Regression Modeling

Prediction of travel time for flights from Taiwan to Hong Kong

Overview

1. Problem Statement
2. Exploratory Data Analysis
3. Feature Engineering
4. Baseline Regression Modeling
5. More Regression Modeling
6. Conclusion and Recommendations

Problem Statement

To predict the estimated time taken for an aircraft, plying the route from from Taiwan Taoyuan International Airport (IATA: TPE) to Hong Kong International Airport (IATA: HKG), at any point in time at which it is detected on the flight radar to arrive to Hong Kong.

Exploratory Data Analysis

Summary flight info

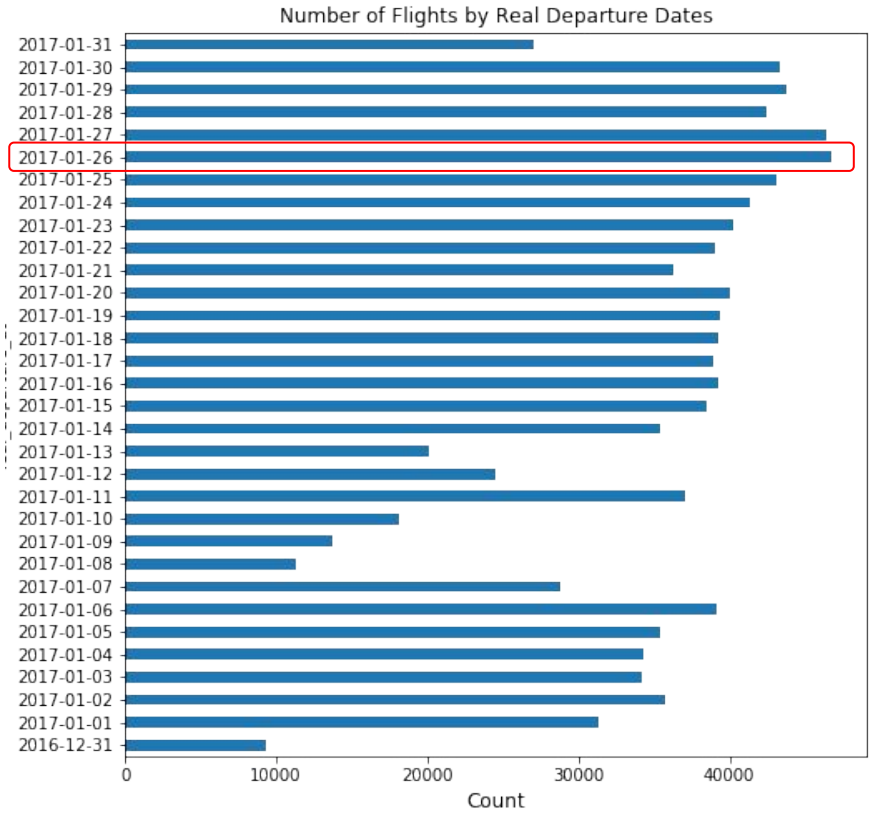
Feature	Type	Description
flight_id	*object*	Unique identifier for every flight record
flight_callsign	*object*	Used by Air Traffic Control to denote a specific flight
aircraft_model	*object*	Aircraft model of the flight
aircraft_registration	*object*	Each aircraft is assigned a registration number—often called a tail number—that is unique to the aircraft. Registration numbers are based on the country of registration, with the United Kingdom employing 'G' as their first letter identifier, followed by a hyphen and four further letters e.g. G-STBA. In the United States 'N' is used as the first letter followed by either letters or numbers e.g. N463UA.
airline	*object*	Airline operating the flight
origin	*object*	Country which the flight departed from
destination	*object*	Country which the flight arrived at
scheduled_departure_utc	*float*	Scheduled departure in Unix Time (UTC); elapsed seconds since January 1st 1970 00:00:00 UTC (Universal Time)
scheduled_arrival_utc	*float*	Scheduled arrival in Unix Time (UTC); elapsed seconds since January 1st 1970 00:00:00 UTC (Universal Time)
real_departure_utc	*float*	Real departure in Unix Time (UTC); elapsed seconds since January 1st 1970 00:00:00 UTC (Universal Time)
estimated_arrival_utc	*float*	Estimated arrival in Unix Time (UTC); elapsed seconds since January 1st 1970 00:00:00 UTC (Universal Time)
real_flight_duration	*float*	Duration in which the plane is in the air (in seconds)

Flight radar info

Feature	Type	Description
flight_id	*object*	Unique identifier for every flight record
timestamp_utc	*int*	Time of observation in Unix Time (UTC); elapsed seconds since January 1st 1970 00:00:00 UTC (Universal Time)
timestamp_dt	*int*	Timestamp of observation (YYYY-MM-DD HH:MM)
latitude	*float*	Latitude of aircraft's coordinate (WGS 84) at time of observation
longitude	*float*	Longitude of aircraft's coordinate (WGS 84) at time of observation
altitude	*int*	Altitude of aircraft (feet above mean sea level) at time of observation
heading	*int*	Compass heading of aircraft, denoted as 0-359
speed	*int*	Speed of the aircraft over the ground, measured in Knots (nautical miles per hour)

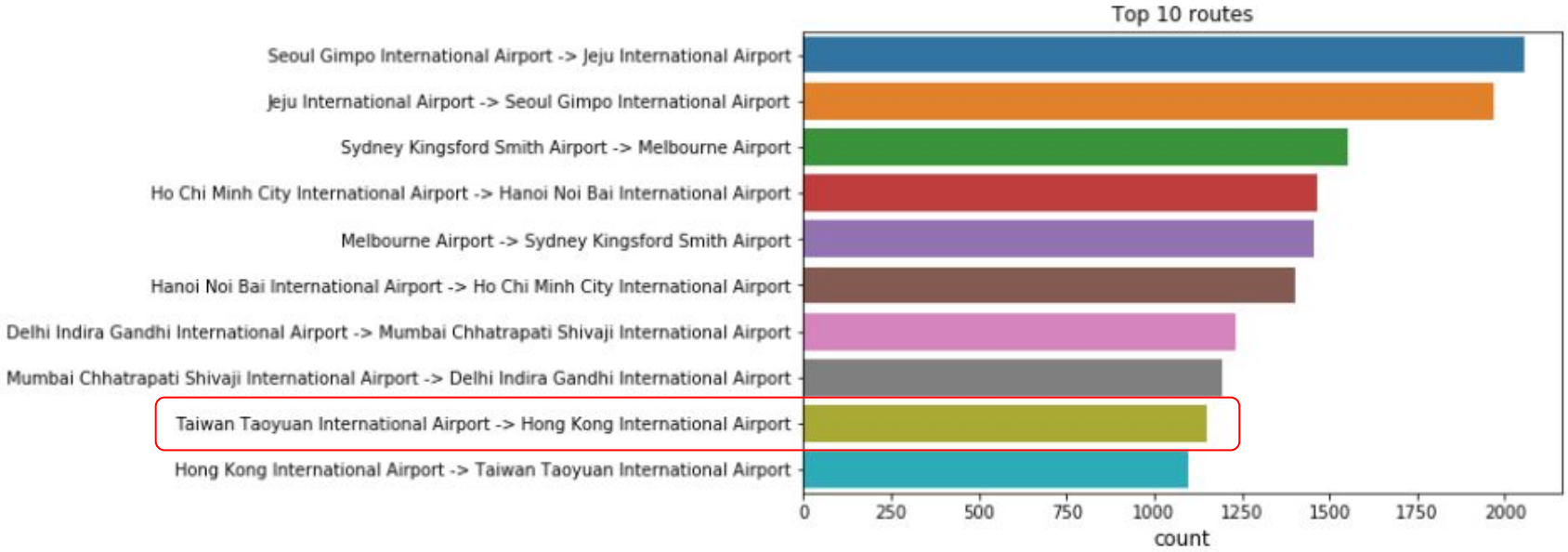
Exploratory Data Analysis

- Most number of departures happen to be on 2017-01-26



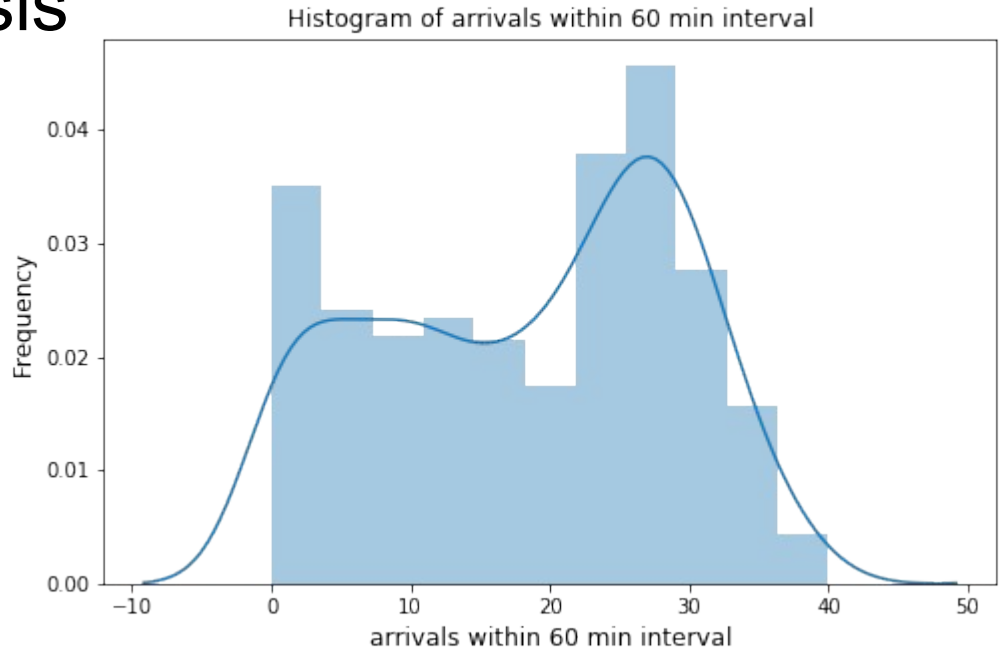
Exploratory Data Analysis

- Top international route in Jan 2017



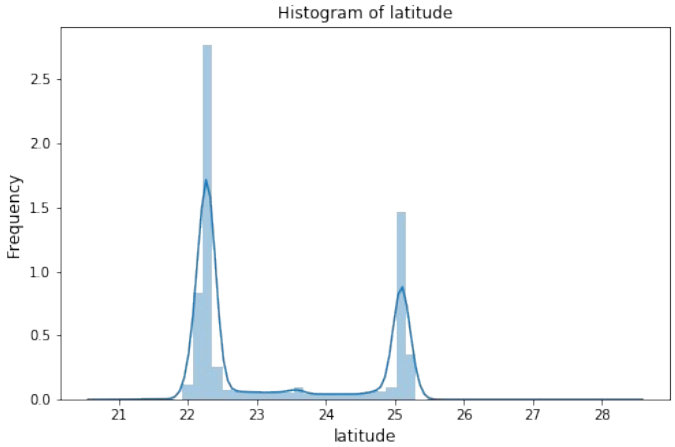
Exploratory Data Analysis

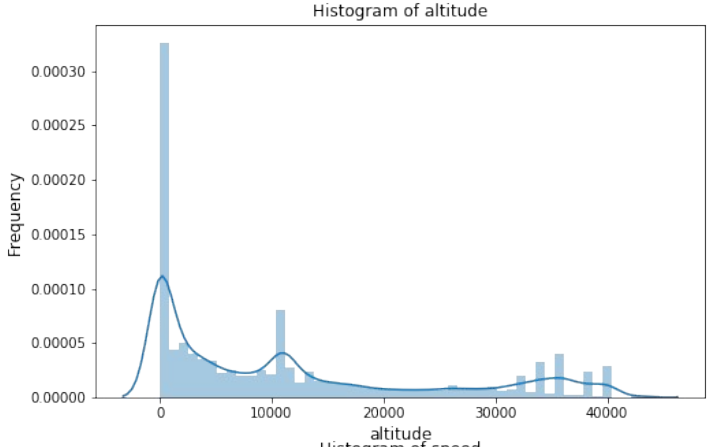
- Not normally distributed
- Bi-modal

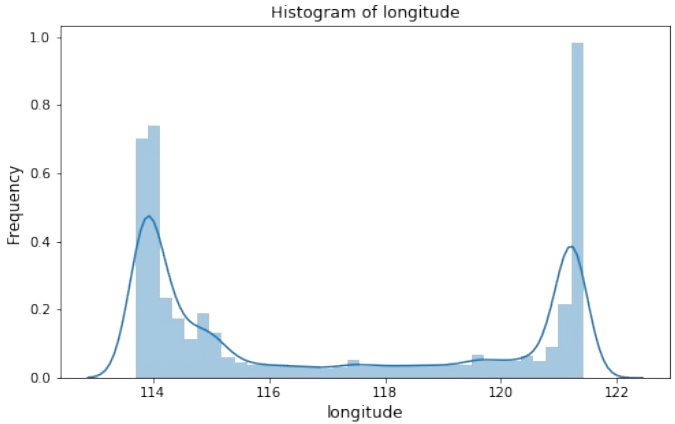


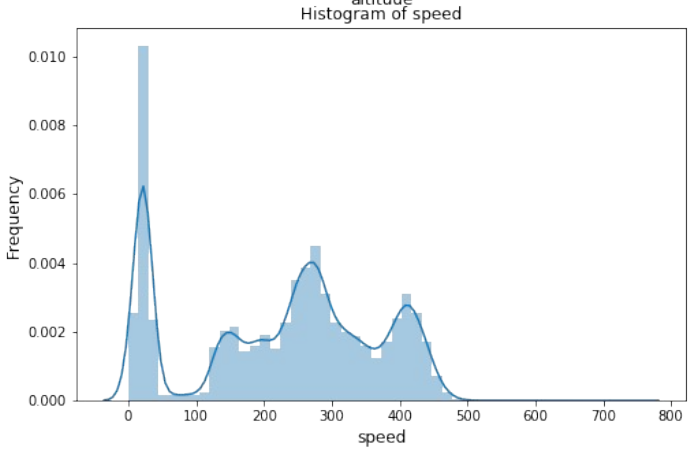

```
df_arrivals_at_hkg['forecasted_arrival_dt'] = \
(df_arrivals_at_hkg['real_departure_dt'] + pd.to_timedelta(df_arrivals_at_hkg['real_flight_duration'], unit='s'
.fillna(df_arrivals_at_hkg['real_departure_dt'] + pd.to_timedelta(df_arrivals_at_hkg['scheduled_flight_duration
.fillna(df_arrivals_at_hkg['scheduled_departure_dt'] + pd.to_timedelta(df_arrivals_at_hkg['real_flight_duration
.fillna(df_arrivals_at_hkg['scheduled_arrival_dt'])\
.fillna(df_arrivals_at_hkg['estimated_arrival_dt'])
```

Exploratory Data Analysis

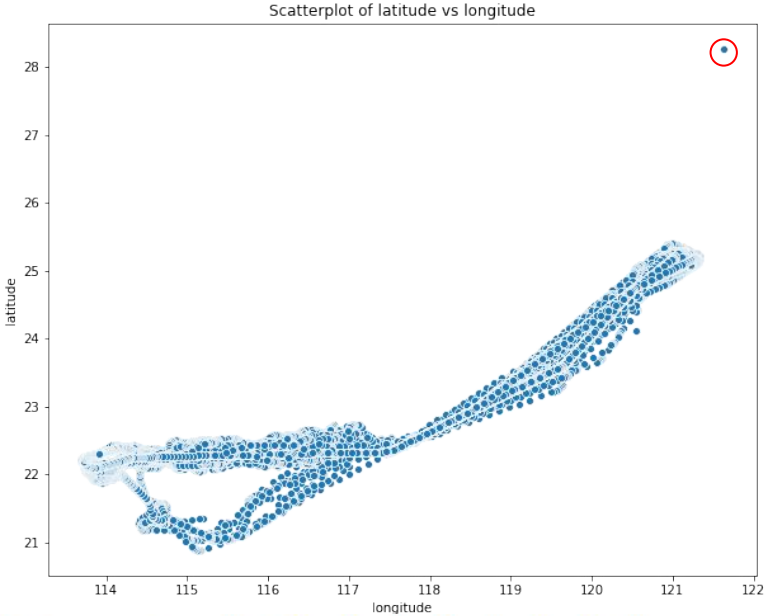








Exploratory Data Analysis



	flight_id	timestamp_utc	latitude	longitude	altitude	heading	speed	timestamp_dt
27371	c31cf95	1484662603	28.258541	121.633163	11600	8	0	2017-01-17 14:16:43
27372	c31cf95	1484662641	24.921299	120.893318	19450	256	0	2017-01-17 14:17:21
27373	c31cf95	1484662667	24.853016	120.848541	20684	230	390	2017-01-17 14:17:47
27374	c31cf95	1484662677	24.837517	120.830132	21338	219	454	2017-01-17 14:17:57
27375	c31cf95	1484662694	24.831850	120.808067	22190	258	329	2017-01-17 14:18:14

Feature Engineering

	flight_id	timestamp_dt	real_departure_dt	time_since_real_departure	latitude	longitude	altitude	heading
152164	c33e9e8	2017-01-18 11:53:40	2017-01-18 11:53:45	-5.0	25.067184	121.244965	0	49
152165	c33e9e8	2017-01-18 11:53:46	2017-01-18 11:53:45	1.0	25.070297	121.248878	50	48
152166	c33e9e8	2017-01-18 11:53:54	2017-01-18 11:53:45	9.0	25.076294	121.256813	525	50
152167	c33e9e8	2017-01-18 11:54:00	2017-01-18 11:53:45	15.0	25.079367	121.260925	775	51

	flight_id	timestamp_dt	real_departure_dt	time_since_real_departure	latitude	longitude	altitude	heading
152445	c33e9e8	2017-01-18 13:29:27	2017-01-18 11:53:45	5742.0	22.287163	113.872627	450	74
152446	c33e9e8	2017-01-18 13:29:33	2017-01-18 11:53:45	5748.0	22.288513	113.876900	375	70
152447	c33e9e8	2017-01-18 13:30:01	2017-01-18 11:53:45	5776.0	22.294333	113.895111	50	71
152448	c33e9e8	2017-01-18 13:30:10	2017-01-18 11:53:45	5785.0	22.296242	113.900909	0	69

Feature Engineering

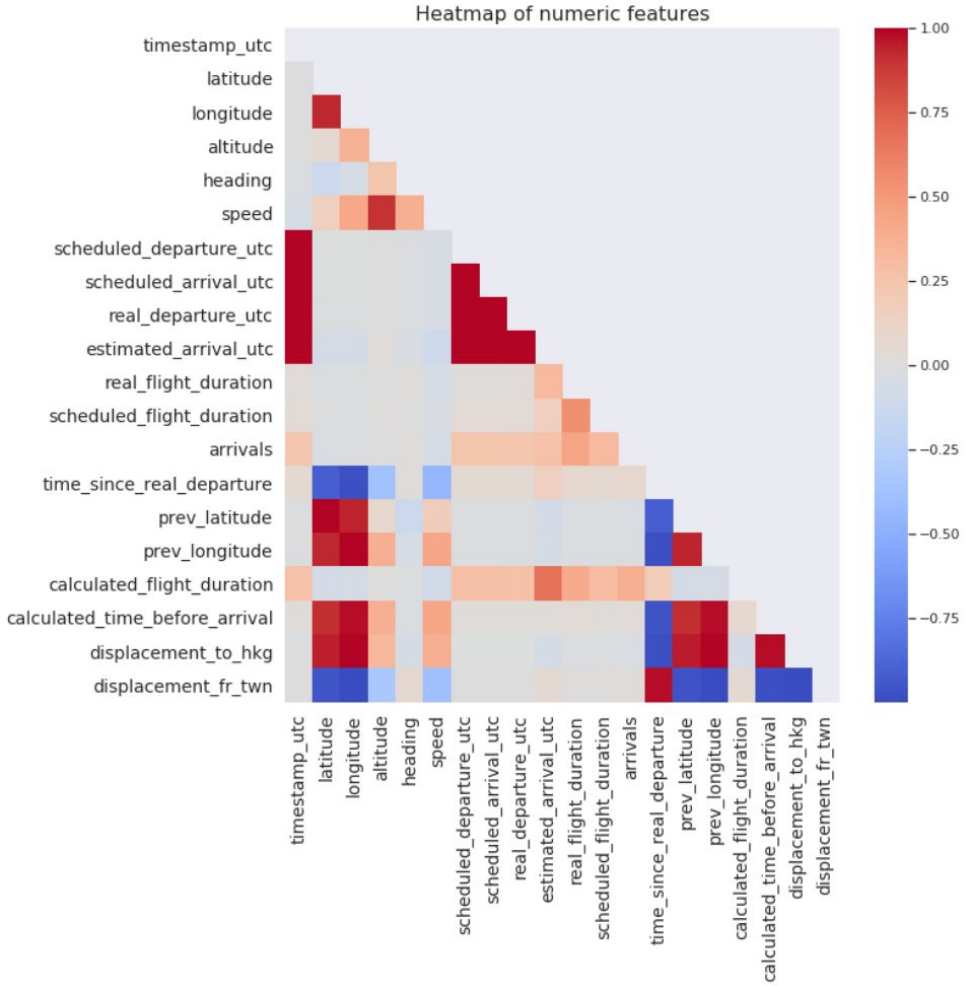
```
# Filter to get records where
# plane has taken off; i.e. speed > 0
# plane's time_since_real_departure >= 0
temp = merged_df.loc[(merged_df['flight_id'] == unique_flight_id) &
                     (merged_df['time_since_real_departure'] >= 0) &
                     (merged_df['speed'] > 0)].sort_values(by='timestamp_utc')


# time_since_real_departure after arrival
# where altitude = 0

possible_arrival_df = temp.loc[(temp['altitude'] == 0) &
                               (temp['time_since_real_departure'] > 11)]
if possible_arrival_df.shape[0] > 0:
    complete_flights.append(unique_flight_id)
    calculated_flight_duration = possible_arrival_df.iloc[0]['time_since_real_departure']
    temp = temp[temp['time_since_real_departure'] <= calculated_flight_duration]
    temp = temp.assign(calculated_flight_duration=calculated_flight_duration)
    filtered_df = filtered_df.append(temp, ignore_index=True)
else:
    incomplete_flights.append(unique_flight_id)
```

Feature Engineering

- Heatmap was plotted to investigate multicollinearity
- There are features in the dataset that describe a common attribute of the displacement_to_hkg, displacement_fr_twn, latitude, longitude.
- These features tend to exhibit high correlation among themselves



Baseline Regression Modeling

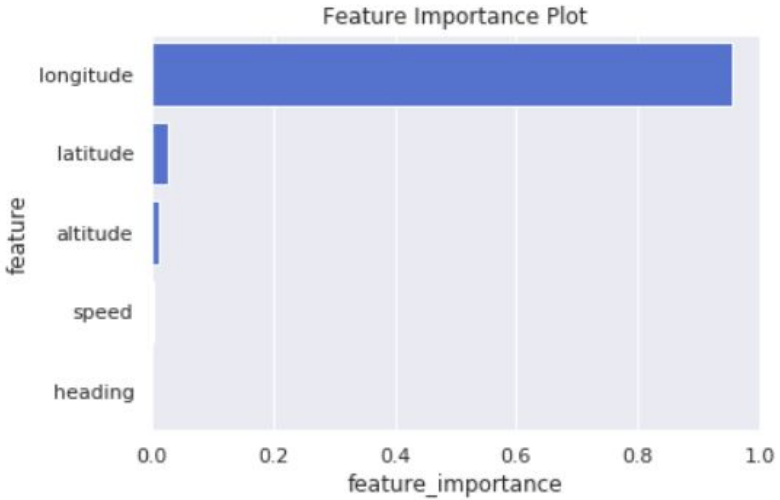
- Features: latitude, longitude, altitude, heading, speed
- RMSE value: 1909.991128
- Adjusted R2: 0

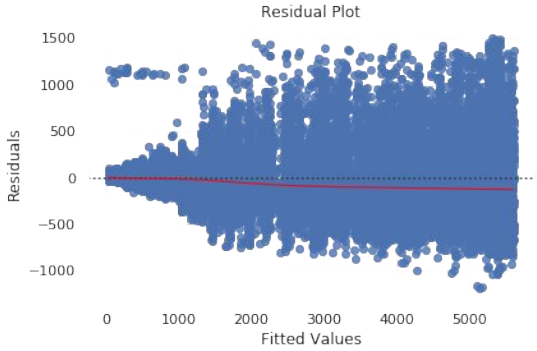
More Regression Modeling

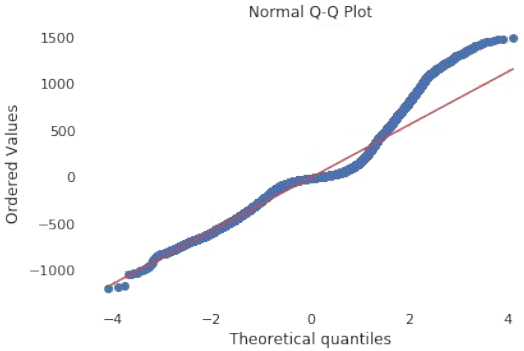
	model_name	mse	rmse	mae	r2	r2_adjusted
0	Random Forest Regression	8.970662e+04	299.510628	86.043145	0.975403	0.975399
1	Linear Regression	1.307992e+05	361.661743	184.162588	0.964136	0.964130
2	Lasso Regression	1.307993e+05	361.661815	184.162983	0.964136	0.964130
3	Ridge Regression	1.307998e+05	361.662606	184.201100	0.964135	0.964130
4	AdaBoost Regression	1.695393e+05	411.751498	322.838671	0.953513	0.953506
5	ElasticNet Regression	2.524455e+05	502.439582	311.995118	0.930781	0.930771
6	Baseline	3.648066e+06	1909.991128	1666.500000	-0.000281	-0.000427

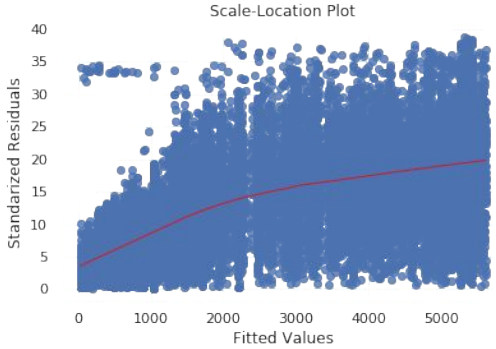
Regression Modeling

- Random Forest Regression









Conclusion and Recommendation

- The Random Forest Regressor had the lowest RMSE
- This model may not necessarily be reliable when predicting outside the time periods of Jan 2017, as this model omits potentially important time-dependent variables such as enroute weather information
- Obtain more information of the flight journey if financially feasible