CS 174 Project – Flight Delay Prediction
Final write-up
Huiting Yu

1.Preprocessing of Data

The dataset on http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236&DB_Short_Name=On-Time has flight on-time data of more than 20+ years. Since the airline performance usually varies significantly over the years, we choose the flights in one month (2012, January) to study. Also, the performance of different carriers varies much, so I choose all the flights operated by American Airlines in that specific month as my dataset. The original dataset involves almost all airports in U.S, to decrease the complexity of the program, I choose only flights involving the top ten busiest airports in U.S, which are ATL, ORD, LAX, DFW, DEN, JFK, IAH, SFO, LAS, PHX. After the filtering, the final dataset I use in this project has 8000+ flight entries in total. About 1300+ flight are classified (by whether delay time is larger than 15 minutes) as delayed . 6500 flights which are before 23th, January are used as training set and 2263 flights which are after 23th, January are used as testing set.

The eight flight attributes we are interested in: (originally in online dataset, not processed yet)

Arributes	Description
Origin	Airport code
Dest	Airport code
Scheduled departure time	hhmm
Air time	duration of the flight, in minutes
Distance	The distance from Origin airport to Dest airport, in miles
Day of week	from 1 to 7, integers
Day of Month	from 1 to 31, integers
Delay	delay of flight, in minutes

The attributes that need to be processed:
Origin/Dest : the airport code needs to be represented in numeral form. Some number reflecting the overall performance of the airport probably is useful in the prediction of flight delay. So I replace the airport code by the average delay time of the airport. The average time is computed from the whole dataset I have chosen. The figure 1-1 below shows the average departure delay (red) and average arrival delay(blue) of the ten

airports involved in the dataset. On x-axis 0-9 represents 10 airports. y-axis is for delay time.

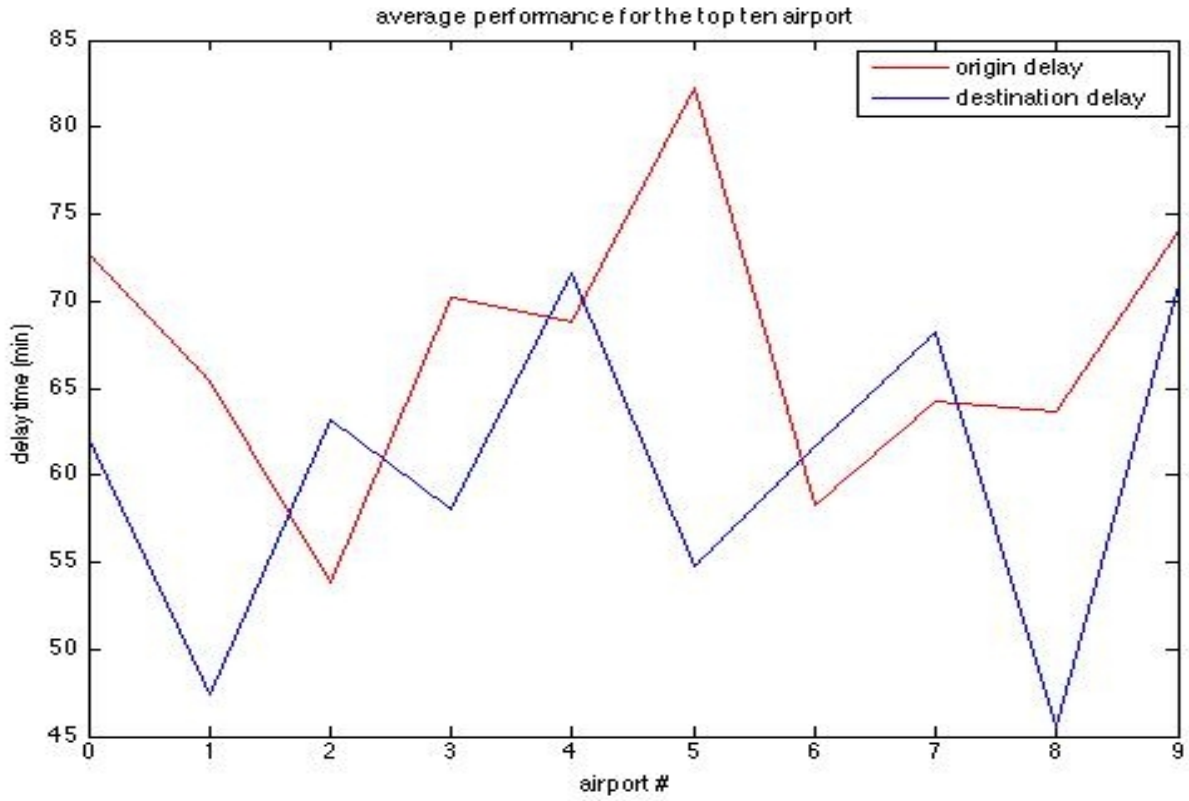

Fig 1-1 average performances for the top ten airport in U.S

Scheduled departure time: the original time is divided by 100 to preserve only the hour number, since the minute number seems to have much less influence on the prediction than the hour number.

Day of Month: the days which are holidays (in January, 2012) are labeled 1, otherwise they are labeled 0. I changed the day-of-month-number to the holiday label because whether a day is holiday matters much more in terms of flight delay.

2.Analysis of training set

To investigate the significance of every attribute on the delay, I evaluated the performance by computing the average delay related to different values of every attributes. Followings are the plots of the analysis of the training set (implementation in trainSetAnalysis.m), which give some intuitions about the relationship between each of the attributes and the actual flight delay.

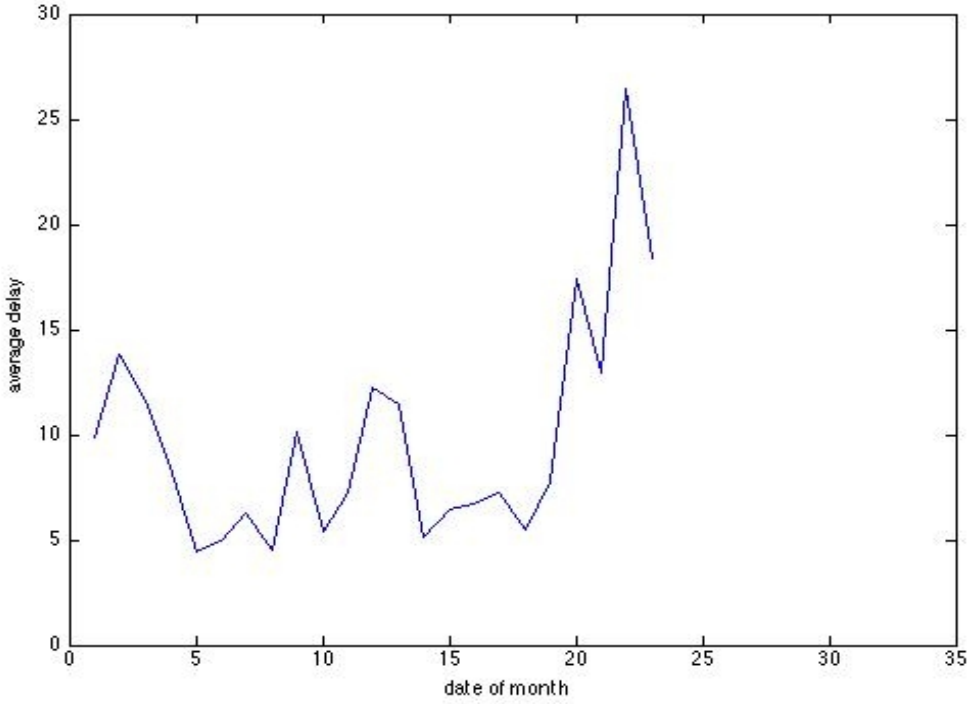

Fig. 2-1 date_of_month v.s delay

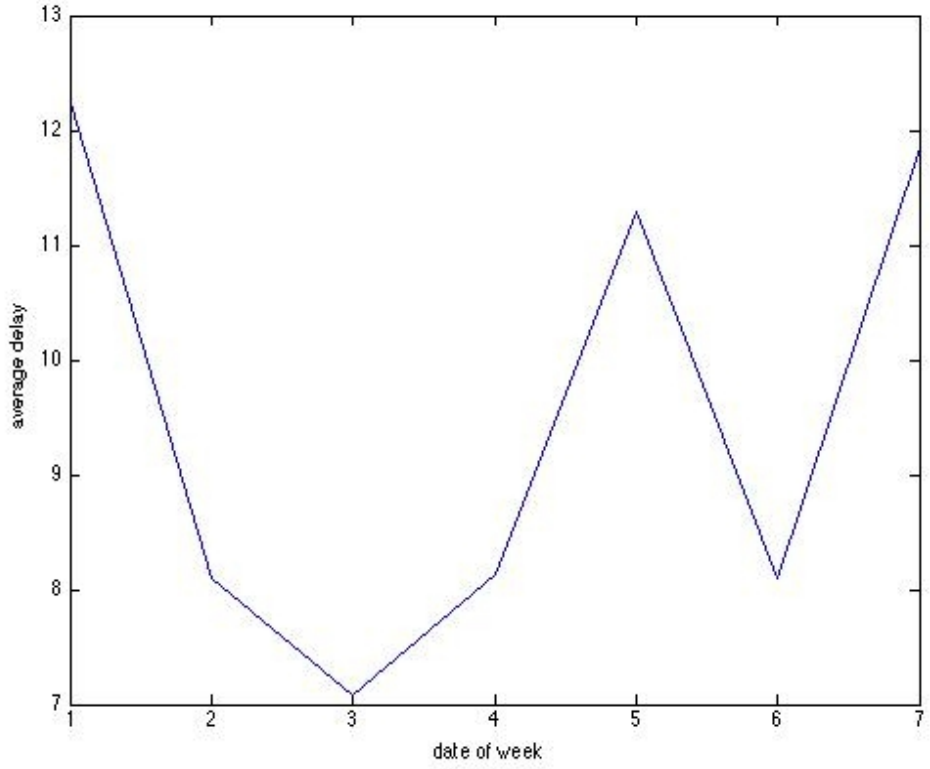

fig 2-2. day_of_week v.s delay

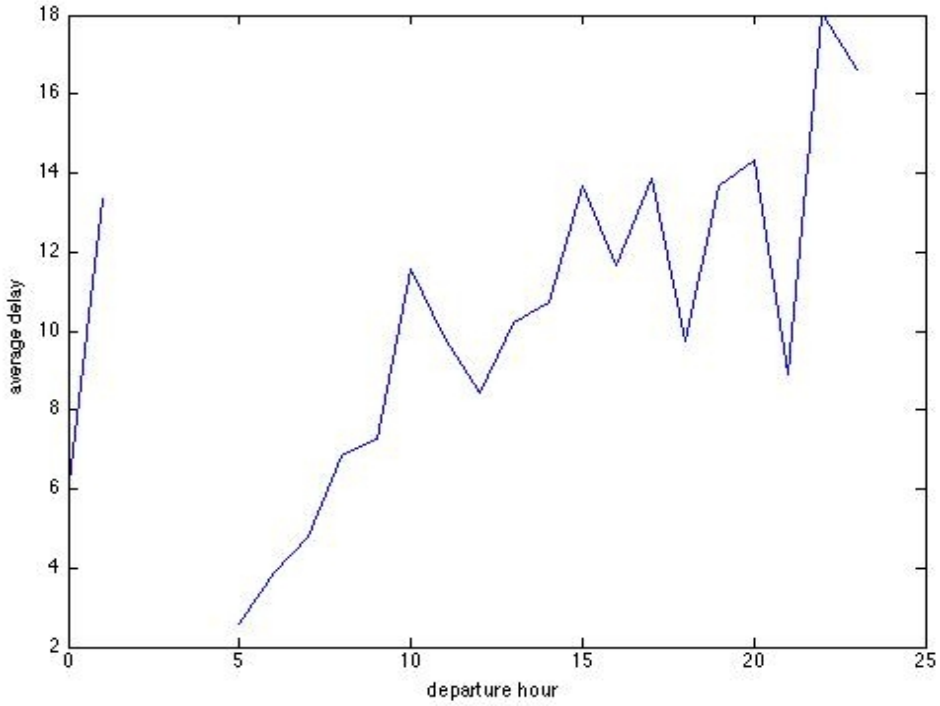

fig 2-3 departure time v.s delay

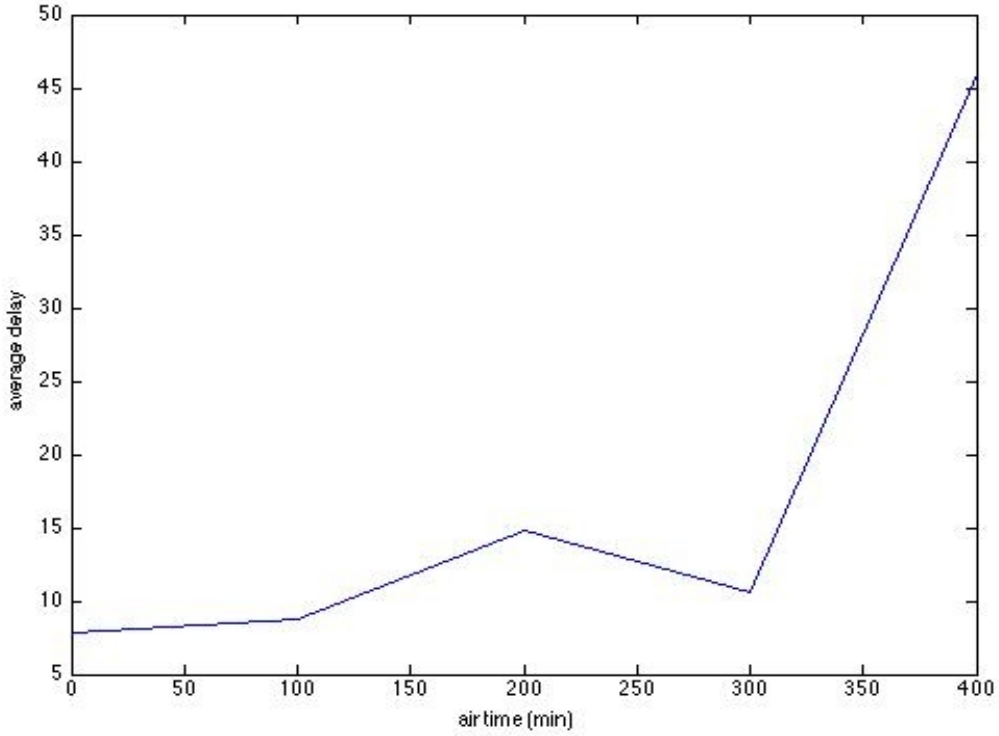

fig 2-4 air time v.s delay

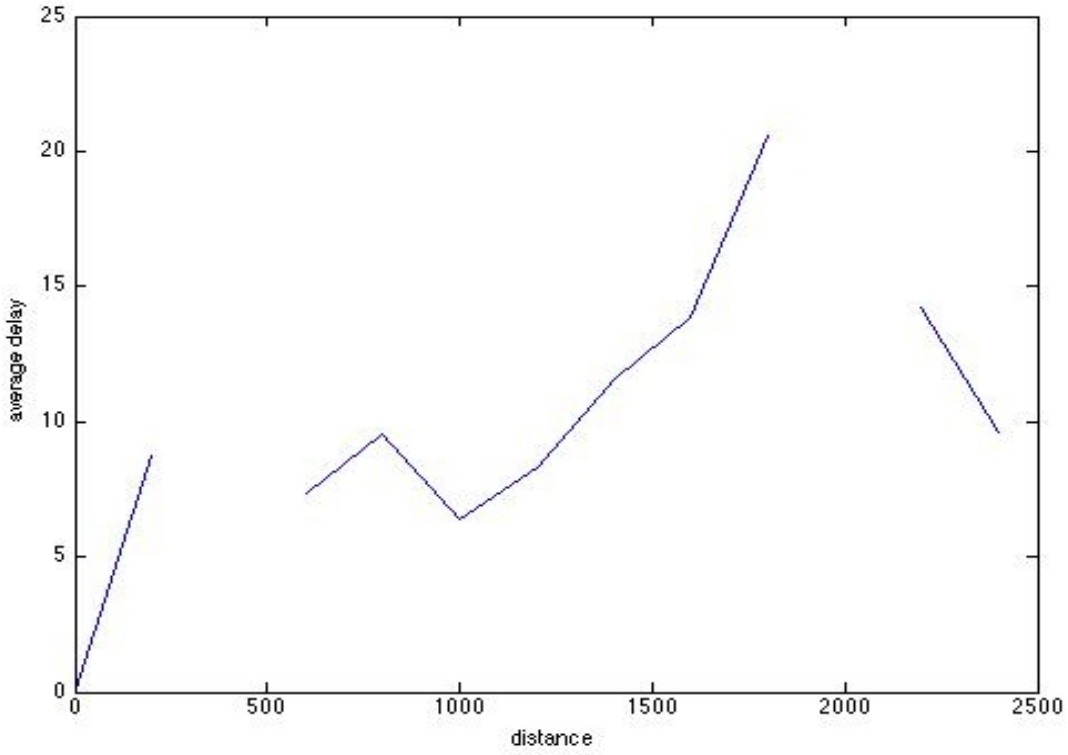

fig 2-5 distance v.s delay

From these plots, I found that with some values of the attributes the average delay is significantly large. For example, for the day_of_week attribute, if the value is 1, 5 or 7, the average delay is very large, which makes sense since there are much more air travels on weekends than on weekdays. Another example is in figure 2-3, the flights departing in early morning tend to have much shorter delay, which can affirm our common sense of non-busy hours. These insights will help when deciding which regression method to choose and which attributes should be given more weights when making prediction.
On the other hand, some curves cannot be explained confidently, for example in fig 2-1 the average delay tends to change sharply from day to day and I guess it has something to do with the weather, which makes our prediction task harder.

3.Prediction methods

Three methods are used – linear regression, kNN and segment regression.

3.1 Linear Regression

Implementation: LR_predictor.m. Linear regression with regularization (lamda = 0.1) is used. The results don't vary much if lamda's value changes.

result:
average error of prediction on training set: 13.96 min
average error of prediction on testing set: 17.66 min

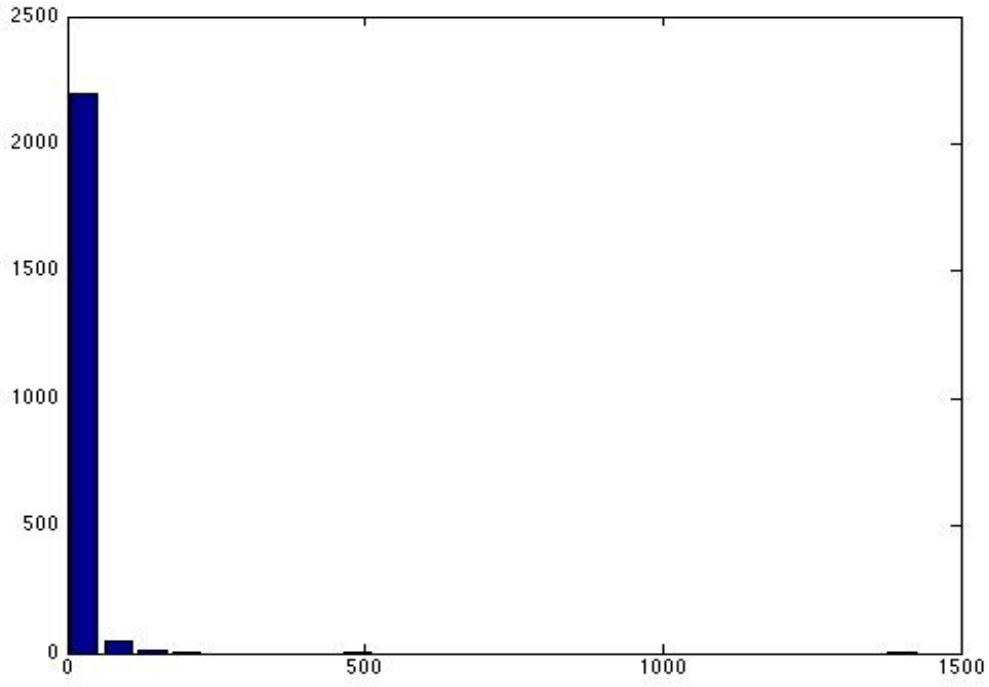

fig 3-1 histogram of the testing error

Bin value	29	86	143	200	258	314	372	429	486	544	601	658
Flight count	2198	46	12	6	0	0	0	0	0	0	0	0
Bin value	715	772	829	886	943	1001	1058	1115	1172	122	1287	1344
Flight count	0	1	0	0	0	0	0	0	0	0	0	1

Table 3-1 distribution of testing errors

The histogram and table above show that most predictions are in the range of [0,30] minutes. Only two predictions have very large errors, around 500 and 1400 minutes respectively. The histogram gives us confidence in terms of the average error of the

method. However, there are a few predictions having errors from 80 to 200 minutes.

3.2 kNN method

Implementation: kNN_predictor.m. A flight delay is predicted by the mean of its k nearest records (i.e. k nearest neighbors) . The distance of two records, X1 and X2, is measured by norm(X1-X2). Due to the large loop, a small testing set (300 flights) is randomly chosen from the original testing set.

Result:

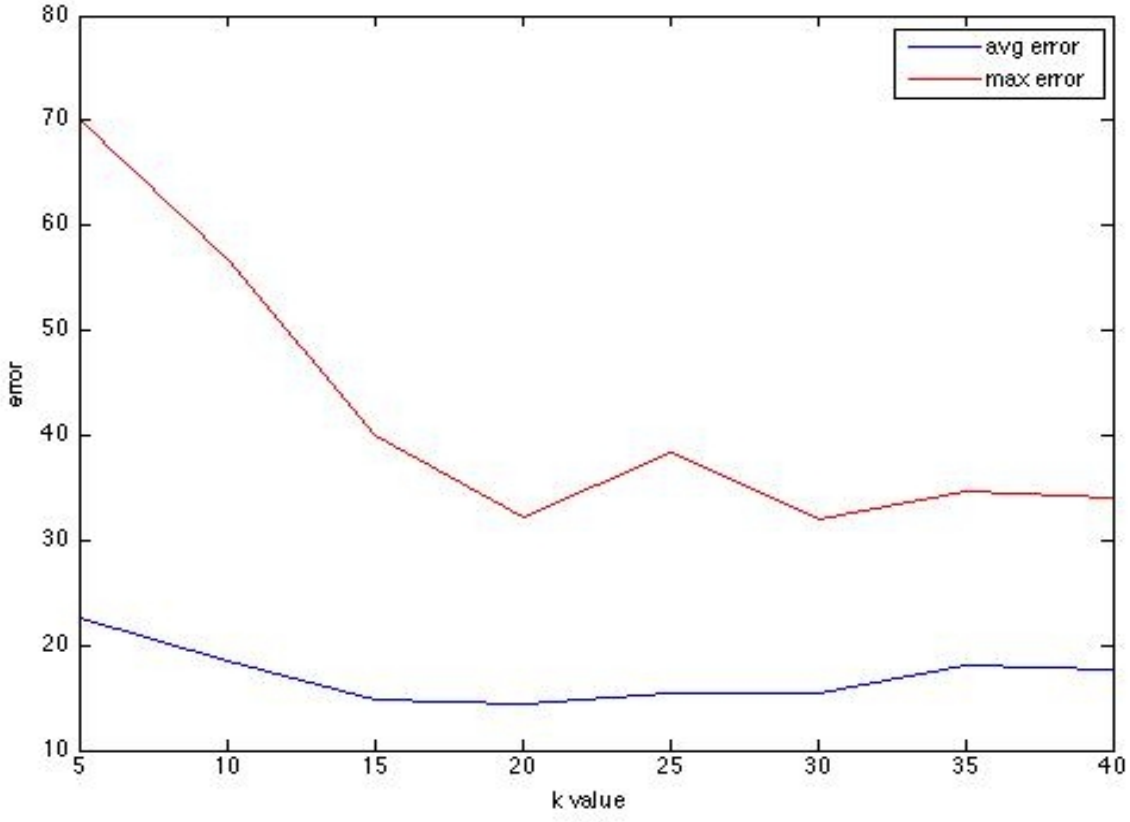

fig 3-2 testing errors with different values of k

Figure 3-2 shows that the average testing error does not change much as k changes from 5 to 40. As k > 15, the average testing error is around 15 minutes, which is acceptable. The maximum error over all testing data, however, is improved greatly as k increases. As k > 20, the maximum testing error is around 30 minutes.

3.3 Segment Regression

Overview: (refer to http://en.wikipedia.org/wiki/Segmented_regression)
The delay of a flight usually is shorter on weekday than weekends, and shorter in the

morning than in the evening. If whether the day or the time is in “busy time” is considered when making prediction, it is of great chance that the results can be improved. Regarding this thought, the segment regression method is implemented to partition the data into segments and fit different line segments to different partitions. The prediction of a testing data is based on which segment the data belongs to and the corresponding set of parameters regarding this segment.

The boundary of segments is called break-point, whose value needs to be determined by maximizing Cd, which is the coefficient of determination for all data (Cd) . Cd is found from:

- $Cd = 1 - \text{sum} \{ (\mathbf{y} - Yr)^2 \} / \text{sum} \{ (\mathbf{y} - Ya)^2 \}$

where Yr is the expected (predicted) value of **y** according to the former regression equations and Ya is the average of all **y** values.
The Cd coefficient ranges between 0 (no explanation at all) to 1 (full explanation, perfect match).

Implementation:
I tried two attributes – day_of_week and departure_hour to partition the data. The selection of the two is based on the analysis of the training set, where it looks like each of the two attributes can be belong to two groups according to the value. For example, in fig 2-2, the day_of_week attribute seems can be divided into two groups.

Result :
SRPredDayOfWeek.m: The training data is partitioned into two segments based on the value of DayOfWeek The first segment has values [3 2 6 4]. The second segment has values [1 5 7]. The average testing error is around 16 minutes. The maximum testing error is around 350 minutes. The figure 3-3(a) shows the distribution of the testing errors, it's obvious that it's greatly improved compared with the linear regression.

SRPredDepHour.m: The training data is partitioned into two segments based on the value of DayOfWeek The first segment has values [5 6 7 0 8 9]. The second segment has all the other values representing hours. The average testing error is around 16 minutes.

From the histograms below we can see the prediction almost won't fall into the bins after the first bin, indicating that most predictions are good. However, one or two predictions are very bad.

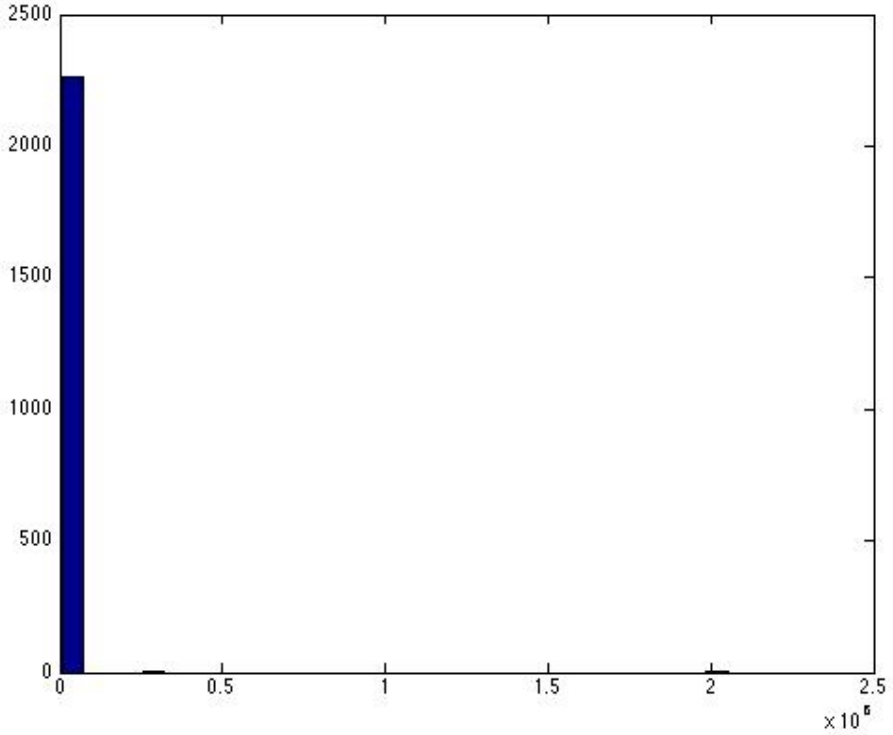

Fig 3-3(a) histogram of testing errors with segment regression using day_of_week

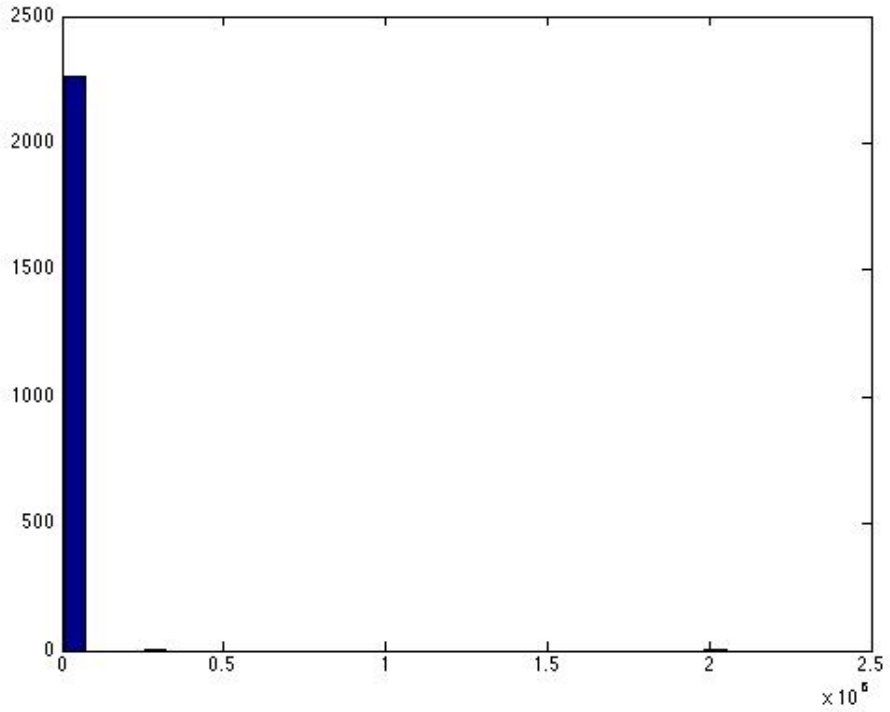

Fig 3-3(b) histogram of testing errors with segment regression using departure_hour

4.Discussion

All of the three methods can obtain acceptable average testing error value. In terms of the maximum testing error value, none of the three methods produces good result probably because some flights are extremely delayed and cannot be predicted by the learned parameters. The percentage of testing data cannot be well predicted is as small as 5%.

5.Reference

1). http://www.transtats.bts.gov
2). http://en.wikipedia.org/wiki/Segmented_regression
3). Predicting Flight Delays Through Data Mining, Tim Stefanski
4).Alarming Large Scale of Flight Delays: an Application of Machine Learning, Zonglei Lu.