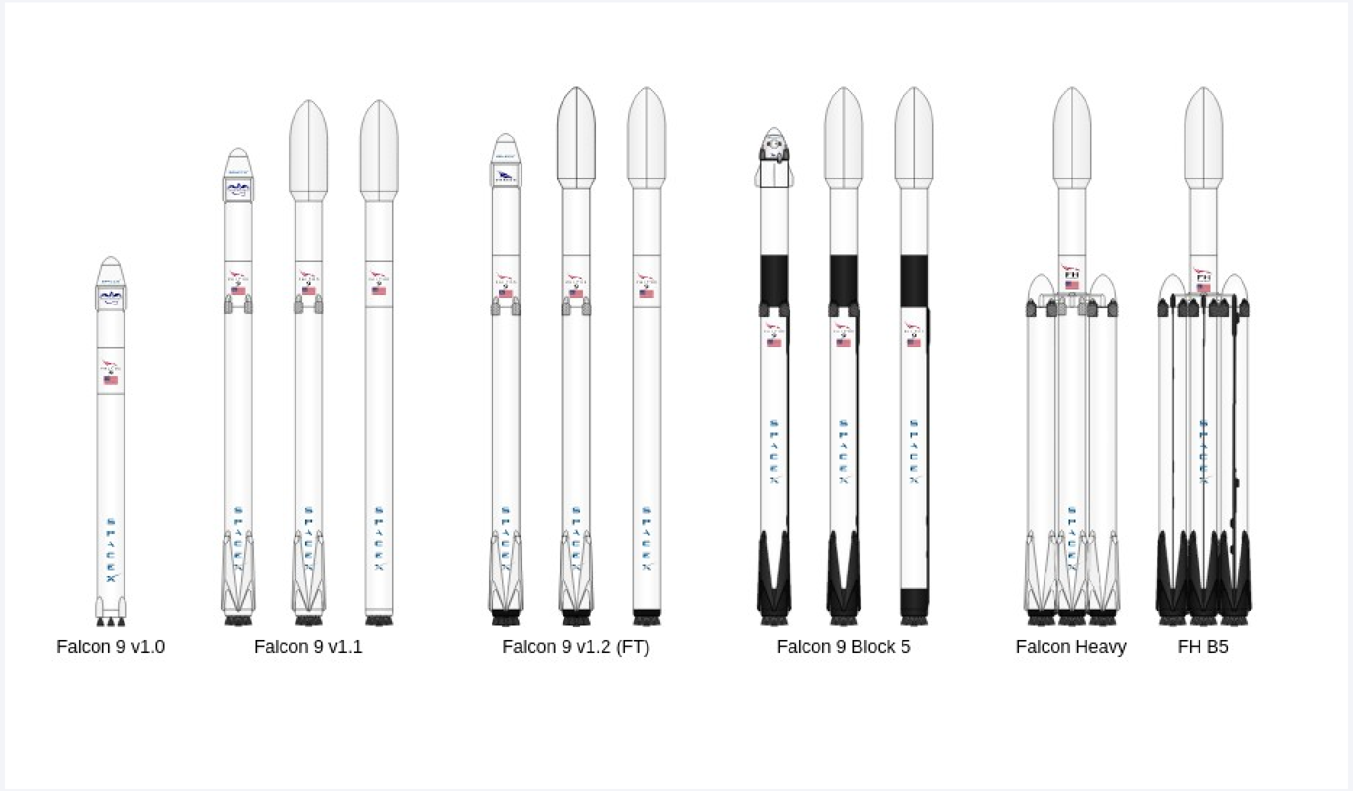
IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

J3_14
5/2/23

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Insights drawn from EDA
- Launch sites proximity analysis
- Build a Dashboard with Plotly
- Predictive Analysis (Classification)
- Conclusion
- Appendix



Python scripts were uploaded to Github in place of Jupyter Notebooks. See Github ReadMe.

Executive Summary

- Summary of methodologies
  - Data is filtered for useful information using python.
  - The data is plotted to help visualize relationships.
  - A dashboard is created to help analyze the data.
  - Machine learning techniques are used to analyze relationships.
- Summary of all results
  - Correlation exists between variables such as orbit, payload, and successful recovery of the booster engines.
  - Orbit levels and payloads have increased over time.
  - Success rates have generally improved over time.

Introduction

- Project background and context
  - SpaceX has helped to decrease the cost of space flight by creating booster engines that can be recovered and reused.
  - The success of the recovery can depend on variables such as orbit, booster version, and payload.
  - Data science and machine learning can help us to predict these relationships.

- Problems you want to find answers to
  - What are the relationships among the variables such as payload and success?
  - Has the success rate improved over time?
  - Which machine learning techniques best predict the data?

Section 1

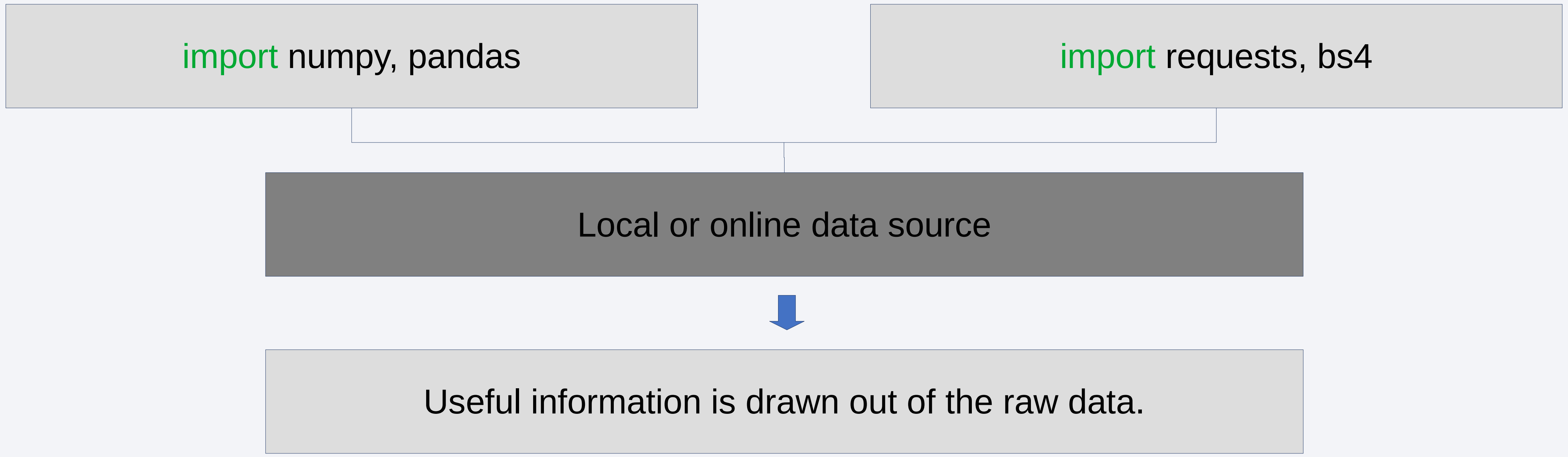
Methodology

Methodology

Executive Summary

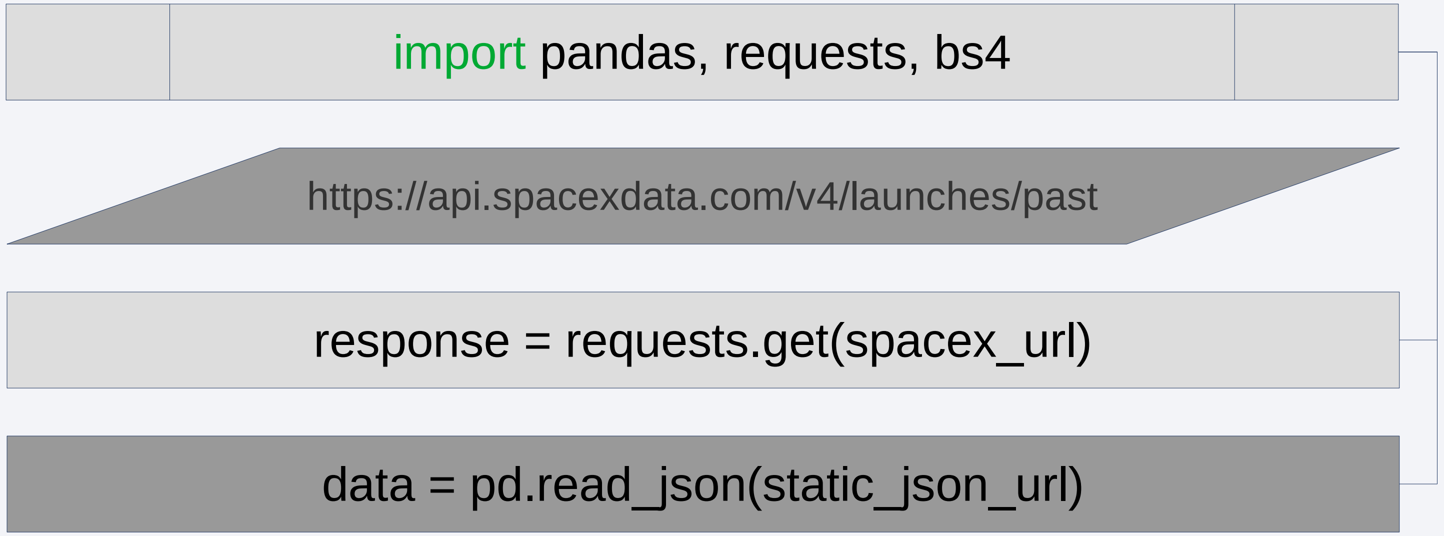
- Data collection methodology:
  - Packages like pandas and requests are used to gather data
- Perform data wrangling
  - Functions are called on html and data frames to parse raw data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Machine learning techniques are used to analyze the data

Data Collection


import numpy, pandas
import requests, bs4
Local or online data source
Useful information is drawn out of the raw data.

In this project information was gathered from web resources related to the SpaceX launches. The data can be stored online as html, csv, json, and other formats that are not easily read by humans. Some of the python packages shown above can help to get and refine the raw data.

Data Collection – SpaceX API



```
import pandas, requests, bs4
https://api.spacexdata.com/v4/launches/past
response = requests.get(spacex_url)
data = pd.read_json(static_json_url)
```

Python libraries are imported to extend functionality.

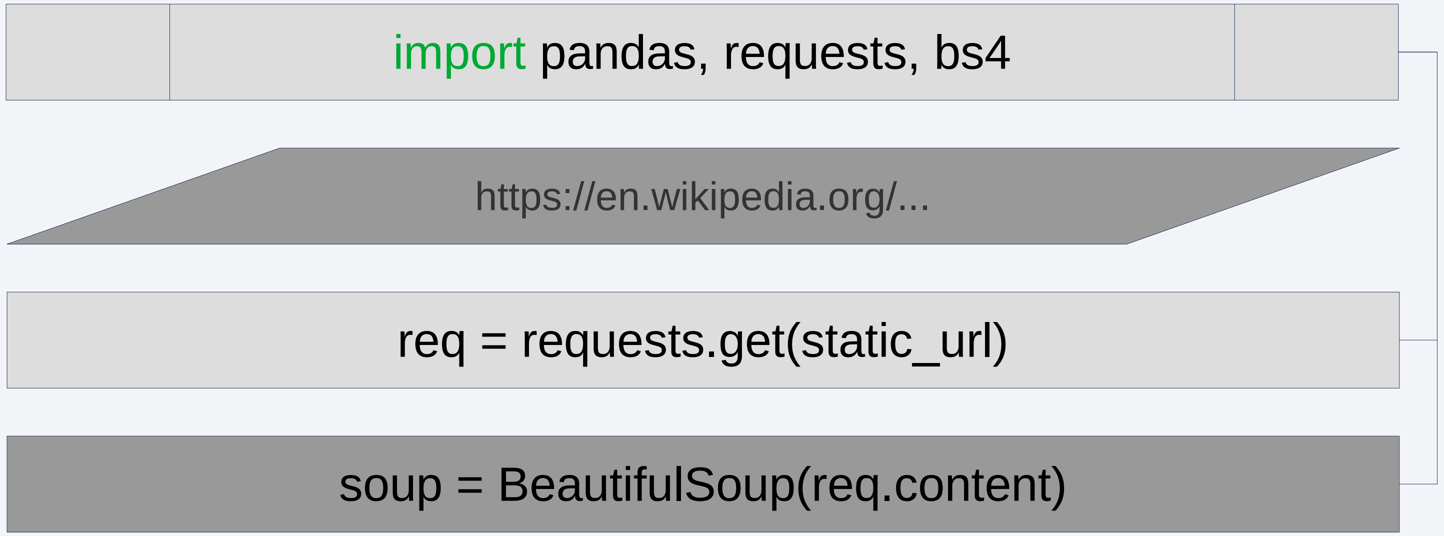
Pandas is very useful for data analysis while requests and bs4 help to work with html.

Requests can be used to get info from a website or data can be read into a pandas dataframe.

Once the data has been loaded it can then be wrangled for useful information.

https://github.com/J-3-14/P4DS-Final/blob/main/API.py

Data Collection - Scraping



```
import pandas, requests, bs4
```

```
https://en.wikipedia.org/...
```

```
req = requests.get(static_url)
```

```
soup = BeautifulSoup(req.content)
```

With the Web Scraping notebook we use BeautifulSoup to parse a requests response.

The tables from the html are used to build a dictionary object.

The dictionary is then converted into a data frame for further analysis.

https://github.com/J-3-14/P4DS-Final/blob/main/spacex-webscraping.py

Data Wrangling

```
import numpy, pandas

df = pd.read_csv(“dataset_part_1.csv")

df["Class"].mean()
```

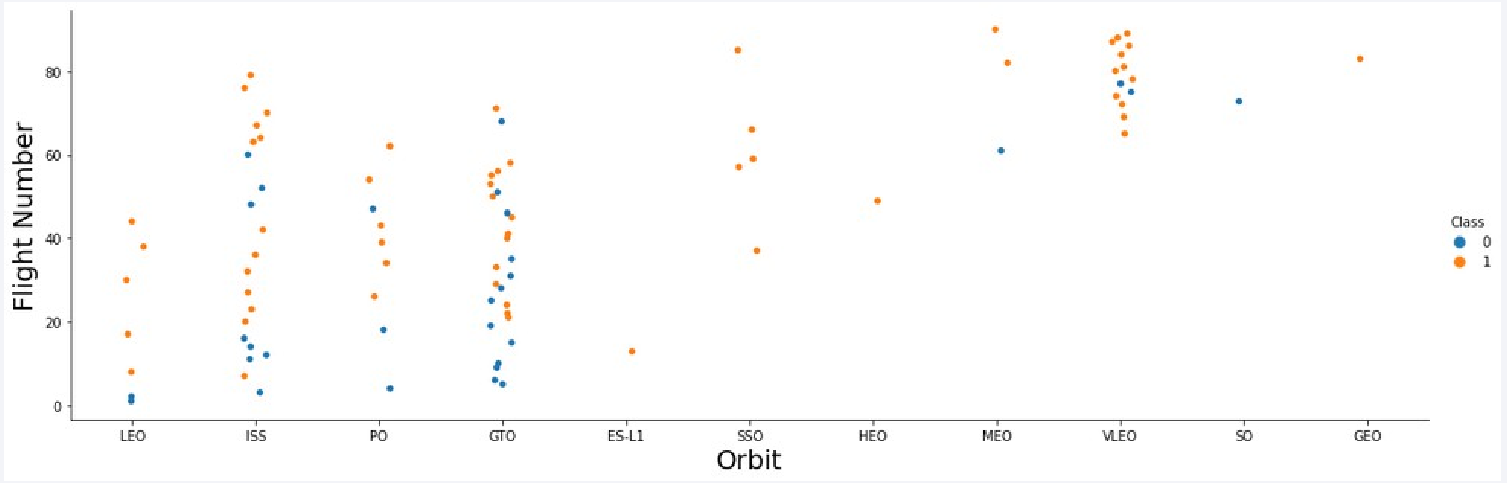
With the Data Wrangling notebook we use pandas to import a csv file.

Numpy provides additional functionality when working with numerical data.

Methods such as ‘value_counts’ can be called on the data to provide useful information.

https://github.com/J-3-14/P4DS-Final/blob/main/data%20wrangling.py

EDA with Data Visualization



Various plots were created to explore relationships within the data.

Packages such as matplotlib and seaborn can be used to help visualize these relationships.

Variables such as Flight Number, Payload, and Orbit are used to analyze effects on success.

The data shows that relationships do exist between certain variables that affect success.

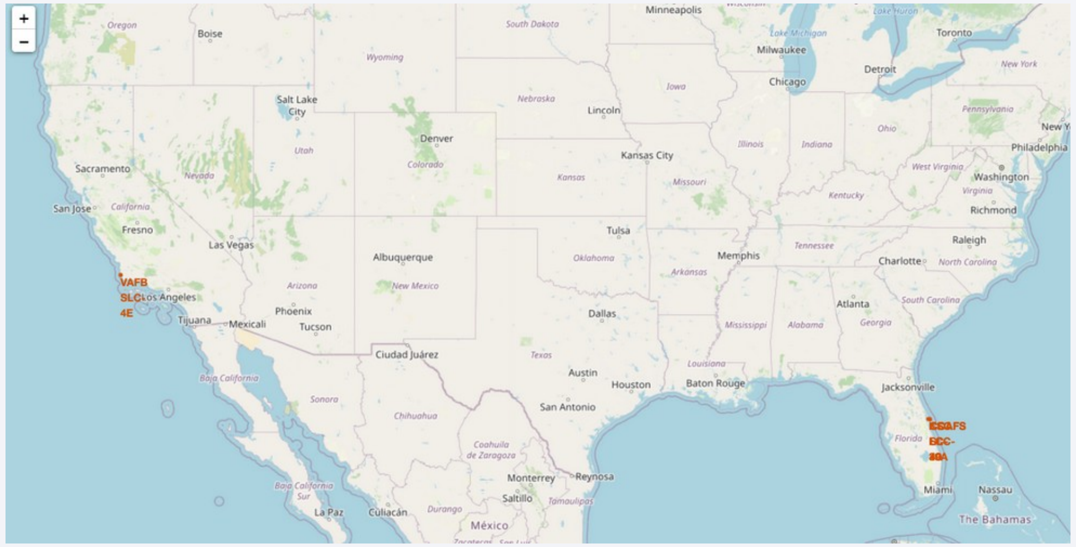
https://github.com/J-3-14/P4DS-Final/blob/main/spacex-eda-dataviz.py

EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'KSC'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date where the succesful landing outcome in drone ship was acheived.
- List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass.
- List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

https://github.com/J-3-14/P4DS-Final/blob/main/spacex-eda-sql.py

Build an Interactive Map with Folium

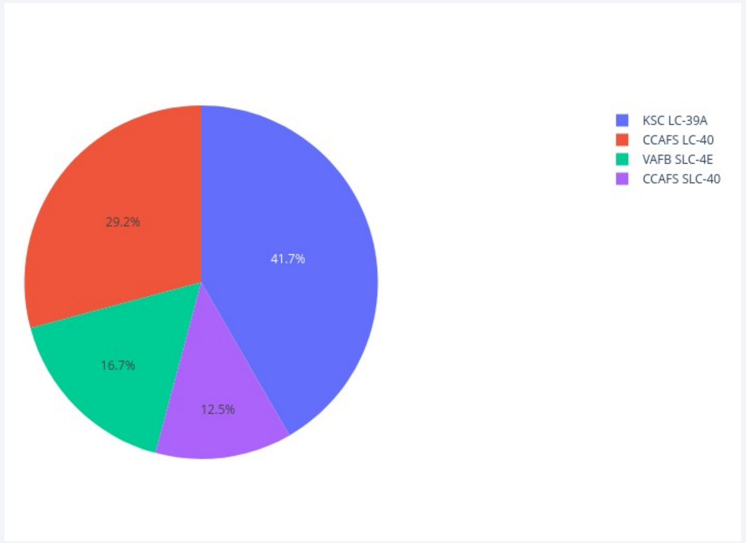


Mapping the launches can help us get a sense of where progress and setbacks took place.

- Mark all launch sites on a map
- Mark the success/failed launches for each site on the map
- Calculate the distances between a launch site to its proximity

https://github.com/J-3-14/P4DS-Final/blob/main/spacex-launch_site_location.py

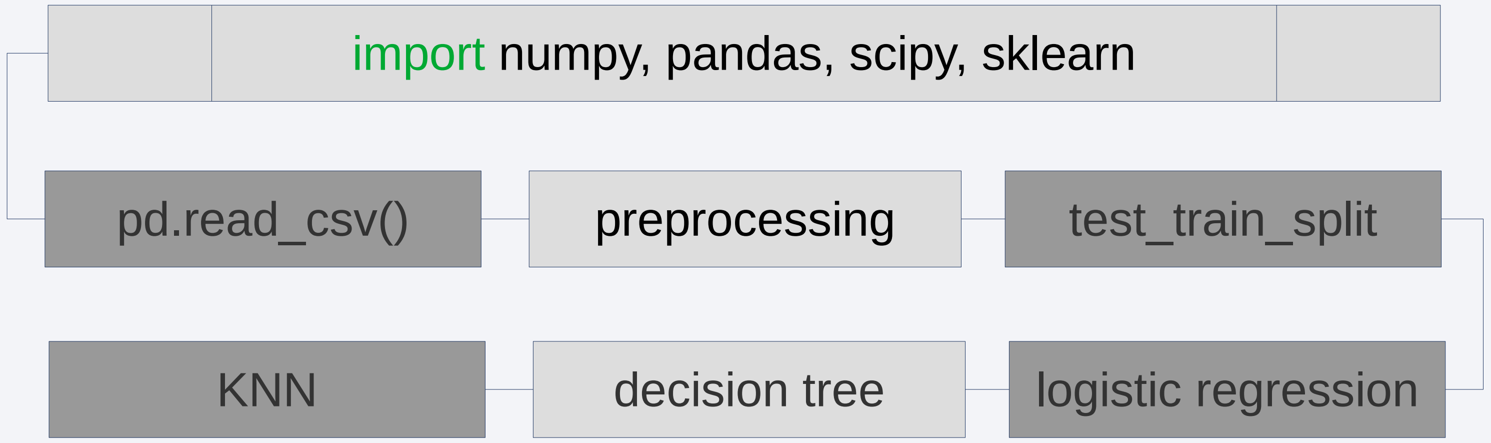
Build a Dashboard with Plotly Dash



- A dropdown list and a slider were added to dig into the data.
- The successes per launch site are plotted in a pie chart.
- A scatter plot shows the correlation between payload and launch success.

https://github.com/J-3-14/P4DS-Final/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)



In predictive analysis a confusion matrix plotting function was created and the data was split into test and train sets. GridSearchCV was the primary method used to fit and score the data training data which was then verified against the test data.

https://github.com/J-3-14/P4DS-Final/blob/main/spacex-Machine_Learning_Prediction.py

Results

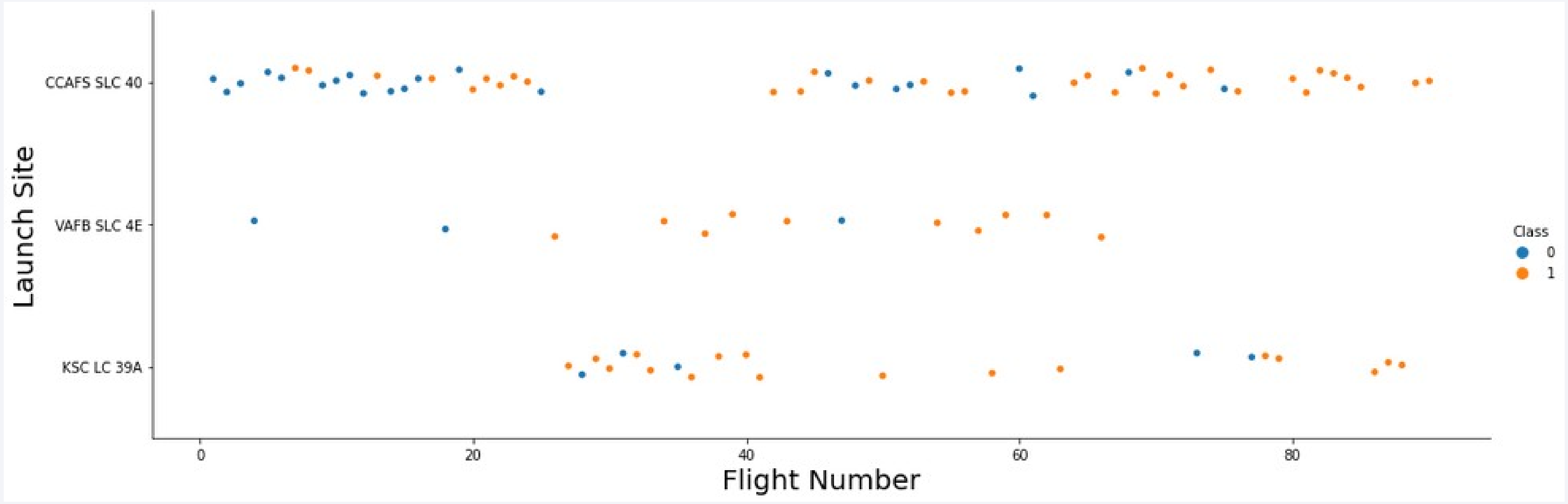
- Exploratory data analysis results
  - Gradual improvement in success rates over time
  - Success varies with payload and orbit
- Interactive analytics demo in screenshots
  - Launches took place near coastal areas with varying degrees of success
- Predictive analysis results
  - K – Nearest Neighbor clustering was the most effective model

Section 2

Insights drawn from EDA

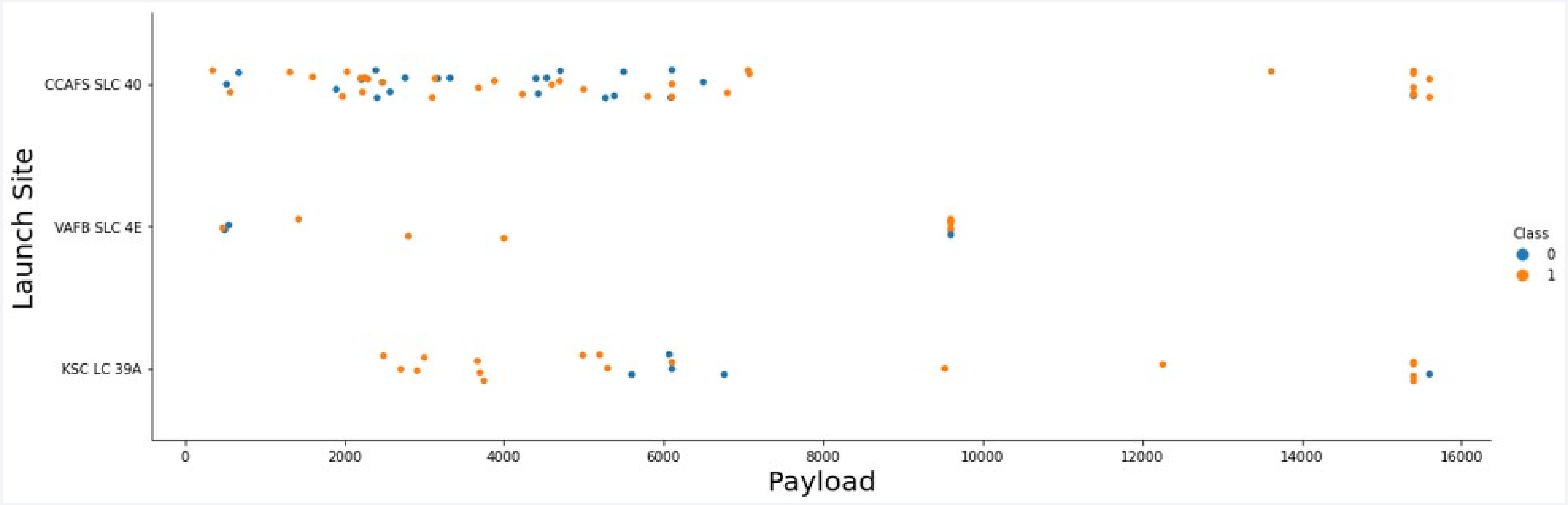
Flight Number vs. Launch Site

From this plot we can see that most of the launches originated from the CCAFS SLC-40 Launch Site. It also indicates that the success rate improved after the first 20 flights.



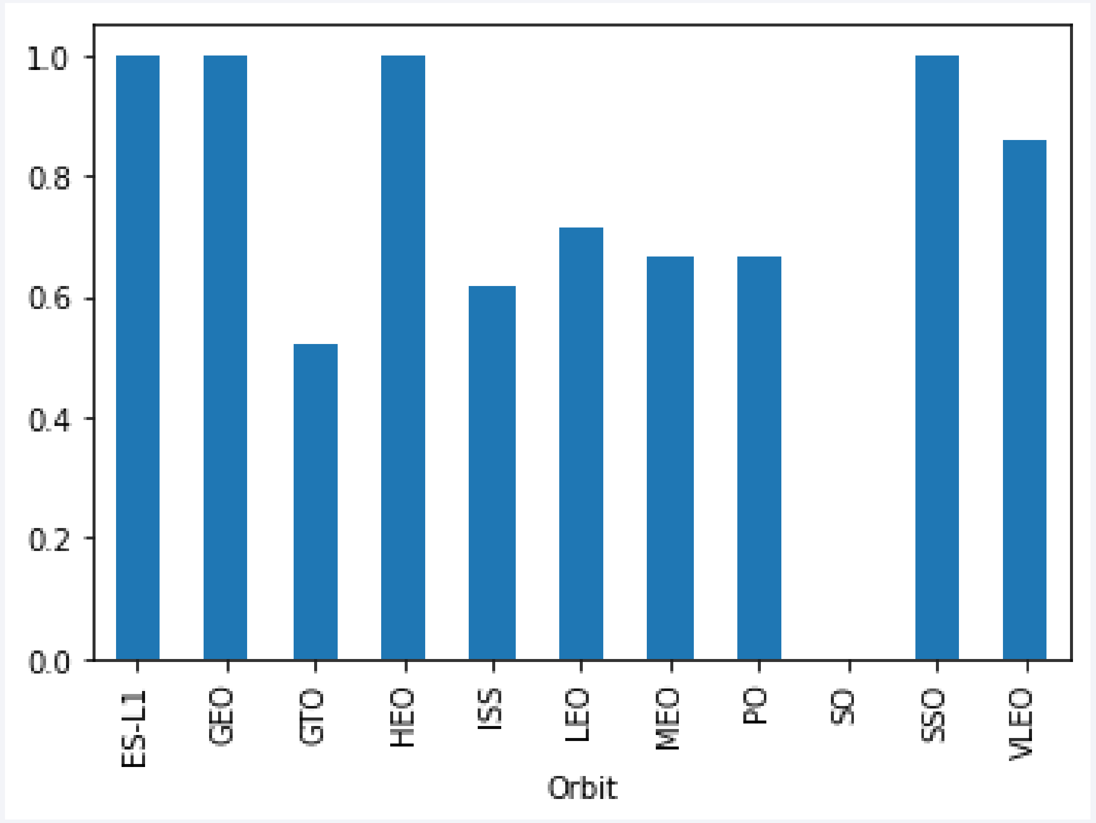
Payload vs. Launch Site

- For each of the Launch Sites a blue dot indicates a failure and orange a success
- From this we see that VAFB SLC 4E had the fewest launches and no payloads over 10,000



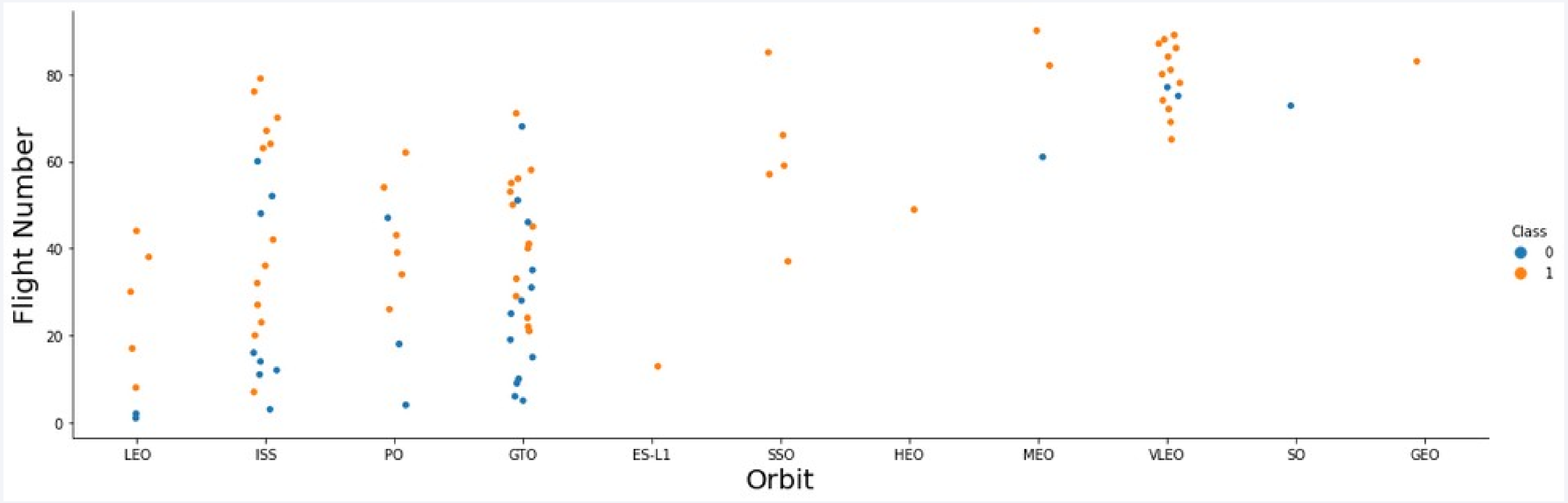
Success Rate vs. Orbit Type

- This bar plot shows that the Success Rate varies according to Orbit.
- The range varies from 100% success rate to 0% with perhaps no attempts to deploy at the ‘SO’ Orbit.



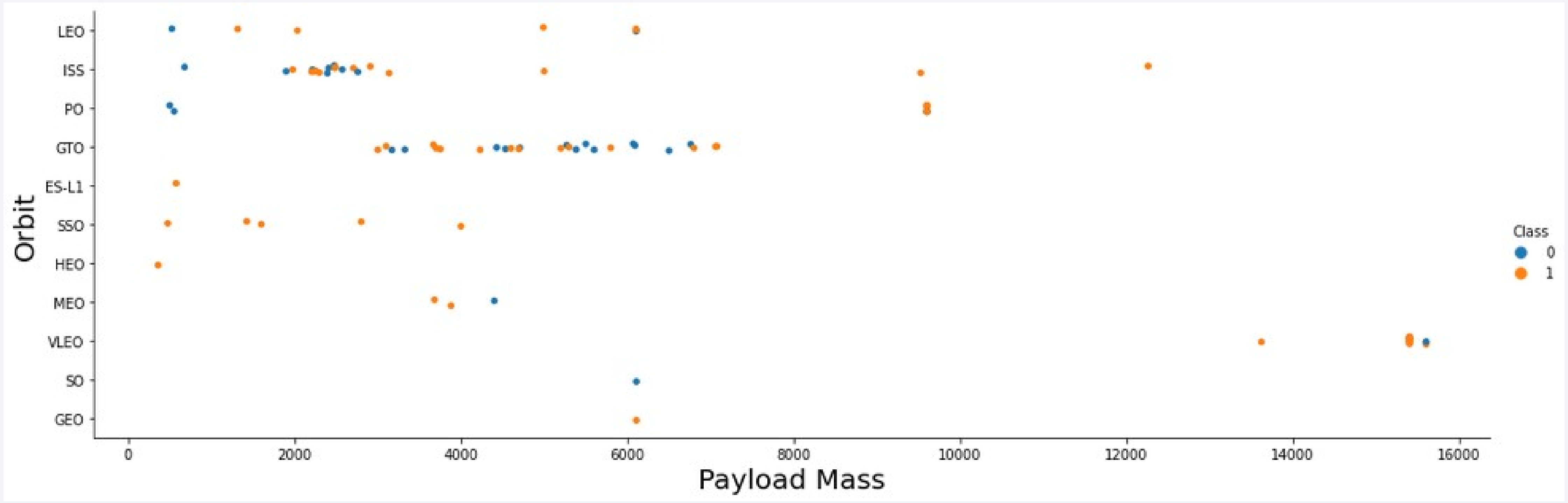
Flight Number vs. Orbit Type

- For each of the Orbits a blue dot indicates a failure and orange a success.
- From this we see that certain Orbits such as ‘VLEO’ were only attempted in later flights.



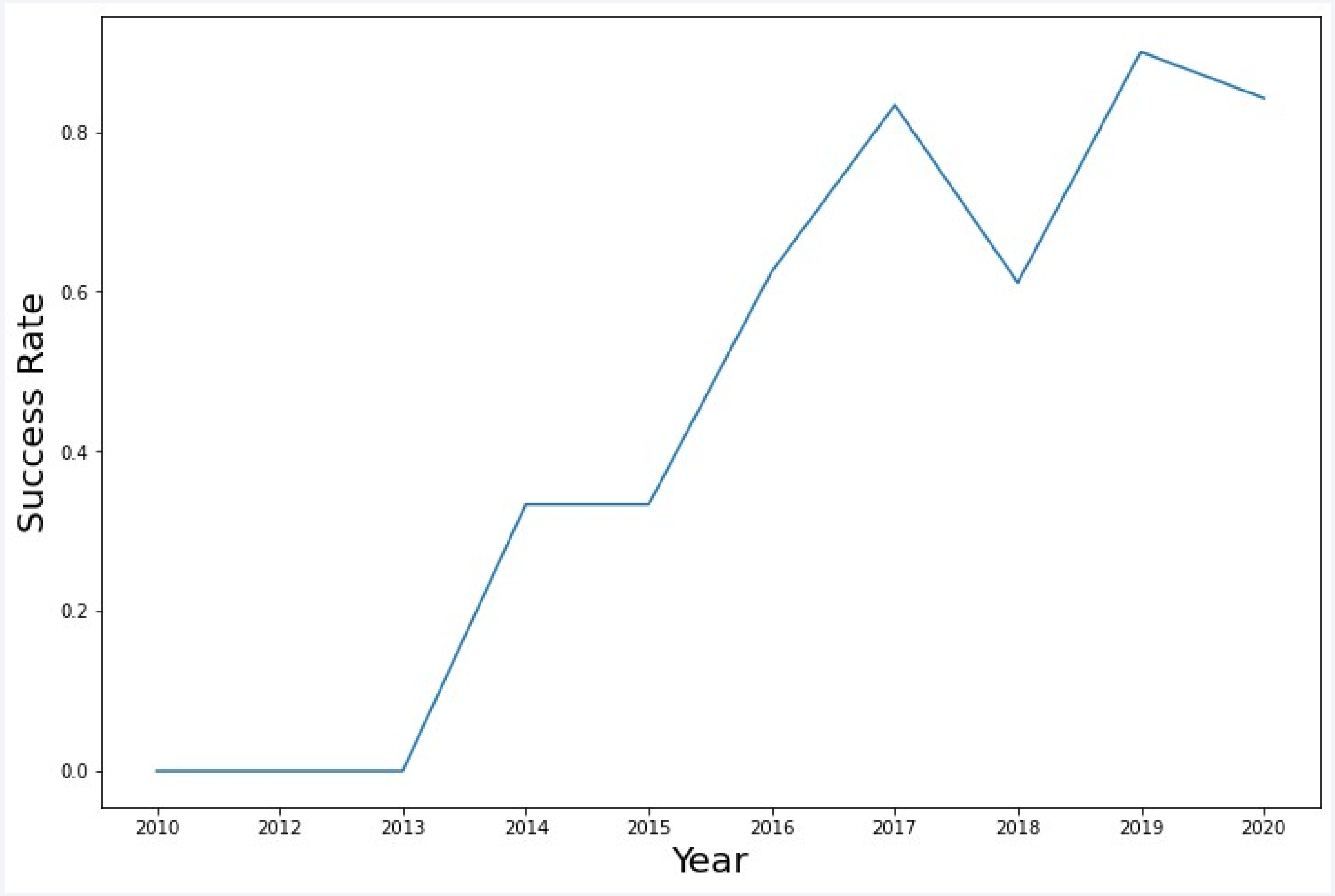
Payload vs. Orbit Type

- For each of the Orbits a blue dot indicates a failure and orange a success
- From this we see a wide variation in success rate in the ‘GTO’ Orbit with payloads all below 8,000



Launch Success Yearly Trend

This line chart indicates a gradual improvement in the Success Rate over time.



All Launch Site Names

```
df['Launch_Site'].unique()
```

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

Calling the function “unique” on the Data Frame column “Launch_Site” returns all of the unique elements within the column.

Launch Site Names Begin with 'KSC'

```
KSC = [ ]
for i in range(len(df)):
    if 'KSC' in df['Launch_Site'][i]:
        KSC.append(df['Booster_Version'][i])
KSC[0:5]
```

‘F9 FT B1031.1’, 'F9 FT B1030', 'F9 FT B1021.2', 'F9 FT B1032.1, 'F9 FT B1034'

For the length of the Data Frame, if the string ‘KSC’ is in the column ‘Launch_Site’, then the ‘Booster_Version’ on the same row is appended to a list.

The first 5 elements in the list index are then displayed.

Total Payload Mass

```
CRS = [ ]
for i in range(len(df)):
    if '(CRS)' in df['Customer'][i]:
        CRS.append(int(df['PAYLOAD_MASS__KG_'][i]))
CR = np.array(CRS)
CR.sum()
```

```
48213
```

For the length of the Data Frame, if the string ‘(CRS)’ is in the column ‘Customer’, then append the ‘PAYLOAD_MASS_KG_’ to a list.
The list is then converted to a numpy array and the function ‘sum’ is called.

Average Payload Mass by F9 v1.1

```
F = [ ]
for i in range(len(df)):
    if 'F9 v1.1' in df['Booster_Version'][i]:
        F.append(int(df['PAYLOAD_MASS__KG_'][i]))
FA = np.array(F)
FA.mean()
```

```
2534.6666666666665
```

For the length of the Data Frame, if the string ‘F9 v1.1’ is in the column ‘Booster_Version’, then append the ‘PAYLOAD_MASS_KG_’ to a list.
The list is then converted to a numpy array and the function ‘mean’ is called.

First Successful Ground Landing Date

```
D = [ ]
for i in range(len(df)):
    if 'Success (drone ship)' in df['Landing _Outcome'][i]:
        D.append(df['Date'][i])
D[0]
```

```
'08-04-2016'
```

For the length of the Data Frame, if the string ‘Success (drone ship)’ is in the column ‘Landing_Outcome’, then append the ‘Date’ to a list.
The first index in the list is then called to display the first date.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
GP = [ ]
for i in range(len(df)):
    if 'Success (ground pad)' in df['Landing _Outcome'][i]:
        GP.append([df['Booster_Version'][i], df['PAYLOAD_MASS__KG_'][i]])
for i in range(len(GP)):
    if 4000 < int(GP[i][1]) < 6000:
        print(GP[i])
```

```
['F9 FT B1032.1', 5300]
['F9 B4 B1040.1', 4990]
['F9 B4 B1043.1', 5000]
```

For the length of the Data Frame, if the string 'Success (ground pad)' is in the column 'Landing_Outcome', then append the 'Booster_Version' and 'PAYLOAD_MASS__KG_' to a list.
For the length of the list, if the 'PAYLOAD_MASS__KG_' is greater than 4,000 and less than 6,000, print the 'Booster_Version' and 'PAYLOAD_MASS__KG_'.

Total Number of Successful and Failure Mission Outcomes

```
df['Mission_Outcome'].value_counts()
```

```
- Success                              98
- Success (payload status unclear)     1
- Failure (in flight)                   1
- Success                               1
```

Calling the function “value_counts” on the Data Frame column “Mission_Outcome” returns all of the count of elements within the column.

Boosters Carried Maximum Payload

```
for i in range(len(df)):
    if df['PAYLOAD_MASS__KG_'][i] == df['PAYLOAD_MASS__KG_'].max():
        print(df['Booster_Version'][i])
```

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

For the length of the Data Frame, if the column 'PAYLOAD_MASS_KG_' is the maximum, then print the 'Booster_Version.

2017 Launch Records

```
t = [ ]
for i in range(len(df)):
    if '2017' in df['Date'][i] and 'Success (ground pad)' in df['Landing _Outcome'][i]:
        t.append([df['Date'][i], df['Booster_Version'][i], df['Launch_Site'][i]])
t
```

```
['19-02-2017', 'F9 FT B1031.1', 'KSC LC-39A'],
['01-05-2017', 'F9 FT B1032.1', 'KSC LC-39A'],
['03-06-2017', 'F9 FT B1035.1', 'KSC LC-39A'],
['14-08-2017', 'F9 B4 B1039.1', 'KSC LC-39A'],
['07-09-2017', 'F9 B4 B1040.1', 'KSC LC-39A'],
['15-12-2017', 'F9 FT  B1035.2', 'CCAFS SLC-40']
```

For the length of the Data Frame, if the string ‘2017’ is in the column ‘Date’, and the string ‘Success (ground pad)’ is in the column ‘Landing_Outcome’, then append the ‘Date, 'Booster_Version' and 'Launch_Site' to a list. Then display the list.

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

```
Z = [ ]
DF = df[:][0:30]
for i in range(len(DF)):
    if 'Success' in DF['Landing _Outcome'][i]:
        Z.append(DF['Date'][i])
print(Z)
```

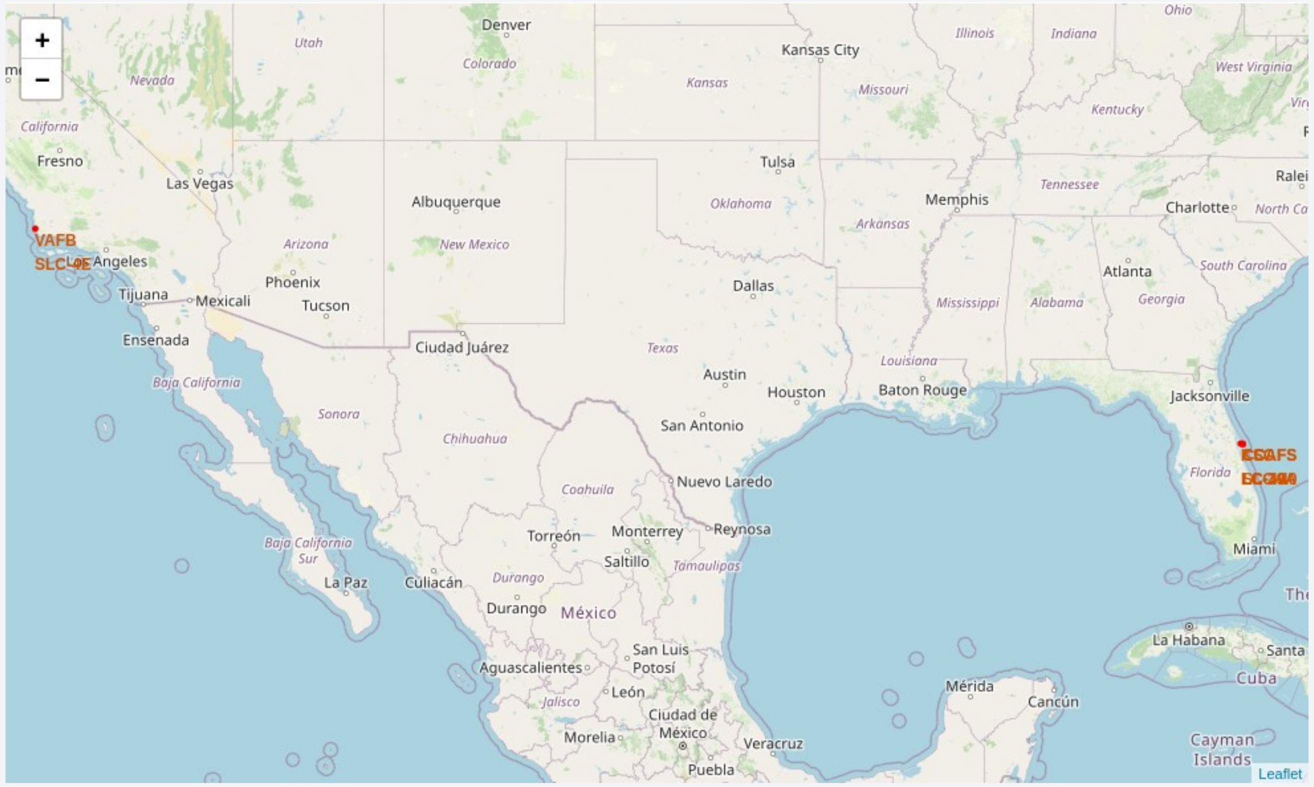
```
['22-12-2015', '08-04-2016', '06-05-2016', '27-05-2016', '18-07-2016', '14-08-2016', '14-01-
2017', '19-02-2017']
```

For the length of the Data Frame up to the index position of the end date, if the string ‘Success’ is in the column ‘Landing_Outcome’, then append the 'Date' to a list. Then print the list of dates.

Section 3

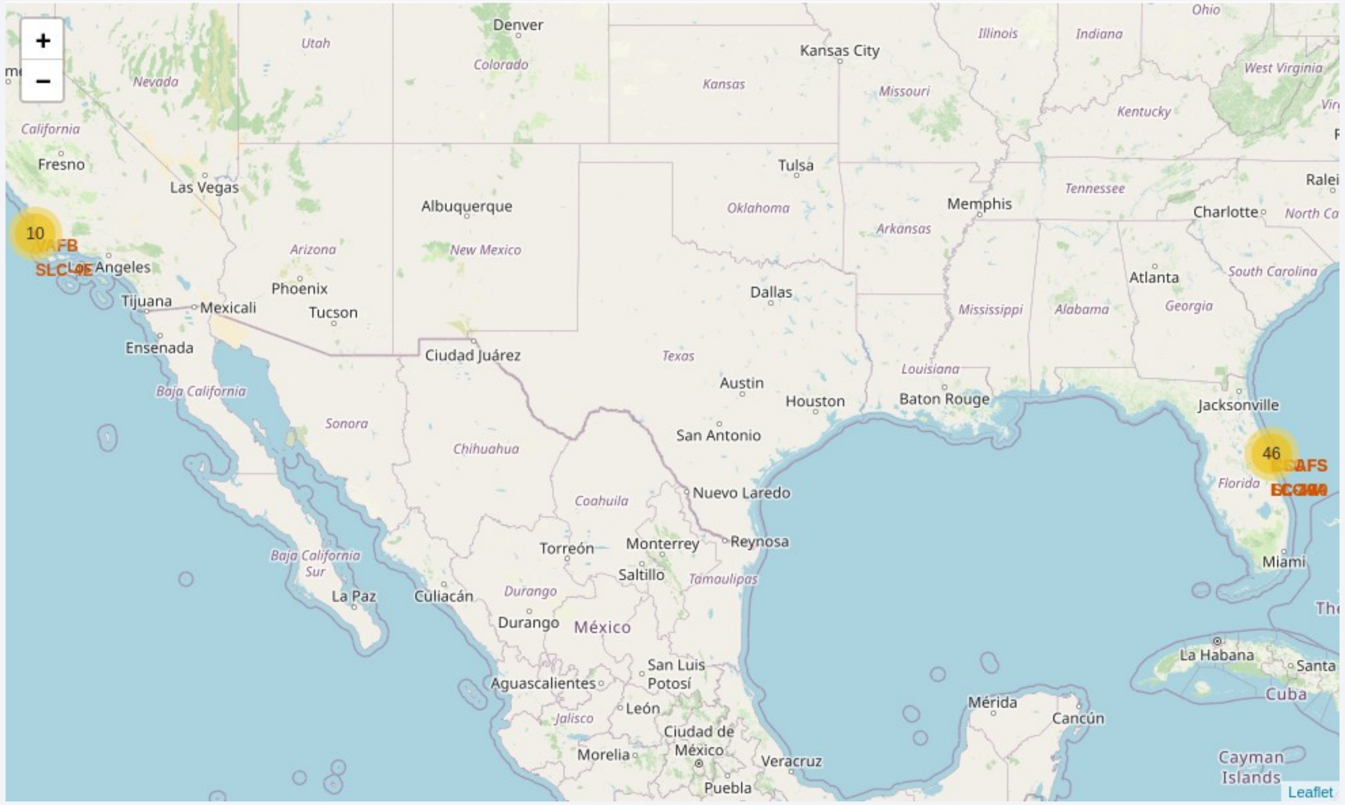
Launch Sites Proximities Analysis

Map of Launch Sites



SpaceX launches take place near the coastlines of Florida and California. Proximity to the ocean can help improve safety in the event of problems and a method of booster recovery involves floating platforms on open water.

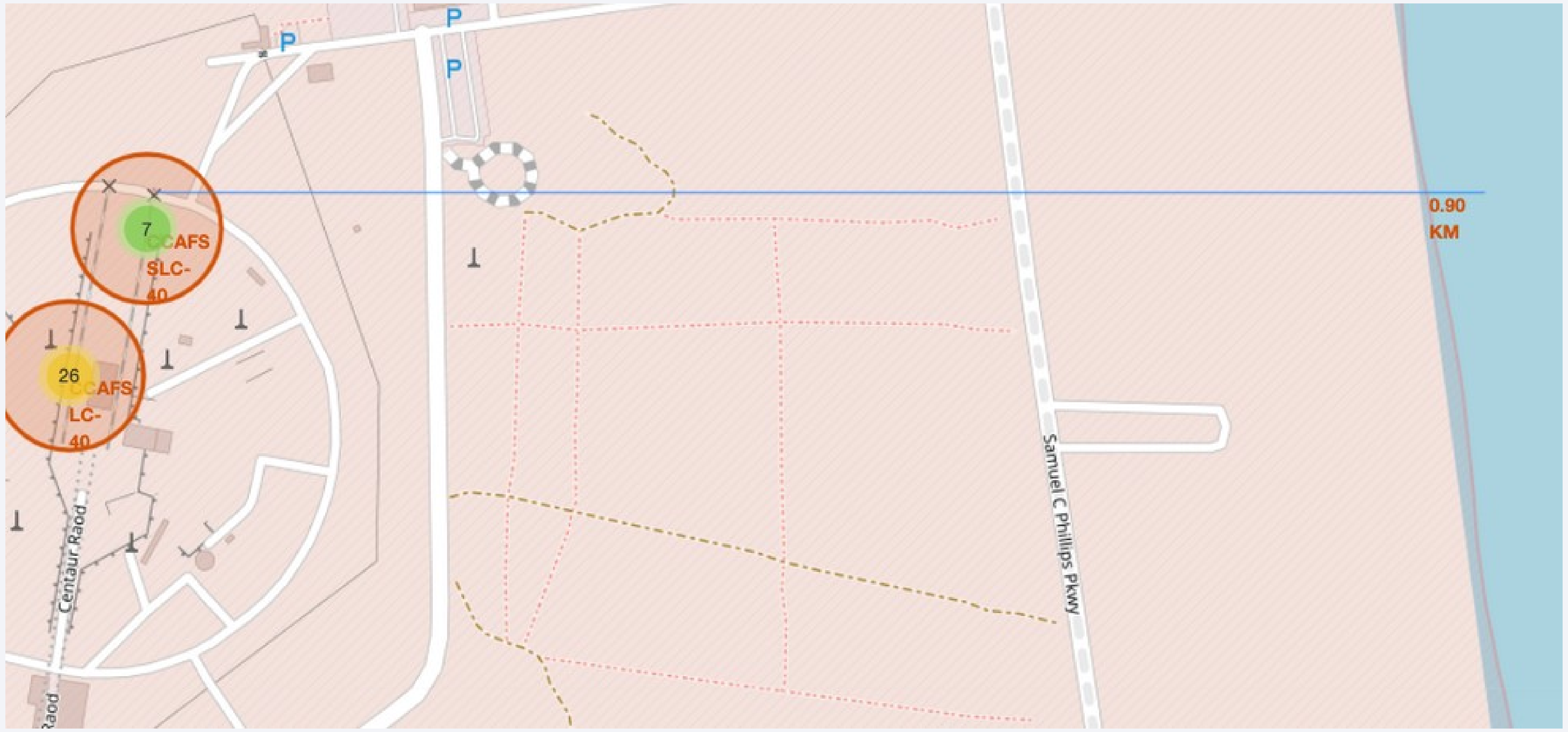
Map of Launch Outcomes



The majority of SpaceX launches took place in Florida.

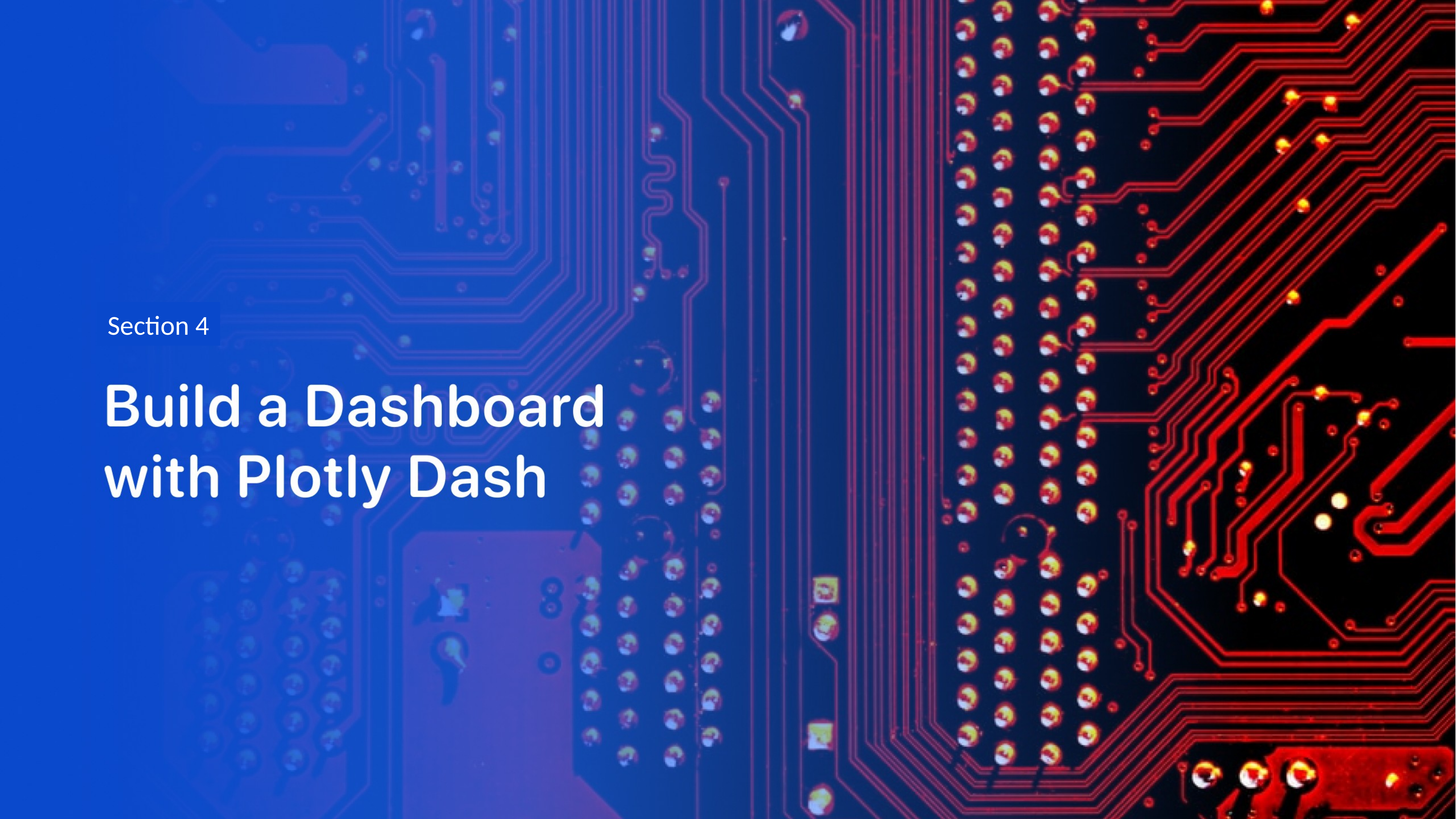
The launch site with the most launches is CAFS LC-40 with 26 which included 7 successful recovery outcomes.

Zoom of coastline proximity



The SpaceX launch sites are located in close proximity to coastlines.

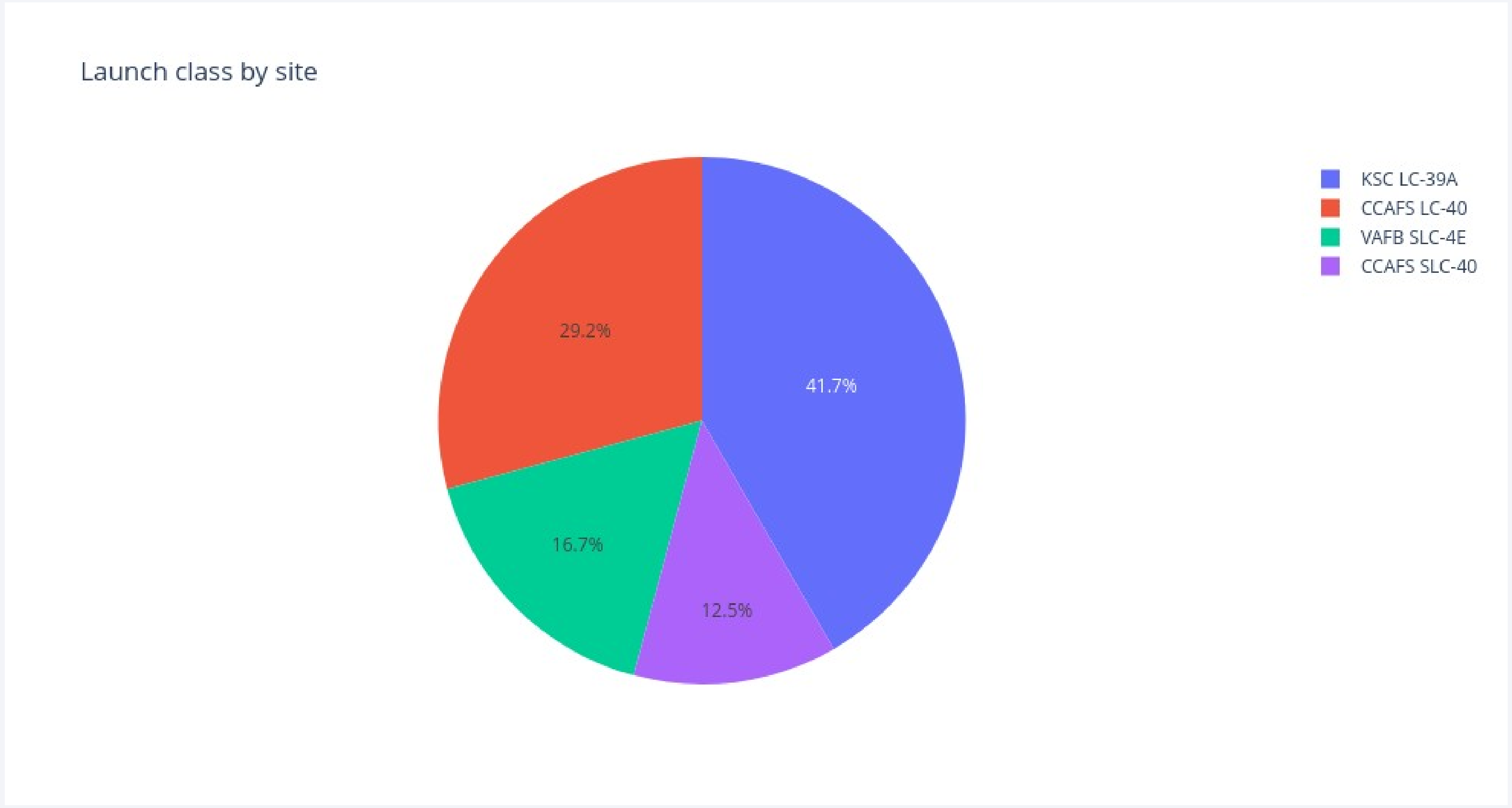
Some sites are near an airport while still at a distance from major cities.



Section 4

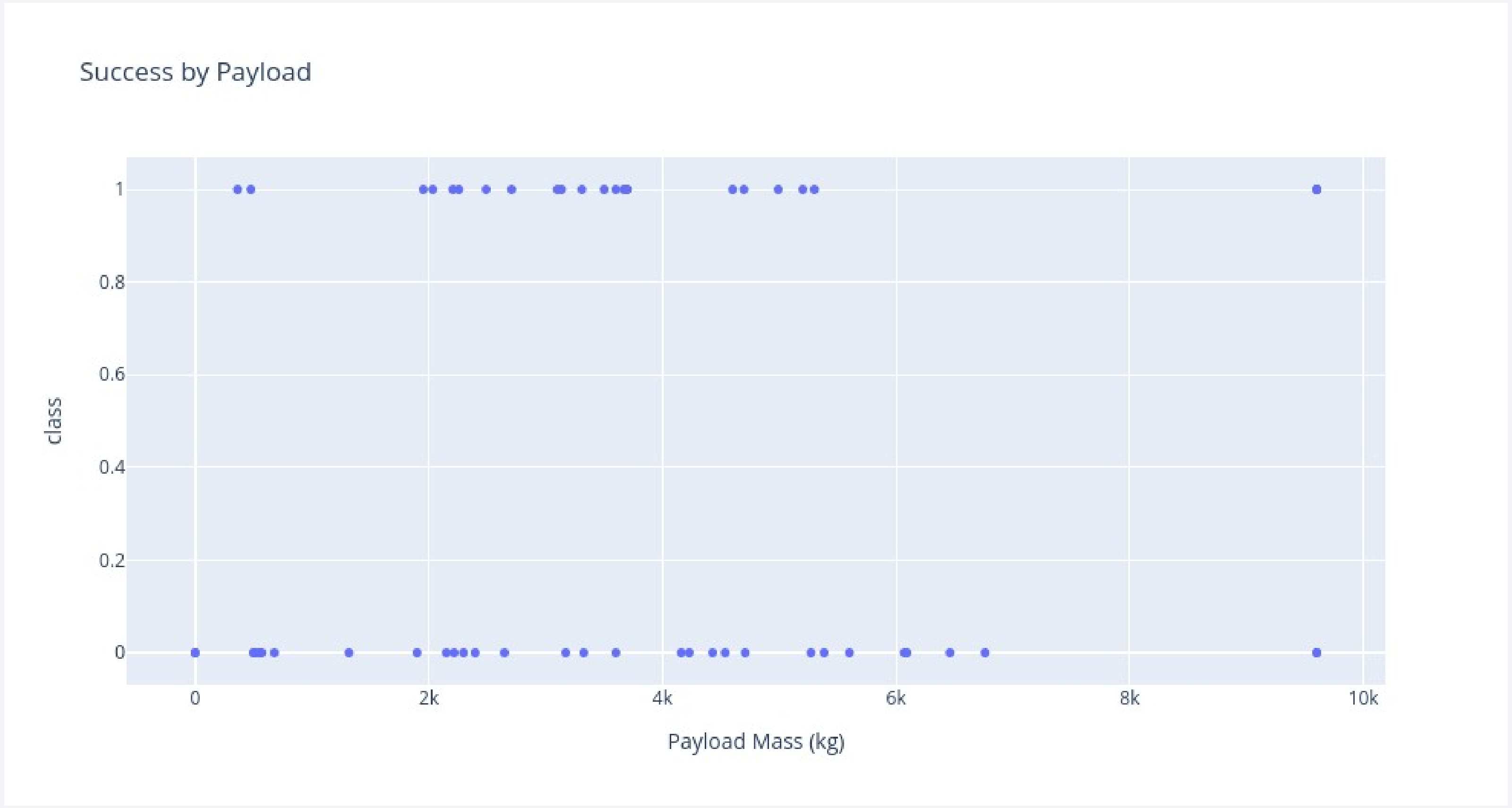
Build a Dashboard with Plotly Dash

Launch Success for All Sites



This pie chart shows us that CCAFS SLC-40 was the launch site with the most successes.

Success according to payload

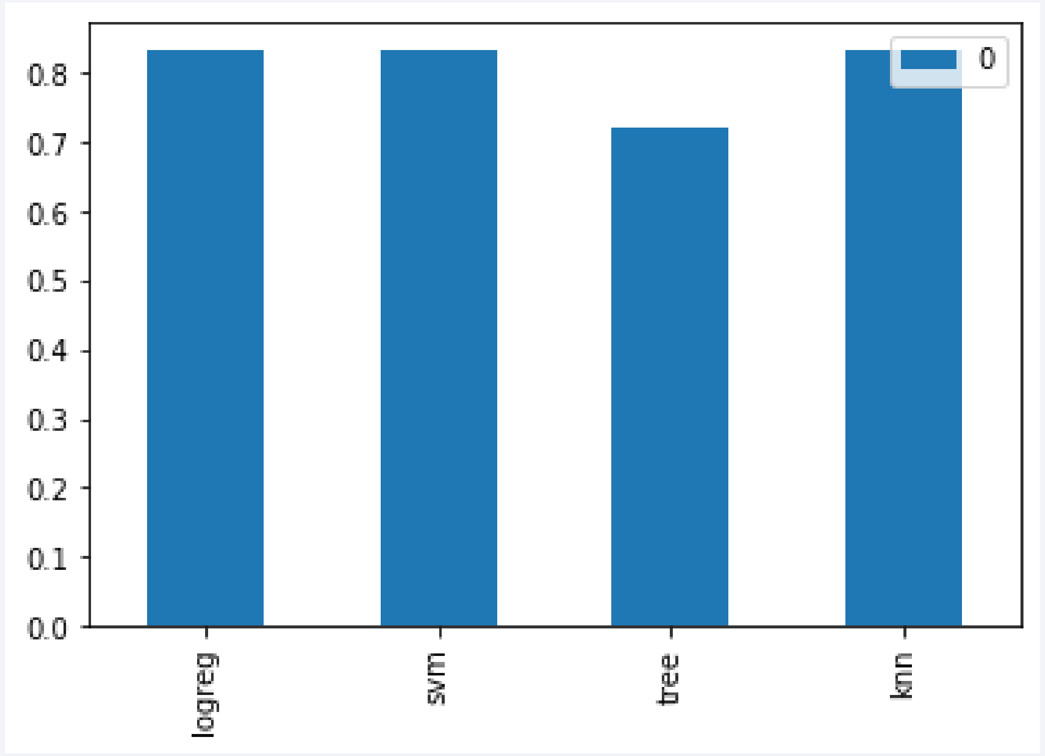


The success rate varies according to the payload carried with most successes in the range of 2k to 4k.

Section 5

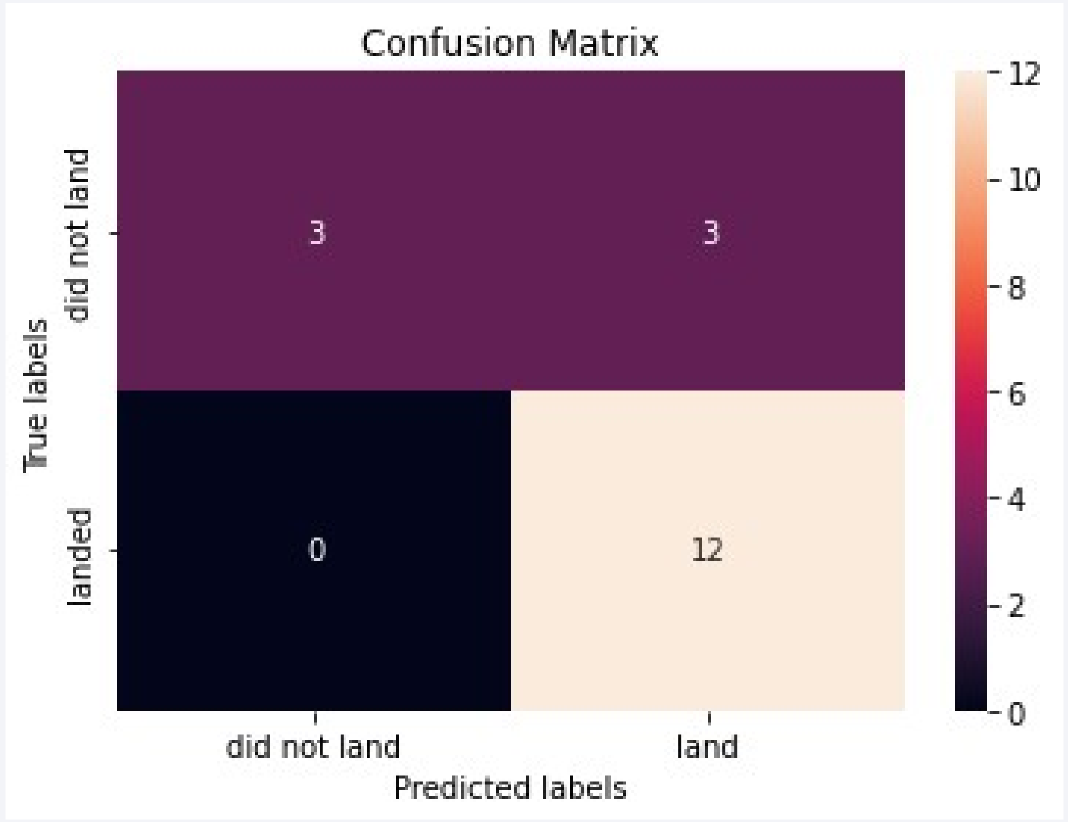
Predictive Analysis (Classification)

Classification Accuracy



The K-Nearest Neighbor model scored the best on the test data.

Confusion Matrix



The KNN model performed the best after being fitted on the training data and then scored on the test data. This confusion matrix shows the difference between the predicted and true labels.

Conclusions

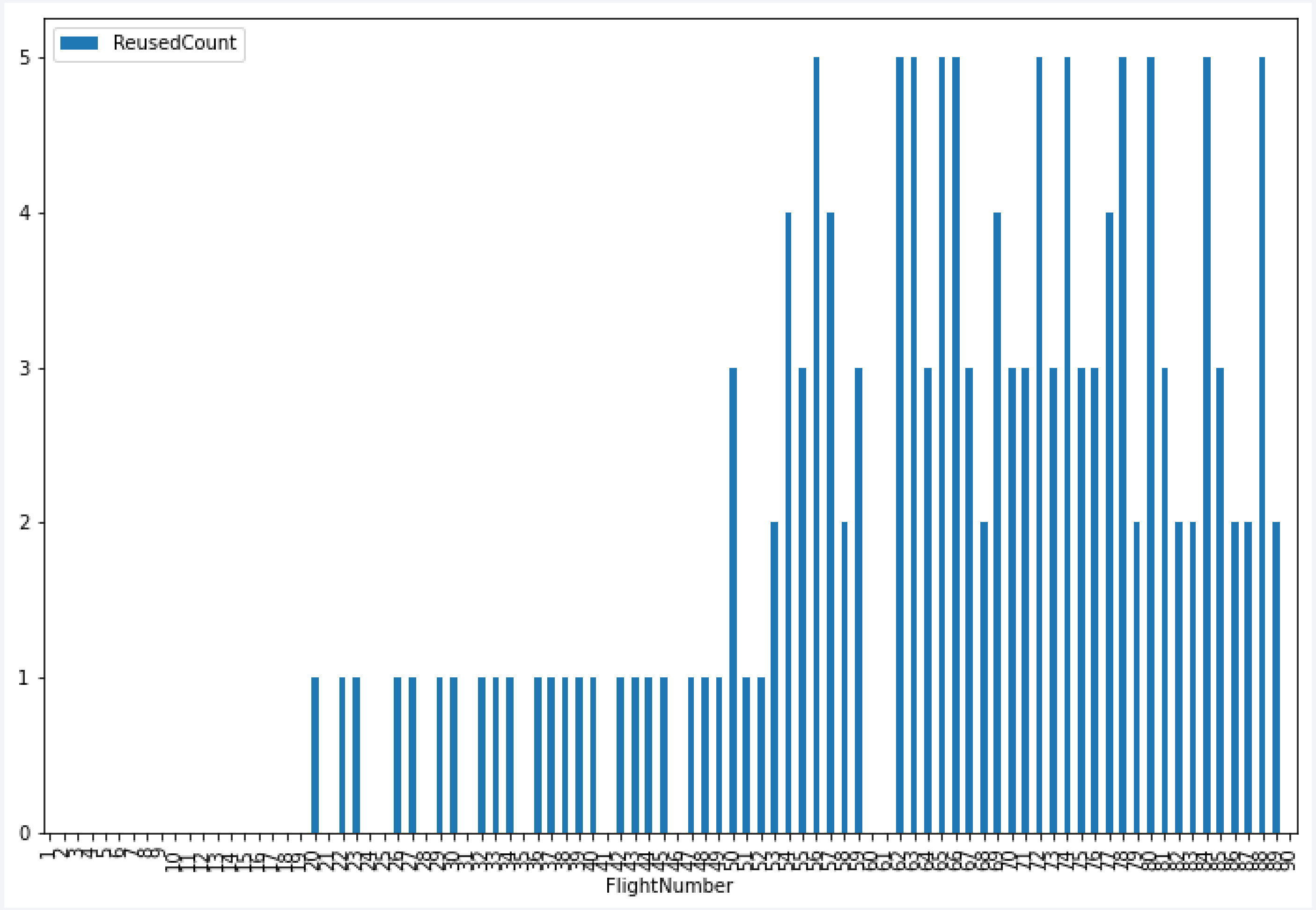
The success rate of SpaceX rocket recovery is a determining factor in the cost of launches. Overall success rates have generally improved over time.

Certain variables such as payload weight and orbit achieved can have important impacts on the success rate. Generally larger payloads and certain orbits were only attempted with later launches.

SpaceX launches take place in California and Florida in close proximity to coastlines.

KNN clustering proved to be the most effective model for predicting SpaceX data patterns.

Appendix



Thank you!