IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Cannon Johnson
8/29/2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

  Space X 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 Space X 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 space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers

  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collected was done through the SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One hot encoding for classification models
  - Clean and preprocess data for ML models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- Data Collection from SpaceX API
    - Information is called from the Space X REST API URL (api.spacexdata.com/v4/)
    - The API is read using .json and turned into a pandas dataframe using .json_normalize.
    - From here the data was cleaned and exported to use for the ML models.
- Web scraping from Wikipedia
    - Information is found at https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
    - Perform Request and Get HTML from Wikipedia.
    - Extract data with Beautiful Soup
    - Create a pandas data frame
    - Extract and load data for ML models

Data Collection – SpaceX API

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

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

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

```
[ ] launch_dict = {'FlightNumber': list(data['flight_number']),
    'Date': list(data['date']),
    'BoosterVersion':BoosterVersion,
    'PayloadMass':PayloadMass,
    'Orbit':Orbit,
    'LaunchSite':LaunchSite,
    'Outcome':Outcome,
    'Flights':Flights,
    'GridFins':GridFins,
    'Reused':Reused,
    'Legs':Legs,
    'LandingPad':LandingPad,
    'Block':Block,
    'ReusedCount':ReusedCount,
    'Serial':Serial,
    'Longitude': Longitude,
    'Latitude': Latitude}
```

```
[ ] # Create a data from launch_dict
    launch_df = pd.DataFrame([launch_dict])
```

- [Link to Notebook]()

Data Collection - Scraping

[Link to Code]

```
html_data = requests.get(static_url)
html_data.status_code
```

```
soup = BeautifulSoup(html_data.text, 'html.parser')
```

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

Data Wrangling

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv")
df.head(10)
```

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

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []

for key, value in df['Outcome'].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

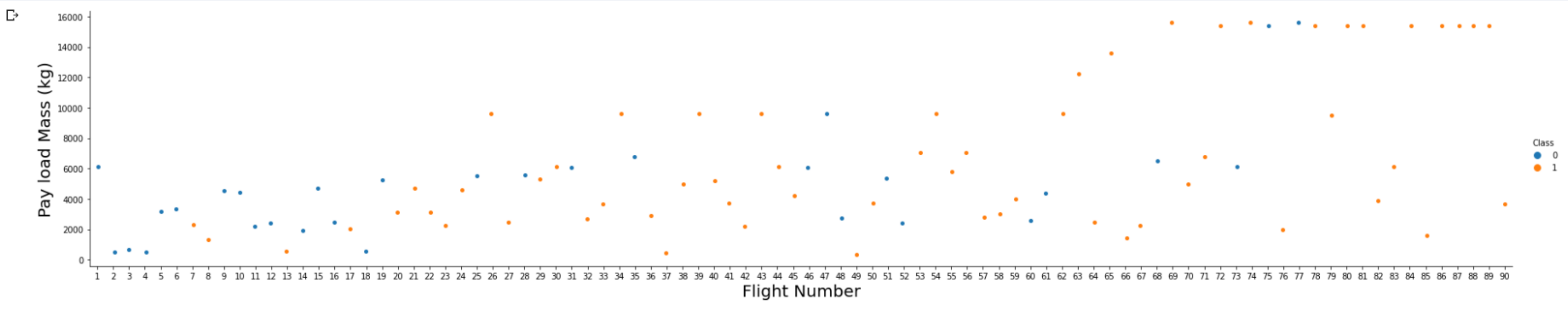
```
df['Class']=landing_class
df[['Class']].head(8)
```

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

```
0.6666666666666666
```

EDA with Data Visualization

- We began the lab analyzing pay load mass vs flight number
- This lead the basis for exploration of other relationships between features
  - These relationships are shown later in the presentation and included in the final code commit



EDA with SQL

- SQL Queries performed:
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch site begins with the strong ‘CCA’
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mas carried by booster version F9 V1.1
  - List the data when the first successful landing outcome in ground pad was achieved
  - List the names of the boosters which have success in drone ship 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 carries the maximum payload mass. Use a subquery
  - List the records which display the months names, failure landing outcomes in drone ship, booster versions, launch site for the months in year 2015
  - Rank the count of successful landing outcomes between the data 04-06-2010 and 20-03-2017 in descending order
- Link to code

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- Link to Code
- George,2022

Build a Dashboard with Plotly Dash

- Dashboard has dropdown, pie chart, range slider and scatter plot components
  - Dropdown allows a user to choose the launch site or all sites (dash core components)
  - Pie chart shows the total success and total failure for the launch site chose with the dropdown component (Plotly express)
  - Range slider allows a user to select a payload mass in a fixed range
  - Scatter chart shows relationship between two variables, in particular success vs payload mass
- Link to Code

Predictive Analysis (Classification)

- After the EDA and data wrangling process we used sklearn to split and train our data
- From this point we built four models with different parameters using GridSearchCV
  - Logistic Regression
  - K- Nearest Neighbors
  - Decision Tree
  - Support vector machine (SVM)

Results

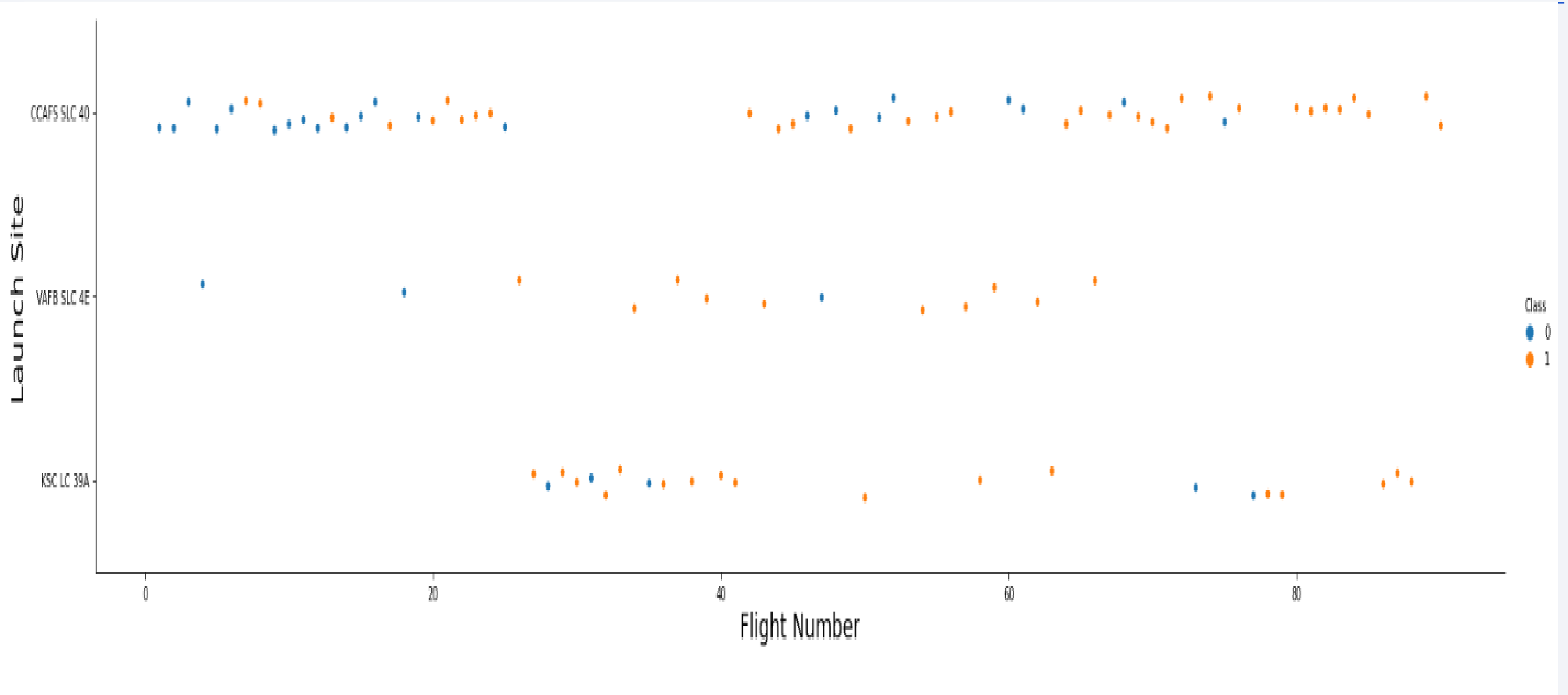
- The model with the highest accuracy is the decision tree.

```
[ ]  models = {'KNeighbors':knn_cv.best_score_,
               'DecisionTree':dtc_cv.best_score_,
               'LogisticRegression':logreg_cv.best_score_,
               'SupportVector': svm_cv.best_score_}

     bestalgorithm = max(models, key=models.get)
     print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])

     Best model is DecisionTree with a score of 0.875
```

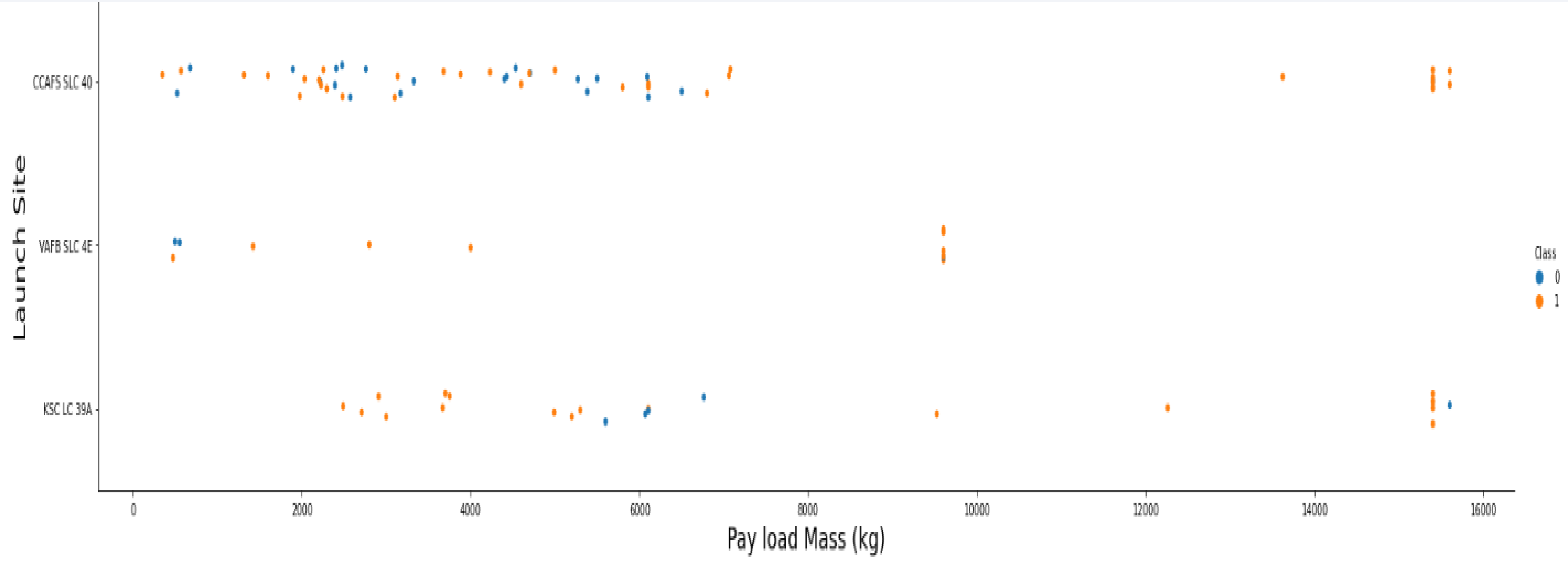
Section 2

Insights drawn from EDA

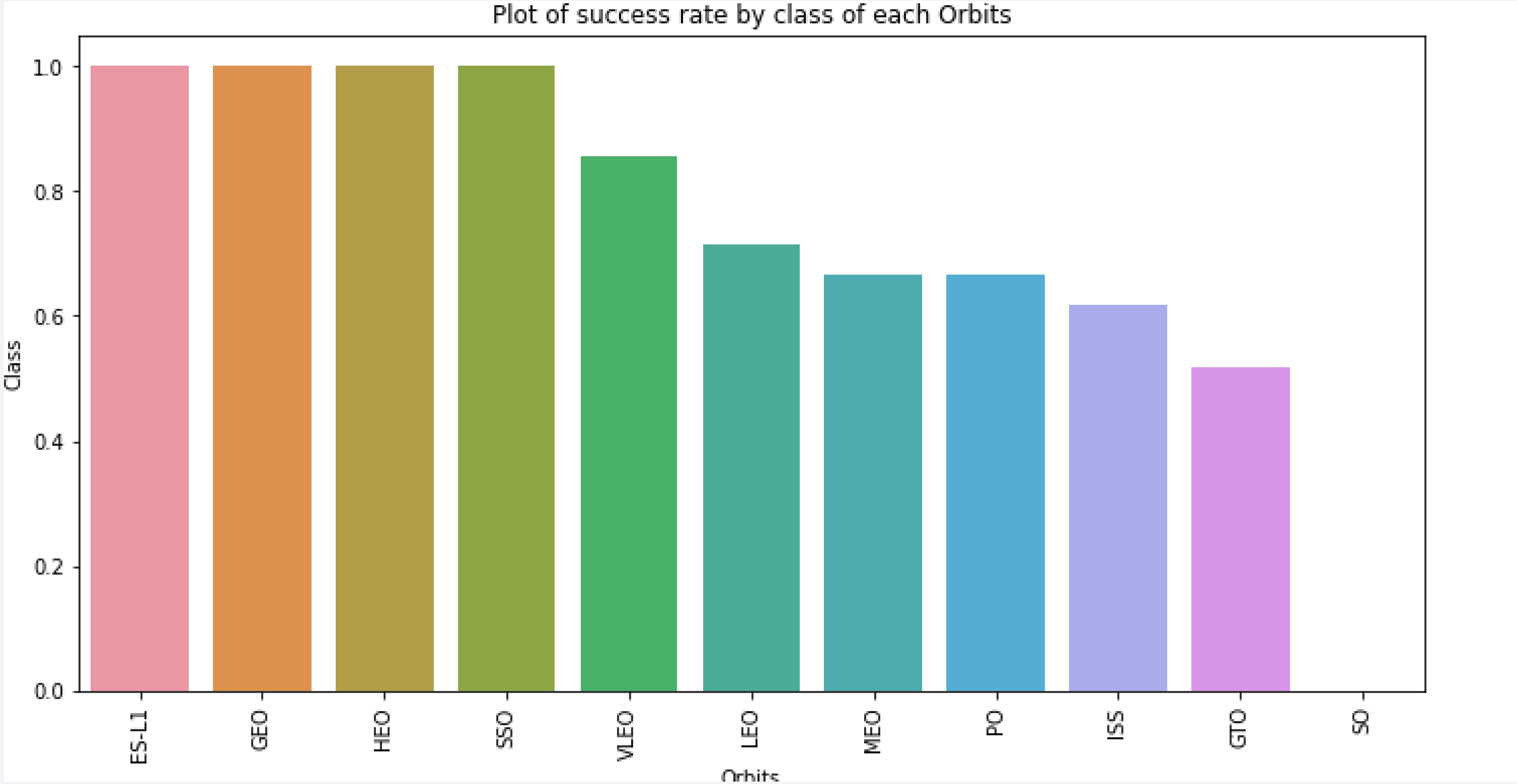
Flight Number vs. Launch Site



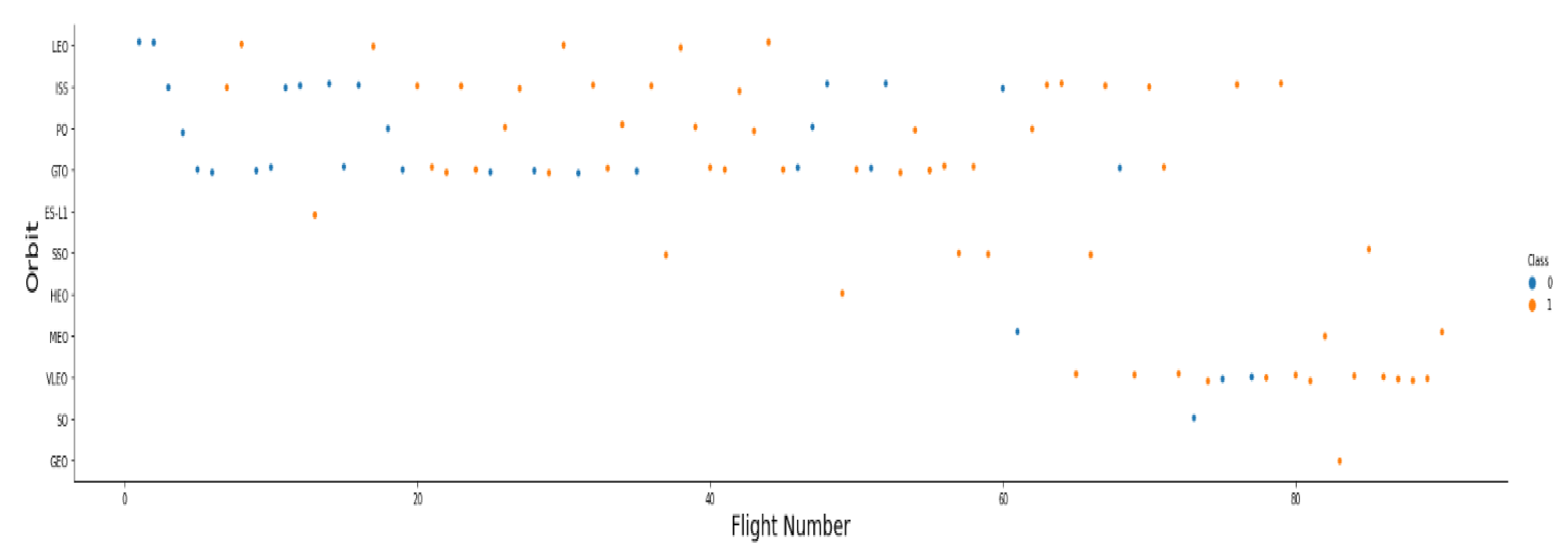
Payload vs. Launch Site



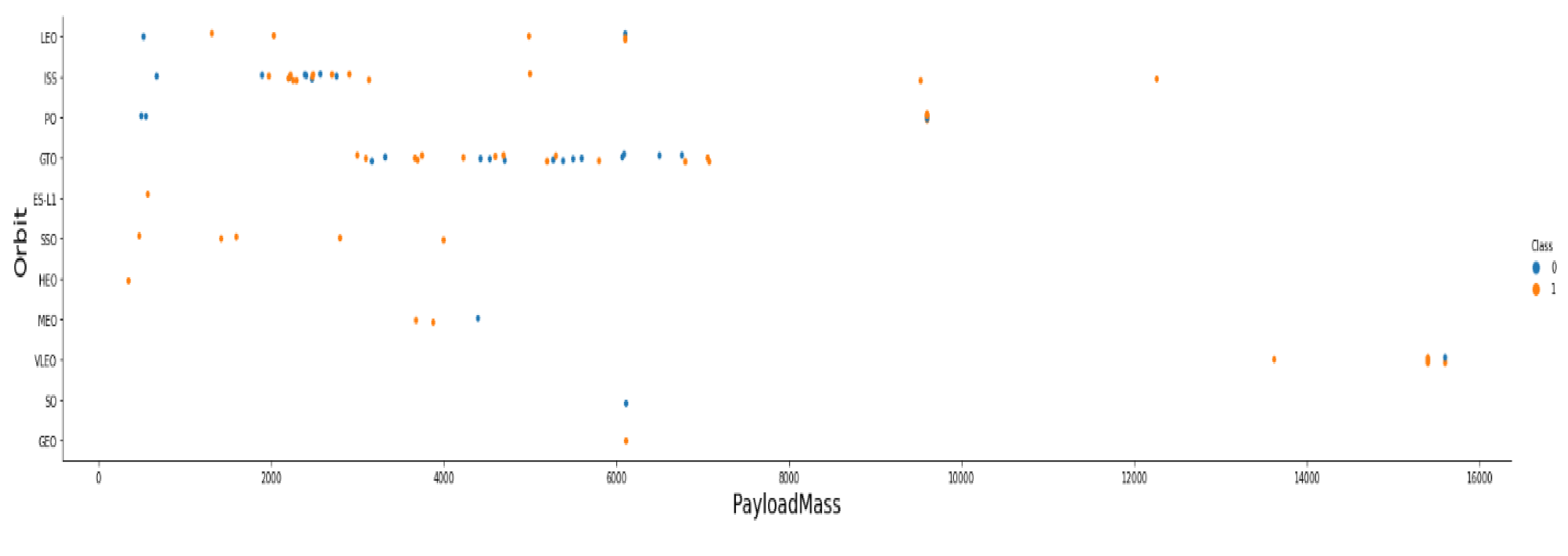
Success Rate vs. Orbit Type



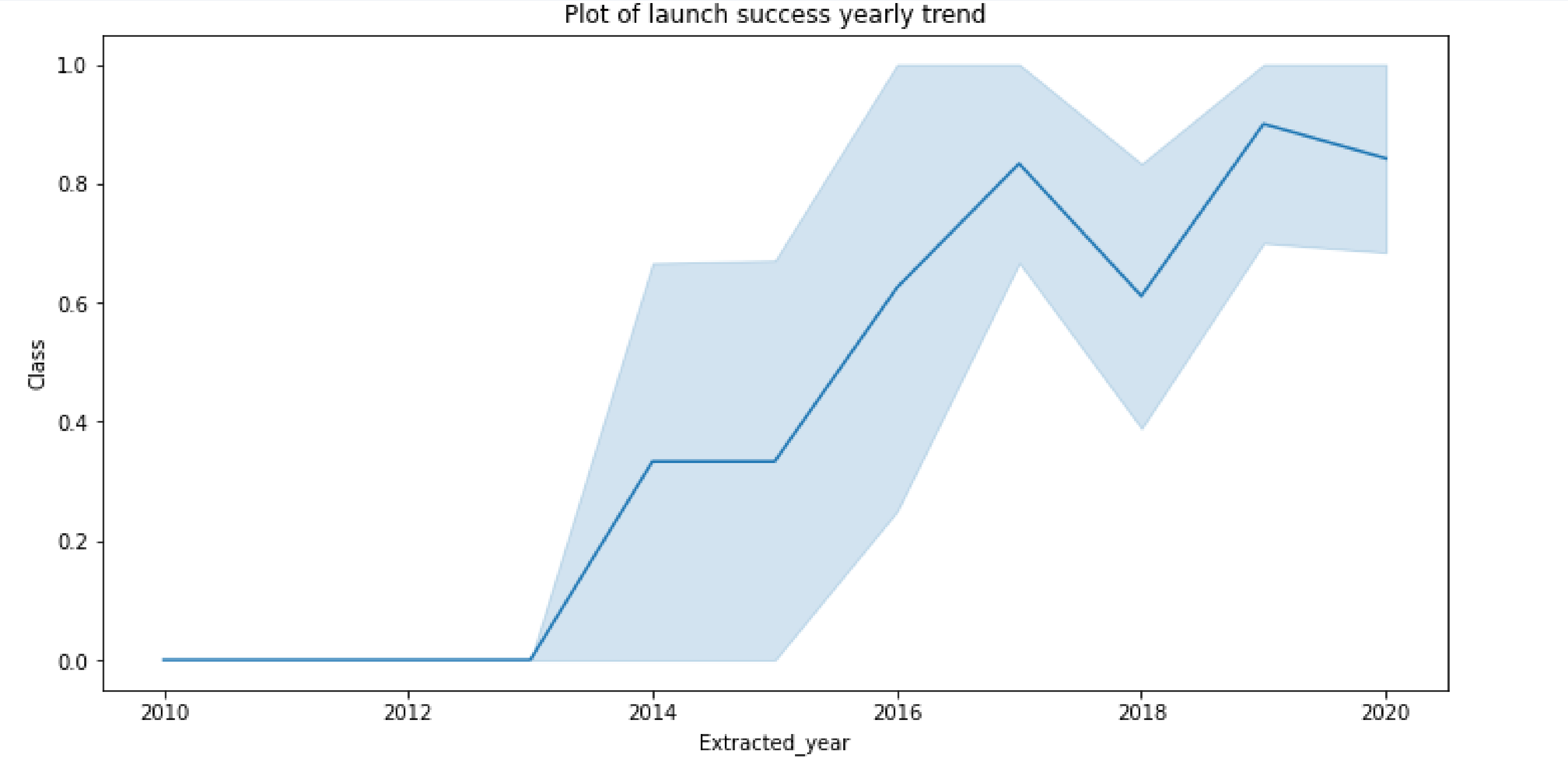
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

```
%sql SELECT DISTINCT launch_site FROM SPACEXTBL
```

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

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTBL WHERE launch_site LIKE 'CCA%' LIMIT 5;
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer LIKE '%NASA (CRS)%';
```

SUM(PAYLOAD_MASS__KG_)
48213

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1';
```

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

```
%sql SELECT min(Date) as Date , Mission_Outcome FROM SPACEXTBL WHERE Mission_Outcome ='Success';
```

Date	Mission_Outcome
01-03-2013	Success

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT Mission_Outcome, COUNT(Mission_Outcome) FROM SPACEXTBL GROUP BY Mission_Outcome;
```

Mission_Outcome	COUNT(Mission_Outcome)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

Booster_Version
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

2015 Launch Records

```
%sql SELECT substr(Date, 4, 2) as Month, booster_version, launch_site FROM
 (SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Failure%' and substr(Date,7,4)='2015') GROUP BY substr(Date,4,2);
```

Month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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

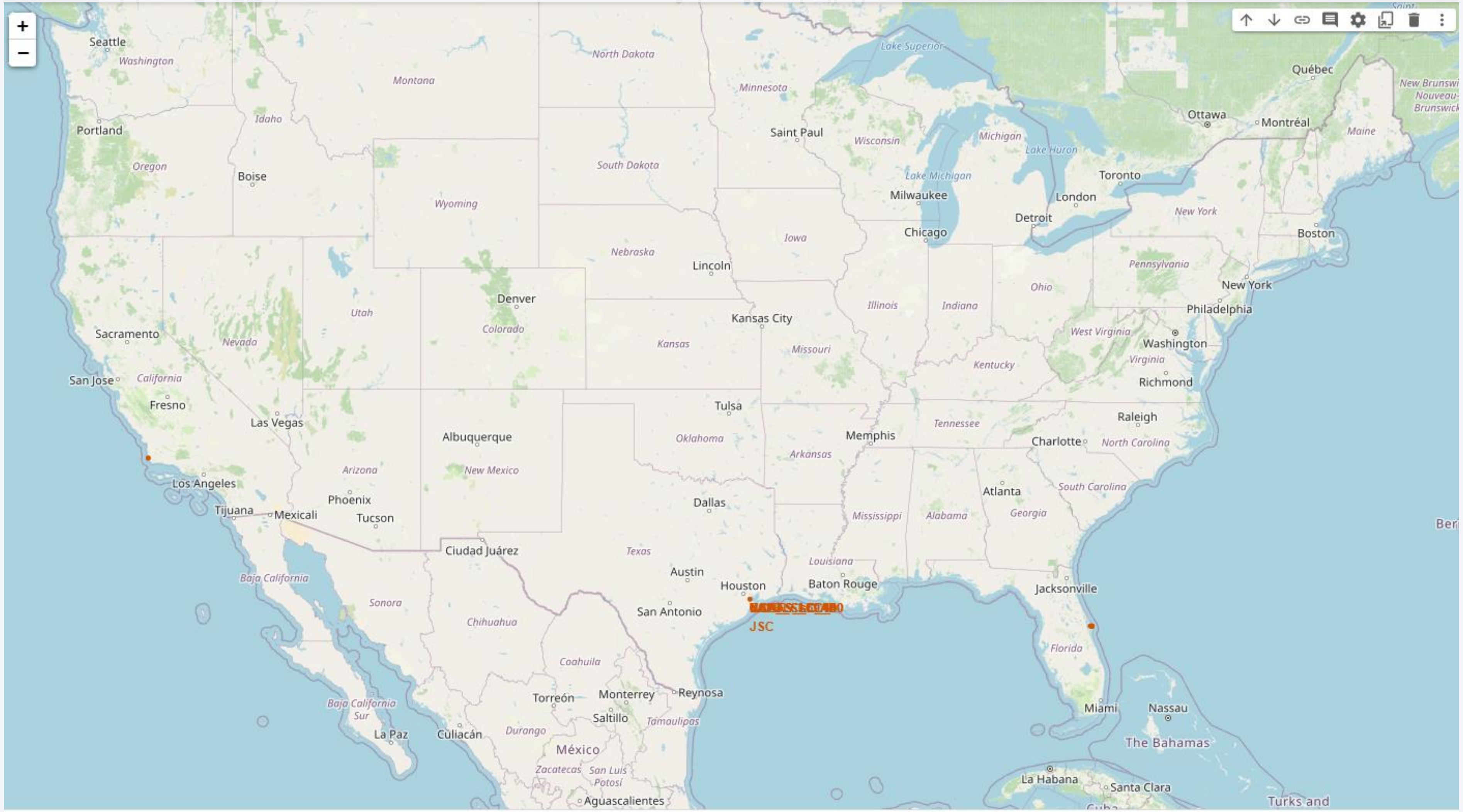
```
%sql SELECT "Landing _Outcome", COUNT(*) AS COUNT FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND DATE BETWEEN '04-06-2010' AND '20-03-17'
GROUP BY "Landing _Outcome" ORDER BY COUNT DESC;
```

Landing _Outcome	COUNT
Success	20
Success (drone ship)	8
Success (ground pad)	6

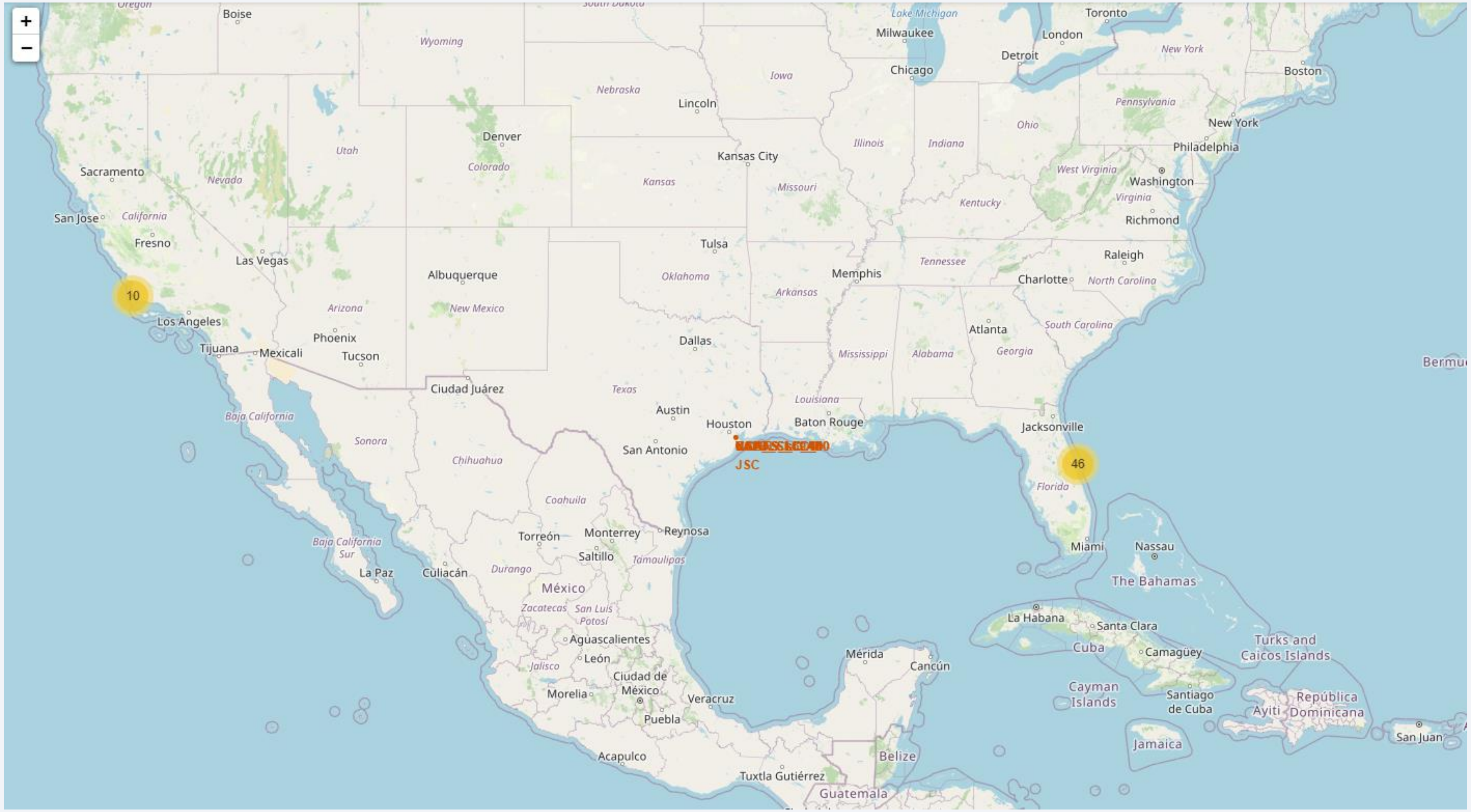
Section 3

Launch Sites Proximities Analysis

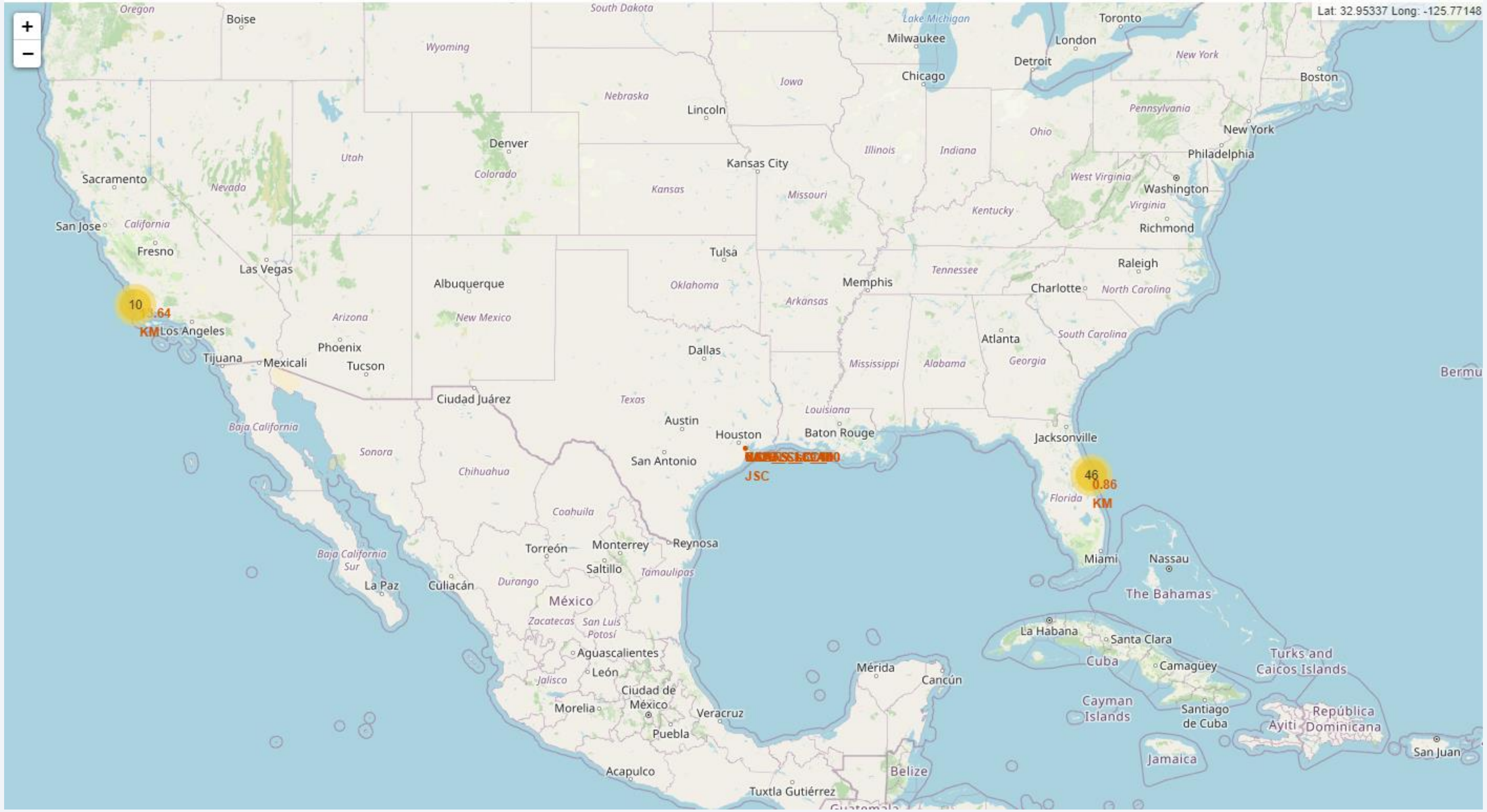
Folium Map with Launch Site markers



Folium Map with success/failure markers



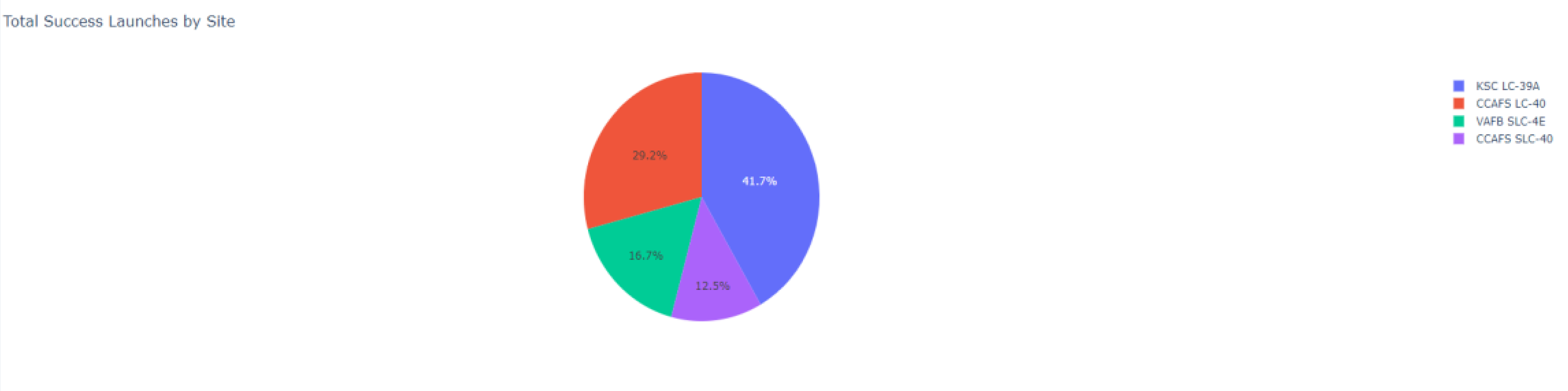
Folium Map with distance markers



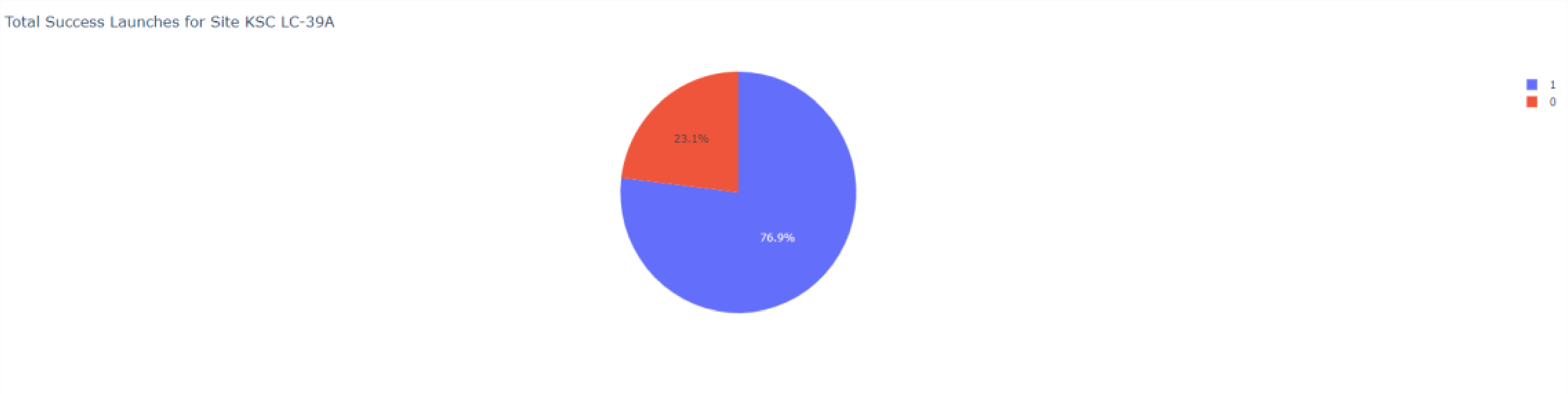
Section 4

Build a Dashboard with Plotly Dash

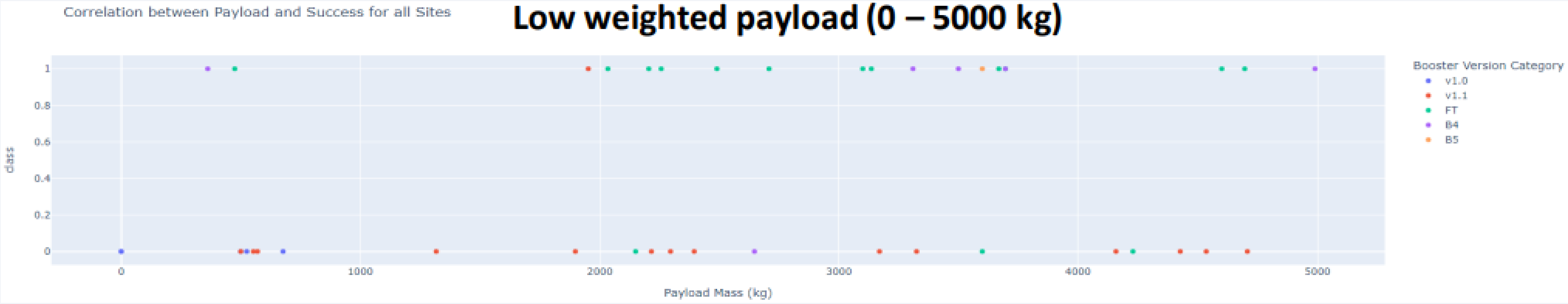
Total success by site

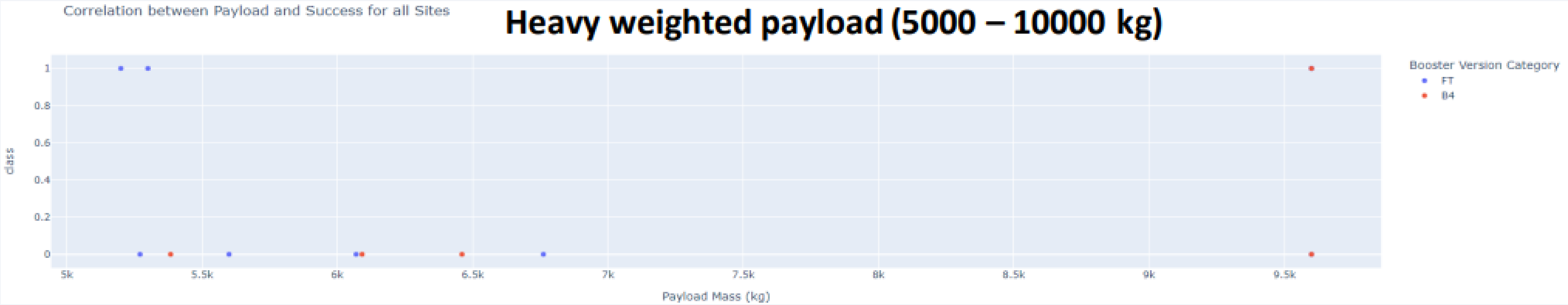


Total success launches for site KSC LC- 39A



Payload Mass KG vs Launch Outcome

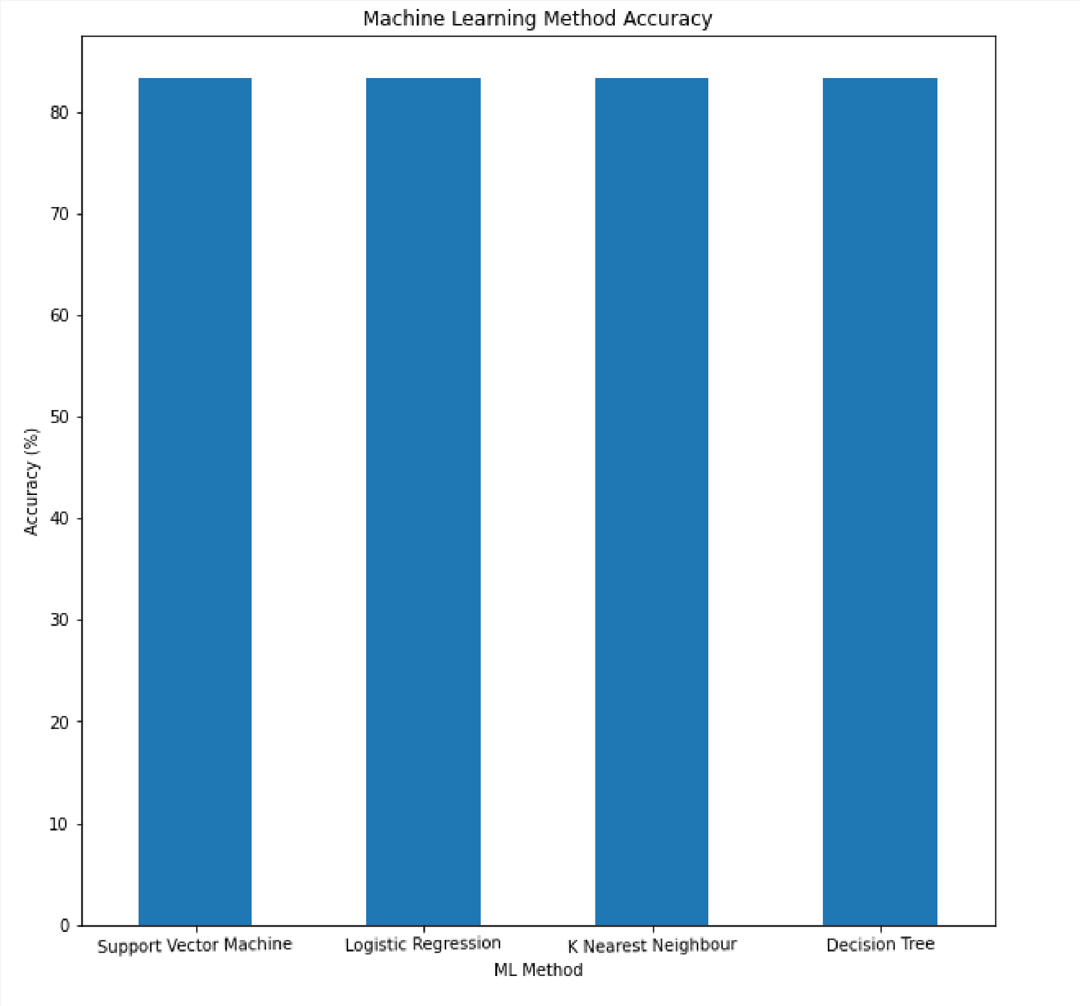




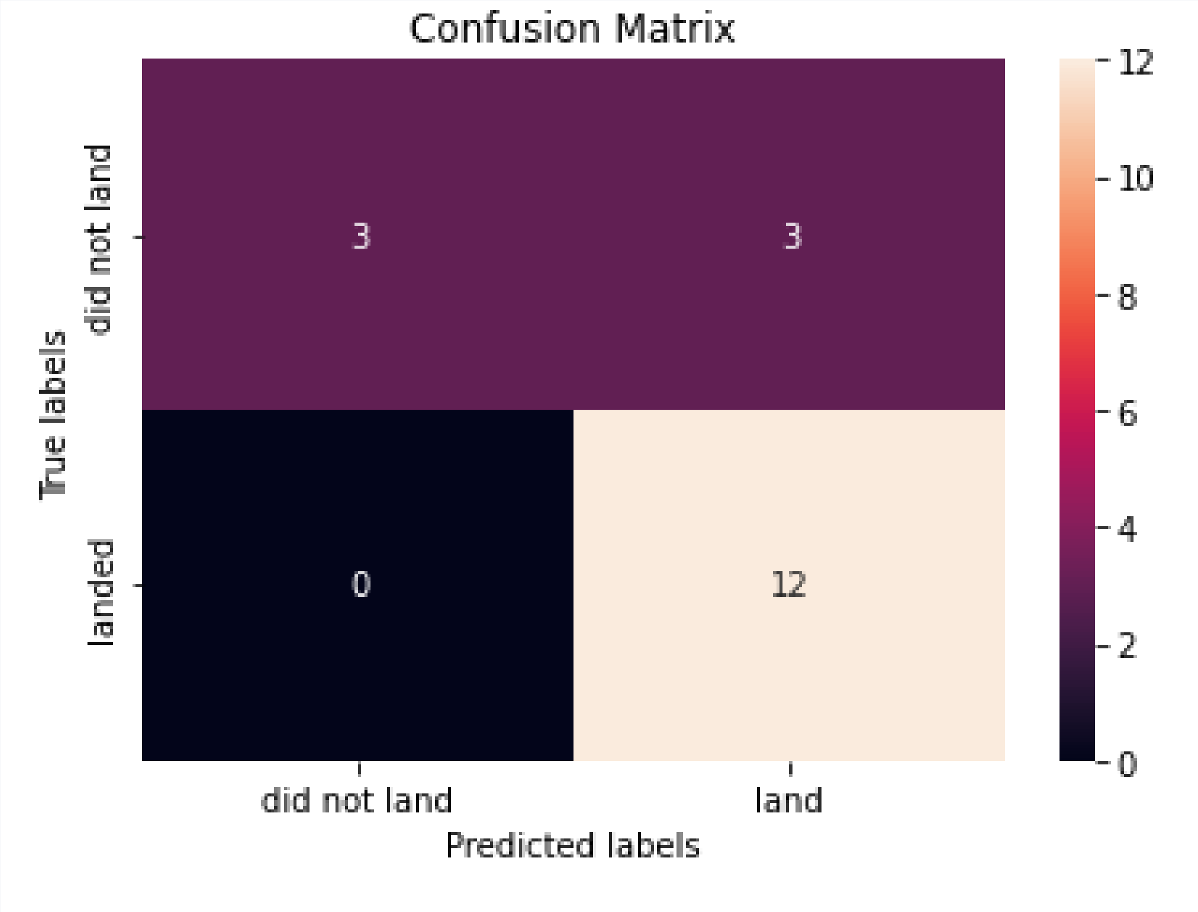
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Conclusions

- From the data we can see a large factor of mission success comes from launch site, the orbit, and the payload mass
- From our EDA process we were able to locate the launch sites with the highest probability of success
- Attempting a total of 4 ML models has resulted in similar accuracy levels. The highest and most accurate model being the decision tree algorithm
- With an accuracy score of 90.35% from testing the decision tree model could begin deployment for SpaceX

Thank you!