataset, and the model will satisfy the demand of GD first because it takes a larger portion than LD.

Another thing is that, without the feature month, GD will perform better compared with dropping other features. On the other hand, LD will perform worse with this feature than with other features. This can illustrate when the purchase date is far from the departure date; the changing of the price will go the same no matter the travelers decide to depart in April or September. Moreover, in order to make WD, GD, and LD getting the best performance, we should use different feature sets. So our hypothesis, which is the datasets that are split by the key-point, can have different characters that are even different from the whole dataset is correct.

In the previous study on this problem, there is a large portion of work focusing on providing customer buy or wait suggestions or giving prediction on the flight price will fall or rise in the future. Although some researchers have given methodology on making the prediction price available, what they were using is one single model on this problem.

Comparing with the previous methods, we provide a mechanism to combine multiple models. Our algorithm is useful not only on the flight price prediction problem but also can be effective in other problems such as oil prediction, consumer product predictions and etc. The key-point is the chosen of the base algorithms. It is easy to know when the predictive value of the base algorithm has the same trend with the real value, and the weighted average can always have a better or equal performance compared with the base algorithms.

Another thing is, in the case of focusing on the flight prediction, we found a key point of the flight ticket. In the previous work, researchers have already proved that when the collect date is near the departure date, the prediction will become less effective. However, they did not provide a way to address this problem. In our study, from the result, we can tell that by splitting the dataset basing on the key date can make the final result more precise. On the other hand, our dataset only contains April and September flight data, and they have the same key date. However, if the flights in different months have a different key date, following the data split strategy, we should merge the months which have the same key date together and treat it as a whole dataset. Moreover, if the flights in one month that have multiple key points, we should split the data into multiple parts, but we have to guarantee that every part has enough amount data because of the basic rule of applying the machine learning technique.

Apparently, all the algorithms in our model are independent of each other. That is, our RWA algorithm time complexity would be equal to the sum of the time complexity of XGboost, Random Forest, Bagging Regressor, Gradient Boosting trees, and Linear Regression. The training time complexity of XGboost is [34]:

$$\boldsymbol{O}(\boldsymbol{KD}||\boldsymbol{x}||_{\boldsymbol{0}}\boldsymbol{log}(\boldsymbol{N}))$$

where K is the number of trees it generates, D is the depth of each tree, $||\boldsymbol{x}||_{\boldsymbol{0}}$ is the number of records which do not contain any null value and n is the total number of records. Apparently, K and D are constant number and because before running our algorithm, we removed all the records which have null value, $|\boldsymbol{x}_{\boldsymbol{0}}|$ would be equal to $\boldsymbol{n}$ in our case. So the training time complexity is:

$$\boldsymbol{O}(\boldsymbol{Nlog}(\boldsymbol{N}))$$

From [53], we can know the training time complexity of Random Forest is:

$$\boldsymbol{O}(\boldsymbol{VKNlog}(\boldsymbol{N}))$$

Where K is total number of the trees, V is the number of features, and n is the number of the records. Because in our case, we set the K and V to be two constants, the training time complexity would be:

$$\boldsymbol{O}(\boldsymbol{Nlog}(\boldsymbol{N}))$$

As we mentioned in Chapter 2, the only difference between Bagging Regressor and Random Forest is when constructing the trees, the number of features used is different. Thus, the training time complexity of Bagging Regressor is:

$$\boldsymbol{O}(\boldsymbol{Nlog}(\boldsymbol{N}))$$

In [54], the author introduced that the training time complexity for Gradient Boosting trees is:

$$\boldsymbol{O}(\boldsymbol{VN})$$

where N is the number of samples (i.e., the number of records) and V is the number of dimensions (i.e., the number of features). So in our case, the training time complexity for Gradient Boosting trees is:

$$\boldsymbol{O}(\boldsymbol{N})$$

Linear Regression that we used to assign weight has training time complexity of Linear Regression is [55]:

$$\boldsymbol{O}(\boldsymbol{V}^{\mathbf{2}} * (\boldsymbol{N} + \boldsymbol{V}))$$

where k is the number of features. Thus, the training time complexity would be:

$$\boldsymbol{O}(\boldsymbol{N})$$

In our experiment, when it comes to the Linear Regression part, the input data only contains four features, so the running time is very small comparing to other algorithms. Furthermore, the difference of running time in a practical experiment of our ensemble algorithms is because the value tree depth and the number of trees for each algorithm are different. For example, the running time BR is longer than RF is because RF only takes part of the features as the input when constructing the trees.

In summary, our RWA algorithm's training time complexity would be the sum of the time complexity of these algorithms:

$$\boldsymbol{O}(\mathbf{3} * \boldsymbol{Nlog}(\boldsymbol{N}) + \mathbf{2} * \boldsymbol{N})$$

which is:

$$\boldsymbol{O}(\boldsymbol{Nlog}(\boldsymbol{N}))$$

CHAPTER 5 LIMITATIONS AND FUTURE WORK

In this chapter, we present the limitations of the proposed flight prediction scheme and our future work.

6.1 LIMITATIONS

However, our study has several limitations on this problem. First of all, our dataset is focusing on the April and September's airline on Expedia. As we all know, the different agencies can give different prices of one specific airline, because sometimes, traveling agency might provide coupons on some specific airlines. So other agencies' prices might be more closed to the model of airline companies. Furthermore, our data can only illustrate the trend of the air ticket price in April and September, which means we cannot conclude other months' price trends are the same as these two months.

Secondly, airline companies might have different price strategy on internal flights and domestic flights. Hence, our model cannot apply to international flights.

Thirdly, due to our data collection process, it is easy to know that LD's data size is smaller than GD's. That might be the reason which causes the result of LD is worse than the result of GD.

6.2 FUTURE WORK

Focusing on addressing these limitations. Our future work is as follow:

- In the future, we will make comparisons between different data sources. Moreover, we will collect all years' flight information.
- According to [19], there are 26.4 million international passengers in the past year, and the amount is still increasing. So in the future, we will expand our study not only the domestic flights in a whole year but also the international flights.

- To balance the data size of LD and GD. The first thing we can do is we can randomly drop some data in GD to make it has the same size as LD. Alternatively, when the days before departure is less than the key-days, in each day, we can do the data collection more than once a day to increase the size of LD.

CHAPTER 6 CONCLUSIONS

In our research, we proposed a slope-based dataset spitting method and a regression-based weighted average algorithm, RWA, for flight price prediction.

Flight price prediction can be regarded as a time series problem. In the old study, all the studies always treat the data as a whole part. Comparing with their method, we split the data into two parts according to the trend of the price curve and make a hypothesis that the different datasets can have different characters. To get the best performance of each dataset, we should have different feature sets on different datasets.

Then we use a two-phase experiment to get the best feature set for each dataset. In phase one, we use a brute force algorithm to search the feature combination in two redundant feature sets, which has the best performance on our datasets. Moreover, in phase two, we test the importance of different features for each dataset and remove one more feature, which has the worst influence on the prediction result to get the best performance. The result shows that the best feature sets for each dataset are different from each other.

Inspired by Xia et al. work, a weighted average method was also proposed. After getting the prediction result of four mature algorithms, we use linear regression to assign each algorithm's result a weight to get the predicted price approach to the real price. After that, we combined the data split strategy to find the highest accuracy for three datasets.

In the last stage of our approach, we make a comparison between GD and GD part in WD and a comparison between LD and LD part in WD.

Our experiment result shows that the performance of our data split strategy works, and the RWA algorithm's result is better than any base algorithms that we use as the input for the linear regression. Furthermore, it can bring accuracy to over 94%. This means if the flight price is $1,000, on average case, there will only be only a $60 error.

In summary our algorithm is effective on the flight prediction problem and our data split strategy can overcome the problem that when the purchase date is close to the departure date, prediction is less effective. From the result, we can tell when the purchase date is before the key date, using $\boldsymbol{M_1}$ and $\boldsymbol{M_2}$ actually have no difference. But when the purchase date is after the key date, using $\boldsymbol{M_3}$ will archive a better result.

Bibliography

[1] Mahapatra, D. M., & Patra, S. K. (2019). A New Destination of Online Travel Business: A Case Study. SEDME (Small Enterprises Development, Management & Extension Journal), 46(2), 130-137.

[2] Malighetti, P., Paleari, S., & Redondi, R. (2009). Pricing strategies of low-cost airlines: The Ryanair case study. Journal of Air Transport Management, 15(4), 195-203.

[3] Malighetti, P., Paleari, S., & Redondi, R. (2010). Has Ryanair's pricing strategy changed over time? An empirical analysis of its 2006–2007 flights. Tourism management, 31(1), 36-44.

[4] Smith, B. C., Leimkuhler, J. F., & Darrow, R. M. (1992). Yield management at American airlines. interfaces, 22(1), 8-31.

[5] Daudel, S., Vialle, G., & Humphreys, B. K. (1994). Yield Management: Applications to Air Transport and Other Service Industries; this New English Version is Published with Additional Material and Updated Statistics. Presses de l'Institut du Transport aérien.

[6] McGill, J. I., & Van Ryzin, G. J. (1999). Revenue management: Research overview and prospects. Transportation science, 33(2), 233-256.

[7] Pritscher, L., & Feyen, H. (2001). Data mining and strategic marketing in the airline industry. Data Mining for Marketing Applications, 39.

[8] Avineri, E., & Ben-Elia, E. (2015). Prospect theory and its applications to the modelling of travel choice. Bounded Rational Choice behavior: Applications in Transport, 233.

[9] Bishop, C. M. (2006). Pattern recognition and machine learning. springer.

[10] Russell, S., & Norvig, P. (2002). Artificial intelligence: a modern approach.

[11] Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). Foundations of Machine Learning. Adaptive computation and machine learning. MIT Press, 31, 32.

[12] Alpaydin, E. (2020). Introduction to machine learning. MIT press

[13] Tucker, A. B. (Ed.). (2004). Computer science handbook. CRC press.

[14] Al-Jarrah, O. Y., Yoo, P. D., Muhaidat, S., Karagiannidis, G. K., & Taha, K. (2015). Efficient machine learning for big data: A review. Big Data Research, 2(3), 87-93.

[15] Deo, R. C. (2015). Machine learning in medicine. Circulation, 132(20), 1920-1930.

[16] Buczak, A. L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications surveys & tutorials, 18(2), 1153-1176.

[17] Díaz, Z., Segovia, M. J., & Fernández, J. (2005). Machine learning and statistical techniques: an application to the prediction of insolvency in Spanish non-life insurance companies.

[18] Etzioni, O., Tuchinda, R., Knoblock, C. A., & Yates, A. (2003, August). To buy or not to buy: mining airfare data to minimize ticket purchase price. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 119-128).

[19] Groves, W., & Gini, M. (2013, June). Optimal airline ticket purchasing using automated user-guided feature selection. In Twenty-Third International Joint Conference on Artificial Intelligence.

[20] Zouaoui, F., & Rao, B. V. (2009). Dynamic pricing of opaque airline tickets. Journal of Revenue and Pricing Management, 8(2-3), 148-154.

[21] Liu, B., Tan, Y., & Zhou, H. (2016, December). A Bayesian predictor of airline class seats based on multinomial event model. In 2016 IEEE International Conference on Big Data (Big Data) (pp. 1787-1791). IEEE.

[22] Liu, T., Cao, J., Tan, Y., & Xiao, Q. (2017, December). ACER: An adaptive context-aware ensemble regression model for airfare price prediction. In 2017 International Conference on Progress in Informatics and Computing (PIC) (pp. 312-317). IEEE.

[23] Wohlfarth, T., Clémençon, S., Roueff, F., & Casellato, X. (2011, December). A data-mining approach to travel price forecasting. In 2011 10th International Conference on Machine Learning and Applications and Workshops (Vol. 1, pp. 84-89). IEEE.

[24] Mazareanu, E. (2019, July 9). Topic: Air transportation in Canada. Retrieved March 17, 2020, from https://www.statista.com/topics/2890/air-transportation-in-canada/

[25] Chen, Y., Cao, J., Feng, S., & Tan, Y. (2015, October). An ensemble learning based approach for building airfare forecast service. In 2015 IEEE International Conference on Big Data (Big Data) (pp. 964-969). IEEE.

[26] Santana, E., & Mastelini, S. (2017, May). Deep regressor stacking for air ticket prices prediction. In Anais do XIII Simpósio Brasileiro de Sistemas de Informação (pp. 25-31). SBC.

[27] Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., & Vlahavas, I. (2016). Multi-target regression via input space expansion: treating targets as inputs. Machine Learning, 104(1), 55-98.

[28] Tziridis, K., Kalampokas, T., Papakostas, G. A., & Diamantaras, K. I. (2017). Airfare prices prediction using machine learning techniques. In 2017 25th European Signal Processing Conference (EUSIPCO) (pp. 1036-1039). IEEE.

[29] Janssen, T., Dijkstra, T., Abbas, S., & van Riel, A. C. (2014). A linear quantile mixed regression model for prediction of airline ticket prices. Radboud University.

[30] Geraci, M., & Bottai, M. (2014). Linear quantile mixed models. Statistics and computing, 24(3), 461-479.

[31] Lantseva, A., Mukhina, K., Nikishova, A., Ivanov, S., & Knyazkov, K. (2015). Data-driven modeling of airlines pricing. Procedia Computer Science, 66, 267-276.

[32] Taylor, J. W., McSharry, P. E., & Buizza, R. (2009). Wind power density forecasting using ensemble predictions and time series models. IEEE Transactions on Energy Conversion, 24(3), 775-782.

[33] Papadopoulos, S., & Karakatsanis, I. (2015, February). Short-term electricity load forecasting using time series and ensemble learning methods. In 2015 IEEE Power and Energy Conference at Illinois (PECI) (pp. 1-6). IEEE.

[34] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[35] Schapire, R. E. (1999, July). A brief introduction to boosting. In Ijcai (Vol. 99, pp. 1401-1406).

[36] Lewis, R. J. (2000, May). An introduction to classification and regression tree (CART) analysis. In Annual meeting of the society for academic emergency medicine in San Francisco, California (Vol. 14).

[37] Nielsen, D. (2016). Tree boosting with xgboost-why does xgboost win" every" machine learning competition? (Master's thesis, NTNU).

[38] Gumus, M., & Kiran, M. S. (2017, October). Crude oil price forecasting using XGBoost. In 2017 International Conference on Computer Science and Engineering (UBMK) (pp. 1100-1103). IEEE.

[39] Zheng, H., Yuan, J., & Chen, L. (2017). Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation. Energies, 10(8), 1168.

[40] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.

[41] Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21(3), 660-674.

[42] Horning, N. (2013). Introduction to decision trees and random forests. Am. Mus. Nat. Hist, 2, 1-27.

[43] Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.

[44] Ho, T. K. (1998). The random subspace method for constructing decision forests. IEEE transactions on pattern analysis and machine intelligence, 20(8), 832-844.

[45] Louppe, G., & Geurts, P. (2012, September). Ensembles on random patches. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 346-361). Springer, Berlin, Heidelberg.

[46] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.

[47] Ridgeway, G. (2007). Generalized Boosted Models: A guide to the gbm package. Update, 1(1), 2007.

[48] Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. Transportation Research Part C: Emerging Technologies, 58, 308-324.

[49] Expedia Travel: Search Hotels, Cheap Flights, Car Rentals & Vacations. (n.d.). Retrieved from http://www.expidia.com/

[50] Pels, E. (2008). Airline network competition: Full-service airlines, low-cost airlines and long-haul markets. Research in transportation economics, 24(1), 68-74.

[51] Xia, L., Jie, Y., Lei, C., & Ming-Rui, C. (2016, October). Prediction for air route passenger flow based on a grey prediction model. In 2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CYBERC) (pp. 185-190). IEEE.

[52] Seber, G. A., & Lee, A. J. (2012). Linear regression analysis (Vol. 329). John Wiley & Sons.

[53] Louppe, G. (2014). Understanding random forests: From theory to practice. arXiv preprint arXiv:1407.7502.

[54] Si, S., Zhang, H., Keerthi, S. S., Mahajan, D., Dhillon, I. S., & Hsieh, C. J. (2017, August). Gradient boosted decision trees for high dimensional sparse output. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 3182-3190). JMLR. org.

[55] Preacher, K. J., Curran, P. J., & Bauer, D. J. (2006). Computational tools for probing interactions in multiple linear regression, multilevel modeling, and latent curve analysis. Journal of educational and behavioral statistics, 31(4), 437-448.