Capstone Project

Krittisak Chaiyakul
Nov 11th, 2017

I. Definition

Project Overview

Federal Aviation Administration is the United States Department of Transportation. They provide information about the transportation, including flights. Their website shows the real-time flight delay information, which consists of flight delay time. However, the information covers only major airports in the United States.

In this project, I will cover all airports across the United States. I apply k-nearest neighbor regression to learn on previous the flight delay information from kaggle (https://www.kaggle.com/usdot/flight-delays/data) and predict the expected delay time of the flights. It will be a lot useful if we know the likely arrival time of our flights.

Problem Statement

In this project, I download and preprocess the data for training. Then, I apply machine learning model to predict the flight delay time by training with the previous flight delay information and using grid search cross validation to choose the parameter for our model. I measure the performance of the model by predicting the delay time in minutes on testing set and calculating the error rate. Therefore, our goal is to find the optimal model to minimize the error rate.

Metrics

I use mean absolute deviation (MAD) as our metric.

$$mad = \frac{1}{N}\sum_{i=1}^{N}|out_i - pred_i|$$

$where\ N\ is\ the\ total\ number\ of\ predicting\ flights,$
$out_i\ is\ the\ previous\ recorded\ flight\ delay$

$$pred_i\ is\ the\ predicted\ delay\ time\ of\ flight\ i\ from\ our\ model$$

I calculate the mad value of predicted delay from both benchmark model and our solution model. In other words, the mad value is the average error delay of the predicted time in minutes.

II. Analysis

Data Exploration

The flight information comes from kaggle (https://www.kaggle.com/usdot/flight-delays/data). Firstly, I explore the dataset and total number of flights is 5,819,079. Then I discard all flights that are not necessary for our model, such as the number of cancelled flights, unknown origin/destination flights and unknown delay time. Finally, I count the total number of final flights which is 5,231,130.
For features, I use a total of 11 features

- *'MONTH', 'DAY', 'DAY_OF_WEEK', 'FLIGHT_NUMBER', 'SCHEDULED_DEPARTURE', 'SCHEDULED_TIME', 'DISTANCE', and 'SCHEDULED_ARRIVAL'* - Numeric
- *'AIRLINE', 'ORIGIN_AIRPORT', and 'DESTINATION_AIRPORT'* - String

Then I calculate the statistics of delay time.

```
Statistics for delays:

Minimum delay: -87.00 minutes
Maximum delay: 1,971.00 minutes
Mean delay: 4.89 minutes
Standard deviation of delay: 39.79 minutes
```

Exploratory Visualization

In this plot, it shows the distribution of the number of the flights based on the delay time of the flights. I notice that the distribution is centered around 0 and bends toward right.

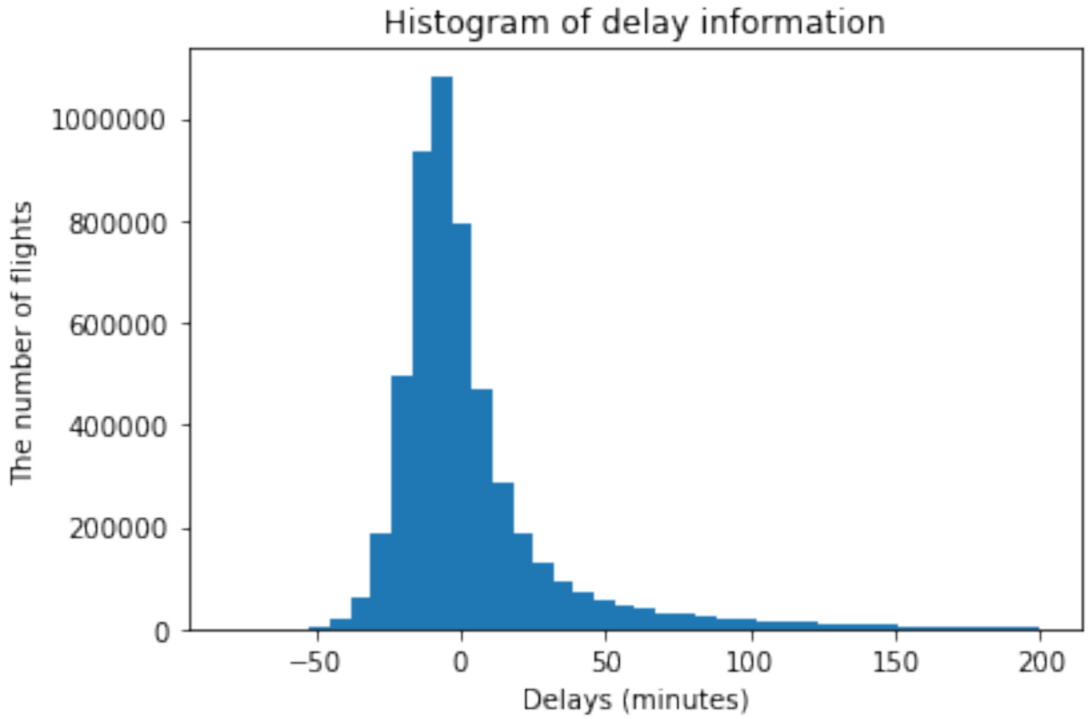

Figure 1: the histogram shows distribution of the number of flights based on the delay time.

Algorithms and Techniques

In this project, I use k-nearest neighbor regression as our model. Firstly, I will split the dataset into training and testing set with ratio 0.8:0.2. Then I train k-nearest neighbor regression to predict the flight delay time with the previous flight delay information and applying grid search cross validation to choose the best k parameter. Finally, I test the final model with testing set and calculate the error rate of the model with our evaluation function above.

Benchmark

Our benchmark model will be the average delay time of all flights. I try to get the better predicted results with our proposed model.

III. Methodology

Data Preprocessing

First, I filter the data and discard all abnormal flights.

Total number of flights: 5819079
Total number of cancelled flights: 89884
Total number of nan flights: 15187
Total number of unknown-origin flights: 482878
Total number of unknown-destination flights: 0
Total number of final flights: 5231130

Then, for features I use a total of 11 features because I know this information and the information are guaranteed to be unchanged before the flights begin.

- *'MONTH', 'DAY', 'DAY_OF_WEEK', 'FLIGHT_NUMBER', 'SCHEDULED_DEPARTURE', 'SCHEDULED_TIME', 'DISTANCE', and 'SCHEDULED_ARRIVAL'* - Numeric
- *'AIRLINE', 'ORIGIN_AIRPORT', and 'DESTINATION_AIRPORT'* - String

Finally, I apply label encoder to encode all strings (*'AIRLINE', 'ORIGIN_AIRPORT', and 'DESTINATION_AIRPORT'*) to numbers.

Implementation

Training section:

1. Load the data into memory.
2. Preprocess the data as shown above.
3. Split the dataset into training/testing set with ratio 0.8/0.2.
4. Develop a benchmark model which equals to an average of all flights' delays.
5. Develop our solution model with KNeighborsRegressor from scikit-learn.
6. Apply GridSearchCV from scikit-learn to choose the best parameter k for our model on training set.

Testing section:

7. Predict the delay time on testing set with our both benchmark and solution models.
8. Calculate the performance with absolute_mean_error from scikit-learn.

Refinement

As shown above, I apply grid search cross validation to improve the k-nearest neighbor regressor by trying many possible k parameters, such as 1, 5, 10, 20, 50, 100. The cross validation shows that k=100 is the best parameter for our model.

IV. Results

Model Evaluation and Validation

The final model and parameters are chosen based on the data exploration and cross validation.

The number of training set is 4,184,904
The number of testing set is 1,046,226
The number of features is 11
The parameter k for knn regressor is 100

Justification

The mean absolute error of benchmark algorithm is 21.70 minutes and the mean absolute error of k-nearest neighbor regressor is 21.12 minutes. Both results are very similar and close because from the histogram above the wide of the range is more than 200 minutes. The final delay errors are very small that people can wait.

V. Conclusion

Free-Form Visualization

From the histogram below, I notice that the mean predicted delay is around 0 and t he range of predicted delay is very small.

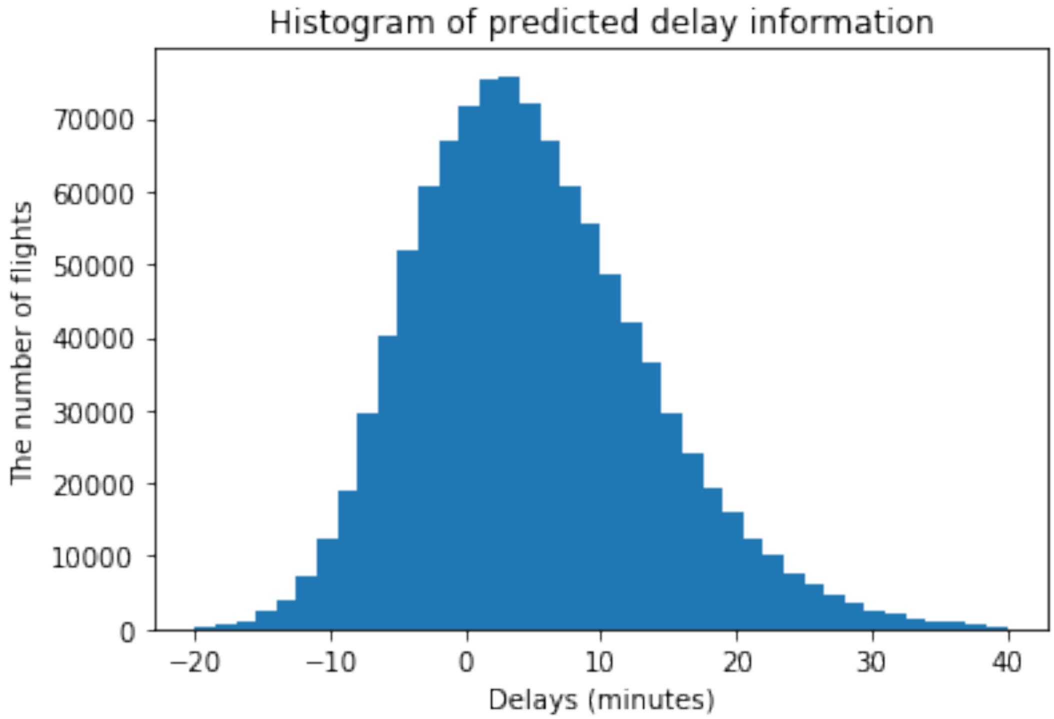

Figure 2: the histogram shows distribution of the number of flights based on the predicted delay time.

I see that most of our predicted delays have the errors less than 10 minutes and more than half are less than 30 minutes.

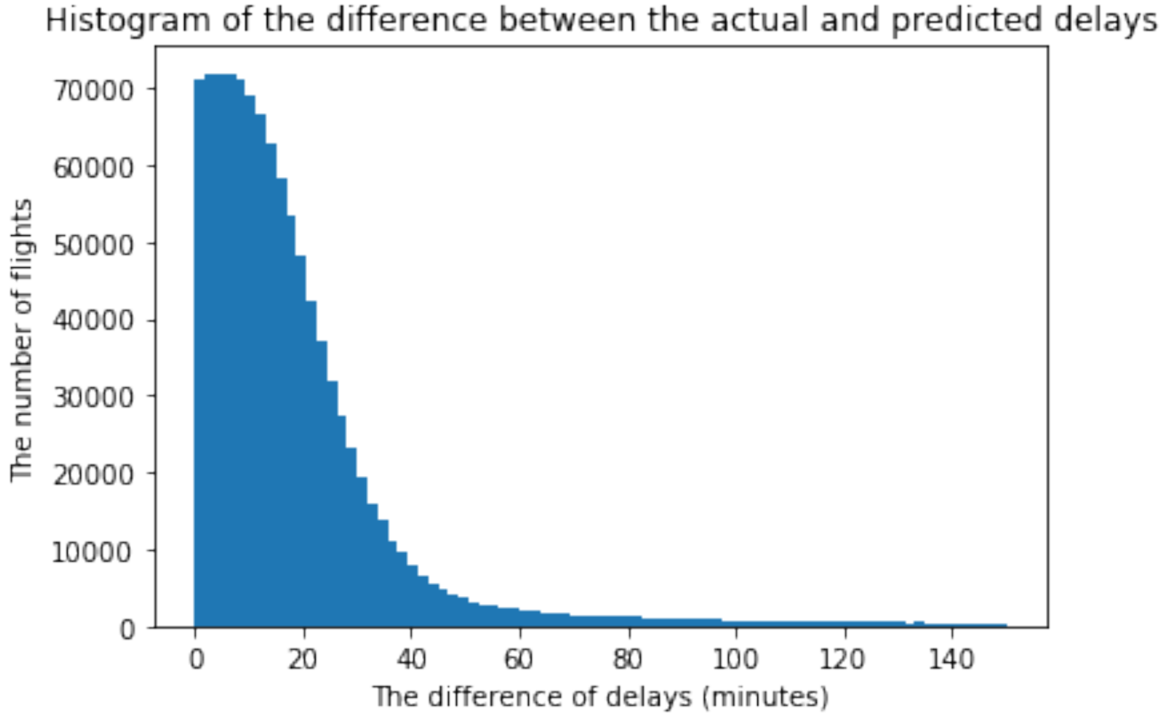

Figure 3: the histogram shows distribution of the difference between the actual delay and the predicted delay.

Reflection

I can summarize the process into 4 big steps

1. Explore the data
2. Prepare the data
3. Develop both benchmark and solution model
4. Evaluate the performance

I found that step 2 and 3 are very hard. For step 2, I tried to use one-hothead encoder to encode the strings into numbers, but the number of features go up to 633 which is too big because the number of all flights are more than a million and the program ran out of memory. For step 3, in grid search cross validation I tried larger parameters k, but again the program ran out of memory. Moreover, when I ran decision tree regressor and random forest regressor to train on more than a million data, the program took forever and had no sign of finishing. Therefore, I decided to train only k-nearest neighbor regressor.

Improvement

For dataset, it includes only all flights from year 2015. We can try a dataset from many years to improve our model. For preprocessing, as I mentioned above, we should try one-hothead encoder on the computer which has a very very very large memory. One more thing, we can normalize all features that are numbers. For models, as I mentioned above, we should get a larger pc and try a larger parameter k for k-nearest neighbor regressor. Addionally, a decision tree regressor and a random forest are good models to try on this problem.