ETL PROJECT REPORT
2015 Flight Delays and Cancellations Data Analysis

November 26, 2020

Carol Figueroa
Meixi Yu
UCF Data Visualization Bootcamp

Contents

0.1 Project Roadmap

The main task of the project is to explore 2015 flights delays and cancellation and compare the results with another data set where people tweets about their flights experience.

1. The first step of the project is **extract**. To find the appropriate data from different sources and brainstorm what we want to know from them, we browsed through some pages on kaggle.com and decided on the Flights data and Tweets data, where both specify their airlines, and dates. Since both datasets are from 2015, we see the potentials to understand customers' tweets behaviors regarding the delays and cancellations within that year.
   We have 5 files:

   - Airlines has information about airline id (IATA Code) and airline name;
   - Airports has information about airport id, airport, city, state, country, latitude, longitude;
   - Flights contains information about date, day of the week,airline id, flight number,tail number,origin airport, destination airport, scheduled departure, actual departure time,delay(in minutes), scheduled arrival, Arrival delay, and cancellation.
   - Cancellation code file provides information about cancellation reason.
   - And finally, tweets has information about tweets regarding the flight experience,airline, attitude date of tweets and etc.

2. The second step of the project is to **transform**. To do so, we did the following:

   - Use Jupyter notebook to load the files, and use pandas to convert them into data-frames.
   - By checking the column titles and the size of each data-frame, understand what's the main information in each of them.
   - Identify the main data frame, check and update null value cells.
   - Subset the dataframe and update labels within the columns.
   - Convert columns about date to appropriate form.
   - Merge airline data with flights data for readability.
   - Separate delays and cancellation data frame.

- Update tweets data' airline name to match with previous labels.
- Fill NaN cells with texts or 0.

3. The third step of the project is to **load**. We use sqlalchemy to create engine and connect to sql database. The data are now saved as tables.

References

https://www.kaggle.com/gauravmehta13/airline-flight-delays
https://www.kaggle.com/crowdflower/twitter-airline-sentiment

0.2 Project Updates

Update when anything related to the roadmap is addressed

- Nov 21, 2020. Completed the step 1.
- Nov 23, 2020. Completed the step 2.
- Nov 25, 2020. Completed the step 3.

0.3 Results

Main Results can be found on github (Checkout https://github.com/Masiesworld/ETL-challenge)

0.4 Things We wish we Knew

1. It will be better to have feedback at the end of each step during the project.
2. We would like to know how to decide what's the right one to start (especially within such a short time span). We rushed a little bit because we don't want to waste our nine hours.
3. We stuck at the merging step. How to merge the datasets when one is much bigger than the other (with only two columns in common)?

0.5 Process Diary

```python
import pandas as pd
import numpy as np
from sqlalchemy import create_engine
import sqlite3

airline_file = "./Resources/airlines.csv"
airline_df = pd.read_csv(airline_file)
airline_df.columns = ['AIRLINE_ID','AIRLINE']

airline_df
```

Listing 1: Python codes

	AIRLINE_ID	AIRLINE
0	UA	United Air Lines Inc.
1	AA	American Airlines Inc.
2	US	US Airways Inc.
3	F9	Frontier Airlines Inc.
4	B6	JetBlue Airways
5	OO	Skywest Airlines Inc.
6	AS	Alaska Airlines Inc.
7	NK	Spirit Air Lines
8	WN	Southwest Airlines Co.
9	DL	Delta Air Lines Inc.
10	EV	Atlantic Southeast Airlines
11	HA	Hawaiian Airlines Inc.
12	MQ	American Eagle Airlines Inc.
13	VX	Virgin America

Figure 1: This is an image of airline data

```python
airports_file = "./Resources/airports.csv"
airports_df = pd.read_csv(airports_file)
airports_df.columns =['AIRPORT_ID','AIRPORT','CITY','STATE','COUNTRY','
    LATITUDE','LONGITUDE']
# airports_df.set_index('AIRPORT_ID',inplace=True)
airports_df.head()
```

Listing 2: Python codes

```python
flights_file = "./Resources/flights.csv"
```

	AIRPORT_ID	AIRPORT	CITY	STATE	COUNTRY	LATITUDE	LONGITUDE
0	ABE	Lehigh Valley International Airport	Allentown	PA	USA	40.65236	-75.44040
1	ABI	Abilene Regional Airport	Abilene	TX	USA	32.41132	-99.68190
2	ABQ	Albuquerque International Sunport	Albuquerque	NM	USA	35.04022	-106.60919
3	ABR	Aberdeen Regional Airport	Aberdeen	SD	USA	45.44906	-98.42183
4	ABY	Southwest Georgia Regional Airport	Albany	GA	USA	31.53552	-84.19447

Figure 2: This is an image of airport data

```
flights_df = pd.read_csv(flights_file)
flights_df.head()
```

Listing 3: Python codes

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	...
0	2015	1	1	4	AS	98	N407AS	ANC	SEA	5	...
1	2015	1	1	4	AA	2336	N3KUAA	LAX	PBI	10	...
2	2015	1	1	4	US	840	N171US	SFO	CLT	20	...
3	2015	1	1	4	AA	258	N3HYAA	LAX	MIA	20	...
4	2015	1	1	4	AS	135	N527AS	SEA	ANC	25	...

Figure 3: This is an image of flights data

```
# Drop rows of NaN
# There isn't a row that's completely NaN
all_nan_df = flights_df[flights_df.isna().all(axis=1)]

# However, for the flights that's not delayed/ cancelled, those columns are NaN
any_nan_df = flights_df[flights_df.isna().any(axis=1)]

# For readibility, we replace NaN with 0
flights_df = flights_df.fillna(0)
flights_df[flights_df.isna().any(axis=1)]

# DROP DELAY columns
flights_df = flights_df.drop(columns=['AIR_SYSTEM_DELAY', 'SECURITY_DELAY', 'AIRLINE_DELAY',
                                      'LATE_AIRCRAFT_DELAY','WEATHER_DELAY'])

# Merge DateTime to one column
flights_df['DATE'] = pd.to_datetime(flights_df[['YEAR','MONTH','DAY']])
```

```
flights_df = flights_df.drop(columns=['YEAR','MONTH','DAY'])

# Change Day of week from num to string
flights_df.loc[flights_df["DAY_OF_WEEK"] == 1, "DAY_OF_WEEK"] = "Monday"
flights_df.loc[flights_df["DAY_OF_WEEK"] == 2, "DAY_OF_WEEK"] = "Tuesday"
flights_df.loc[flights_df["DAY_OF_WEEK"] == 3, "DAY_OF_WEEK"] = "Wednesday"
flights_df.loc[flights_df["DAY_OF_WEEK"] == 4, "DAY_OF_WEEK"] = "Thursday"
flights_df.loc[flights_df["DAY_OF_WEEK"] == 5, "DAY_OF_WEEK"] = "Friday"
flights_df.loc[flights_df["DAY_OF_WEEK"] == 6, "DAY_OF_WEEK"] = "Saturday"
flights_df.loc[flights_df["DAY_OF_WEEK"] == 7, "DAY_OF_WEEK"] = "Sunday"

flights_air = pd.merge(flights_df,airline_df,left_on='AIRLINE',right_on='
    AIRLINE_ID',how='left')
flights_air = flights_air.drop(columns=['AIRLINE_ID'])
flights_air = flights_air.rename(columns={'AIRLINE_x': 'AIRLINE_ID', '
    AIRLINE_y': 'AIRLINE'})
flights_air.head(5)
```

Listing 4: Transformation

```
# The total df of delay flights
arrival_Delay = flights_air.loc[flights_air['ARRIVAL_DELAY'] > 0].
    reset_index(drop=True)  # negative value means arrive earlier
departure_Delay=flights_air.loc[flights_air['DEPARTURE_DELAY']> 0].
    reset_index(drop=True)
#  for dep_Delay: 2125618 rows
#  for arr_Delay: 2086896 rows

# Both arrival and departure are delayed
delay_Flights = flights_air.loc[(flights_air['ARRIVAL_DELAY'] > 0) & (
    flights_air['DEPARTURE_DELAY'] > 0)].reset_index(drop=True)
delay_Flights
# 1508147 total
# Further Narrow down the dataframe
delay_Flights = delay_Flights[['AIRLINE_ID','AIRLINE','DATE','DAY_OF_WEEK',
    'TAIL_NUMBER','FLIGHT_NUMBER','ORIGIN_AIRPORT',
                                   'DESTINATION_AIRPORT','DEPARTURE_DELAY
    ','ARRIVAL_DELAY']]
delay_Flights.head(5)
```

Listing 5: Subset the delays

```
# Which Airline has the most Delays?
```

	AIRLINE_ID	AIRLINE	DATE	DAY_OF_WEEK	TAIL_NUMBER	FLIGHT_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT	DEPARTURE_DELAY	ARRIVAL_DI
0	NK	Spirit Air Lines	2015-01-01	Thursday	N525NK	520	LAS	MCI	25.0	
1	NK	Spirit Air Lines	2015-01-01	Thursday	N528NK	597	MSP	FLL	12.0	
2	AA	American Airlines Inc.	2015-01-01	Thursday	N3HRAA	2392	DEN	MIA	21.0	
3	NK	Spirit Air Lines	2015-01-01	Thursday	N629NK	168	PHX	ORD	72.0	
4	B6	JetBlue Airways	2015-01-01	Thursday	N307JB	2134	SJU	MCO	95.0	

Figure 4: This is an image of delayed flights data

```
delay_Flights.groupby('AIRLINE').count().sort_values(by='FLIGHT_NUMBER',
    ascending=False)['FLIGHT_NUMBER']
```

Listing 6: Delays by Airline

```
AIRLINE
Southwest Airlines Co.          391005
Delta Air Lines Inc.            175530
American Airlines Inc.          172258
United Air Lines Inc.           156697
Skywest Airlines Inc.           139880
Atlantic Southeast Airlines     138185
JetBlue Airways                  75849
American Eagle Airlines Inc.     74409
US Airways Inc.                  46519
Spirit Air Lines                 43033
Alaska Airlines Inc.             31541
Frontier Airlines Inc.           29070
Hawaiian Airlines Inc.           17594
Virgin America                   16577
Name: FLIGHT_NUMBER, dtype: int64
```

Figure 5: This is an image of delayed flights by airline

```

flights_air['CANCELLED'].value_counts()

cancelled_flights = flights_air.loc[flights_air['CANCELLED']==1].
    reset_index(drop=True)
# Count total number of cancelled flights
cancelled_flights.count()['CANCELLED'] # 89884
# Remove irrelevant columns
cancelled_flights = cancelled_flights[['AIRLINE_ID','AIRLINE','DATE','
    DAY_OF_WEEK','TAIL_NUMBER','FLIGHT_NUMBER','ORIGIN_AIRPORT',
                                       'DESTINATION_AIRPORT','DEPARTURE_DELAY
    ','ARRIVAL_DELAY']]
cancelled_flights.head(5)
```

```
# Which Airline has the most cancallations?
cancelled_flights.groupby('AIRLINE').count().sort_values(by='FLIGHT_NUMBER'
    , ascending=False)['FLIGHT_NUMBER']
```

Listing 7: cancellations by Airline

```
AIRLINE
Southwest Airlines Co.          16043
Atlantic Southeast Airlines     15231
American Eagle Airlines Inc.    15025
American Airlines Inc.          10919
Skywest Airlines Inc.            9960
United Air Lines Inc.            6573
JetBlue Airways                  4276
US Airways Inc.                  4067
Delta Air Lines Inc.             3824
Spirit Air Lines                 2004
Alaska Airlines Inc.              669
Frontier Airlines Inc.            588
Virgin America                    534
Hawaiian Airlines Inc.            171
Name: FLIGHT_NUMBER, dtype: int64
```

Figure 6: This is an image of cancelled flights by airline

```
tweets_file = "./Resources/Tweets.csv"
tweets_df = pd.read_csv(tweets_file)
tweets_df = tweets_df[['tweet_id','airline_sentiment','negativereason','
    airline','tweet_created']]
# Update the airline Name so it matches the flights_df
tweets_df.loc[tweets_df["airline"] == "American", "AIRLINE"] = "American
    Airlines Inc."
tweets_df.loc[tweets_df["airline"] == "Delta",    "AIRLINE"] = "Delta Air
    Lines Inc."
tweets_df.loc[tweets_df["airline"] == "Southwest","AIRLINE"] = "Southwest
    Airlines Co."
tweets_df.loc[tweets_df["airline"] == "US Airways","AIRLINE"] = "US Airways
     Inc."
tweets_df.loc[tweets_df["airline"] == "United", "AIRLINE"] = "United Air
    Lines Inc."
tweets_df.loc[tweets_df["airline"] == "Virgin America", "AIRLINE"] = "
    Virgin America"

# Clean the tweet_created column to get the Date
tweets_df['DATE'] = pd.to_datetime(tweets_df['tweet_created'])
tweets_df['DATE']=pd.DatetimeIndex(tweets_df['DATE']).date
tweets_df = tweets_df.drop(columns=['tweet_created','airline'])

# replace NaN with "No comments"
```

```
tweets_df['negativereason'] = tweets_df['negativereason'].fillna('No
    comments')
tweets_dft
```

Listing 8: tweets

	tweet_id	airline_sentiment	negativereason	AIRLINE	DATE
0	570306133677760513	neutral	No comments	Virgin America	2015-02-24
1	570301130888122368	positive	No comments	Virgin America	2015-02-24
2	570301083672813571	neutral	No comments	Virgin America	2015-02-24
3	570301031407624196	negative	Bad Flight	Virgin America	2015-02-24
4	570300817074462722	negative	Can't Tell	Virgin America	2015-02-24
...	...	...	...	...	...
14635	569587686496825344	positive	No comments	American Airlines Inc.	2015-02-22
14636	569587371693355008	negative	Customer Service Issue	American Airlines Inc.	2015-02-22
14637	569587242672398336	neutral	No comments	American Airlines Inc.	2015-02-22
14638	569587188687634433	negative	Customer Service Issue	American Airlines Inc.	2015-02-22
14639	569587140490866689	neutral	No comments	American Airlines Inc.	2015-02-22

14640 rows × 5 columns

Figure 7: This is an image of tweets data

```
tweets_df.groupby(['AIRLINE','airline_sentiment']).count()
# Apparently, United AirLines has the most negative reviews
# Virgin America has the least nagative reviews
```

Listing 9: tweets

AIRLINE	airline_sentiment	tweet_id	negativereason	DATE
American Airlines Inc.	negative	1960	1960	1960
	neutral	463	463	463
	positive	336	336	336
Delta Air Lines Inc.	negative	955	955	955
	neutral	723	723	723
	positive	544	544	544
Southwest Airlines Co.	negative	1186	1186	1186
	neutral	664	664	664
	positive	570	570	570
US Airways Inc.	negative	2263	2263	2263
	neutral	381	381	381
	positive	269	269	269
United Air Lines Inc.	negative	2633	2633	2633
	neutral	697	697	697
	positive	492	492	492
Virgin America	negative	181	181	181
	neutral	171	171	171
	positive	152	152	152

Figure 8: This is an image of tweets data groupby reviews and airline

Connect to MongoDB

```
]: import pymysql
   pymysql.install_as_MySQLdb()
   import pymongo
   import MySQLdb
```

```
]: conn = 'mongodb://localhost:27017'
   client = pymongo.MongoClient(conn)
```

```
]: db = client.flights_db
```

```
]: # Insert tweets data
   db.tweets.insert_many(tweets_df.to_dict('records'))
```

```
]: <pymongo.results.InsertManyResult at 0x7fe4144f9b40>
```

```
]: # db.tweets.find()
   for index,row in tweets_df.iterrows():
       print(row)
       db.tweets.insert_one(dict(row))
```

```
airline                         Virgin America
tweet_created       2015-02-24 11:15:36 -0800
AIRLINE                         Virgin America
Name: 3, dtype: object
tweet_id                  570300817074462722
airline_sentiment                   negative
negativereason                   Can't Tell
airline                         Virgin America
tweet_created       2015-02-24 11:14:45 -0800
AIRLINE                         Virgin America
Name: 4, dtype: object
tweet_id                  570300767074181121
airline_sentiment                   negative
negativereason                   Can't Tell
airline                         Virgin America
tweet_created       2015-02-24 11:14:33 -0800
AIRLINE                         Virgin America
Name: 5, dtype: object
tweet_id                  570300616901320704
airline_sentiment                   positive
```

```
]: # Airports

   db.airports.insert_many(airports_df.to_dict('records'))

   for index,row in airports_df.iterrows():
       print(row)
       db.airports.insert_one(dict(row))
```

Figure 9: This is an image of connecting to MongoDB

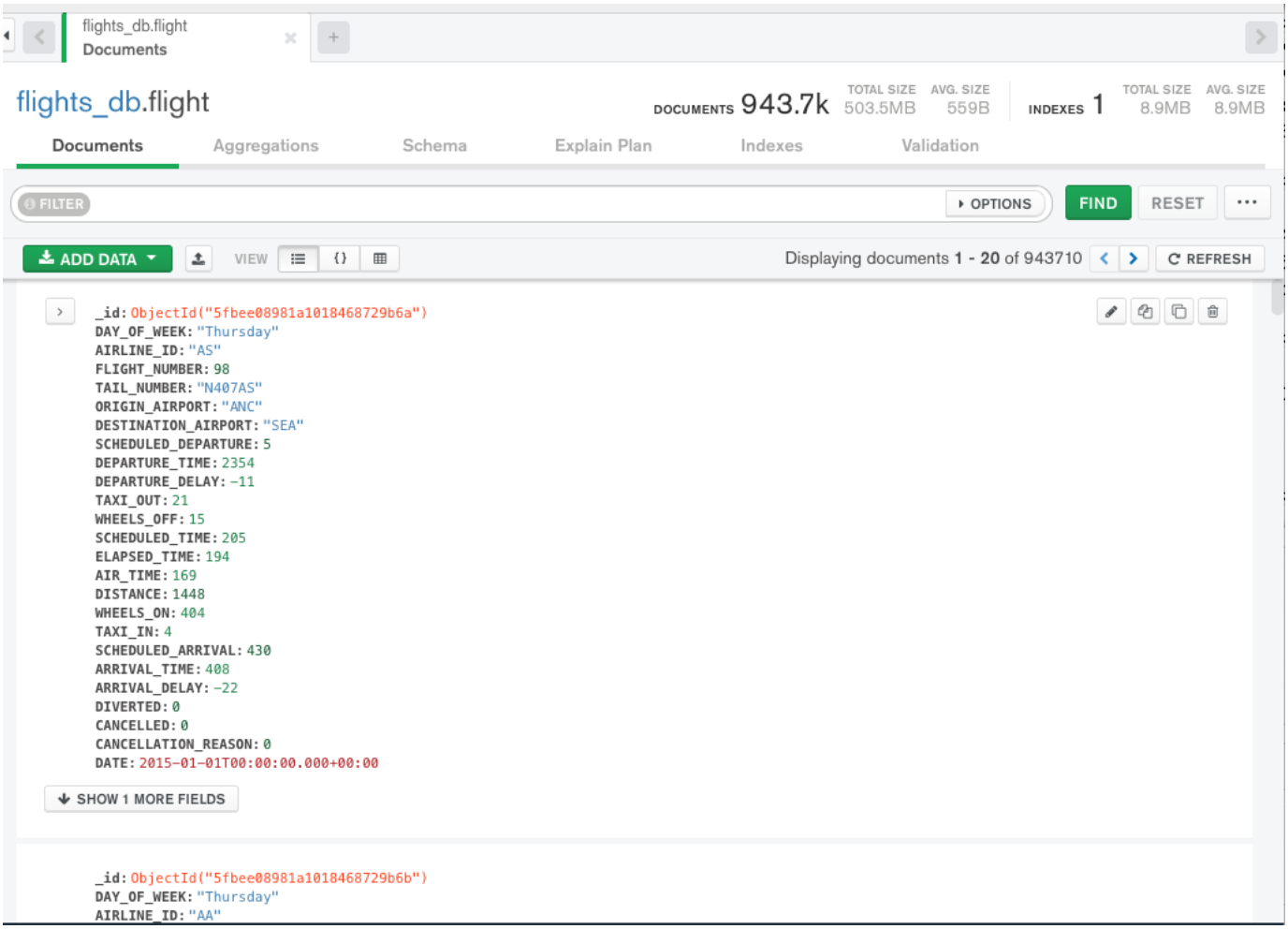

Figure 10: This is an image of MongoDB page