IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sahil Ahuja
April 08,2022

Outline

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

Executive Summary

- Summary of methodologies
    1. Collection of data via web scrapping
    2. Collection of Data through API
    3. Data Wrangling
    4. Exploratory data analysis with SQL
    5. Exploratory data analysis with Visualization
    6. Interactive dashboards with Folium
    7. Classification using machine learning algorithms
- Summary of all results
    - EDA results
    - Dashboard results
    - Classification results

Introduction

- **Project background and context**

SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers

What factors contribute to the successful landing of the rockets

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using web scrapping and SpaceX API
- Perform data wrangling
  - Irrelevant data was removed and one hot encoding was applied
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- The data was extracted from the table of Falcon launch wikipage
- It was requested using a GET method and then all the columns were extracted and fit into a data frame
- Further SpaceX API was used to get more data
- The response was in JSON format which was further noramalized and converted into a data frame

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting

- Githublink:https://github.com/sahilahuja09/IBM_data_science_capstone/blob/c8d732c0e9377c9cb0ed5b02733ef6bc774f72d5/data%20collection%20api%20notebook%20.ipynb

Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

Check the content of the response

To make the requested JSON results more consistent, we will use the followir

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-st
```

We should see that the request was successfull with the 200 status response

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it i

```
# Use json_normalize meethod to convert the json result into a
data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- Github Link: https://github.com/sahilahuja09/IBM_data_science_capstone/blob/c8d732c0e9377c9cb0ed5b02733ef6bc774f72d5/Data%20Collection%20with%20Web%20Scraping.ipynb

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an H

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a respons
soup = BeautifulSoup(response.content,'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
# Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

After you have fill in the parsed launch record values

```
]: df=pd.DataFrame(launch_dict)
```

We can now export it to a **CSV** for the next section, b

Data Wrangling

- All the null values were identified
- Data type of each column was noted
- Number of launches on each site was calculated
- Occurrence of different orbits was counted
- Mission outcome was noted

Github link : https://github.com/sahilahuja09/IBM_data_science_capstone/blob/c8d732c0e9377c9cb0ed5b02733ef6bc774f72d5/EDA.ipynb

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

False ASDS means the mission outcome was unsuccessfully landed to a

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

EDA with Data Visualization

- Catplot used to display relation between flight number and launch site ,Payload and launch site, payload and orbit type
- Bar chart to visualize relationship between success rate of each orbit
- Line chart to visualize successful yearly trend
- Githublink: https://github.com/sahilahuja09/IBM_data_science_capstone/blob/c8d732c0e9377c9cb0ed5b02733ef6bc774f72d5/EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

- SQL used for :
  - Display unique launch sites
  - Launch sites beginning with 'CCA'
  - Calculate total payload carried by NASA Boosters
  - Display average payload mass
  - Boosters with successful drone ship landing
  - Total outcomes(successful and failure)

- Github Link:https://github.com/sahilahuja09/IBM_data_science_capstone/blob/c8d732c0e9377c9cb0ed5b02733ef6bc774f72d5/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Marker , Circles, Lines were added to the folium map
- Green color represented successful outcome whereas red meant failure
- Line was used to represent distance between coastline and launch site
- Github Link :https://github.com/sahilahuja09/Space-X-Falcon-9-First-Stage-Landing-Prediction/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

- Plotted a pie chart to show number of launches at each site
- Plotted a scatter plot to show relation between outcome and payload mass for different booster version
- Github link :https://github.com/sahilahuja09/Space-X-Falcon-9-First-Stage-Landing-Prediction/blob/master/app.py

Predictive Analysis (Classification)

- The data was standardized then split into training and testing
- Then data was trained on the following classification algorithms :
  - Logistic regression
  - Support vector classifier
  - Decision tree
  - KNN
- GridsearchCV was used to determine the best hyperparameter and at the end it was found that decision tree had the best score
- Github Link :https://github.com/sahilahuja09/Space-X-Falcon-9-First-Stage-Landing-Prediction/blob/master/Machine%20Learning%20Prediction.ipynb

Results

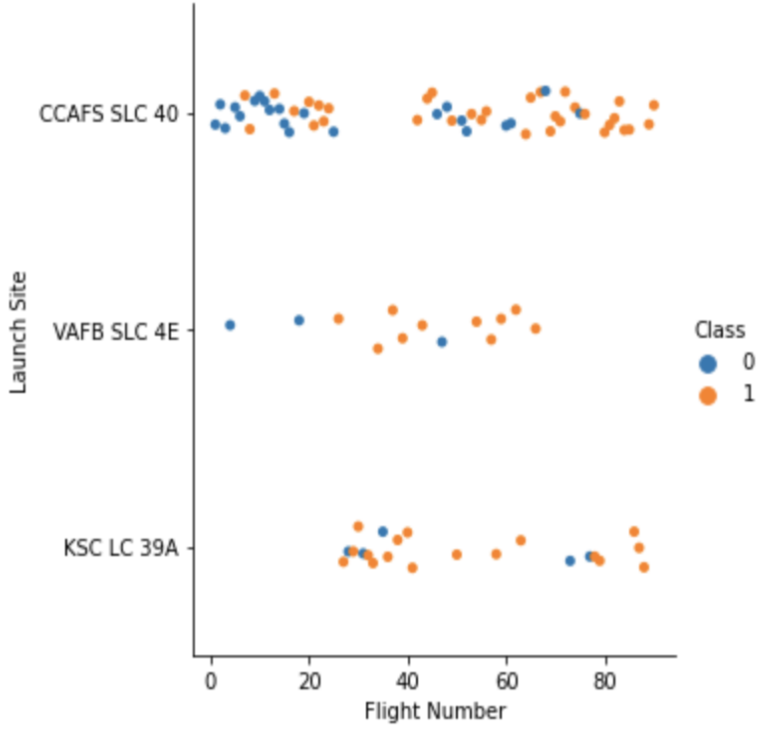
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(x = 'FlightNumber' ,y = 'LaunchSite' , hue ='Class' , data = df)
plt.xlabel("Flight Number" )
plt.ylabel("Launch Site")
plt.show()
```

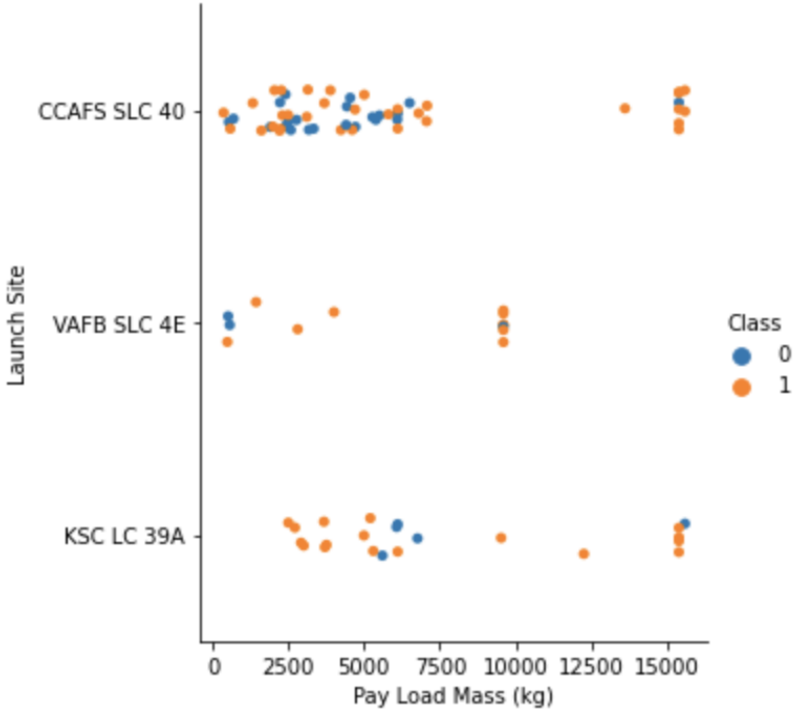


It was found that the majority of flights took off from CCAFS SLC 40 launch site

Payload vs. Launch Site

[6]:

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(x = 'PayloadMass' ,y = 'LaunchSite' , hue ='Class' , data = df)
plt.xlabel("Pay Load Mass (kg)" )
plt.ylabel("Launch Site")
plt.show()
```

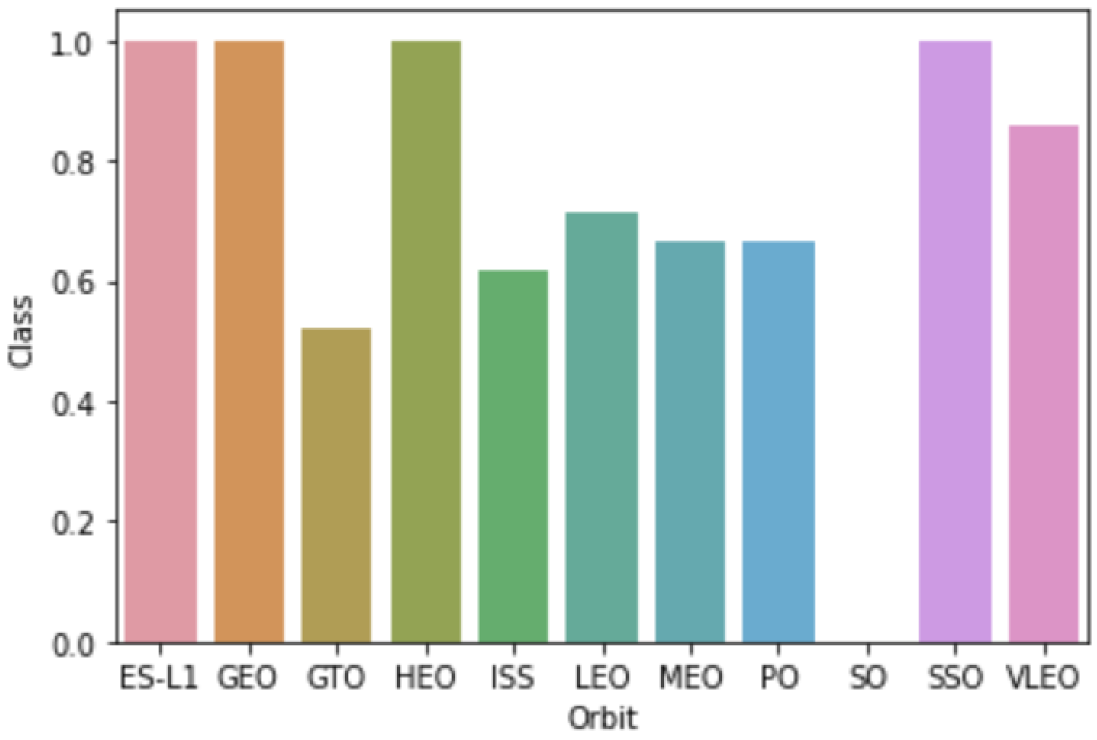


At all the sites , flights with higher mass had higher success rate

Success Rate vs. Orbit Type

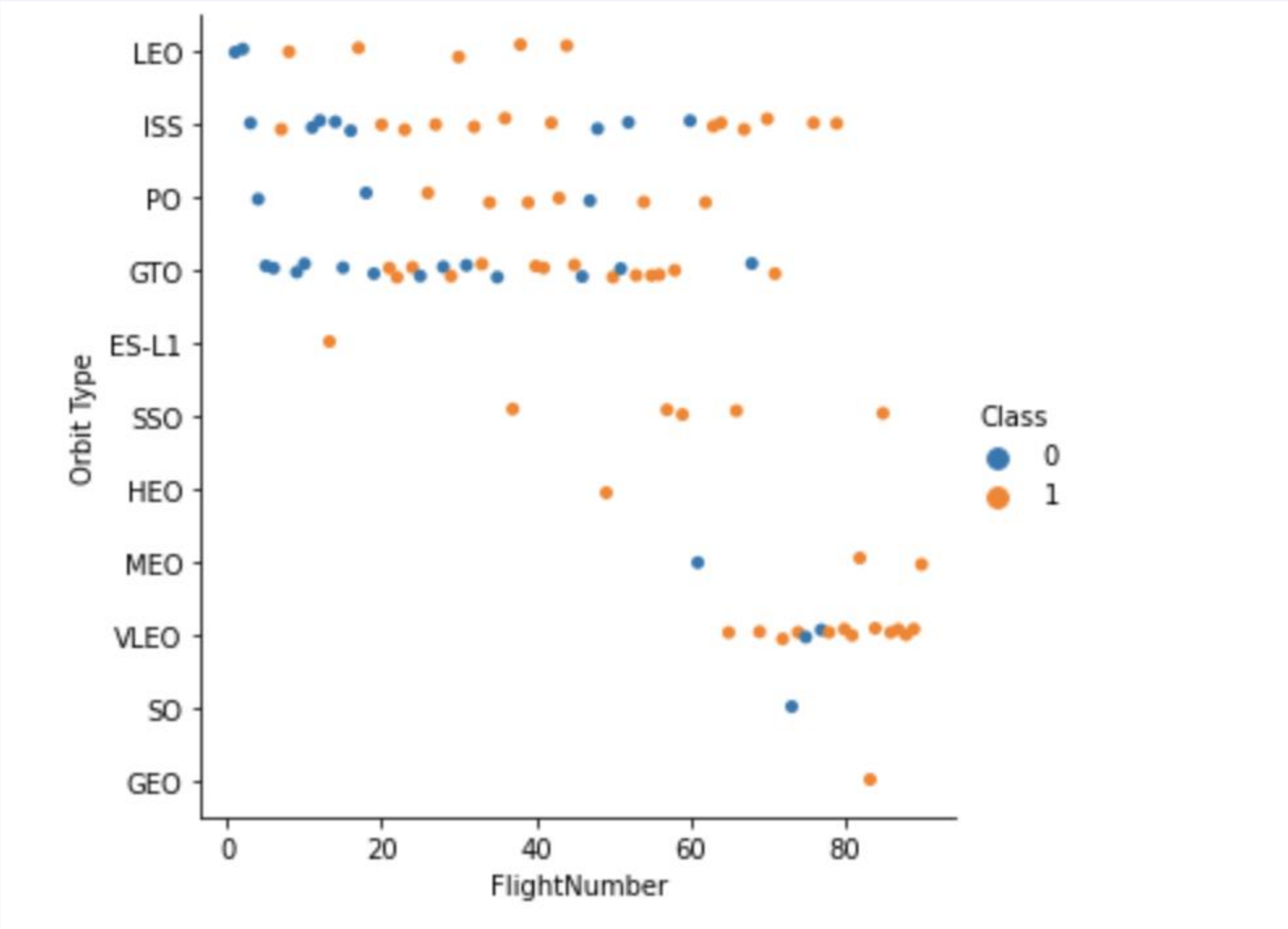
```
# HINT use groupby method on Orbit column and get the mean of Class column
orbits = df.groupby('Orbit').mean()
sns.barplot(x =  orbits.index , y =orbits['Class'] , data = orbits)
```

```
<AxesSubplot:xlabel='Orbit', ylabel='Class'>
```



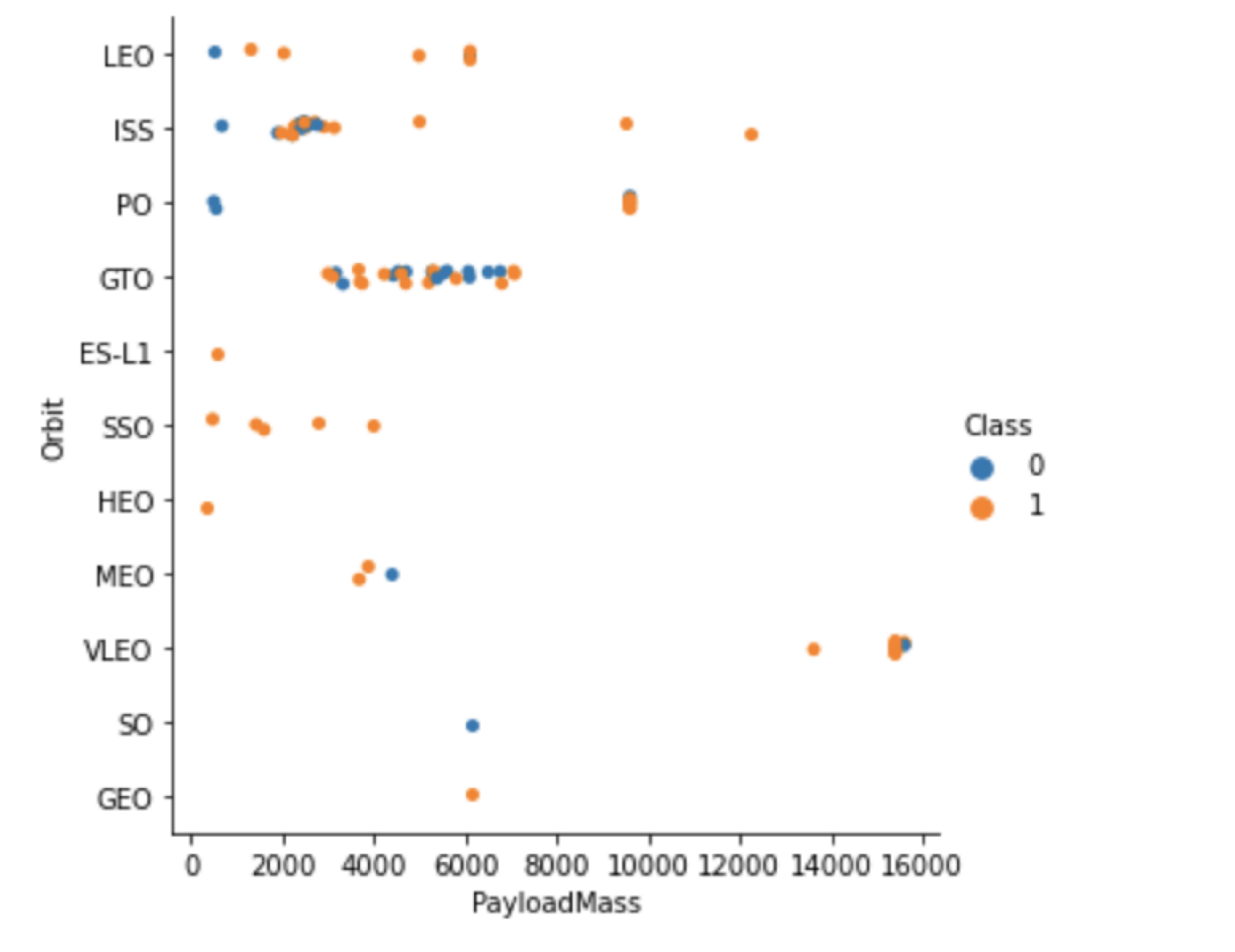
ES-L1 ,GEO , HEO , SSO had the highest success rate

Flight Number vs. Orbit Type



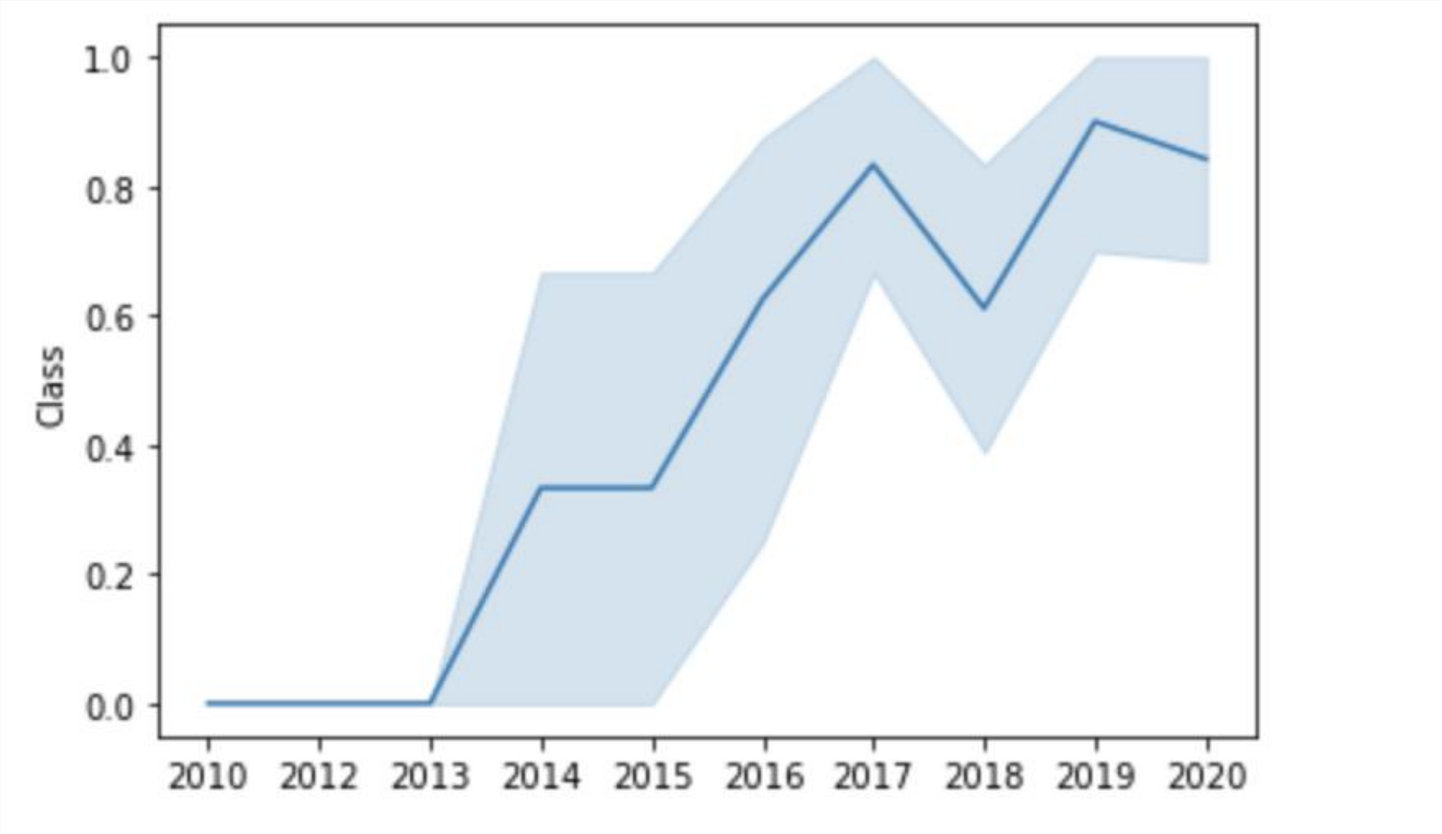
In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



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

Launch Success Yearly Trend



Sucess rate since 2013 kept increasing till 2020

All Launch Site Names

[6]:

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

There are 4 unique launch sites

Launch Site Names Begin with 'CCA'

]:

```
%%sql
select launch_site
from spacextable
where launch_site like 'CCA%'
limit 5
```

```
 * ibm_db_sa://rzs32041:***@6667d8e9-9d4d-4ccb-ba3
Done.
```

]:

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

Total Payload Mass

```
[8]:
%%sql
select sum(PAYLOAD_MASS__KG_)
from spacextable
where customer = 'NASA (CRS)'

 * ibm_db_sa://rzs32041:***@6667d8e9-9d4d-
Done.
[8]:
```

1
45596

Average Payload Mass by F9 v1.1

```
]:  %%sql
    select avg(PAYLOAD_MASS__KG_)
    from spacextable
    where booster_version = 'F9 v1.1'

     * ibm_db_sa://rzs32041:***@6667d8e9-9
    Done.
]:      1
    2928
```

First Successful Ground Landing Date

```
]: %%sql
   select min(date)
   from spacextable
   where landing_outcome = 'Success (ground pad)'

 * ibm_db_sa://rzs32041:***@6667d8e9-9d4d-4ccb-ba32
Done.
```

]:

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
select unique(booster_version)
from spacextable
where landing_outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000
```

```
 * ibm_db_sa://rzs32041:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databa
Done.
```

booster_version
F9 FT B1021.2
F9 FT B1031.2
F9 FT B1022
F9 FT B1026

Total Number of Successful and Failure Mission Outcomes

```
%%sql
select count(landing_outcome) as number_of_success
from spacextable
where landing_outcome like 'Success%'
```

```
 * ibm_db_sa://rzs32041:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5a
Done.
```

number_of_success
61

```
%%sql
select count(landing_outcome) as number_of_failure
from spacextable
where landing_outcome like 'Failure%'
```

```
 * ibm_db_sa://rzs32041:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5a
Done.
```

number_of_failure
10

Boosters Carried Maximum Payload

```
%%sql
select unique(booster_version)
from spacextable
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextable)
```

```
 * ibm_db_sa://rzs32041:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lq
Done.
```

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

```
%%sql
select landing_outcome, booster_version,launch_site
from spacextable
where landing_outcome = 'Failure (drone ship)' and year(date) = 2015
```

```
 * ibm_db_sa://rzs32041:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd(
Done.
```

landing_outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
%%sql
select landing_outcome , count(landing_outcome) as count
from spacextable
where date between '2010-06-04 ' and '2017-03-20'
group by landing_outcome
order by count desc
```

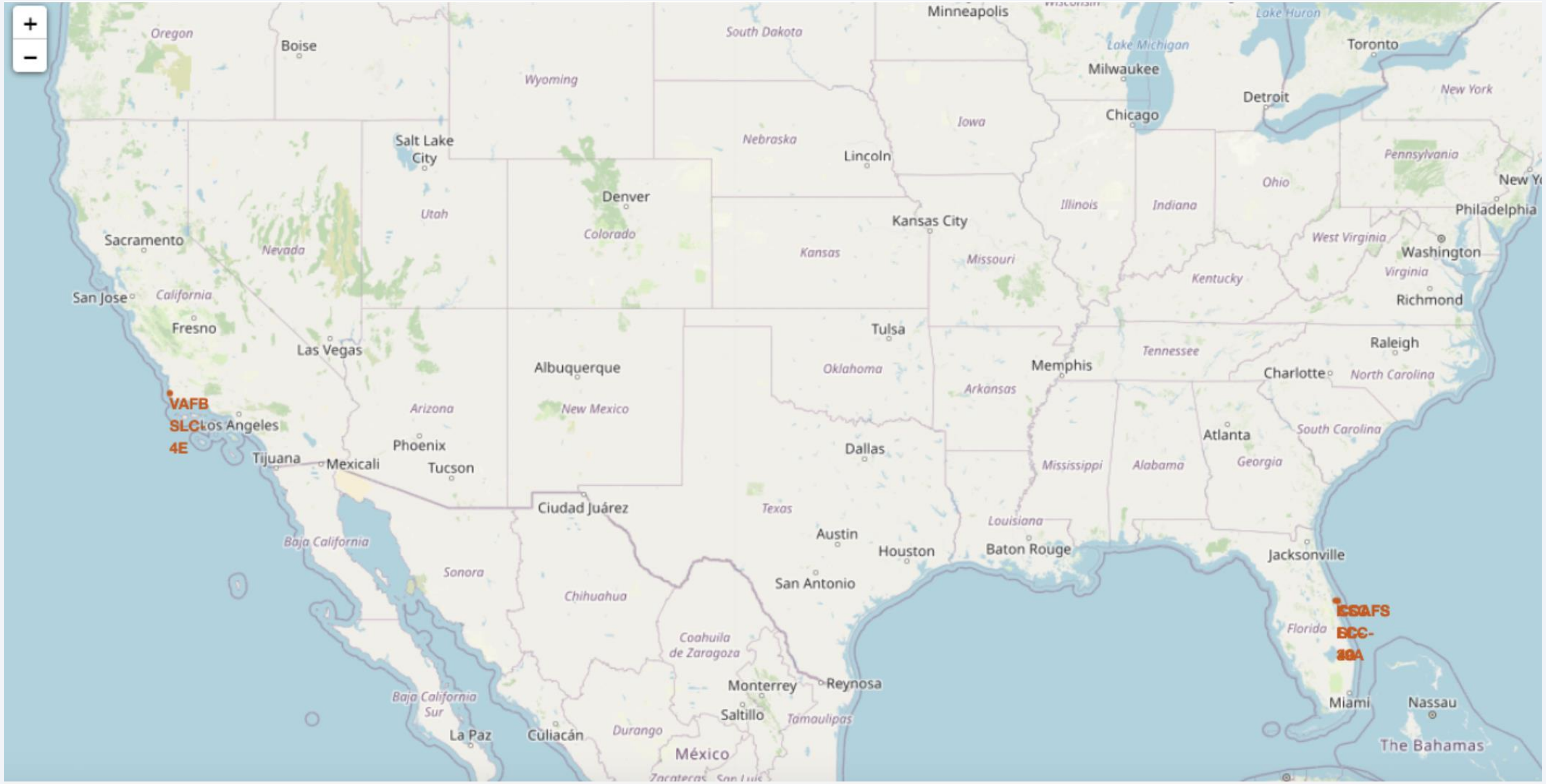
```
 * ibm_db_sa://rzs32041:***@6667d8e9-9d4d-4ccb-ba32-21da3b
Done.
```

landing_outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

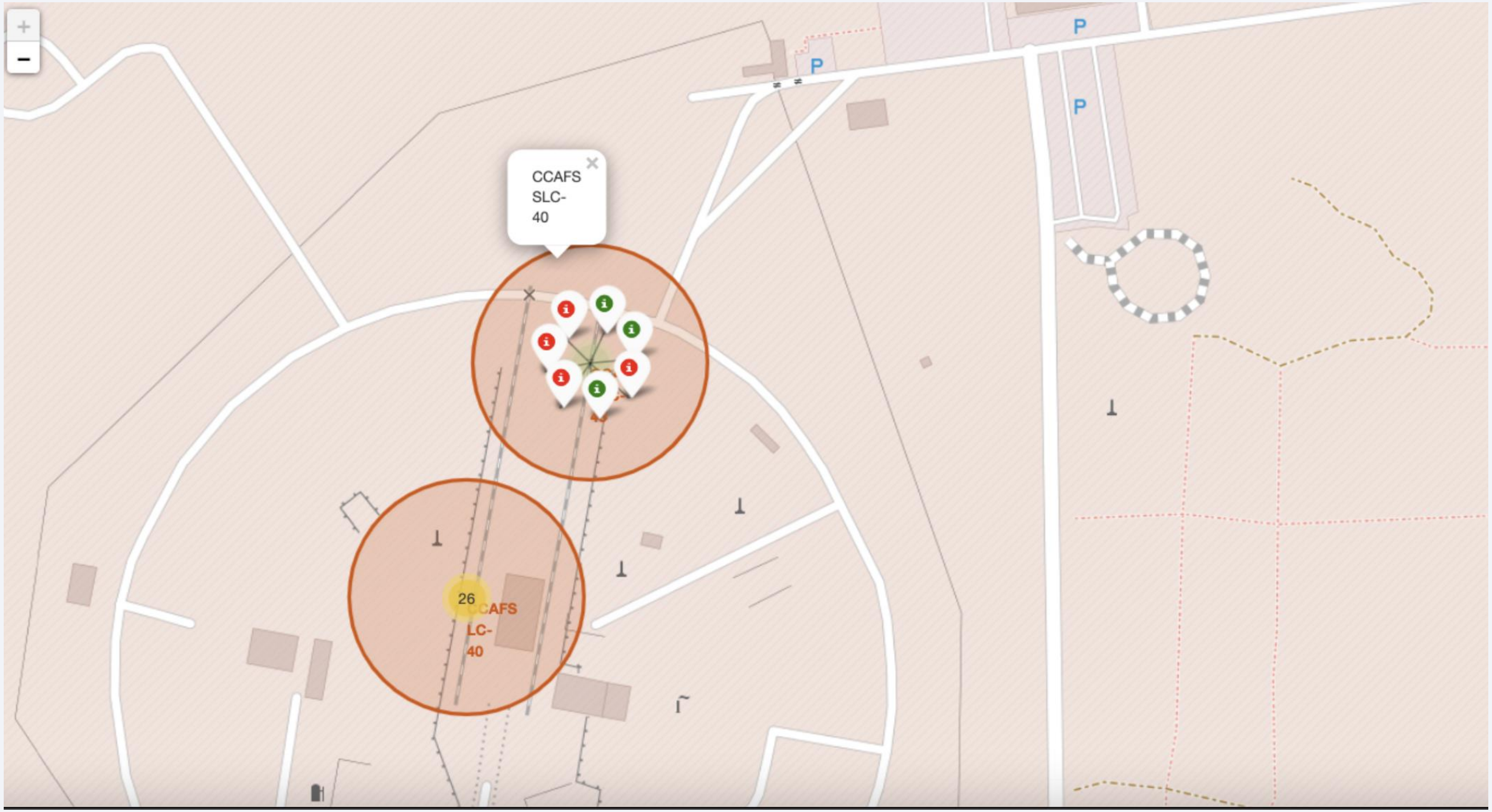
Section 3

Launch Sites Proximities Analysis

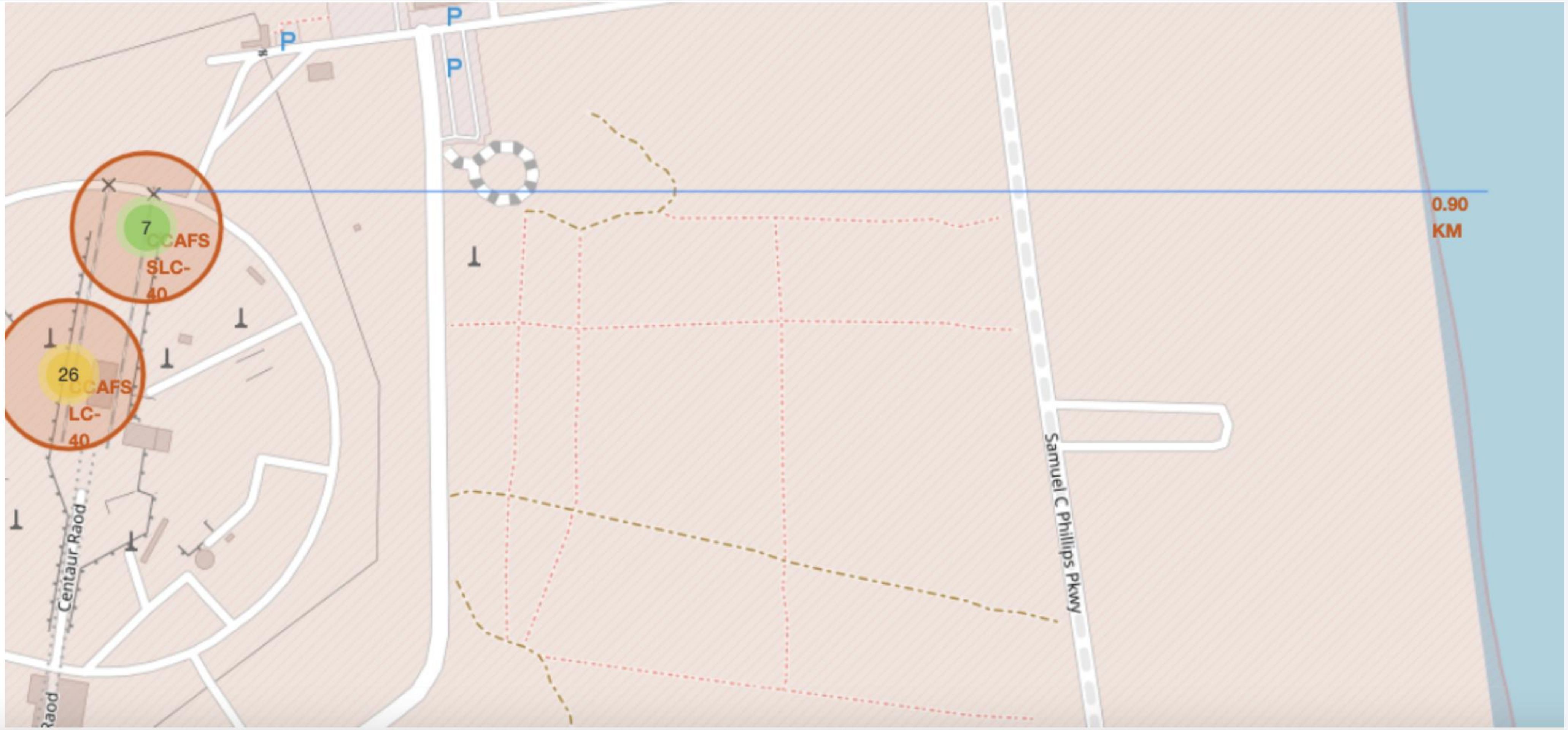
Launch sites in the US



Color coded marker clusters



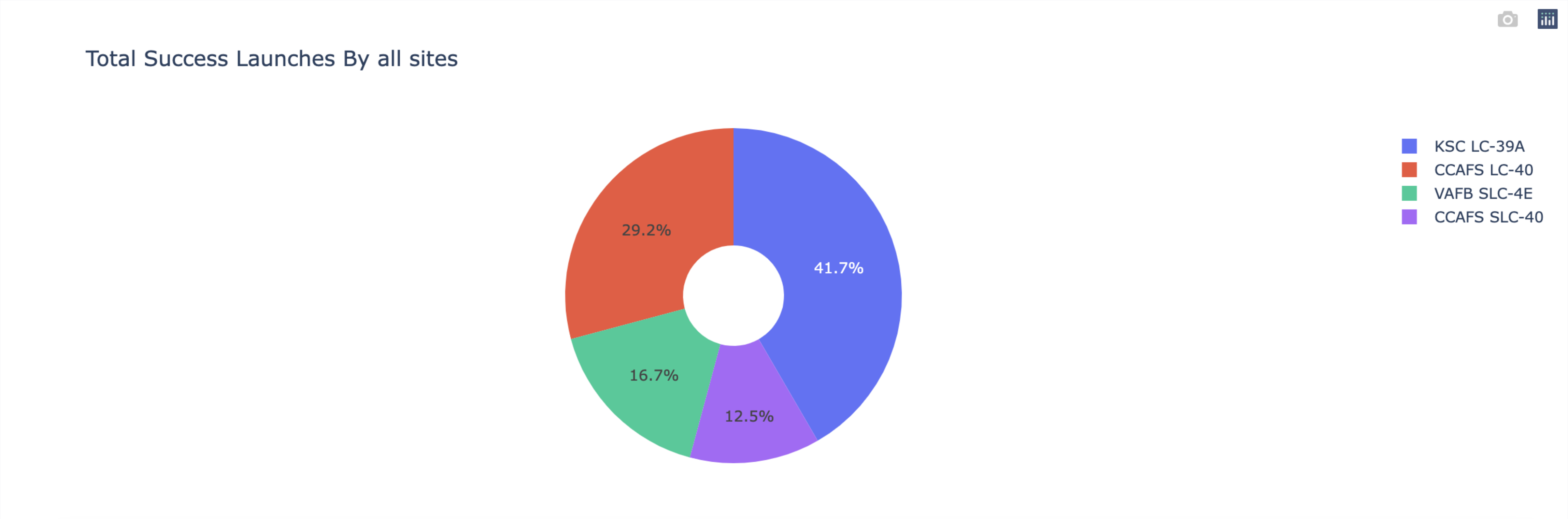
Distance between launch site and coast



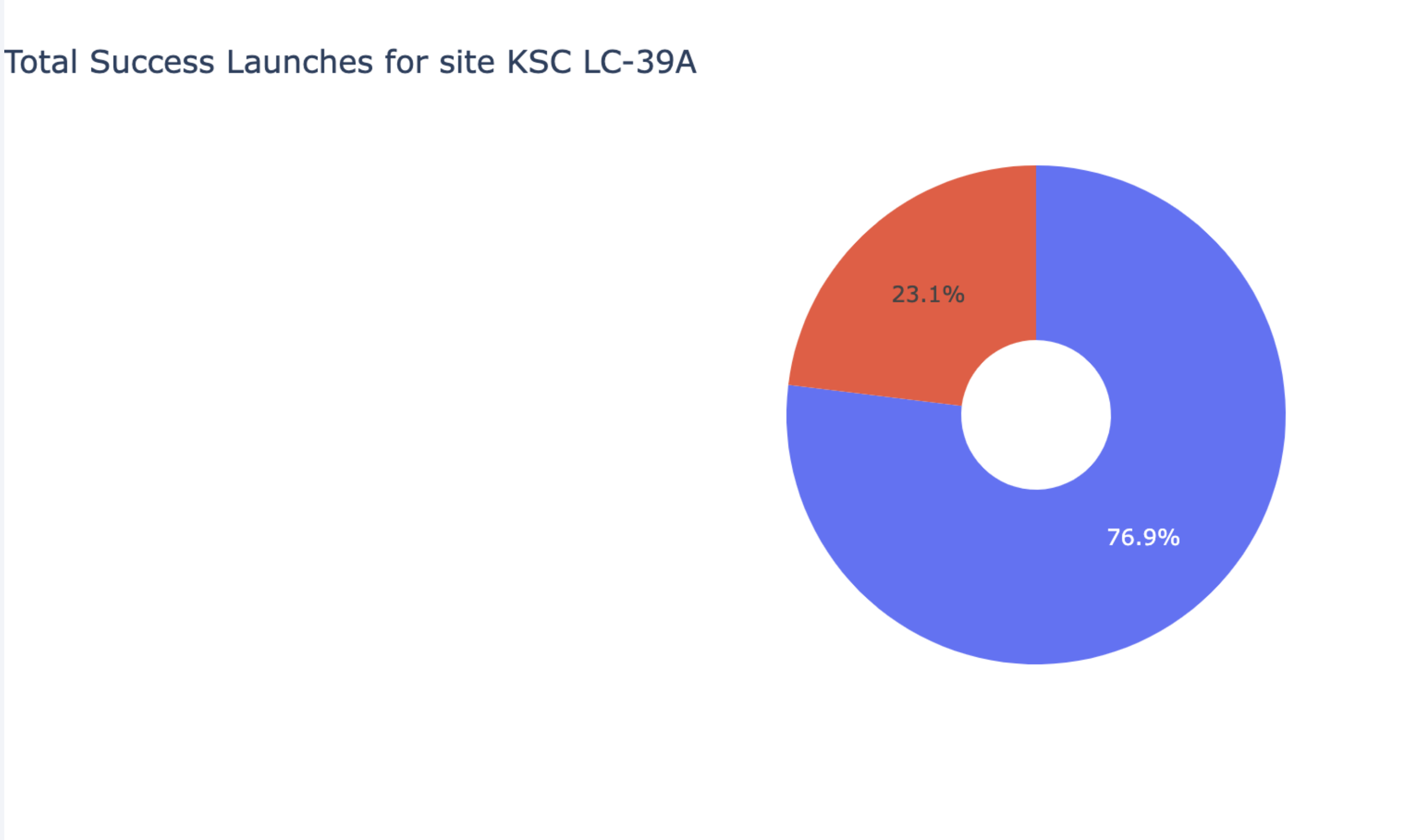
Section 4

Build a Dashboard with Plotly Dash

Succes launch for all sites



Site with highest success ratio



Payload vs. Launch Outcome scatter plot

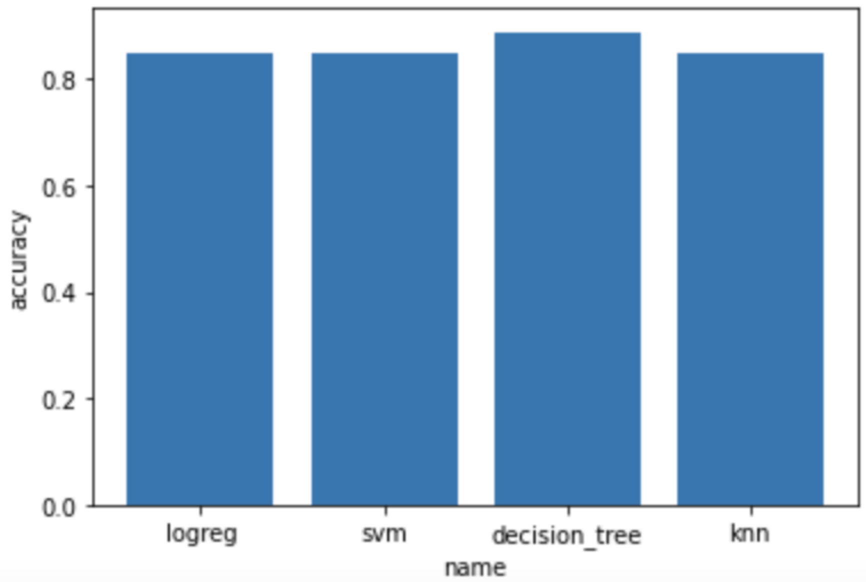
Section 5

Predictive Analysis (Classification)

Classification Accuracy

```
acc=[{"name":"logreg","accuracy":logreg_cv.best_score_},
     {"name":"svm","accuracy":svm_cv.best_score_},
     {"name":"decision_tree","accuracy":tree_cv.best_score_},
     {"name":"knn","accuracy":knn_cv.best_score_}
    ]
df=pd.DataFrame.from_dict(acc)
plt.bar(df['name'],df['accuracy'])
plt.xlabel('name')
plt.ylabel('accuracy')
```

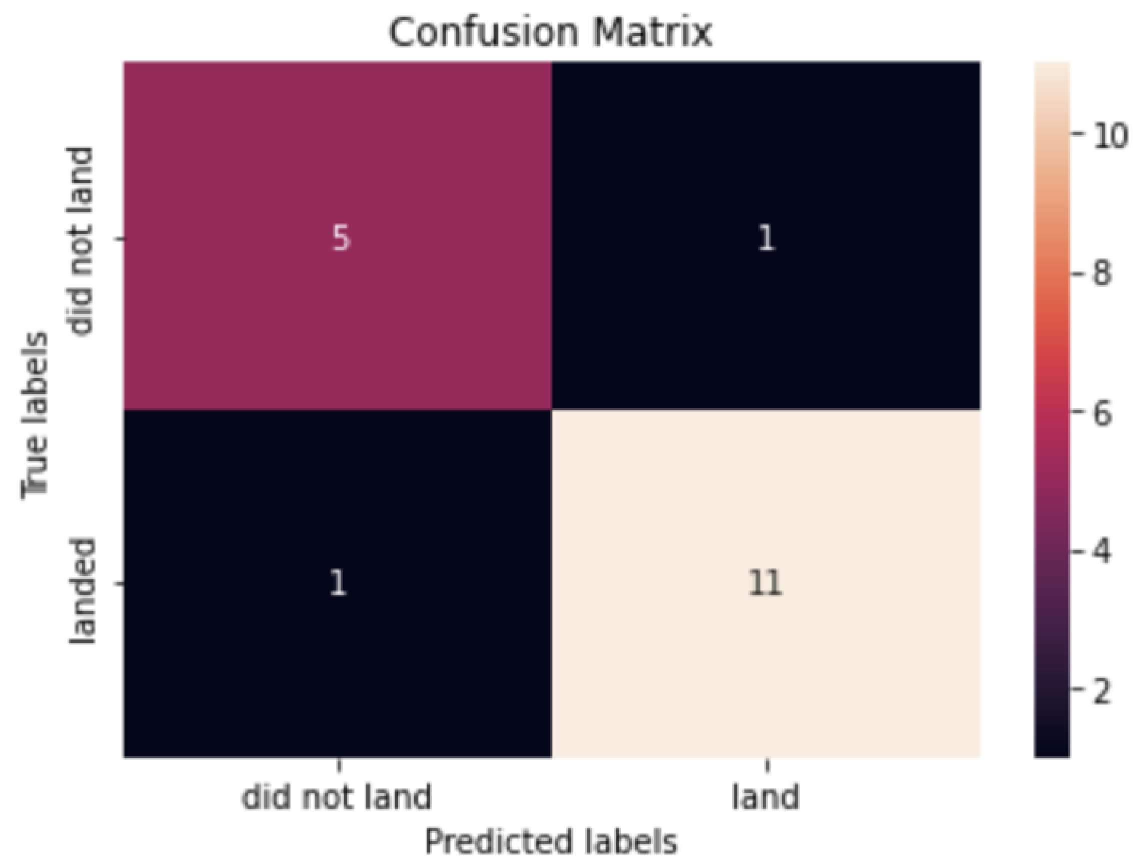
```
]: Text(0, 0.5, 'accuracy')
```



Decision tree has the highest accuracy

Confusion Matrix

```
yhat = tree.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Although decision tree gives the best accuracy it still gives 1 false positive and 1 false negative result

Conclusions

- Larger the flight amount at a site greater the success rate
- Launch success rate started to increase from 2013
- ES-L1 ,GEO , HEO , SSO orbits had the highest success rate
- Decision tree was the classifier with the highest accuracy

Thank you!