IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Ali El Hanandeh>
<4/01/2022>

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Historical data available from spaceX were analyzed using various data exploration and analysis methods: data cleaning; wrangling; visualization; interrogation using SQL queries and application of machine learning techniques to predict the successful landing of Falcon 9 stage 1.
- In total, 90 records were retrieved for Falcon 9 flights. The LandingPad feature had more than 40% missing records. The Cape Canaveral Space Launch Complex 40 was the most frequently used launching location (55 launches), followed by Vandenberg Air Force Base Space Launch Complex 4E (22) and the rest were launched from the Kennedy Space Center Launch Complex 39A (13). The Cape Canaveral had the lowest success rate (60%) compared to the other 2 locations which had a success rate of 70%.
- More than 53% of the launched satellites were distained for The GTO and ISS orbits but had the lowest success rates. Nevertheless, the success rate improved overtime. Sixty landing attempts (66.7%) were successful. Drone ship landing was the most popular method for landing with success rate of 83.6% .
- Classification Trees method returned the most accurate predictions (accuracy score=0.8875 (cv) and 0.944 on the test set); meanwhile, SVM, logistic regressions and KNN methods were less accurate in their predictions (accuracy score =0.83 on the test set)
- Further detail available from: https://github.com/alimas2/Capstone-Project

Introduction

- SpaceX is an enterprise which focuses on launching satellite into orbit at a reduced cost. Falcon 9 is their premier rocket which is advertised to launch a satellite at 62 million dollars which is a steep reduction over their competition. A major contributor to the SpaceX cost reduction is the re-use of stage one of the launching rocket. Therefore, the aim of this project is to predict the success rate of launching are recovering stage 1 of Falcon 9.
- Objectives:
  - Identify launch pad stations and their success rate
  - Identify common satellite orbits and their respective launch success rates
  - Identify different landing methods and their respective success rates
  - Identify features that have high correlations to launch and recovery success rates
  - Construct different machine learning models and identify the most accurate model for predicting the landing outcome.

Section 1

Methodology

Methodology

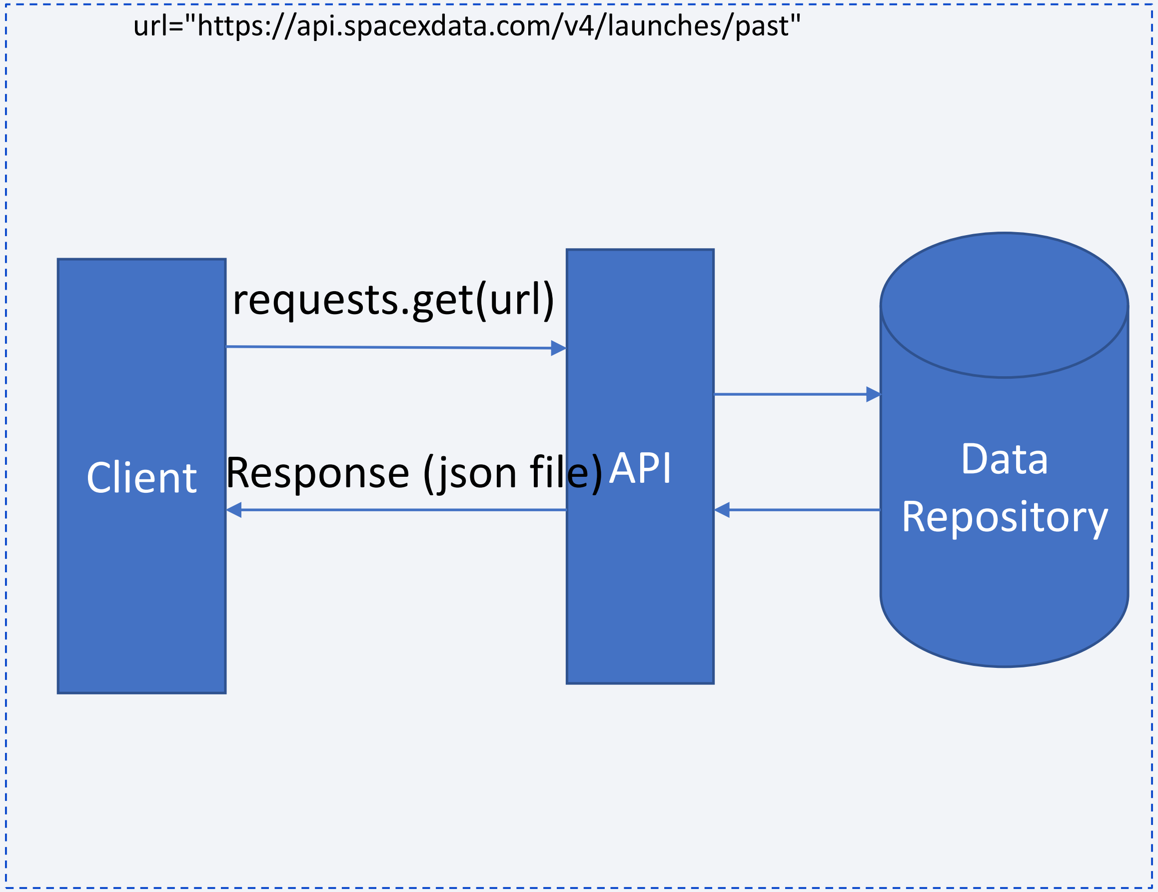
Executive Summary

- Data collection methodology:
  - Historical data retrieved from https://api.spacexdata.com/v4/launches/past
- Perform data wrangling
  - The json file was converted to a pandas dataframe using pandas.json_normalize() method. Then, relevant fields were extracted from the dataframe, excluding Multi-core rockets, records >(13/11/2020) and records not corresponding to Falcon 9.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Use cross-validation and accuracy metrics and confusion matrix

Data Collection - SpaceX API

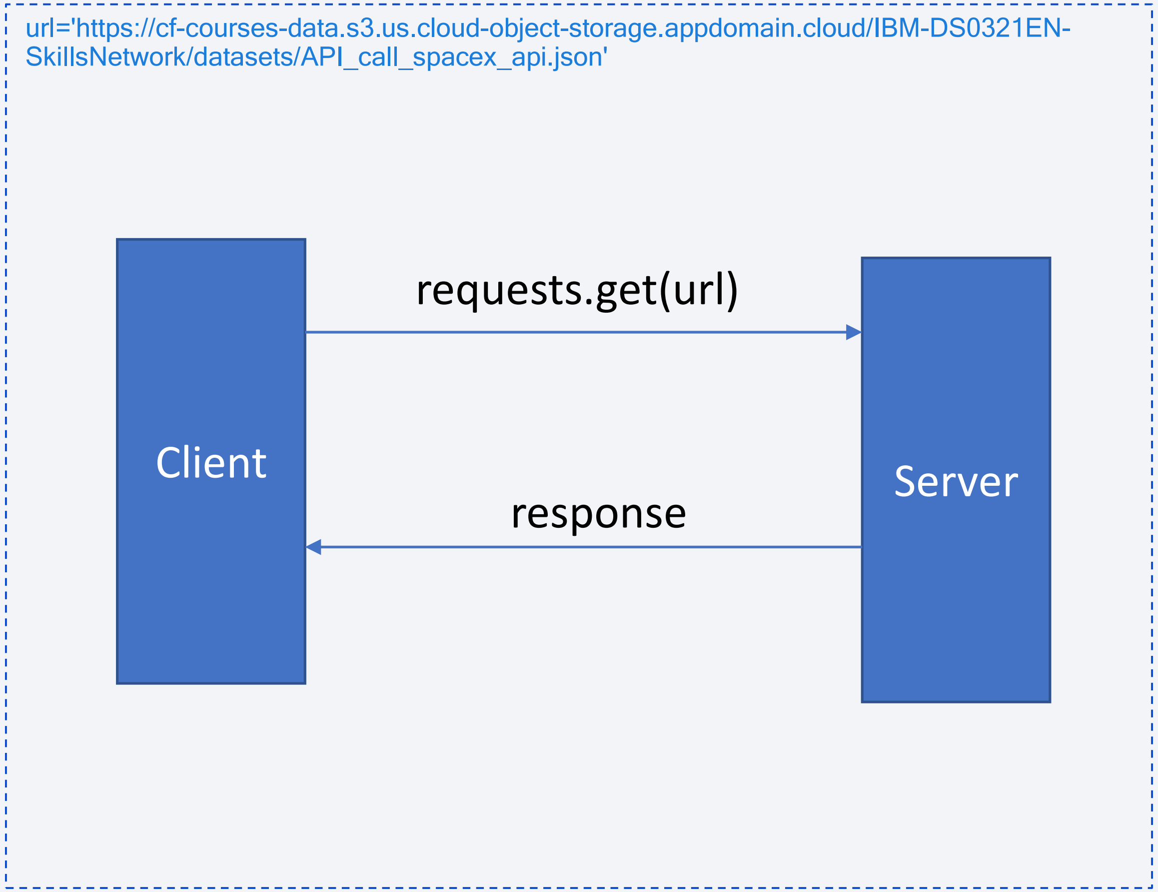
- A call API via requests.get(url) is performed. The API returns a json file as a response.

- https://github.com/alimas2/Capstone-Project/blob/master/Capsone_project.ipynb



Data Collection - Scraping

- Client makes direct call to the file on the server using get method. The server returns the requested file as response.
- https://github.com/alimas2/Capstone-Project/blob/master/Capsone_project.ipynb



Data Wrangling

- First, the data were returned as json response. Data was converted to a pandas dataframe using *pandas.json_normalize()* method.
- Second, an overview of the data was presented using the *dataframe_name.head()* method.
- Important features (fields) were selected *dataframe_name=dataframe_name[[feature1, feature2...,feature_n]]*
- The dataframe was cleaned by removing multi-core launches and converting date to utc format and limiting the data to <13/11/2020.
- The dataframe was filtered to include Falcon 9 only.
- Data was then inspected for missing values using *isnull()* method. PayloadMass missing values were replaced by the mean value.
- https://github.com/alimas2/Capstone-Project/blob/master/Capsone_project.ipynb

Data Analysis

- Data inspected to determine missing data and data types.
- Number of launches per launch site were counted using *value_counts()* method
- The landing outcome for each orbit were calculated using the *value_counts()* method
- Class field was added based on the landing outcome (0: failed landing and 1: successful landing)

- https://github.com/alimas2/Capstone-Project/blob/master/Lab2.ipynb

EDA with Data Visualization

- To visualize the relationships between the different features and the **landing outcome** the following plots were constructed:
  - Flight Number vs Payload Mass
  - Flight Number vs Launch site
  - Payload vs Launch site
  - Orbit type vs the average success rate
  - Flight Number vs Orbit type
  - Payload vs Orbit type
- https://github.com/alimas2/Capstone-Project/blob/master/lab4.ipynb

EDA with SQL

- Data was interrogated to determine the launch sites
- Top 5 records where the launch site begins with ‘KSC’ were displayed using the LIKE and LIMIT conditions
- The total payload mass carried by boosters launched by NASA (CRS) was calculated using the sum() function in SQL
- Average payload mass carried by booster version F9 v1.1 was calculated using the ave() function and the *where* conditional statement
- The date where the first successful landing outcome in drone ship was achieved was listed, using the min() function
- Names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000 were listed using composite condition ‘AND’and ‘BETWEEN’ statement
- https://github.com/alimas2/Capstone-Project/blob/master/lab%203.ipynb

EDA with SQL ..continued

- The names of the booster_versions which have carried the maximum payload mass were listed using the ‘group by’
- the month names when succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017 were displayed using MONTHNAME() and year() functions
- the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order were performed
- https://github.com/alimas2/Capstone-Project/blob/master/lab%203.ipynb

Build an Interactive Map with Folium

- Map objects including markers, circles, lines, were created and added to a folium map
- Circles were created to mark launch site locations, clusters were crated to mark success/failed launches and lines created to measure the distance of the locations to their proximities (roads, coast line, train tracks, ..etc).

- https://github.com/alimas2/Capstone-Project/blob/master/lab5.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- First class column was defined as 0 for failed and 1 for successful landing. The data was standardized using StandardScaler() method and finally split into 2 sets: **training** (80%) and **test** (20%).
- Four machine learning methods were used: LogisticRegression(); SVC(); DecisionTreeClassifier() and KNeighborsClassifier().
- Best Hyperparameter were identified using the GridSearchCV() method with cv=10. Confusion matrix and accuracy_score() were used to identify the best performing model.
- Model development process:

Load data→Y=class→Standardize data (0 to 1)→split data into X_train and Y_train, X_test and Y_test→define parameters for classifier→create classifier object→search best hyperparameters→train the model using X_train,Y_train→test the model using X_test → visualize the performace using the confussion matrix

- https://github.com/alimas2/Capstone-Project/blob/master/lab6.ipynb

Results

- Exploratory data analysis results
- 90 records were retrieved for Falcon 9 flights.
- The LandingPad feature had more than 40% missing records.
- The Cape Canaveral Space Launch Complex 40 was the most frequently used launching location (55 launches), followed by Vandenberg Air Force Base Space Launch Complex 4E (22) and the rest were launched from the Kennedy Space Center Launch Complex 39A (13).
- The Cape Canaveral had the lowest success rate (60%) compared to the other 2 locations which had a success rate of 70% with an overall success rate of 66.7%
- The average payload mass carried by booster version F9 v1.1 was 2928 kg

Results

- Interactive analytics demo in screenshots

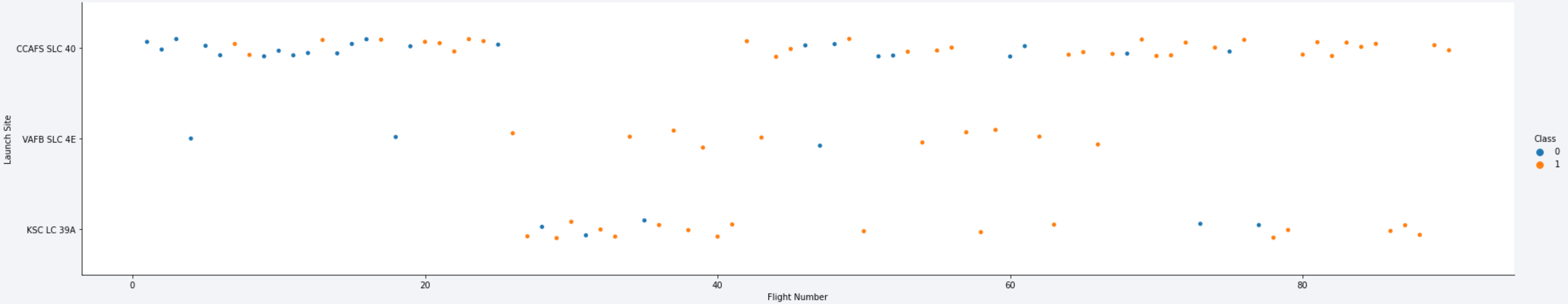
Results

- Predictive analysis results
  - Classification Trees method returned the most accurate predictions (accuracy score=0.8875 (training) and 0.944 on the test set);
  - SVM, logistic regressions and KNN methods were less accurate in their predictions (accuracy score =0.84 (training) and 0.83 on the test set)

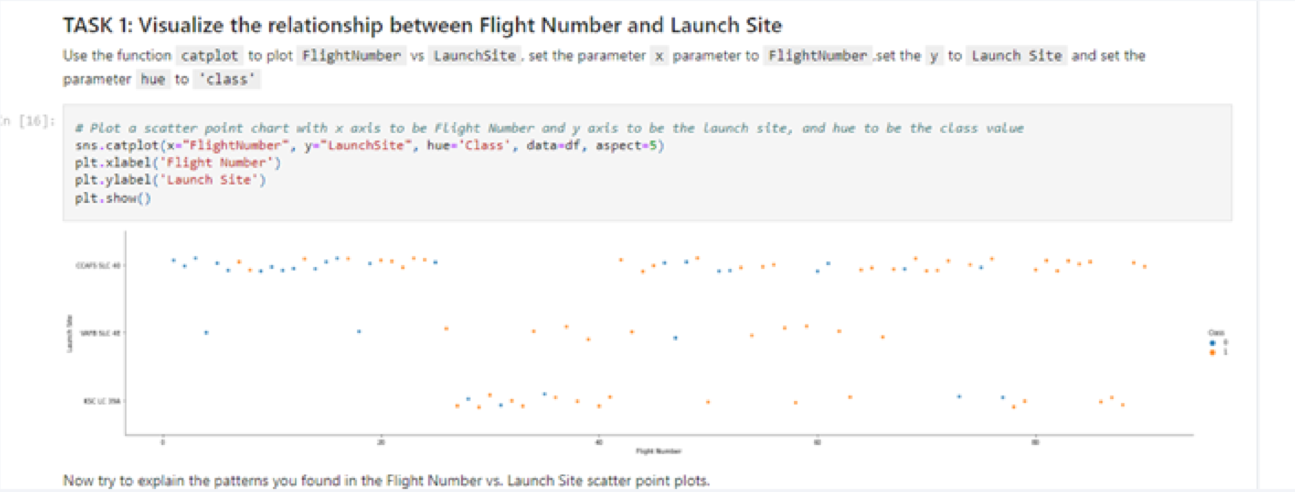
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



Scatter plot of Flight Number vs. Launch Site



Screenshot of the scatter plot

CCAFS SCL 40 had the largest number of launches

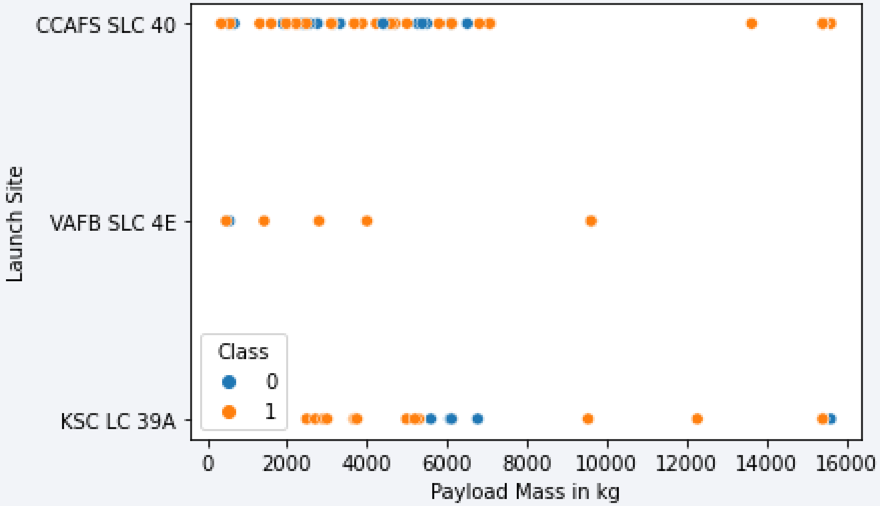
Most of the earlier launches (Flight Number <20) had unsuccessful landings.

Most of the earlier launches were attempted from the CCAFS SCL 40.

Success rate improved after Flight Number 20.

VAFB SLC 4E had the least number of launches and the least failures.

Payload vs. Launch Site



Payload vs. Launch Site

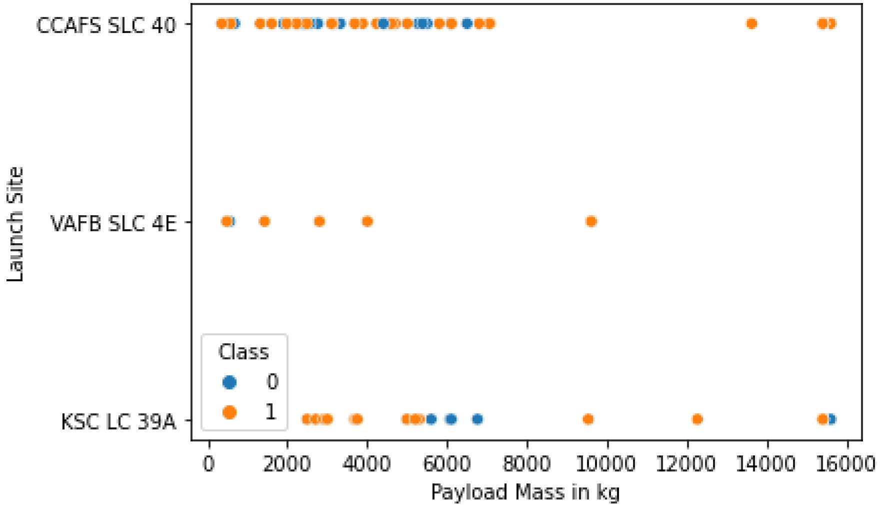
No heavy loads (>10,000 kg) were launched from VAFB SLC 4E
No failure of heavy loads launched from CCAFS SLC 40
KSC LC 39A had better success rate with small loads (<5500 kg)

TASK 2: Visualize the relationship between Payload and Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

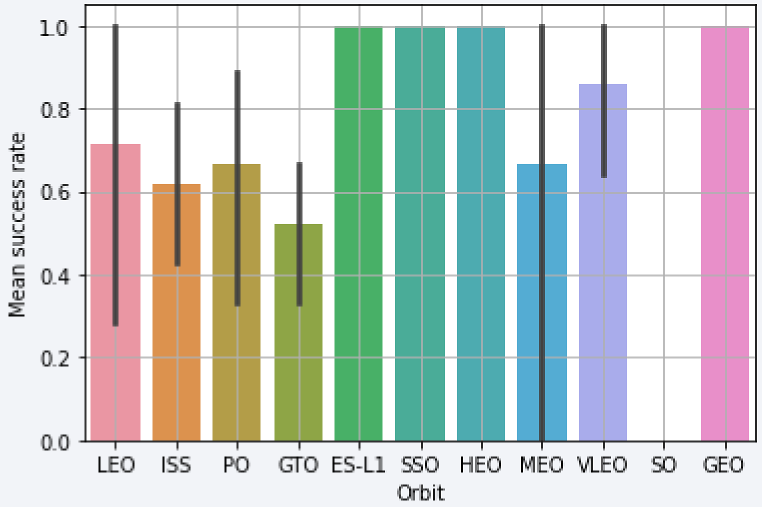
[17]:

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the
sns.scatterplot(x='PayloadMass', y='LaunchSite', hue='Class', data=df)
plt.xlabel('Payload Mass in kg')
plt.ylabel('Launch Site')

plt.show()
```



Screenshot of scatter plot

Success Rate vs. Orbit Type



Success rate of each orbit type

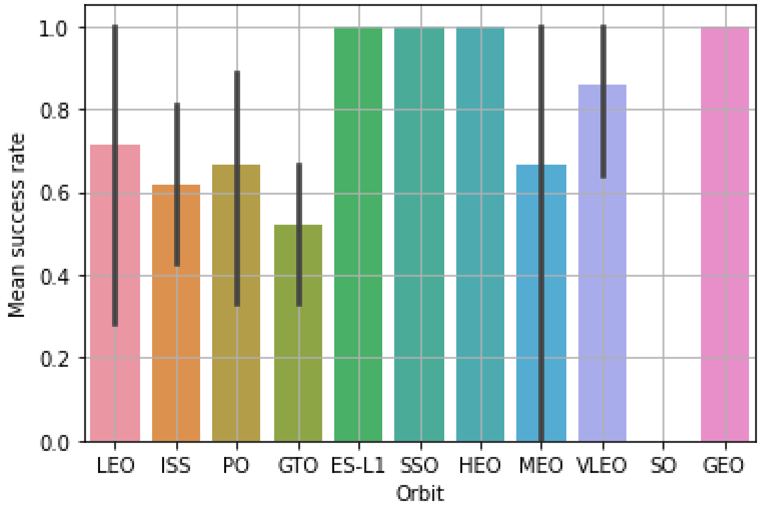
GEO, HEO, SSO andES-L1 had the highest success rate (100%).
GTO had the lowest success rate.

TASK 3: Visualize the relationship between success rate of each orbit type

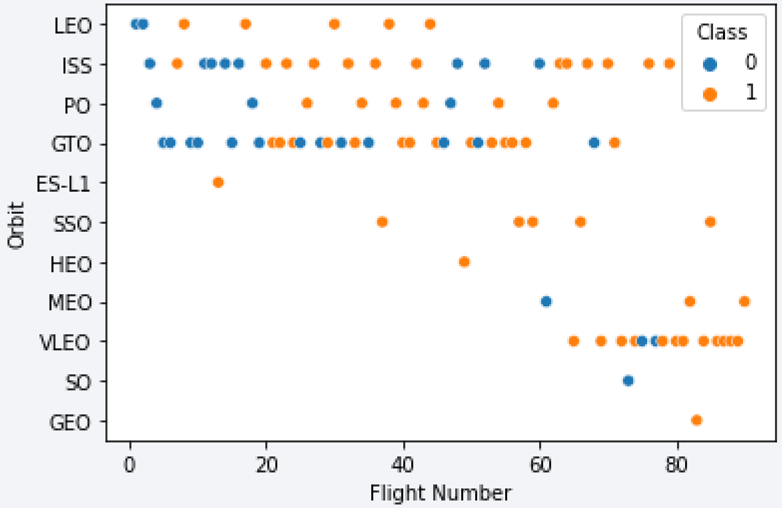
Next, we want to visually check if there are any relationship between success rate and orbit type.

Let's create a `bar chart` for the sucess rate of each orbit

```
# HINT use groupby method on Orbit column and get the mean of Class column
from numpy import mean
sns.barplot(x='Orbit', y='Class', data=df, estimator=mean)
plt.xlabel('Orbit')
plt.ylabel('Mean success rate')
plt.grid()
plt.show()
```



Flight Number vs. Orbit Type



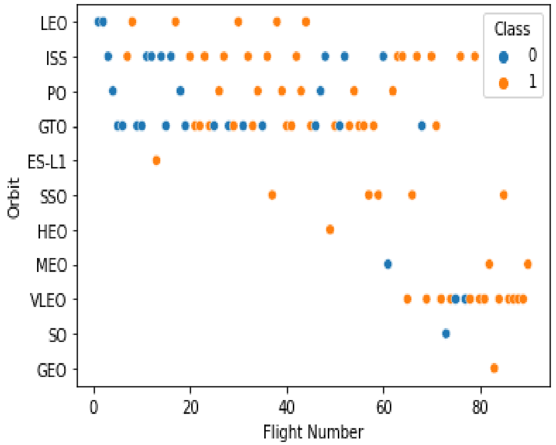
Flight number vs. Orbit type

TASK 4: Visualize the relationship between FlightNumber and Orbit type

For each orbit, we want to see if there is any relationship between FlightNumber and Orbit type.

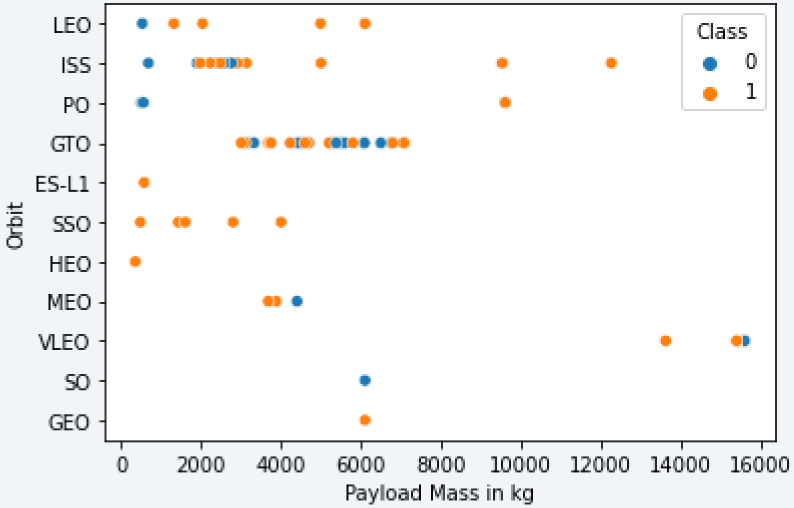
n [30]:

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.scatterplot(x='FlightNumber', y='Orbit', hue='Class', data=df)
plt.xlabel('Flight Number')
plt.ylabel('Orbit')
plt.show()
```



You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type

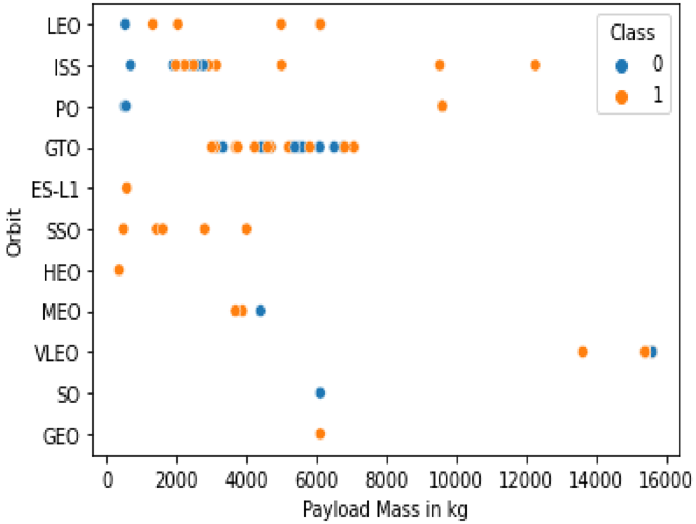


Payload vs. orbit type

TASK 5: Visualize the relationship between Payload and Orbit type

Similarly, we can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and Orbit type

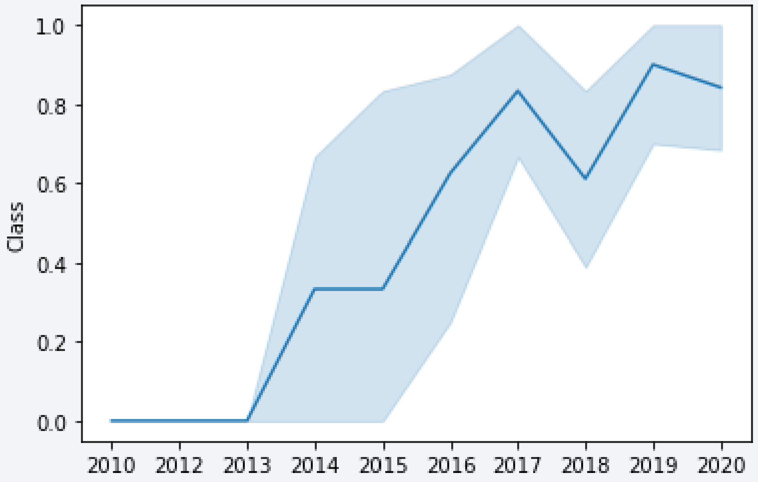
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.scatterplot(x='PayloadMass',y='Orbit', hue='Class', data=df)
plt.xlabel('Payload Mass in kg')
plt.ylabel('Orbit')
plt.show()
```



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



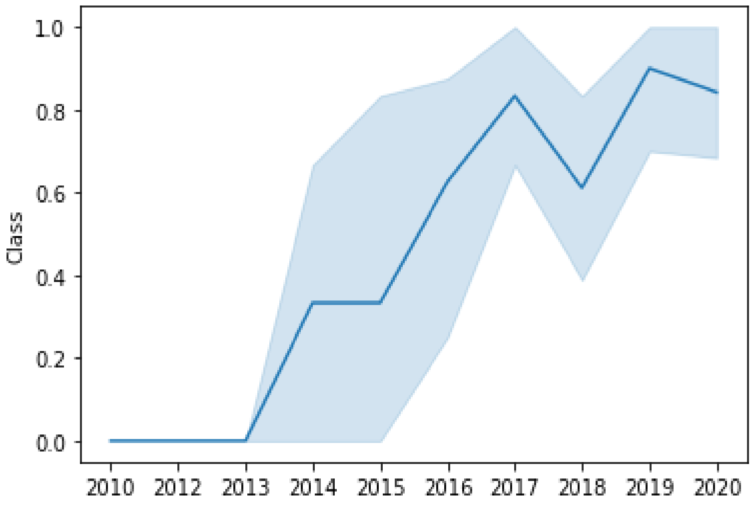
Line chart of yearly average success rate

In [48]:

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
year=[]
yy=Extract_year(df['Date'])

sns.lineplot(x=yy, y='Class', data=df, estimator=mean)
```

Out[48]:

```
<AxesSubplot:ylabel='Class'>
```



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

Display the names of the unique launch sites in the space mission

```
import pandas as pd
%sql SELECT unique launch_site from spacex
```

```
 * ibm_db_sa://tgg71884:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databa
Done.
```

launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Select statement with keyword unique to prevent repeat of names
From the spacex (database) from the field (column) ‘launch_site’

Launch Site Names Begin with 'CCA'

```
In [9]: %sql SELECT launch_site from spacex where launch_site LIKE 'CCA%' LIMIT 5;

 * ibm_db_sa://tgg71884:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.dat
Done.
```

Out[9]:

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Select the records from the launch_site field in the spacex database for records that start with ‘CCA’ and limit the returned records to the first 5.

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

```
: %sql select sum(PAYLOAD_MASS__KG_) from spacex where customer='NASA (CRS)';

 * ibm_db_sa://tgg71884:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.
Done.

10]:
```

1
45596

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from spacex where Booster_Version='F9 v1.1'
```

```
 * ibm_db_sa://tgg71884:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.app
Done.
```

]:

1
2928

First Successful Ground Landing Date

```
%sql select min(Date) from spacex where landing__outcome='Success (drone ship)';
```

```
 * ibm_db_sa://tgg71884:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.a
Done.
```

1
2016-04-08

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select booster_version from spacex where landing__outcome='Success (ground pad)' AND payload_mass__kg_ BETWEEN 4000 and 6000;
```

```
 * ibm_db_sa://tgg71884:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

]:

booster_version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

```
In [14]: %sql select mission_outcome, count(mission_outcome) from spacex where mission_outcome in ('Success', 'Failure (in flight)') group by mission_outcome

 * ibm_db_sa://tgg71884:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[14]:

mission_outcome	2
Failure (in flight)	1
Success	99

Boosters Carried Maximum Payload

In [15]:

```
%sql select distinct booster_version, max(payload_mass__kg_) from spacex group by booster_version
```

```
 * ibm_db_sa://tgg71884:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appd
Done.
```

Out[15]:

booster_version	2
F9 B4 B1039.2	2647
F9 B4 B1040.2	5384
F9 B4 B1041.2	9600
F9 B4 B1043.2	6460
F9 B4 B1039.1	3310
F9 B4 B1040.1	4990
F9 B4 B1041.1	9600
F9 B4 B1042.1	3500

2015 Launch Records

```
6]: %sql select MONTHNAME(Date) as Month_, launch_site, booster_version from spacex where landing__outcome='Success (ground pad)' and year(Date)=2017
```

```
 * ibm_db_sa://tgg71884:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

t[16]:

month_	launch_site	booster_version
February	KSC LC-39A	F9 FT B1031.1
May	KSC LC-39A	F9 FT B1032.1
June	KSC LC-39A	F9 FT B1035.1
August	KSC LC-39A	F9 B4 B1039.1
September	KSC LC-39A	F9 B4 B1040.1
December	CCAFS SLC-40	F9 FT B1035.2

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

```
]: %sql select year(Date) as Year_,count(landing__outcome) cnt from spacex where Date between '2010-06-04' and '2017-03-20'and landing__outcome like 'Success%' group by year(Date) order by cnt desc
```

```
 * ibm_db_sa://tgg71884:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

[17]:

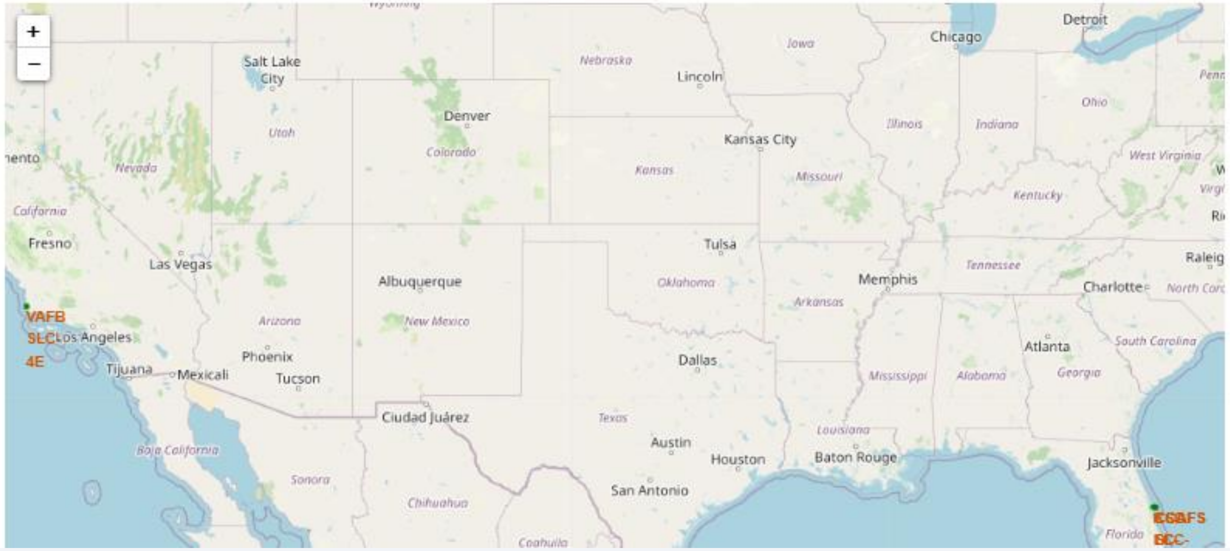
year_	cnt
2016	5
2017	2
2015	1

Section 4

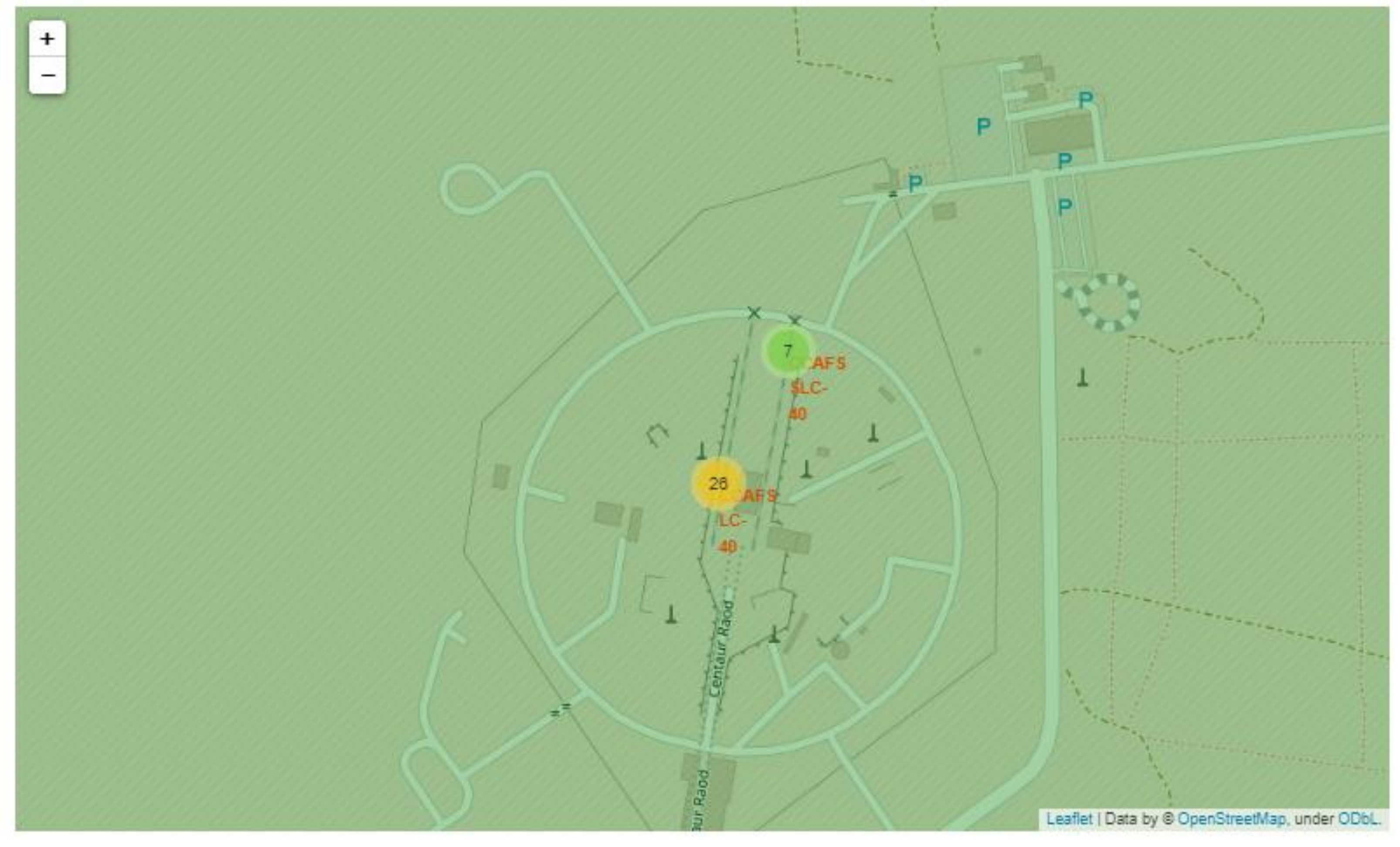
Launch Sites Proximities Analysis

Folium Map Launch Sites

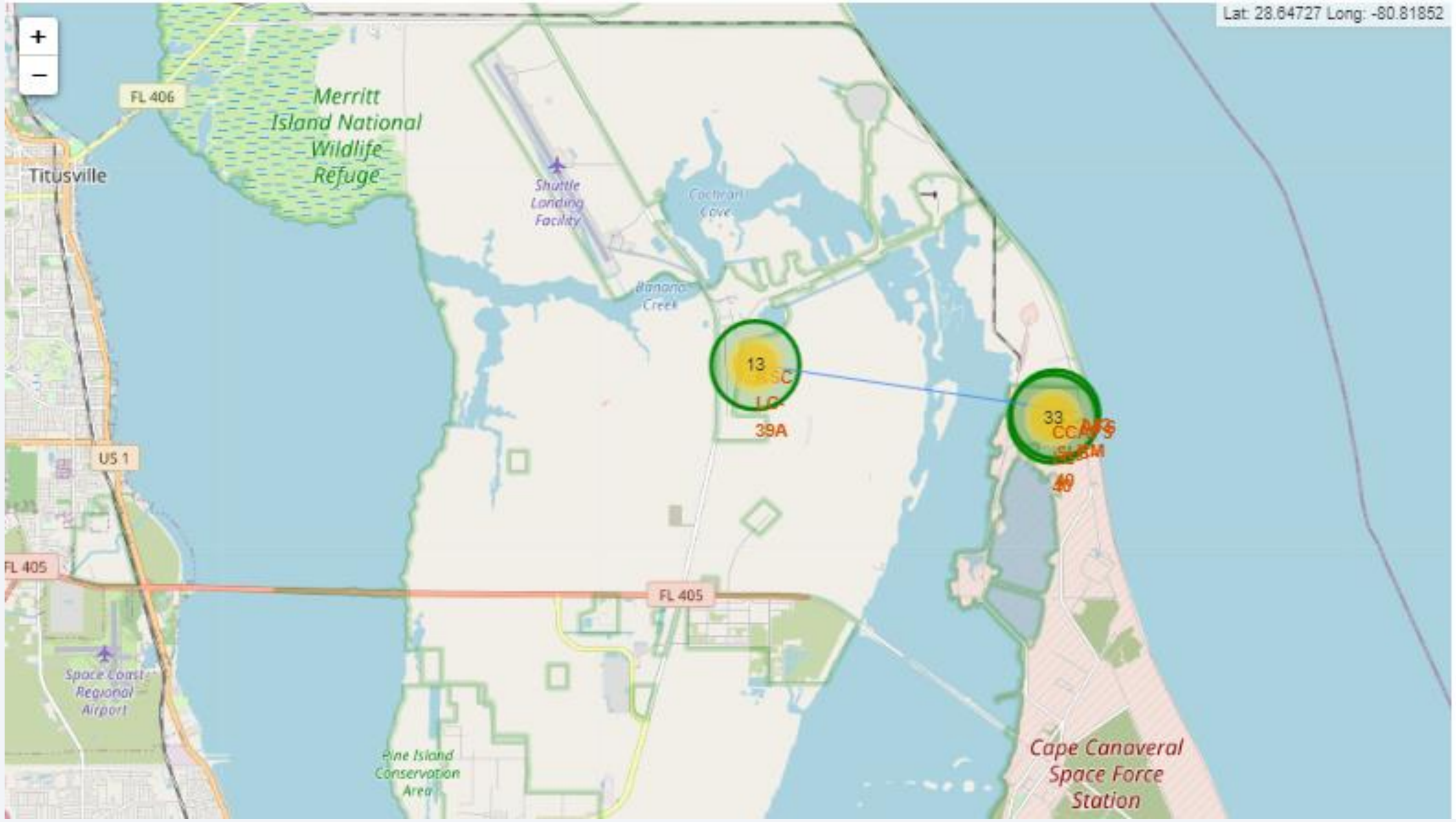
```
# Initial the map
site_map = folium.Map(location=nasa_coordinate, zoom_start=5)
# For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition, add Launch site name as a p
opup label
circle=[]
marker=[]

for i in range(4):
    coordinate=[launch_sites_df['Lat'][i],launch_sites_df['Long'][i]]
    circle.append(folium.Circle(coordinate, radius=1000, color='green', fill=True).add_child(folium.Popup(launch_sites_df['Launc
h Site'][i])))
    marker.append(folium.map.Marker(coordinate, icon=DivIcon(icon_size=(20,20),icon_anchor=(0,0), html='<div style="font-size: 1
2; color:#d35400;"><b>%s</b></div>' %launch_sites_df['Launch Site'][i], )))

#pd.DataFrame(circle)
#pd.DataFrame(marker)

for c in circle:
    site_map.add_child(c)
for m in marker:
    site_map.add_child(m)

site_map
```



Success/failed launches for CCAFS site on the map



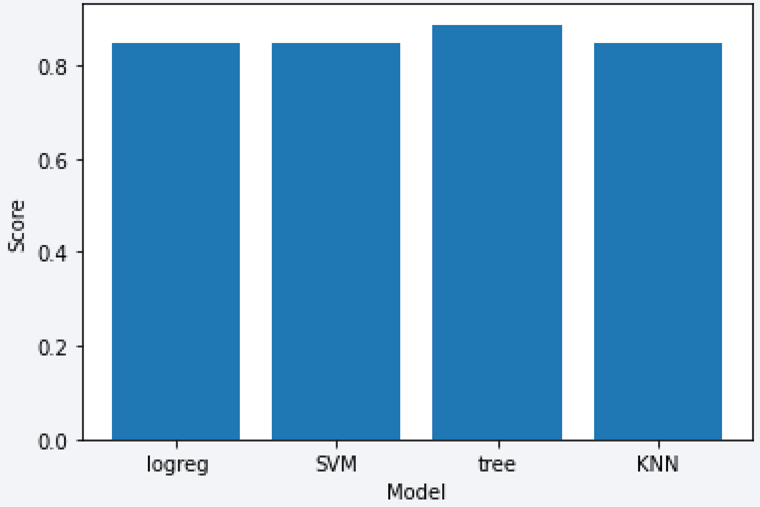
Distance between CCAFS and KSC



Section 6

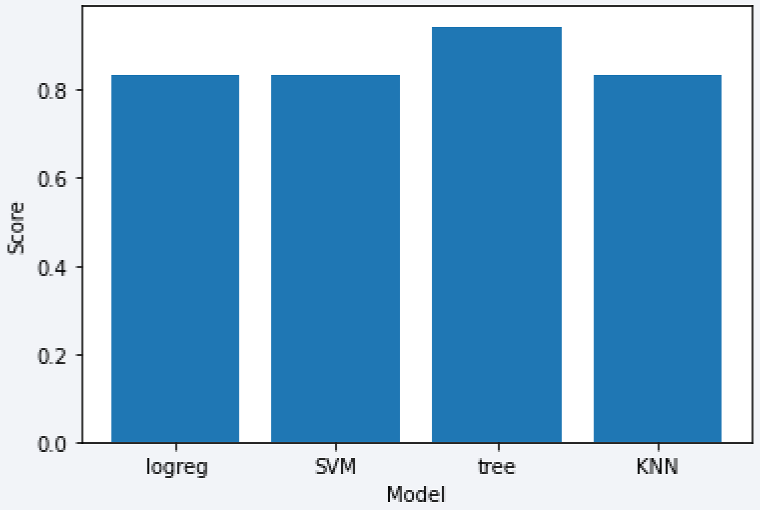
Predictive Analysis (Classification)

Classification Accuracy



Built model accuracy scores (training)

As seen in the figure (left), the decision tree model has the highest accuracy score in training. Therefore, it is the model that would result in the best predictions. This is supported in from accuracy score on the test data set as shown below.



Built model accuracy scores (test)

Confusion Matrix

confusion matrix of the best performing model (tree)

The confusion matrix above shows that the tree model was able to distinguish between the different classes. It further shows that the model predicted true landings with high accuracy. Although, it labeled 1 non-true landing as a landing.

Conclusions

- The Cape Canaveral Space Launch Complex 40 was the most frequently used launching location (55 launches), followed by Vandenberg Air Force Base Space Launch Complex 4E (22) and the rest were launched from the Kennedy Space Center Launch Complex 39A (13).
- The overall landing success rate was 67%. The Cape Canaveral had the lowest success rate (60%) compared to the other 2 locations which had a success rate of 70%.
- More than 53% of the launched satellites were distained for The GTO and ISS orbits but had the lowest success rates. Nevertheless, the success rate improved overtime. Sixty landing attempts (66.7%) were successful. Drone ship landing was the most popular method for landing with success rate of 83.6% .
- Classification Trees method returned the most accurate predictions (accuracy score=0.8875 (cv) and 0.944 on the test set); meanwhile, SVM, logistic regressions and KNN methods were less accurate in their predictions (accuracy score =0.83 on the test set)
- Further detail available from: https://github.com/alimas2/Capstone-Project

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!