# Airline Flights Delay Prediction

Yiluo Qin
UC San Diego
yiq028@ucsd.edu

Yijun Liu
UC San Diego
yil724@ucsd.edu

Yu-Chieh Chen
UC San Diego
yuc399@ucsd.edu

## Abstract

*As the technology grows faster, people tend to travel by planes more often than ever before. Therefore, in order to maximize time efficiency, it is crucial to accurately predict whether a flight will arrive on time. Our work primarily focuses on the key factors contributing to whether a flight will be delayed. The dataset we chose came from Department of Transportation Year 2015 Flights Information across the United States. Within our model selections, we proposed several potentially useful models such as Logistic Regression, K Nearest Neighbors, Decision Tree, etc. Given all the models' performances, we decided that Decision Tree would be the best model because it has the second highest accuracy score and the best F1 score.*

## 1. Dataset Introduction

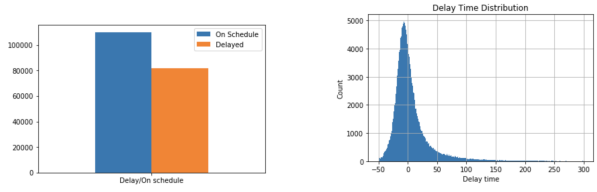
The Flights dataset collected from year 2015 is extremely useful because it provides all the necessary information we need for every specific flight. The data set comes from the US Department of Transformation, which provides general information to the public about what needs to be known to predict whether a flight will be delayed. In addition, the dataset consists important fields including origin airports, destination airports, departure time, and arrival time.

After inspecting the Flights data set, we notice some of the more useful features include day of week, month of year, origin airport, and destination airport. These fields, with some preliminary analysis, can potentially give us a first look at how the delays are distributed. For example, we could tell the distribution of delays based on day of the week or month of the year. Later, we further combine delay reasons and some other features to further determine what else can cause a flight's delay. Additionally, we split arrival time and departure time and convert them to real time to explore whether different times of a day will affect a flight's delay.

Another important aspect we have considered is to include each individual airport's location to find out if there are specific airports tend to have higher ratio of delayed flights than those of others. In order to accomplish this, we have merged two data tables *Flights* and *Airports* to get the longitude and latitude of each airport.

### 1.1. Preliminary Analysis

Note that the entire Flights dataset has nearly 6 million flights distributed across the whole country, which would be a big burden for both computational analysis and model predictions. For simplicity, after careful consideration, we decide to choose Los Angles International Airport(LAX) as the destination airport only. The main reason is that LAX is a well known and large airport, so we can get a fairly good amount of dataset but not too large to train our models. Secondly, among all other big airports in the US, Los Angles is located at a more hospitable area, which means weather condition would not cause significant biases. Additionally, by choosing one specific airport, we can gain more insights particular to the airport rather than averaged results over many other airports. We decide to choose LAX as the destination airport only because we think it is more important for passengers to pre plan their time in a more organized fashion. If they would know a flight would be delayed, they could even change their airlines or choose another departure time. In particular, the subset we obtain contains 194696 rows with 81985 delayed flights and 110151 on schedule flights .




(a) Delay vs On Schedule Split review

(b) Distribution of Delays reviews

Figure 1: Distribution of delays

The distribution of delays regard to LAX as the destination airport in 2015 can be seen in Figure 1. Additionally,

from the distribution of the delays, we see a fairly normal distribution with a center around 0, which means the majority of flights tend to arrive on time.

### 1.2. Delays by Locations

We first want to explore how other airports in the US are connected to LAX. After mapping all airports which have LAX as the final destination, we see that the distribution of airports is very spread out.

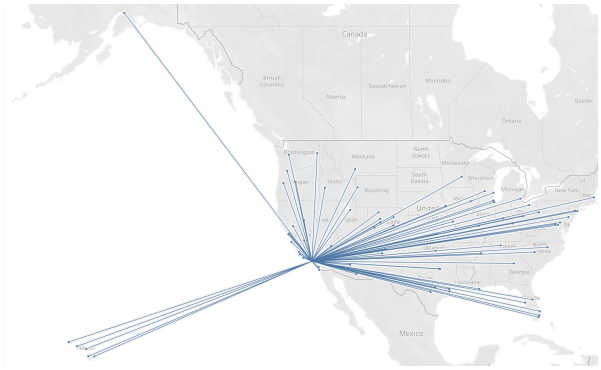



Figure 2: Spread of airports which are connected to LAX

From the graph, we conclude that since all the origin airports connected to LAX are spread out across the country, we can be sure that LAX is considered to be one of the major airports in the US. Such fact implies the data we have chosen is fairly diverse.

Next we want to explore all the origin airports which have LAX as the destination airport regard to their flight delay ratio.

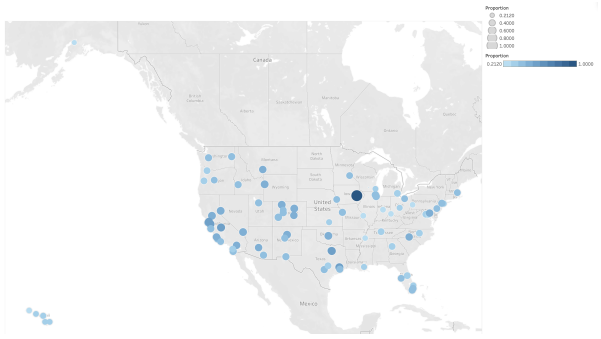
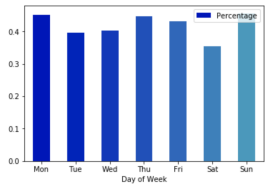


Figure 3: Distributions of Delays across all the airports in the US

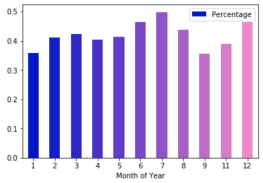
From the heat map chart, we see no clear pattern between the location of a origin airport and having a higher delayed flight ration. This mixed heated map makes sense because we should not expect to see a direct connection between the location of an origin airport and whether a flight to LAX will be delayed. However, we do observe airports on both west and east coasts tend to have a higher chance to have a delayed flight to LAX; however, we should not naively make such assumption since it solely based on the size of each city and its airport size. For example, we see a high concentration of airports which are more likely to have delayed flights in Texas, and the reason is that Houston, Dallas, and Austin are the major cities in the Central US.

### 1.3. Delays by Time

In addition to the inspecting relationship between flight delays and locations, we can also explore if the delays are related to any specific times in terms of the day of a week or the month of a year.




(a) Delays regard to each day review




(b) Delays regard to each month reviews

Figure 4: Distribution of delays for day or month

From the data above, we see that across seven days of the week, there is no significant peak in terms of having a much higher delay rate though it seems like Thursday and Friday have higher than all the other days. However, the difference is not huge enough to simply conclude anything fundamental. We could see that the distributions are fairly evenly distributed, with the highest at 44% and the lowest at 39%. On the right hand side, we have flight delay distributions across 12 months in 2015. We see that during the summer time, in June, July and August the delay rate tends to be higher than other months'. One reason could be that since it is summer time, many students and young adults have their summer vacations and travel to the US. Because Los Angles is considered to be one of the must visit places in the US, higher flow of customers may contribute to a higher delay rate in different origin airports.

### 1.4. Delays by Airlines

One interesting fact we observe from the chart above is that even though American Airline(AA) is considered to one of the biggest airline companies in the US, their flights flying to LAX has a surprisingly low delay rate. Compared to AA, Spirit Airlines(NK) has the highest delay rate, which means even though they have far less flights than those of American Airline, majority of Spirit Airlines flights tend to arrive at LAX behind schedule. Statistically speaking, the difference

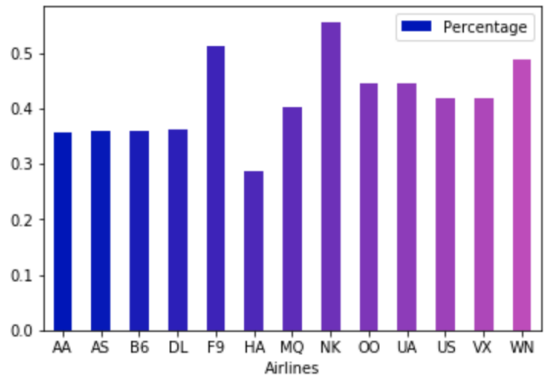



Figure 5: Distributions of delays for airlines

between the highest delay rate almost 30% with the highest at 55% and the lowest at 27%.

## 2. Predictive task

### 2.1. Rating Prediction

We choose to predict whether a flight will be delayed at LAX airport using features such as the origin airport, month of the year, day of the week, etc. The real world application is to help people to decide whether they will arrive at their destinations on time. For example, it is important for business people to maximize their time in each day; therefore, in order to not waste time at the airports, they can predict whether a flight will be delayed. Accordingly, they can reschedule another flight or even change to another airline to make sure they can arrive at their destinations on time.

Since our main prediction task is to predict whether a specific flight will be delayed, we can treat the task as a categorical prediction with two classes. For training, testing, and validation data sets, we decide to split the original data set using a 8:1:1 ratio. We plan to evaluate the performance of our prediction model in two ways. One is the general accuracy of our predictor, which calculates what proportion of our predictions matches the true labels. We can compare our models with the baseline model to gain an insight of the general improvements. The other method is using the average F1 score for our binary classes. F1 score is calculated by taking the harmonic mean of precision and recall of our two classes.

Our simple baseline model would naively choose the majority of the flights' status arriving at LAX airport. From earlier analysis, the majority of the flights are on schedule. This is because on schedule flights count almost 58% of the whole sub dataset we have chosen. In order to maximize our baseline model's accuracy, we choose to predict every flight will be on schedule.

| | *Class 1 Predicted* | *Class 2 Predicted* |
|---|---|---|
| **Class 1 Actual** | **TP** | **FN** |
| **Class 2 Actual** | **FP** | **TN** |

Figure 6: Confusion Matrix

To assess the validity of our prediction models, we visually inspect the confusion matrix of our predictions on the test dataset. The key point is to visually compare the false predictions besides accuracy score to make sure our models are not biased towards one side. In such way, we could try to make our models as neutral as possible.

$$PRE = \frac{TP}{TP + FP}$$

$$REC = TPR = \frac{TP}{P} = \frac{TP}{FN + TP}$$

$$F_1 = 2 \cdot \frac{PRE \cdot REC}{PRE + REC}$$

Figure 7: Model Evaluation Scores

### 2.2. Feature Engineering

Features are initially pre-processed by taking the subset of the original Department Of Transportation dataset. Because each row of data represents a single flight, the subset's features should preserve the same format. In addition, there are several ways to feature engineer our information; however, for different models, the number of features taking into consideration may differ.

One important part for all our models is One Hot Encoding transformation. Basically, we want to one-hot-encode many categorical features and turn them into numeric values that can be processed by models. For example, for each distinct flight, we can represent the airline carrier using a linear vector, with number 1 represents the airline for that specific flight.

Another essential part is to convert the Scheduled Departure and Scheduled Arrival columns, etc to readable Timestamp data. For example, all the data in these time columns are represented as an integer number such as 1250. In order to correctly format the data, we need to first convert the

| | AA | AS | B6 | DL | F9 | HA | OO | UA | US | VX | WN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 8: One Hot Encode Airline Carriers

integer to a string. Then, our next step is to split the string into correct time. Taking 1250 as an example, 1250 will be formatted as 12:50. Additionally, we are using 24 hour clock; therefore, 12:50 means 12:50 P.M.

Last but not least, we also decide to filter out redundant columns to no only preserve our dataset's overall format, but at the same time, to be more memory efficient and time efficient. In order to accomplish this, we have concluded the non significant columns such as Airline Delay Reason, Weather Delay can be dropped because the majority of them are *Nan*. The reason behind it is that unless there are significant issues that have to be reported, these columns are just there for any extra references. However, for the purpose of our model prediction, they are not relevant. As a result, we decide to drop the columns which have majority *Nan* rows.

The four models we choose are Logistic Regression, Decision Tree, K Nearest Neighbors, and Gradient Boosting Classifier because these are some of the most popular models used to predict categorical variables. In addition, compared to more advanced models such as using neutral networks, the first three models are fairly easier to implement, and at the same time, since our featured columns do not compose a large number of dimensions, using these models should be enough for predicting whether a flight will be delayed or not. We include the fourth one because we want to compare our implementation of Gradient Boosting Classifier with the more sophisticated ones in current industry to access what our similarities and limitations are. Doing such comparison can help us to better understand what are some of our dataset's restrictions and what we need to consider if we face such problem in our future.

# 3. Model

## 3.1. Baseline

The simplest baseline model we can come up with is selecting the majority type of the flights arriving at LAX airport, which is on schedule. Therefore, for our baseline model, like we have stated earlier, will always predict the flight arrive with no delays. This can be explained by two things. One is that the proportion of on schedule flights is around 57%. Second is that as long as we are evaluating our model using accuracy, meaning that we do not care about confusion matrix or model biases. This baseline model will give us an accuracy of 0.57.

## 3.2. Logistic Regression

Logistic Regression is the most commonly used statistical model predicting categorical variable. The parameters that are taken into accounts are MONTH, AIRLINE, SCHEDULED DEPARTURE, SCHEDULED TIME, FLIGHT NUMBER, SCHEDULED ARRIVAL, DISTANCE, etc. with one-hot encoding on MONTH, AIRLINE, and FLIGHT NUMBER. As running through different solver, *newton-cg* tends to have higher F1 score and accuracy score comparing to other solvers, which proves that it is good for larger dataset. From the different C values, logistic regression turns out to have best accuracy score and F1 score when C equals to 1. Although the accuracy score improves in the logistic regression, the F1 score still remains low, which means our model is biased towards one sub category of confusion matrix. Therefore, our model is not perfect for scalability because our model would perform even worse on unseen and larger dataset. It is entirely possible that our subset does not necessarily represent the whole dataset. For our logistic regression model, when C equals to 1 and solver equals to *newton-cg*, the accuracy score for testing dataset at 0.634 and F1 score at 0.495.

| solver | f1_score | accuracy score |
|---|---|---|
| newton-cg | 0.502011 | 0.639136 |
| lbfgs | 0.458613 | 0.622170 |
| liblinear | 0.461653 | 0.623731 |
| sag | 0.426290 | 0.611137 |
| saga | 0.423903 | 0.609160 |

(a) Different Solver Used

| C value | f1_score | accuracy score |
|---|---|---|
| 0.01 | 0.472638 | 0.631382 |
| 0.10 | 0.493438 | 0.638407 |
| 1.00 | 0.502011 | 0.639136 |
| 10.00 | 0.502009 | 0.638720 |
| 100.00 | 0.501579 | 0.638616 |

(b) C Values Used

Figure 9: F1 Score and Accuracy Score

## 3.3. Decision Tree

For this classification problem, another general approach is Decision Tree Classifier. The main advantage of Decision Tree [7] is to convert complex relationships between input features and target object to more simplified relationships by dividing original input variables into significant subgroups. When predicting delay of flights, the relationship between the label (*i.e.* 0 for not delayed; 1 for delayed) and features (*i.e.*geographic feature, generic features of flights such as the airline or flight number, and exterior features such as date) are complex. Therefore, Decision Tree will be useful in identifying and ranking useful features. To optimize our Decision

Tree classifier, numerous hyper-parameters are tuned, and 3 parameters have significant influences on predict accuracy: *max depth, min sample split, and max leaf nodes*.

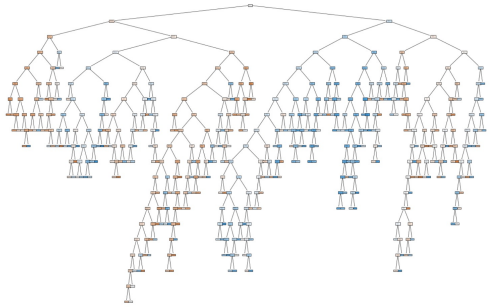


Figure 10: Balanced Decision Tree Graph

To select the best parameters, a Grid Search CV to find all hyper-parameters was performed, and we concluded the best model with F1 score of 0.5725, accuracy of 0.623, with parameters as follows: *class weight* set to 'balanced', *criterion* set to 'entropy, *max depth* equals to 25 ,*min samples split* equals to 300, *max leaf nodes* equals to 250. However, the better performance on validation set, compared to test set, illustrated that this model will potentially overfit. For the scalability of this model, due to the limitation of original dataset, which is restricted only to the data from LAX in 2015, scalability could be an issue.

### 3.4. K Nearest Neighbors

Another common approach is K Nearest Neighbors classifier. The main advantage of this classifier lies on its assumption that similar things exist in close proximity. The selection of K [4], or the number of neighbors, is crucial. Therefore, to optimize this model, we mainly tuned on hyper-parameter K and found when K = 10, we obtained highest accuracy on validation is 0.64 and F1 score is 0.50. The corresponding accuracy on test is 0.6416 and F1 score at 0.50. However, three main drawbacks of this model are considered. Due to the large dataset we obtained, training this model will take a long time, which making this model be not efficient. Second, the scalability of this model heavily relied on the dataset, and when data changed, the corresponding best K in this model will not still be the best. Third, when K is too large, even if the accuracy and F1 score could be improved, this model will possibly overfit the training dataset.

### 3.5. Gradient Boosting Classifier

Last but not least, we choose to use Gradient Boosting Classifier attempting to get a better accuracy score because the model is intrinsically a strong model which combines multiple small models. Another hidden layer of this model is using Decision Tree [2]. We hope that the result will be better than all previous three models, although the performance could turn out to be not as good as we would expect due to the dataset. From the the left table in figure 9, it shows that when the learning rate is 0.750, it has the highest F1 score at 0.518 and accuracy score 0.652. From the right table in figure 9, it shows that when n estimator equals to 1000, it has the highest accuracy score at 0.66 with F1 score at 0.546 on validation dataset. When n estimators is 1000 and the learning rate is 0.750, the accuracy score is 0.655 and F1 score is 0.542 on testing data set. We don't think our model runs into problem of overfitting because the accuracy score we obtained does not significantly differ from the one from the validation set. However, scalabililty could be a problem because the model is again trained on a small amount of dataset compared to the whole dataset.

| Learning Rate | f1_score | accuracy score |
|---|---|---|
| 0.050 | 0.449600 | 0.634556 |
| 0.075 | 0.462996 | 0.637106 |
| 0.100 | 0.475381 | 0.640125 |
| 0.250 | 0.493638 | 0.650013 |
| 0.500 | 0.512768 | 0.652459 |
| 0.750 | 0.518172 | 0.651574 |
| 1.000 | 0.516692 | 0.648139 |

(a) Learning Rate Used

| n_estimators | f1 score | accuracy score |
|---|---|---|
| 500 | 0.540798 | 0.659381 |
| 750 | 0.544089 | 0.658808 |
| 1000 | 0.546287 | 0.659745 |
| 2000 | 0.545780 | 0.656622 |
| 3000 | 0.551246 | 0.656154 |

(b) N Estimator used

Figure 11: F1 Score and Accuracy Score

## 4. Prior Literature

The dataset we choose for all our models directly come from Department of Transportation year 2015. It is very complete dataset which includes all the domestic airlines flights in the US. For the training time purpose, we only used a subset of the whole dataset. We decide to choose the dataset which has LAX as the destination airport and to predict whether an arriving flight will be delayed.

Historically, similar dataset formats have been used in the past to predict whether a flight will be delayed. One of the most recent and successful model is to use multi-class classification [1]instead of using binary classification. However, instead of just using year 2015 dataset, they have accumulated the past three years' data, and as counted in 2018, they have a whole dataset consisting over 40 millions rows of data. Because of the large amount of data, the team was able to well predict the binary classification. Now, they have turned to build a model which is able to predict the magnitude of a flight's delay, resulting into multi-class classification.

Historically, in order to get a higher and non biased model, Gradient Boosting Classifier [2] has been frequently used, and the intuitive idea is to have a series of decision trees to

correct the significant errors caused by the previous trees. Even though we have similarly deployed a Gradient Boosting Classifier, due to some limitations such as failing to find hidden correlation factors and weather data [2], our model's accuracy score is lower than the one they have provided.

Additionally, some state-of-the-art techniques have very similar approach such as conducting Principle Component Analysis to reduce the feature dimensions. Moreover, the use of feature extracting could potentially help to detect latent factor correlations [6], and this approach is mention in McAuley's paper [5]. However, this part could be done with much more sophisticated analysis. Similarly, some other modern techniques include cross validation to maximize the models' neutrality. Last but not least, it is also possible to improve the overall score by one-hot-encoding airports [3] by categorizing airports based on their delay rates. Doing such can minimize some data noise so the model training will be better.

## 5. Final Results and Conclusions

Overall, the Decision Tree model implementation has an accuracy of 0.62. Notably, it achieves the highest F1 score of 0.573, demonstrating that while it's accuracy is not as high as the of Gradient Boosting Classifier, the model tends to be more balanced because F1 score is an indicator of the harmonized score of both precision and recall. Therefore, Decision Tree would be a better than Gradient Boosting Classifier in real practice.

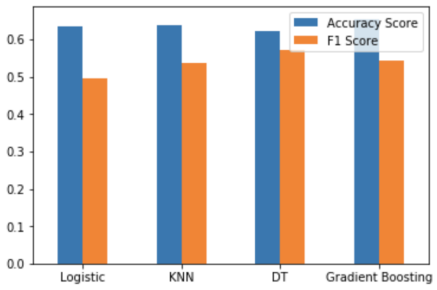



Figure 12: Accuracy Scores and F1 Scores Comparison

For our decision tree model, we performed Grid Search and found that when class weight = 'balanced', criterion = 'entropy, max depth = 25 , min samples split = 300, max leaf nodes =250, the model performed the best. We specifically chose the tree to be balanced because we wanted to make sure the branches would be biased towards one side. In addition, to try to prevent run into problems like overfitting, we restricted our tree depth and max leaf nodes. The result was promising because the accuracy score on the test set actually improved by 0.5%.

In conclusion, several models are proposed including a basic baseline, a logistic regression, a K Nearest Neighbors, a Decision Tree, and a Gradient Boosting Classifier to predict whether a flight will be delayed or not. The logistic regression achieved a fairly high accuracy score but failed the F1 score, and the potential reason is that it failed to remain neutral with the small amount of dataset. Both K Nearest Neighbors and Decision Tree models have a similar accuracy score, but Decision Tree has a higher F1 score due to it is a more balanced tree. Lastly, the Gradient Boosting Classifier does have the highest accuracy score, but it fails to be the most unbiased model among the four. The main reason is that the model trades its low variance with potentially high bias. As a result, we do think Decision Tree is the best model.

## References


[1] I. Cassidy. Applying predictive analytics to flight delays. 2018. 5

[2] N. Chakrabarty. A data mining approach to flight arrival delay prediction for american airlines. CSE, Jalpaiguri Government Engineering College, Jalpaiguri, West Bengal, India, 2011. 5, 6

[3] G. B. C. Eric R. Mueller. Analysis of aircraft arrival and departure delay characteristics. *IEEE Transactions on Industrial Informatics*, 2002. 6

[4] O. Harrison. Machine learning basics with the k-nearest neighbors algorithm. Towards Data Science, 2018. 5

[5] J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013. 6

[6] S. H. R. L. X. L. Q. L. Shaowu Cheng, Yaping Zhang. Study of flight departure delay and causal factor using spatial analysis. 2019:1–11, 2019. 6

[7] Y. L. Yan-yan SONG. Decision tree methods: applications for classification and prediction. pages 130–135. Shanghai Arch Psychiatry, 2015. 4