2015 Flights - Data Analysis

Trials by Fire II

Cole Baugh & Brey Rivera

12.19.2022

Contents

01.

Data Origin

Yes, it’s from Kaggle

2015 Flight Delays and Cancellations

Kaggle.com

This dataset is provided publicly by the Department of Transportation

https://www.kaggle.com/datasets/usdot/flight-delays?select=flights.csv

What data are we using?

	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	...

flights.csv

- 5,819,079 records
- 31 columns

What data are we using?

	IATA_CODE	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

airlines.csv

- 14 records
- 2 columns

What data are we using?

	IATA_CODE	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

airports.csv

- 322 records
- 7 columns

02.

Research Questions

#1

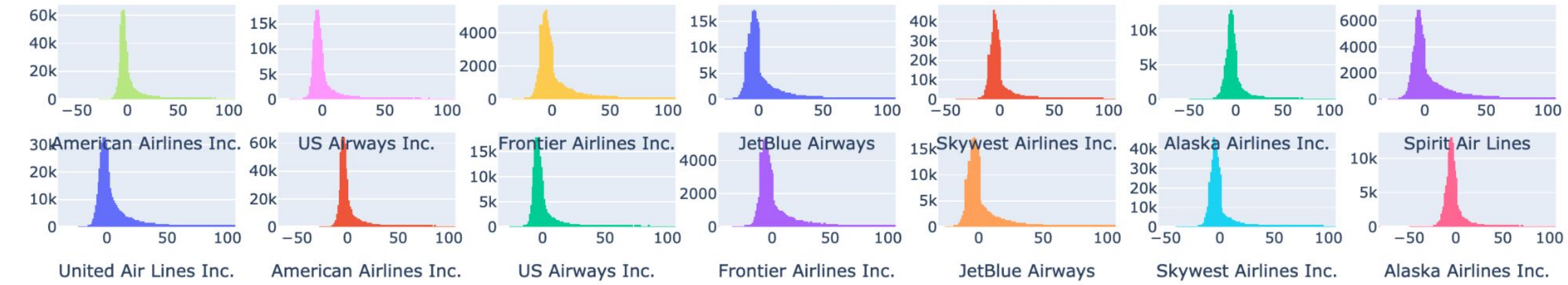
What is the worst airline to fly when it comes to delays?

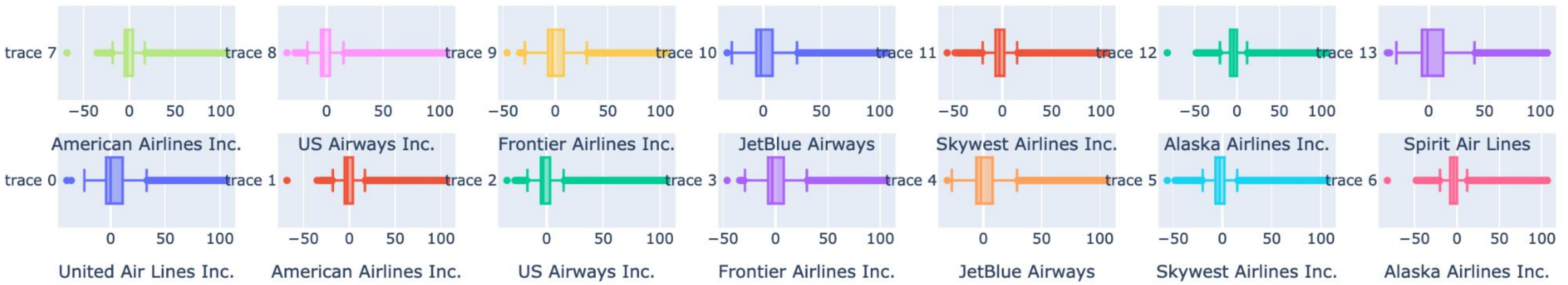
Looking at Departure Delays

	DEPARTURE_DELAY							
	count	mean	std	min	25%	50%	75%	max
AIRLINE								
AA	715598.0	8.900856	41.897429	-68.0	-5.0	-2.0	5.0	1988.0
AS	171910.0	1.785801	26.365575	-82.0	-8.0	-4.0	1.0	963.0
B6	262843.0	11.514353	38.517935	-31.0	-5.0	-2.0	11.0	1006.0
DL	872177.0	7.369254	36.337405	-61.0	-4.0	-1.0	4.0	1289.0
EV	557294.0	8.715934	38.680279	-55.0	-6.0	-3.0	4.0	1274.0
F9	90290.0	13.350858	49.510902	-46.0	-7.0	-2.0	12.0	1112.0
HA	76119.0	0.485713	24.550609	-27.0	-7.0	-4.0	1.0	1433.0
MQ	280282.0	10.125188	40.615207	-36.0	-6.0	-2.0	8.0	1544.0
NK	115454.0	15.944766	43.767651	-37.0	-5.0	-1.0	18.0	836.0
OO	579086.0	7.801104	37.807475	-56.0	-6.0	-3.0	4.0	1378.0
UA	509534.0	14.435441	42.055788	-40.0	-4.0	1.0	13.0	1314.0
US	194825.0	6.141137	29.023259	-35.0	-5.0	-2.0	4.0	759.0
VX	61385.0	9.022595	32.424981	-24.0	-4.0	-1.0	7.0	644.0
WN	1246129.0	10.581986	30.738912	-28.0	-3.0	0.0	11.0	665.0

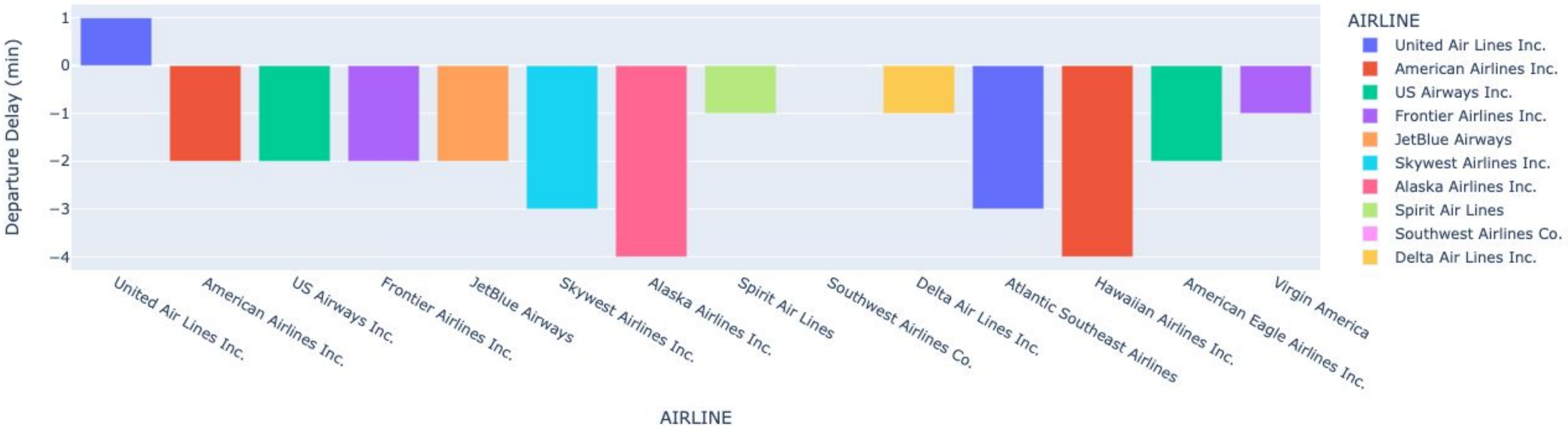
Looking at Departure Delays

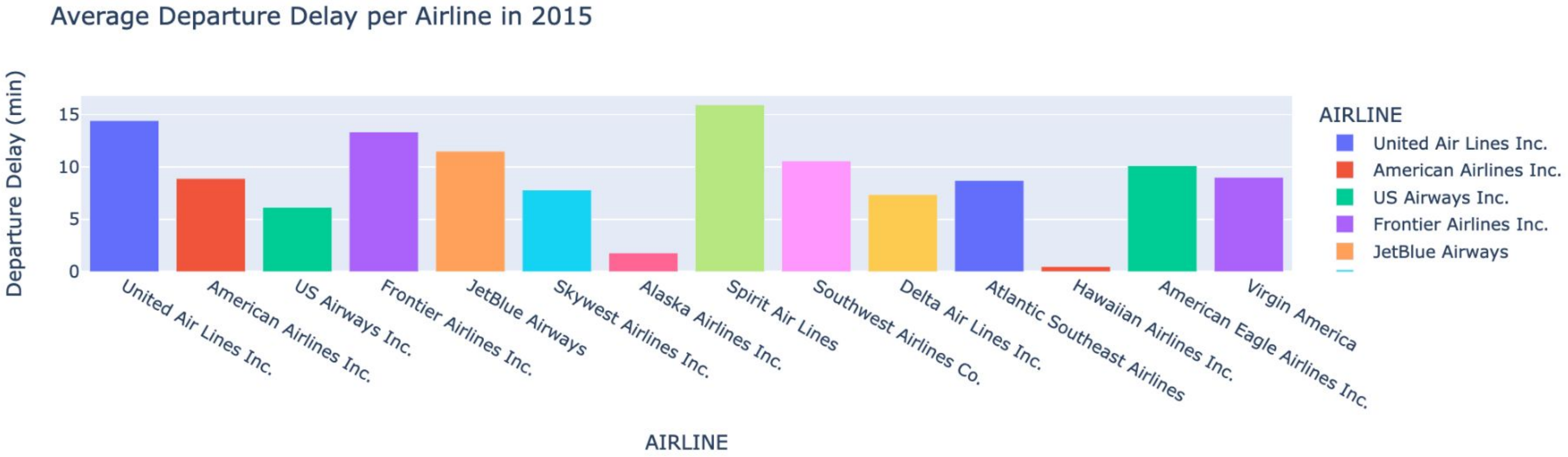
	IATA_CODE	AIRLINE	upper_outlire_bound	lower_outlire_bound
0	UA	United Air Lines Inc.	38.5	-24.5
1	AA	American Airlines Inc.	20.0	-17.0
2	US	US Airways Inc.	17.5	-15.5
3	F9	Frontier Airlines Inc.	40.5	-30.5
4	B6	JetBlue Airways	35.0	-26.0
5	OO	Skywest Airlines Inc.	19.0	-18.0
6	AS	Alaska Airlines Inc.	14.5	-17.5
7	NK	Spirit Air Lines	52.5	-35.5
8	WN	Southwest Airlines Co.	32.0	-21.0
9	DL	Delta Air Lines Inc.	16.0	-13.0
10	EV	Atlantic Southeast Airlines	19.0	-18.0
11	HA	Hawaiian Airlines Inc.	13.0	-16.0
12	MQ	American Eagle Airlines Inc.	29.0	-23.0
13	VX	Virgin America	23.5	-17.5


American Airlines Inc.
US Airways Inc.
Frontier Airlines Inc.
JetBlue Airways
Skywest Airlines Inc.
Alaska Airlines Inc.
Spirit Air Lines
United Air Lines Inc.
American Airlines Inc.
US Airways Inc.
Frontier Airlines Inc.
JetBlue Airways
Skywest Airlines Inc.
Alaska Airlines Inc.


trace 7
trace 8
trace 9
trace 10
trace 11
trace 12
trace 13
American Airlines Inc.
US Airways Inc.
Frontier Airlines Inc.
JetBlue Airways
Skywest Airlines Inc.
Alaska Airlines Inc.
Spirit Air Lines
trace 0
trace 1
trace 2
trace 3
trace 4
trace 5
trace 6
United Air Lines Inc.
American Airlines Inc.
US Airways Inc.
Frontier Airlines Inc.
JetBlue Airways
Skywest Airlines Inc.
Alaska Airlines Inc.

Median Departure Delay per Airline in 2015





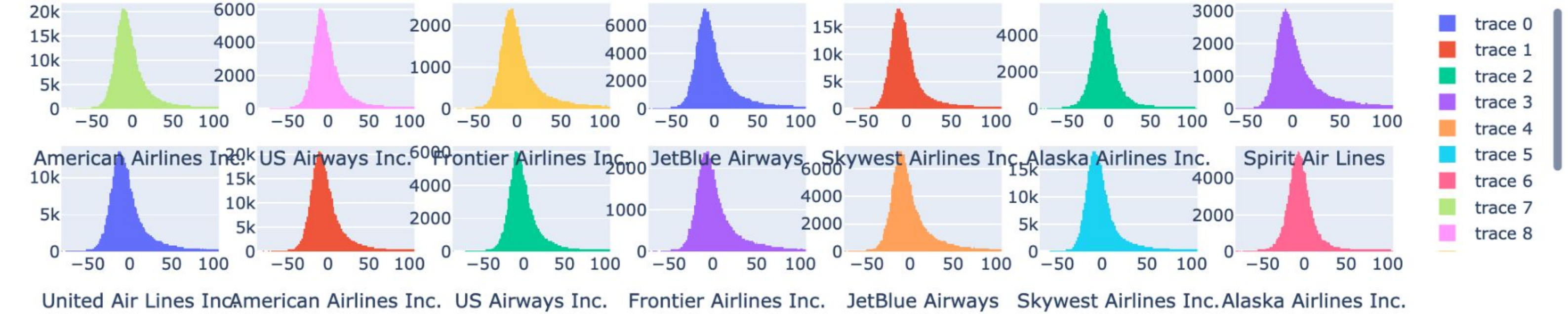
Average Departure Delay per Airline in 2015
Departure Delay (min)
0
5
10
15
United Air Lines Inc.
American Airlines Inc.
US Airways Inc.
Frontier Airlines Inc.
JetBlue Airways
Skywest Airlines Inc.
Alaska Airlines Inc.
Spirit Air Lines
Southwest Airlines Co.
Delta Air Lines Inc.
Atlantic Southeast Airlines
Hawaiian Airlines Inc.
American Eagle Airlines Inc.
Virgin America
AIRLINE
AIRLINE
United Air Lines Inc.
American Airlines Inc.
US Airways Inc.
Frontier Airlines Inc.
JetBlue Airways

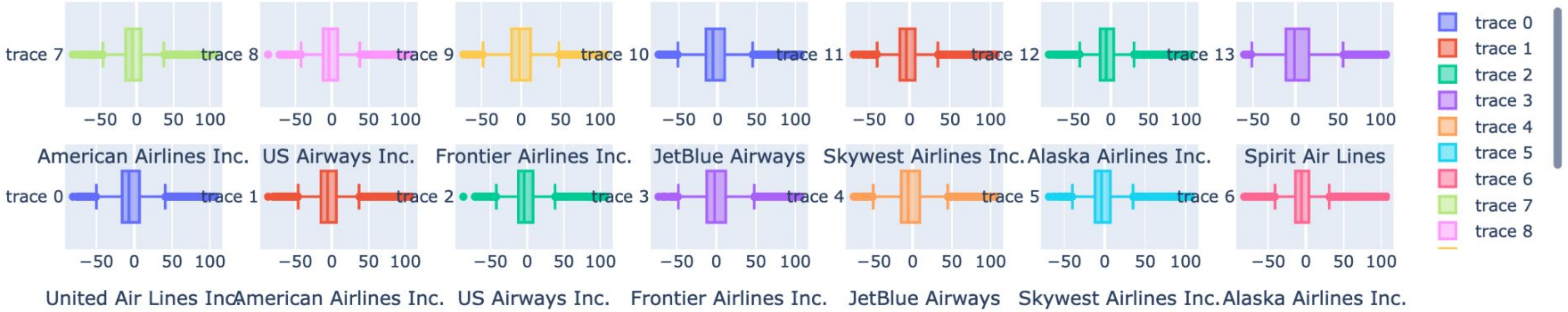
Looking at Arrival Delays

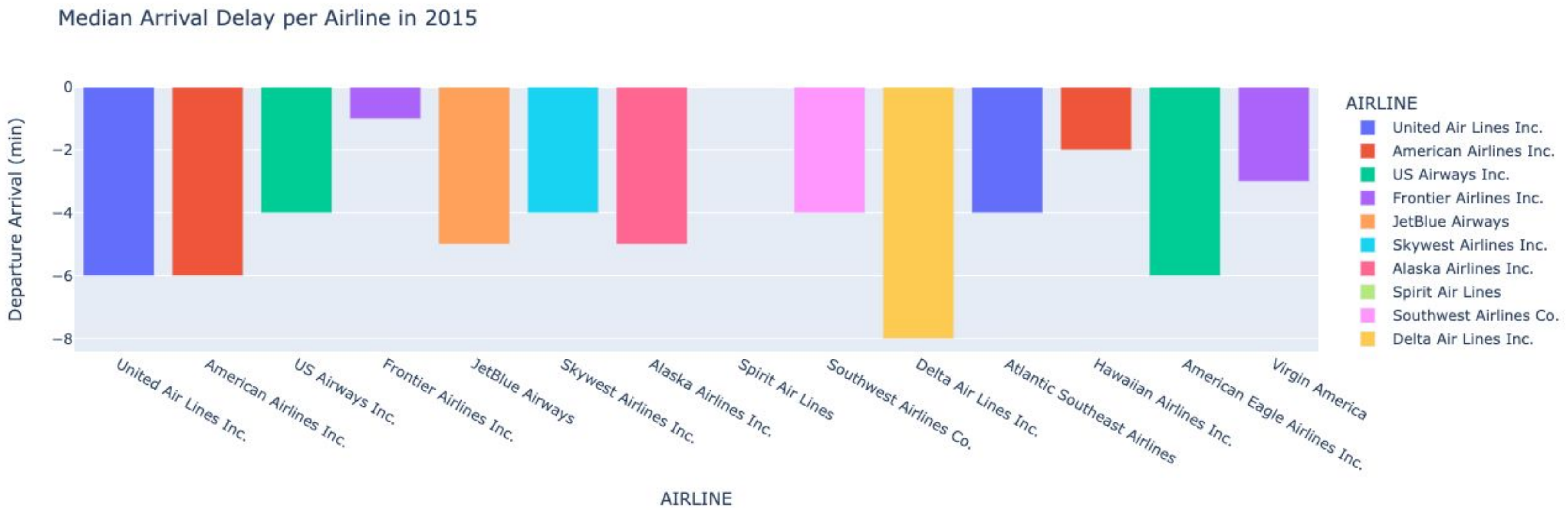
	ARRIVAL_DELAY							
	count	mean	std	min	25%	50%	75%	max
AIRLINE								
AA	712935.0	3.451372	44.266750	-87.0	-15.0	-6.0	7.0	1971.0
AS	171439.0	-0.976563	28.678804	-82.0	-14.0	-5.0	4.0	950.0
B6	262042.0	6.677861	41.400552	-76.0	-14.0	-5.0	12.0	1002.0
DL	870275.0	0.186754	38.439225	-79.0	-15.0	-8.0	3.0	1274.0
EV	554752.0	6.585379	40.682366	-64.0	-12.0	-4.0	9.0	1223.0
F9	90090.0	12.504706	51.561753	-73.0	-11.0	-1.0	16.0	1101.0
HA	76041.0	2.023093	25.714939	-67.0	-6.0	-2.0	5.0	1467.0
MQ	278791.0	6.457873	44.458112	-63.0	-15.0	-6.0	10.0	1528.0
NK	115193.0	14.471800	45.903410	-60.0	-10.0	0.0	20.0	833.0
OO	576814.0	5.845652	39.257694	-69.0	-12.0	-4.0	8.0	1372.0
UA	507762.0	5.431594	44.081214	-81.0	-16.0	-6.0	9.0	1294.0
US	194223.0	3.706209	32.378743	-87.0	-12.0	-4.0	9.0	750.0
VX	61248.0	4.737706	35.621579	-81.0	-12.0	-3.0	9.0	651.0
WN	1242403.0	4.374964	32.774001	-73.0	-12.0	-4.0	8.0	659.0

Looking at Arrival Delays

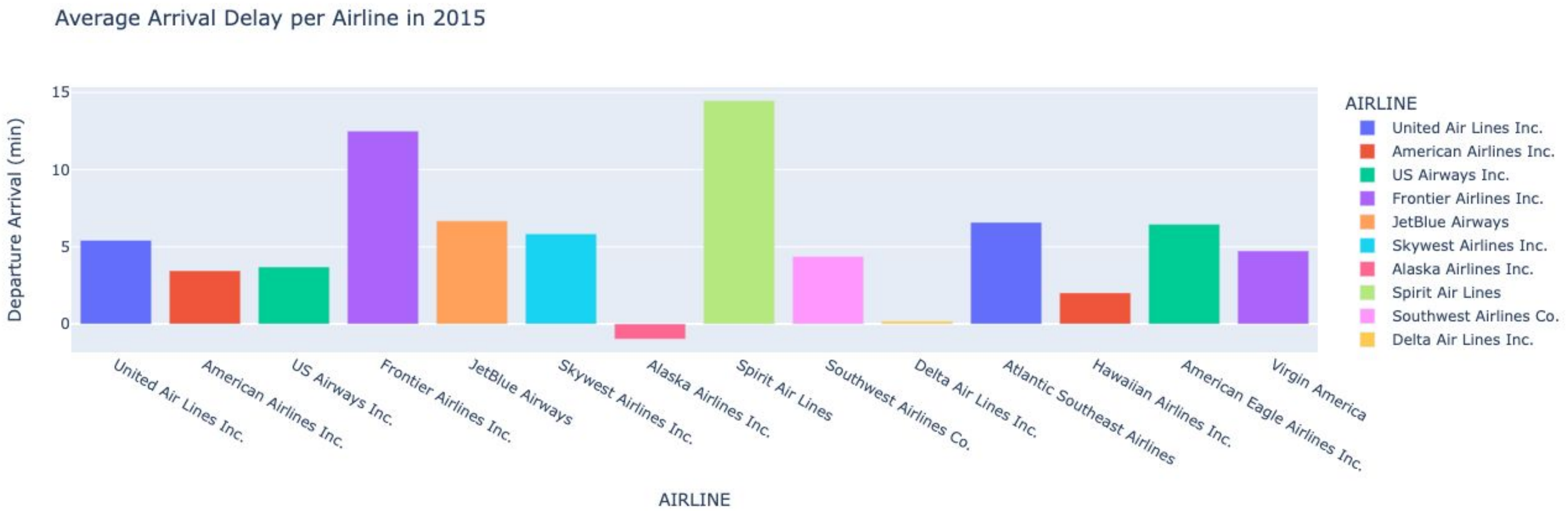
	IATA_CODE	AIRLINE	upper_outlire_bound	lower_outlire_bound
0	UA	United Air Lines Inc.	38.5	-24.5
1	AA	American Airlines Inc.	20.0	-17.0
2	US	US Airways Inc.	17.5	-15.5
3	F9	Frontier Airlines Inc.	40.5	-30.5
4	B6	JetBlue Airways	35.0	-26.0
5	OO	Skywest Airlines Inc.	19.0	-18.0
6	AS	Alaska Airlines Inc.	14.5	-17.5
7	NK	Spirit Air Lines	52.5	-35.5
8	WN	Southwest Airlines Co.	32.0	-21.0
9	DL	Delta Air Lines Inc.	16.0	-13.0
10	EV	Atlantic Southeast Airlines	19.0	-18.0
11	HA	Hawaiian Airlines Inc.	13.0	-16.0
12	MQ	American Eagle Airlines Inc.	29.0	-23.0
13	VX	Virgin America	23.5	-17.5


American Airlines Inc.
US Airways Inc.
Frontier Airlines Inc.
JetBlue Airways
Skywest Airlines Inc.
Alaska Airlines Inc.
Spirit Air Lines
United Air Lines Inc.
American Airlines Inc.
US Airways Inc.
Frontier Airlines Inc.
JetBlue Airways
Skywest Airlines Inc.
Alaska Airlines Inc.
trace 0
trace 1
trace 2
trace 3
trace 4
trace 5
trace 6
trace 7
trace 8


trace 7
trace 8
trace 9
trace 10
trace 11
trace 12
trace 13
American Airlines Inc.
US Airways Inc.
Frontier Airlines Inc.
JetBlue Airways
Skywest Airlines Inc.
Alaska Airlines Inc.
Spirit Air Lines
trace 0
trace 1
trace 2
trace 3
trace 4
trace 5
trace 6
United Air Lines Inc.
American Airlines Inc.
US Airways Inc.
Frontier Airlines Inc.
JetBlue Airways
Skywest Airlines Inc.
Alaska Airlines Inc.
trace 0
trace 1
trace 2
trace 3
trace 4
trace 5
trace 6
trace 7
trace 8



Median Arrival Delay per Airline in 2015
Departure Arrival (min)
0
−2
−4
−6
−8
AIRLINE
United Air Lines Inc.
American Airlines Inc.
US Airways Inc.
Frontier Airlines Inc.
JetBlue Airways
Skywest Airlines Inc.
Alaska Airlines Inc.
Spirit Air Lines
Southwest Airlines Co.
Delta Air Lines Inc.
Atlantic Southeast Airlines
Hawaiian Airlines Inc.
American Eagle Airlines Inc.
Virgin America
AIRLINE
United Air Lines Inc.
American Airlines Inc.
US Airways Inc.
Frontier Airlines Inc.
JetBlue Airways
Skywest Airlines Inc.
Alaska Airlines Inc.
Spirit Air Lines
Southwest Airlines Co.
Delta Air Lines Inc.



Average Arrival Delay per Airline in 2015
Departure Arrival (min)
15
10
5
0
AIRLINE
United Air Lines Inc.
American Airlines Inc.
US Airways Inc.
Frontier Airlines Inc.
JetBlue Airways
Skywest Airlines Inc.
Alaska Airlines Inc.
Spirit Air Lines
Southwest Airlines Co.
Delta Air Lines Inc.
Atlantic Southeast Airlines
Hawaiian Airlines Inc.
American Eagle Airlines Inc.
Virgin America
AIRLINE

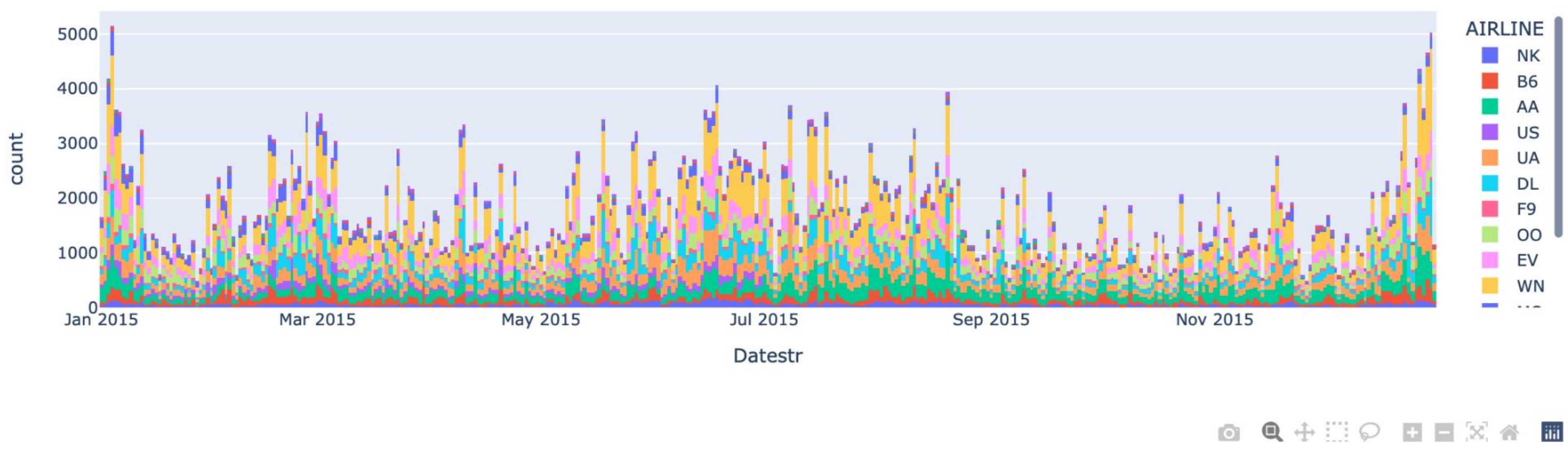
#2

Are flights around the holidays more susceptible to being cancelled?

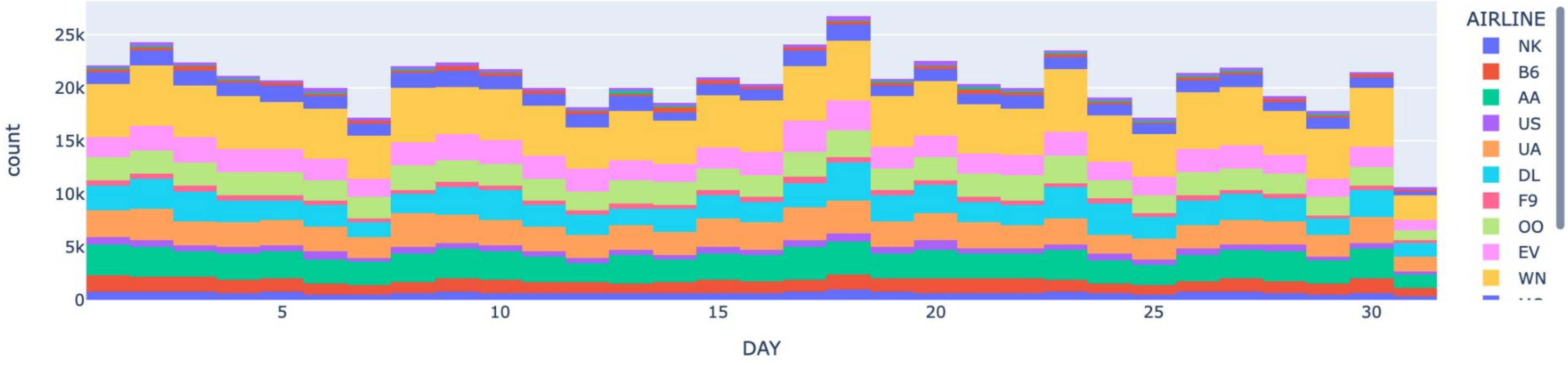
```python
flights_df['Date'] = pd.to_datetime(flights_df['YEAR'].astype('str')+ '-' +
flights_df['MONTH'].astype('str')+ '-' + flights_df['DAY'].astype('str'))
flights_df['Datestr'] = flights_df['YEAR'].astype('str')+ '-' +
flights_df['MONTH'].astype('str')+ '-' + flights_df['DAY'].astype('str')

# looking at holidays
hanukka_time = pd.date_range('2015-12-03','2015-12-15')
christmas_time = pd.date_range('2015-12-23','2016-01-01')
new_years=  pd.date_range('2015-01-01','2015-01-05')
summer_time = pd.date_range('2015-06','2015-09')
turkeytime_time = pd.date_range('2015-11-21','2015-11-29')

#holiday ranges
holidays = flights_df[((flights_df['Date']<christmas_time[-1]) &
(flights_df['Date']>christmas_time[0]))| ((flights_df['Date']
<new_years[-1]) & (flights_df['Date']>new_years[0]))] #
summer = flights_df[((flights_df['Date']<summer_time[-1]) &
(flights_df['Date']>summer_time[0]))]
thanksgive = flights_df[((flights_df['Date']<turkeytime_time[-1]) &
(flights_df['Date']>turkeytime_time[0]))]
hanukka = flights_df[((flights_df['Date']<hanukka_time[-1]) &
(flights_df['Date']>hanukka_time[0]))]

# % of flights that were canceled for each holiday area
display(flights_df['CANCELLED'].mean()*100,
                    holidays['CANCELLED'].mean()*100,
                    summer['CANCELLED'].mean()*100,
                   thanksgive['CANCELLED'].mean()*100,
                    hanukka['CANCELLED'].mean()*100)
```

```
1.5446430612129514

3.6900061837035625

1.213714859345163

1.4082195164760871

0.6062929856934027
```

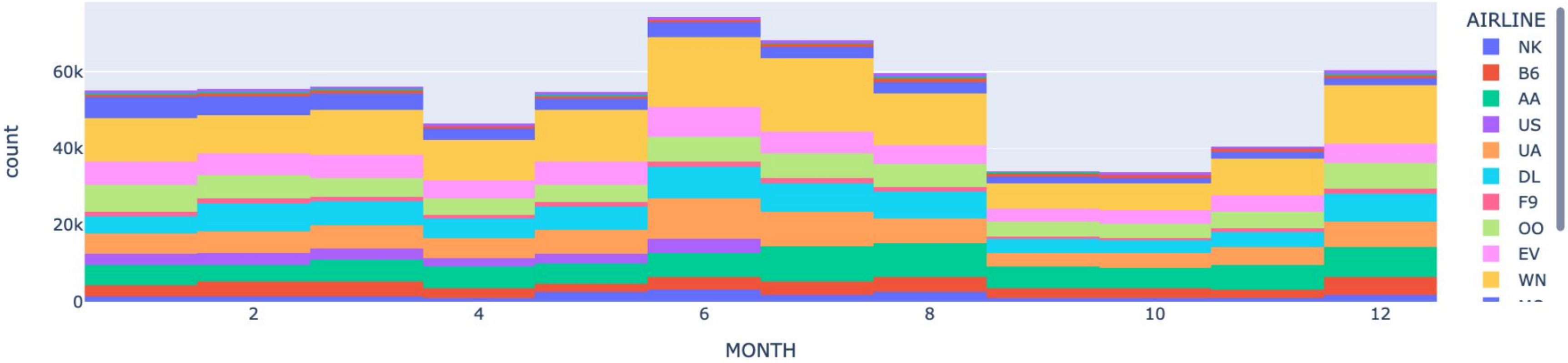
Flight Cancellations per Airline in 2015



Flight Cancellations per airline Grouped by Day of Month in 2015

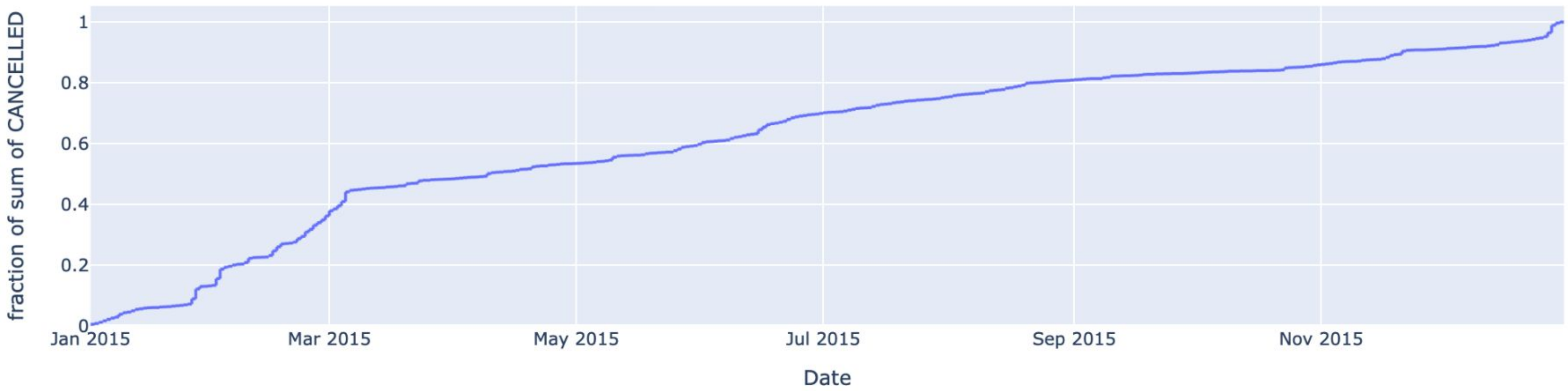


Flight Cancellations per airline Grouped by Month in 2015



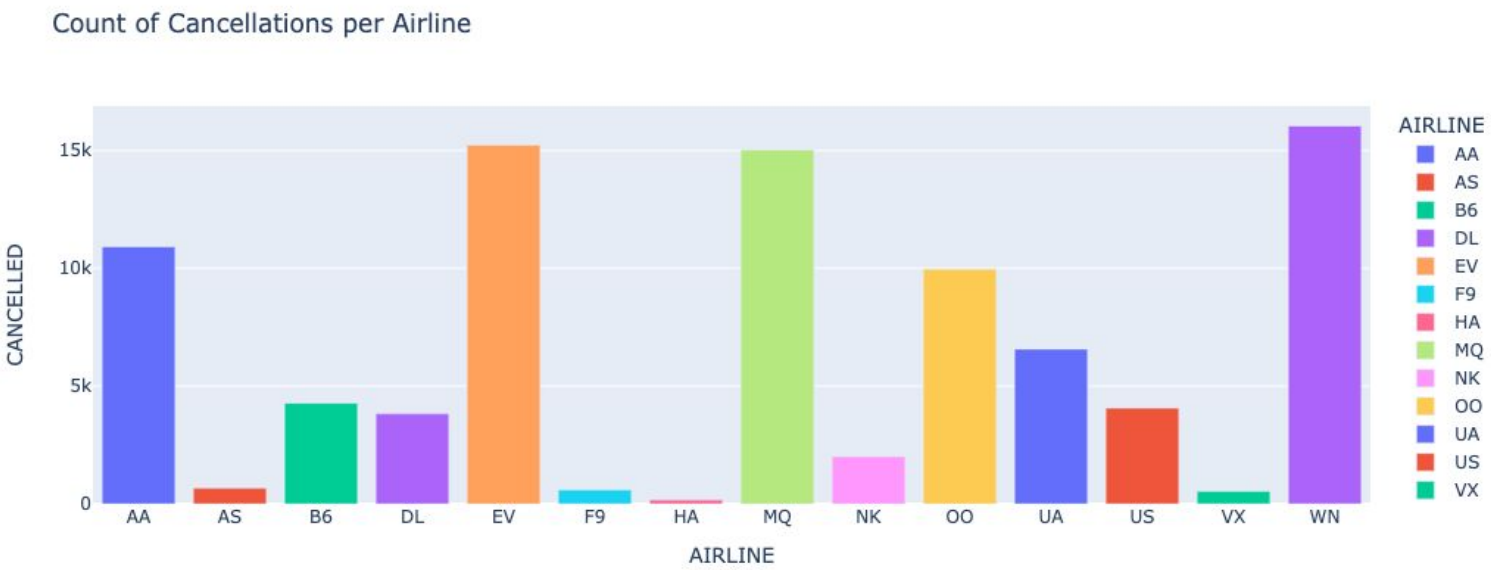
Split View of Cancelled Flights

Combined View of Cancelled Flights



Airline Cancellations

	AIRLINE	CANCELLED
0	AA	10919
1	AS	669
2	B6	4276
3	DL	3824
4	EV	15231
5	F9	588
6	HA	171
7	MQ	15025
8	NK	2004
9	OO	9960
10	UA	6573
11	US	4067
12	VX	534
13	WN	16043



#3

What factors result in longer delays for flights?

Machine Learning - Feature Preprocessing

	AS	AA	US	DL	NK	UA	HA	B6	OO	EV	...	BGM	BGR	ITH	ACK	MVY	WYS	DLG	AKN	GST	HYA
0	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5332909	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
5332910	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
5332911	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
5332912	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
5332913	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5332914 rows × 676 columns

```
def Day(x):
  if x == 0:
    return'Su'
  elif x==1:
    return('M')
  elif x == 2:
    return('T')
  elif x == 3:
    return('W')
  elif x == 4:
    return('Th')
  elif x == 5:
    return('F')
  elif x == 6:
    return('Sa')
def Month(x):
  if x == 0:
    return'Jan'
  elif x == 1:
    return'Feb'
  elif x ==2:
    return'Mar'
  elif x ==3:
    return'Apr'
  elif x ==4:
    return'May'
  elif x ==5:
    return'Jun'
  elif x == 6:
    return 'Jul'
  elif x == 7:
    return 'Aug'
  elif x == 8:
    return 'Sep'
  elif x == 9:
    return 'Oct'
  elif x ==10:
    return'Nov'
  elif x == 11:
    return'Dec'
data = flights_df['DOW'] = flights_df['DAY_OF_WEEK'].apply(Day)
data = flights_df['M'] = flights_df['MONTH'].apply(Month)

filter1= flights_df['ORIGIN_AIRPORT'].str.contains('^[\D]{3}')
filter2= flights_df['DESTINATION_AIRPORT'].str.contains('^\D{3}')
filter3= flights_df['DAY'] >=7
filter4= flights_df['DAY'] <14

X['ORIGIN_AIRPORT']=X['ORIGIN_AIRPORT']+'o'
X['DESTINATION_AIRPORT']=X['DESTINATION_AIRPORT']+'d'

decode = X['DESTINATION_AIRPORT'].unique()
orcode = X['ORIGIN_AIRPORT'].unique()
col = X['AIRLINE'].unique().tolist()
# col += X['DOW'].unique().tolist()
col += X['M'].unique().tolist()
col += decode.tolist()
col += orcode.tolist()

X_nom = X[['AIRLINE','M','DESTINATION_AIRPORT','ORIGIN_AIRPORT']]
onehot = preprocessing.OneHotEncoder(dtype=np.int8,sparse=True)
X_nom = onehot.fit_transform(X_nom).toarray()
X_nom = pd.DataFrame(X_nom,columns=col)
```

Machine Learning - Hyper Parameter Tuning

```python
X_train, X_test, y_train, y_test = train_test_split(X_nom, y,
test_size=0.33, random_state=22)

# Number of trees in random forest
n_estimators = [int(x) for x in range(20,50,5)]
max_features = ['auto', 'sqrt','log2']
# Maximum number of levels in tree
max_depth = [int(x) for x in range(1,23,2)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1,100,10]
# Method of selecting samples for training each tree
bootstrap = [True, False]
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split
               #'min_samples_leaf': []
               }

rf = RandomForestRegressor(random_state=22)
rf_random = RandomizedSearchCV(estimator = rf,
param_distributions = random_grid, n_iter = 10, cv = 3,
verbose=2, random_state=22, n_jobs = 2)
rf_random.fit(X_train, y_train)
```

```
Fitting 3 folds for each of 10 candidates, totalling 30 fits
[CV] END max_depth=None, max_features=sqrt, min_samples_split=5, n_estimators=45; total time= 1.8min
[CV] END max_depth=None, max_features=sqrt, min_samples_split=5, n_estimators=45; total time= 1.8min
[CV] END max_depth=None, max_features=sqrt, min_samples_split=5, n_estimators=45; total time= 1.8min
[CV] END max_depth=15, max_features=auto, min_samples_split=5, n_estimators=45; total time= 9.2min
[CV] END max_depth=15, max_features=auto, min_samples_split=5, n_estimators=45; total time= 9.8min
[CV] END max_depth=15, max_features=auto, min_samples_split=5, n_estimators=45; total time= 9.5min
[CV] END max_depth=21, max_features=auto, min_samples_split=5, n_estimators=45; total time=11.2min
[CV] END max_depth=21, max_features=auto, min_samples_split=5, n_estimators=45; total time=11.9min
[CV] END max_depth=None, max_features=sqrt, min_samples_split=2, n_estimators=45; total time= 1.8min
[CV] END max_depth=None, max_features=sqrt, min_samples_split=2, n_estimators=45; total time= 1.8min
[CV] END max_depth=21, max_features=auto, min_samples_split=5, n_estimators=45; total time=11.6min
[CV] END max_depth=5, max_features=sqrt, min_samples_split=2, n_estimators=40; total time=  14.9s
[CV] END max_depth=5, max_features=sqrt, min_samples_split=2, n_estimators=40; total time=  14.7s
[CV] END max_depth=5, max_features=sqrt, min_samples_split=2, n_estimators=40; total time=  14.8s
[CV] END max_depth=7, max_features=log2, min_samples_split=5, n_estimators=40; total time=  10.1s
[CV] END max_depth=7, max_features=log2, min_samples_split=5, n_estimators=40; total time=  10.1s
[CV] END max_depth=7, max_features=log2, min_samples_split=5, n_estimators=40; total time=  10.2s
[CV] END max_depth=None, max_features=sqrt, min_samples_split=2, n_estimators=45; total time= 1.8min
[CV] END max_depth=None, max_features=auto, min_samples_split=10, n_estimators=20; total time= 8.4min
[CV] END max_depth=None, max_features=auto, min_samples_split=10, n_estimators=20; total time= 9.2min
[CV] END max_depth=1, max_features=sqrt, min_samples_split=10, n_estimators=25; total time=   3.3s
[CV] END max_depth=1, max_features=sqrt, min_samples_split=10, n_estimators=25; total time=   3.4s
[CV] END max_depth=1, max_features=sqrt, min_samples_split=10, n_estimators=25; total time=   3.5s
[CV] END max_depth=None, max_features=log2, min_samples_split=5, n_estimators=25; total time=  45.9s
...
[CV] END max_depth=1, max_features=sqrt, min_samples_split=2, n_estimators=35; total time=   4.3s
[CV] END max_depth=1, max_features=sqrt, min_samples_split=2, n_estimators=35; total time=   4.3s
[CV] END max_depth=1, max_features=sqrt, min_samples_split=2, n_estimators=35; total time=   4.2s
[CV] END max_depth=None, max_features=auto, min_samples_split=10, n_estimators=20; total time= 8.8min

RandomizedSearchCV(cv=3, estimator=RandomForestRegressor(random_state=22),
                   n_jobs=2,
                   param_distributions={'max_depth': [1, 3, 5, 7, 9, 11, 13, 15,
                                                      17, 19, 21, None],
                                        'max_features': ['auto', 'sqrt',
                                                         'log2'],
                                        'min_samples_split': [2, 5, 10],
                                        'n_estimators': [20, 25, 30, 35, 40,
                                                         45]},
                   random_state=22, verbose=2)
```

Machine Learning - Best Model

```
def evaluate(model, test_features, test_labels):
    predictions = model.predict(test_features)
    errors = abs(predictions - test_labels)
    mape = 100 * np.mean(errors / test_labels)
    accuracy = 100 - mape
    print('Model Performance')
    print('Average Error: {:0.4f}
degrees.'.format(np.mean(errors)))
    print('Accuracy = {:0.2f}%.'.format(accuracy))

    return accuracy
# base_model = RandomForestRegressor(n_estimators = 10,
random_state = 22 )
# base_model.fit(X_train, y_train)
best_param = rf_random.best_params_
best_param
```

```
{
  'n_estimators': 45,
  'min_sample_split': 5,
  'max_features': 'auto',
  'max_depth': 15
}
```

Results - Accuracy & Precision

```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
p =metrics.precision_recall_fscore_support(y_test,y_pred)
print('Precision: \nnot cancelled',p[0][0], 'cancelled',p[0][1], '\nRecall:
\nnot canceled',p[1][0],'cancelled',p[1][1],'\nF1score:\nnot cancelled',p[2]
[0],'cancelled',p[2][1])
```

```
Accuracy: 0.9875764469524626
Precision:
not cancelled 0.9875764469524626 cancelled 0.0
Recall:
not canceled 1.0 cancelled 0.0
F1score:
not cancelled 0.9937493961218014 cancelled 0.0
```

	Not Cancelled	Cancelled
Not Cancelled	401118	0
Cancelled	5046	0

Results - Feature Importance

```
feature_imp = pd.Series(classifier.feature_importances_,index=col).sort_values(ascending=False)
sum = 0
monthimp = 0
dowimp = 0
lineimp = 0
orimp=0
deimp=0
for x in feature_imp.index:
  if x in M:
    monthimp += feature_imp.loc[x]
  elif x in DOW:
    dowimp+= feature_imp.loc[x]
  elif x in orcode:
    orimp += feature_imp.loc[x]
  elif x in decode:
    deimp +=feature_imp.loc[x]
  else:
    lineimp+= feature_imp.loc[x]
print('Month importance', monthimp)
print('DOW importance', dowimp)
print('Airline importance', lineimp)
print('Origin importance', orimp)
print('Destination importance',deimp)
```

```
Month importance 0.2976741392667543
DOW importance 0
Airline importance 0.25803290101703297
Origin importance 0.227345824155389
Destination importance 0.2169471355608238
```

03.
Conclusions

Research Question #1

The worst airline to fly in regards to delay times is Spirit Airlines. Spirit Airlines has an delay of ~15 minutes for both arrivals and departures.

Research Question #2

To answer the question of are holidays more susceptible to cancellations, we look at the percentage of during date ranges compared to the dataset as a whole. Overall, 1.5% of all of flights in the file were cancelled. Around christmas and new years 3.7% of flights that were scheduled got . In the Summer 1.2% of flights were cancelled. Around thanksgiving 1.4% of the flights were cancelled. And for Hanukkah about .6% of the flights were cancelled. From this it seems that Christmas was the only time in which flights were cancelled at a high rate. However I don't think we can make any conclusions from this observation on its own without looking at data over multiple years. As there may be confounding variables.

Research Question #3

In the end, we were able to create a classifier with 98.7% accuracy to determine if a flight would be cancelled. However, our classifier is extremely skewed to predict a flight as not being cancelled. We also explored the features selected. The features ranked in order of importance are MONTH (0.3), AIRLINE (0.26), ORIGIN (0.22), DESTINATION (0.217). It would make sense for MONTH to be the most important feature because flights are mostly cancelled because of weather.

04.

Future Analysis

Future **Analysis**

Meteostat

Use Meteostat Python API to predict arrival/departure delays and cancellations based on weather

Major Sporting Events

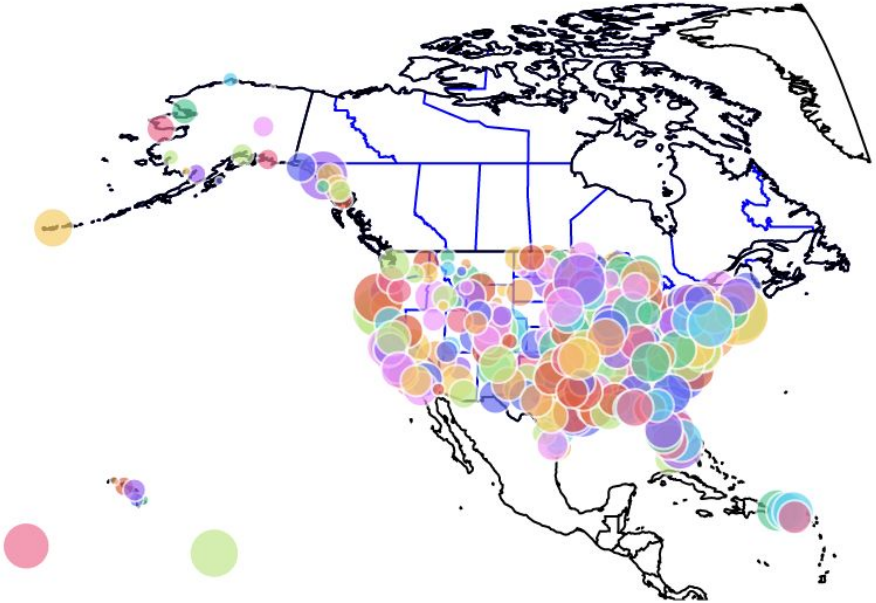
Investigate if major sporting events affect flights delays and cancellations

Meteostat Developers

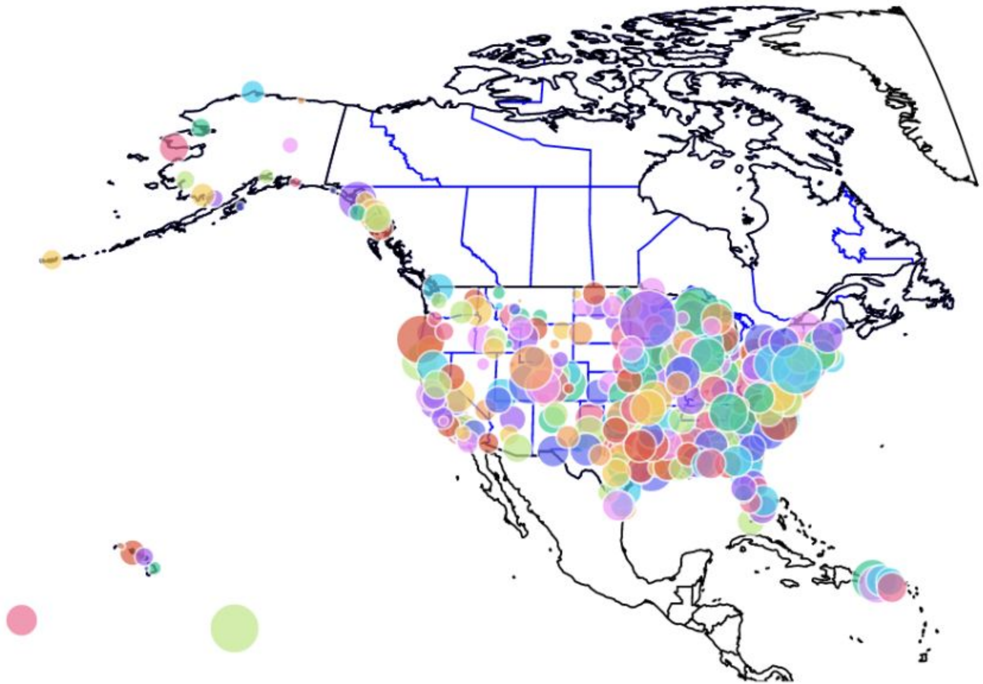
05.

Bubble Maps!

Average Departure Delay per Airport



Average Arrival Delay per Airport



References

[1]

pandas documentation — pandas 1.5.2 documentation. Pandas Documentation. Retrieved December 11, 2022 from

https://pandas.pydata.org/docs/

[2]

scikit-learn user guide. Scikit Learn. Retrieved December 11, 2022 from https://scikit-learn.org/0.21/_downloads/scikit-learn-docs.pdf

[3]

Plotly Python Graphing Library. Plotly Python Graphing Library. Retrieved December 11, 2022 from https://plotly.com/python/

[4]

William Koehrsen. 2018. Hyperparameter Tuning the Random Forest in Python. towards data science. Retrieved December 11, 2022 from

https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

Thank you!