CMPE 255 – Data Mining
Spring 2018

Team Project Report

Instructor:
Dr. David Anastasiu

(Professor, San Jose state University)

Submitted By: Project Team 3

Hongyuan Li (011838151)
Na Yue (011888981)
Fan Wu (012425660)

Section 1 Introduction

Motivation & Objective

Flight delays bring headache to every passenger. We might wonder if there's an application that can predict flight delay ahead of time, so we can avoid booking tickets for flight going to certain destination, during certain time and by certain airlines. With the help of such application, customers can spend their precious time working on valuable tasks, instead of getting stuck in the airport.

Our project explores the viability of predicting flight delay based on historical flight delay data. We also combines weather data in our experiments, in attempt to improve our prediction. Our goal is to build easy-use application that predicts flight delay.

Section 2 System Design & Implementation details

Algorithm(s) considered/selected (and why)

In this project, our algorithm has three parts:

1. Clustering Airports:

As airports have their attributes which include locations and the amount of arrival flights, we considered to cluster airports. And in this assignment, we predict the arrival delay, so the clustering is by the destination airports. We use K-mean and check the different cluster numbers, then use Silhouette coefficient to pick the optimal number of clusters. To visualize the result of clustering, we use LDA which could preserve class discriminatory information to reduce dimensions to two to plot.

2. Classification using XgBoost and Random Forest

Random Forest is used in the initial analysis of the dataset to evaluate the efficiency of clustering and weather data used for classification. The choice is made because this algorithm is relatively fast to train and exceptionally fast for prediction with relatively high f1 score. XgBoost is used to predict whether a flight will delay or not. As XgBoost is applicable for tabular data set and is efficient for large dataset.

3. Classification added weather information with XgBoost

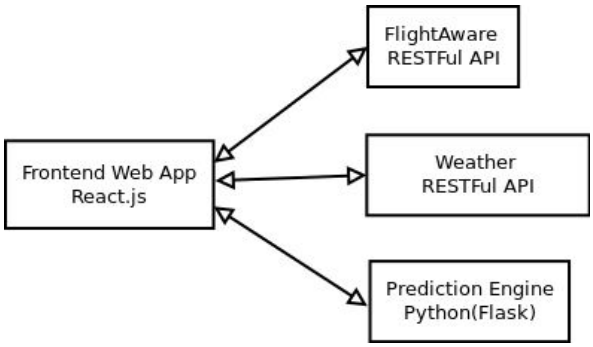
We used XgBoost as the extension of part3 to build models with weather data.

The technologies used which are related to the above algorithms include:

We leveraged well-developed python data mining libraries to tackle our problem. Pandas is used in read data from csv files, Numpy is used to transform pandas table to arrays and matrices. Many algorithms we tried are from the scikit-learn package, including the Random Forest, Decision Tree, K-means, Silhouette coefficient, LDA, and many cross validation function. We also chose XgBoost library as our main algorithm to train and use models. In addition, we

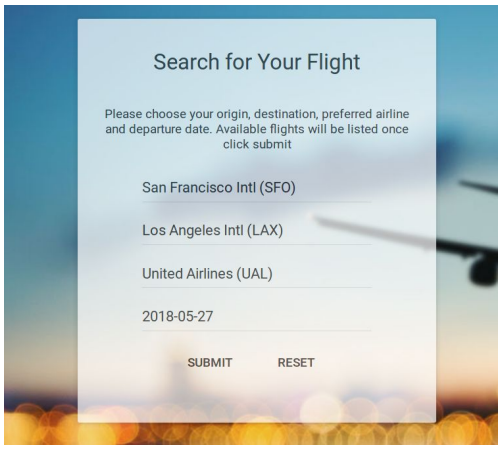
also experimented with Keras neural network API with TensorFlow and LSTM (Long Short-Term Memory layer), though the results aren't as good, because our data isn't inherently sequential.

Application Architecture



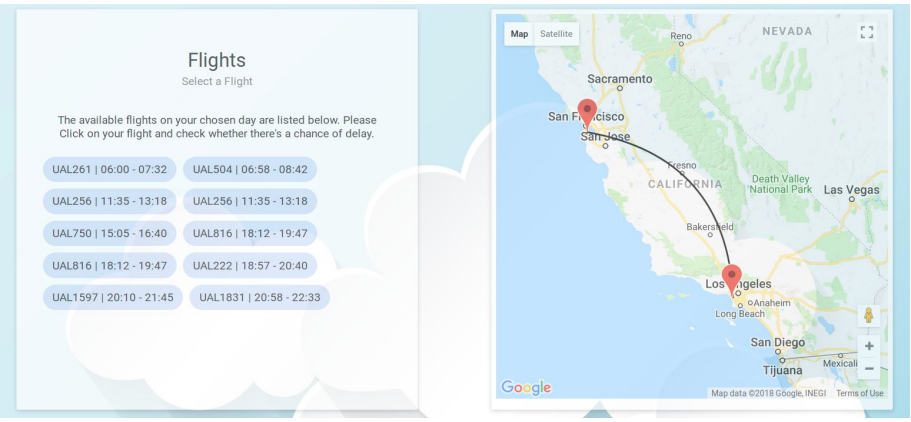
Our Application frontend take user input, including flight origin, destination, airline and date, and query the FlightAware RESTful API to find available flights. After user's selection of a flight schedule, the application queries the weather API for weather forecast or historical weather data. Then our app combines the information provided and send it to the backend Flask prediction engine to offer our delay prediction to the user.

Use cases and GUI



We built an single page application for users to check whether their flight would delay. Users can simply select origin airport, destination airport, airline and departure date. Our app will then send a GET request to FlightAware API to find all available flights.

In the next section, available flights are listed on the left and flight route is plotted on the right. Users can click on one of the chips to select their flight for prediction.



Once a flight is selected, our application contacts the backend weather server to find out weather condition at the destination airport upon the arrival of the plane. Once the weather results returned to the application, the information is combined with other user choices to be sent to our backend prediction engine. Users will receive results on whether their flights will be delayed.

Section 3 Experiments / Proof of concept evaluation

Dataset(s) used

(name, source, type of data, size of data, # of instances/statistics, any preprocessing performed etc.)

1. Delayed Flight Data Set

The Delayed Flight data set comes originally from the U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS) who tracks the on-time performance of domestic flights operated by large air carriers and makes raw data available to public.
The data set has derivable variables removed and is in monthly chunks.

Flight Dataset Introduction
Each entry of the flight_dataset.csv file corresponds to a flight, and 17,111,358 flights have been recorded during 2015 to 2017. These flights are described according to 16 features. A

detailed description of these features can be found from the website of BTS. The below is a brief list of features that will be used in this project:

tab_info

		QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	FL_DATE	UNIQUE_CARRIER	FL_NUM	ORIGIN	DEST	CRS_DEP_TIME	CRS_ARR_TIME	ARR_DEL15	CANCELLED	DIVERTED	DISTANCE
column type	int64	int64	int64	int64	int64	object	object	int64	object	object	int64	int64	float64	float64	float64	float64
null values	0	0	0	0	0	0	0	0	0	0	0	0	279795	0	0	0
null values (%)	0	0	0	0	0	0	0	0	0	0	0	0	1.6351419916525618	0	0	0

- YEAR, MONTH, DAY, DAY_OF_WEEK, FL_DATE: dates of the flight
- UNIQUE_CARRIER: Unique Carrier Code, included 14 major airlines which are ['AA','AS','B6','DL','EV','F9','HA','MQ','NK','OO','UA','US','VX','WN']
- ORIGIN and DEST: code attributed by IATA to identify the airports, included 335 airports
- FL_NUM
- CRS_DEP_TIME and CRS_ARR_Time : scheduled times of take-off and landing(local time: hhmm)
- DISTANCE: distance (in miles)
- ARR_DEL15: Arrival Delay Indicator, 15 Minutes or More
- CANCELLED: Cancelled Flight Indicator
- DIVERTED: Diverted Flight Indicator

Dataset Preprocessing

Step 1. Label entries

The original dataset has features 'ARR_DEL15', 'CANCELLED', and 'DIVERTED', and according to BTS's definition of delay, when the flight's arrival time is 15 minutes or more, this flight would be labeled in feature 'ARR_DEL15'. Furthermore, when a flight is cancelled or diverted, the passengers who order the tickets of the related flights have no chance to arrive on time by that flight. So it is reasonable to label an entry as delay according to the the following condition.(delay = 1, on time = 0)

if (['ARR_DEL15'] > 0) | (['CANCELLED'] == 1) | (['DIVERTED'] == 1) is true,

    labeled the entry as delay

else

    labeled the entry as on time

Step 2. Cleaning dataset

In order to feed data to data mining model, the features UNIQUE_CARRIER, ORIGIN, and DEST have to convert from character data to numeric data, according to the ASCII code. After the cleaning, the dataset is converted as the figure below, and has the size of (17111358, 14).

YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	UNIQUE_CARRIER	FL_NUM	ORIGIN	DEST	CRS_DEP_TIME	CRS_ARR_TIME	AIR_TIME	DISTANCE	LABEL
2015	1	1	22	4	6876	1485	767165	776779	2050	2354	134	950	0
2015	1	1	22	4	6876	1503	678671	776779	1355	1609	102	757	0
2015	1	1	22	4	6876	1509	778380	776779	1112	1527	168	1310	0
2015	1	1	22	4	6876	1510	778380	776779	1918	2328	160	1310	0
2015	1	1	22	4	6876	1585	767165	776779	1829	2142	142	950	0
2015	1	1	22	4	6876	1669	658476	776779	1855	2026	56	404	0
2015	1	1	22	4	6876	1685	767165	776779	1404	1717	141	950	0

Step 3. Split dataset into four categories

As the airports are clustered into four parts, the flight dataset is also splitted into four parts by the feature “DEST”.

2. Weather Dataset

As the experimental part of this project, we want to study the impact of weather conditions on flight delay. Intuitively, we would consider that poor weather conditions, say, heavy rain, strong snow storm or low visibility can have negative effect on flight schedule. Our experiments support this hypothesis to certain degree.

The National Oceanic and Atmospheric Administration (NOAA) hosts a large hourly weather dataset, which is generated by the Automated Surface Observing System (ASOS). This dataset, Integrated Surface Data (ISD), is defined with a complex specification, thus every data contains abundant information. For the purpose of this project, we are mostly interested in time, location, wind, temperature and visibility. These attributes can be easily extracted from the data.

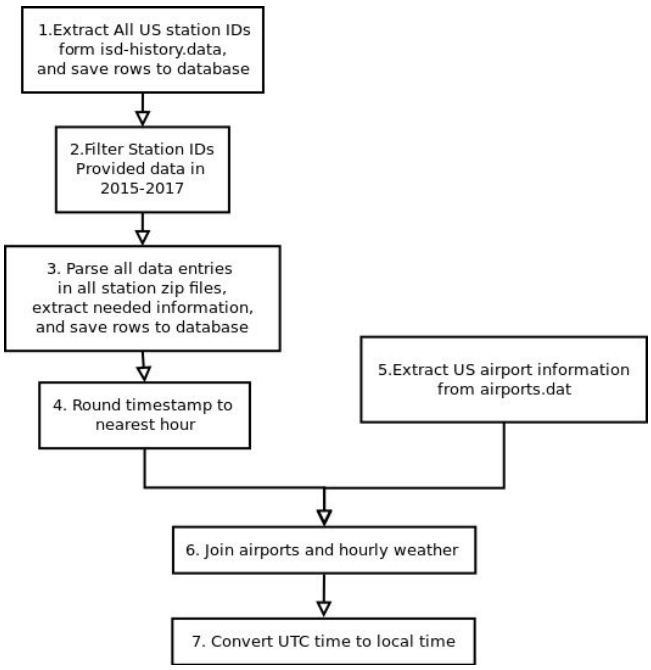
Example:

0248745090232442008010100004+37417-122050FM-15+0012KNUQ
V020360 1N0103 12200019N016093 1N1+0130 1-00601102391ADDGF100991999999999999999
9999KA1999M+01391KA2999N+01061MA1102371999999MD1610161+9999REMMET120ME
TAR KNUQ 312356Z AUTO 36020KT 10SM CLR 13/M06 A3023 RMK AO2 PK WND
36027/2311 SLP239 T01281061 10139 20106 56016 TSNO;EQDQ01 993SCCGA1Q02
099SCCGD1

Position	Value	Meaning
16-23	20080101	Date: 2008/01/01
24-27	0000	Time: 00:00
28	4	USAF Surface Hourly
29-34	+37417	Latitude: +37.417
35-41	-122050	Longitude: -122.050
61-63	360	Wind angle: 360 degree
66-69	0103	10.3 m/s
79-84	016093	Visibility: 16093m
88-92	+0130	Temperature: 13 Celsius

In ADD section, following the mandatory portion, we extract AA1 item for hourly liquid depth and AJ1 item for hourly snow depth. Null values replaced with 0.

To maintain a proper level of workload, we will only process weather data from latest 3 years (2015 - 2017). The data size is around 13GB. Our data pre-processing will only select US weather stations and extract a handful attributes discussed above.



The source dataset is organized by weather stations per year. Each zip file contains the hourly weather data files on each day of a particular weather station in given year. Each zip file is named by USAF and WBAN code, which can be used as filter conditions. The input dataset contains airport around the globe, and therefore should be removed from our data processing. Weather data preprocessing is shown in the flowchart on left.

Step 1. Load all weather stations within United States from isa-history.dat file (part of the metadata of this dataset), with information including, USAF code, WBAN code, latitude, longitude and station name. The extracted data is saved in MySQL database

Step 2. We also need find stations that offers weather information between 2015 and 2017. Many weather stations listed in isa-history.dat file only records for certain years. Our strategy is to traverse through the zip files names to ensure the availability.

Step 3. Once the list of weather stations is finalized, our python algorithm read through all zip files and data entries line by line. Parse each entry and extract useful information discussed earlier. More than 9700K rows are saved in MySQL WEATHER table.

Step 4. Weather data gather is gather on 1 hour intervals, but not necessarily on the hour. SQL Query is used to update round the timestamp of each entry to the nearest hour

3. Airport Dataset

The goal of our weather data processing, is to find out the weather data at each US airport at a given hour between 2015 and 2017. Thus it is necessary to join airport data with our weather data, by calculating great circle distance.

Our airport dataset is taken from Open Flights: https://openflights.org/data.html.

Step 5. Extracting airport IATA, longitude, latitude, and timezone information from airports.dat file and saving data to MySQL database.
Step 6. Join airports and weather data. To achieve this join, we use the latitude and longitude fields on both datasets, calculating the great circle distance with filter condition distance < 15 miles. Joined dataset is aggregated with average, group by airport and hourly timestamp. After this join, 2979K rows are saved in the newly created AIRPORT_WEATHER table.

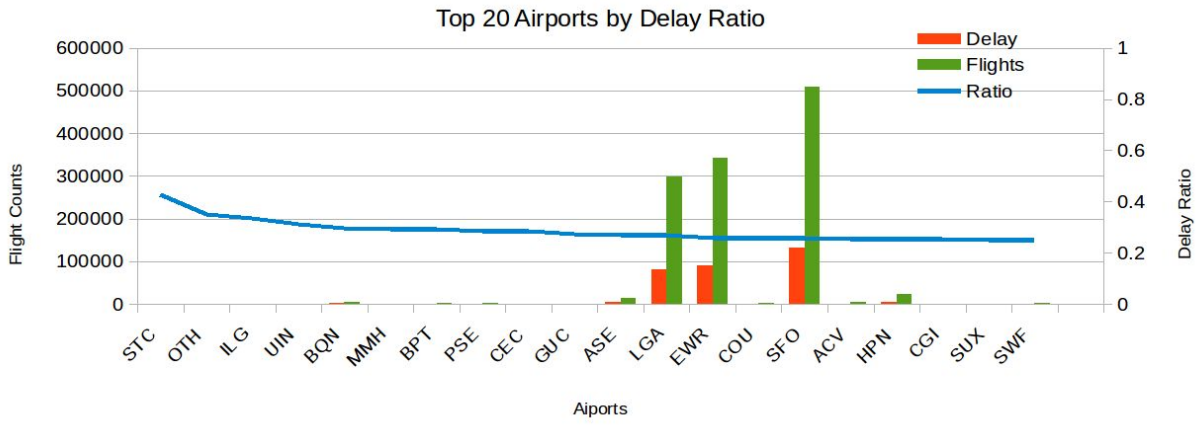
Step 7. Timestamp given in NOAA is GMT, we created a new field local_ts to stored the converted local timestamp based on airport location. Therefore we can eventually join the weather dataset with the flight dataset where timestamps are local by nature.

Methodology followed

1. Delayed Flight Data Set

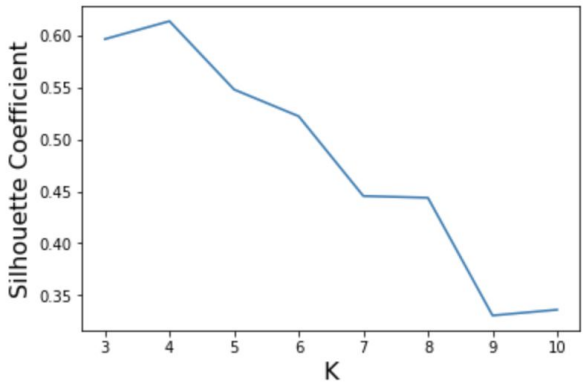
After the preprocessing of dataset, we use 5-fold-cross validation.
The number of folds is five. The size of the training set is (13689086, 13). And the size of the test set is (3422272, 12). The test set is separated from its labels to evaluate the prediction result.

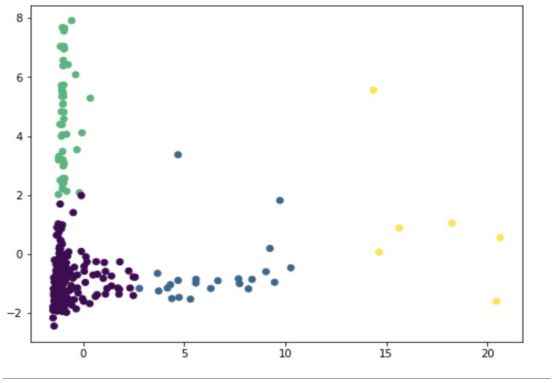
2. Airport Data Set

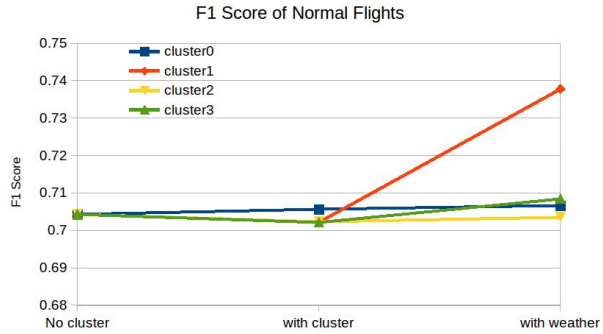


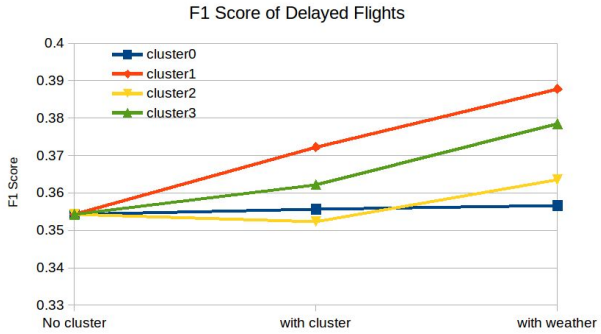
Our Analysis show that flight delay are not necessarily a feature owned solely by large airports. From the top 20 airports by delay ratio, we can see that many small airports with a small number of flights schedule every year can also have high delay rate. From this charts we notice that LaGuardia Airport (LGA), Newark Airport(EWR) in New Jersey and San Francisco International Airport (SFO) also made to top 20. These large airports have features drastically different from small airports. Therefore it is necessary to group similar airports together before we attempt delay prediction to achieve higher F1 score.

We use K-mean to cluster airport and calculate the Silhouette Coefficient of different number of clusters, and then choose the optimal number K = 4, which is shown below on the left. The visualization of the airports clusters is shown below on the right.









We experimented with airport clustering and additional weather features, using Random Forest algorithm and same parameters. From the chart only the left, airport clustering and weather features have limited effect on the f1 score of normal flights, except for cluster 1. Cluster 1 includes mostly large airports. An interesting outlier. On the other side, from the chart on the right, we can see that clustering and weather obviously boosted the F1 scores for predicting delayed flights.

Analysis of results

1. XgBoost:

model selected by cross validation

	common params	gamma	learning rate	min child weight	max depth
cluster 0	base_score=0.5, booster='gbtree', colsample_bylevel=1, objective='binary:logistic',	5.71	0.32	42.50	27
cluster 1		5.68	0.24	1.58	26
cluster 2		5.91	0.35	45.19	28
cluster 3		5.60	0.35	5.32	11

Results of every cluster and without clustering

	XgBoost with clustering				XgBoost without clustering
	cluster 0	cluster 1	cluster 2	cluster 3	entire dataset
F1 score(on time)	0.9	0.9	0.9	0.9	0.83
F1 score(delay)	0.26	0.28	0.33	0.12	0.44
F1 scoure（Total)	0.88	0.87	0.86	0.88	0.76
Count(delay)	956453	876288	1399597	80502	3312840
Count(total)	5058034	4402501	7120476	426316	17007327
Percentage(delay)	18.91%	19.90%	19.66%	18.88%	19.48%

Section 4 Discussion & Conclusions (bullet points as applicable)

- Decisions made
    - Flight Delay Feature Selection: For the original flight delay dataset, some redundant features exist. 1. The feature “QUARTER” can be represented by feature “MONTH”; 2. The feature “AIR_TIME” is related to “DISTANCE”; 3. The feature “FLI_NUM” can be deducted by other features. So we delete features “QUARTER”, “AIR_TIME”, and “FLI_NUM”.
    - Data Store: MySQL. One challenge of this project is to join different datasets together for analysis. We leveraged MySQL relational database engine, a very power tool for relational join. We also added indexes on the joining columns on tables to speed up join operation.

- Difficulties faced

    - Large dataset. For weather data, we started with 12 GB data, loaded and parsed into a table with nearly 1 billion rows. After multiple inner joins, discarding irrelevant records, we still have more than 1100K rows data work with.
    - Algorithm selection.
    - Dataset preparation. Start with raw data from flight status database, to our own SQL database to data for model building, multiple preparations involved.
- Things that worked
    - Clustering Strategy. We clustered airports by their flight delay counts, total flight counts, delay ratio, longitude and latitude. Then we built models for each cluster. Our analysis shows that clustering helps improves the F1 score our prediction. Further, it reduces the amount data needed to build each model, thus saves time
    - Weather Features. By adding weather data, our models' f1 scores improved.

- Conclusion

We learned a lot from working on this project, tackled a big data issue, experimented with various algorithms, built an application that can improve our travel experience.

Section 5 Project Plan / Task Distribution

Member	Contribution
Na Yue	Clustered Airport Dataset. Collected and cleaning flight delay data. Analyzed the correspondence of flight delay data and weather data. Training models with flight delay data using XgBoost. Participated the discussion of processing strategy.
Fan Wu	Preprocessed flight delay data from 201607 - 201712. Worked on Flask backend. Built SVC/KNN/decision tree model for classification model comparison.
Hongyuan Li	Collected and parsed weather data. Joined weather dataset with flight dataset. Set up AWS RDS MySQL for data storage. Analyzed our processing strategy (with clustering and weather) using Random Forest. Trained models with weather data using XgBoost. Built Application Frontend.