IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Fuat Can Akgün
18 March 2022

Outline

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

Executive Summary

- SpaceY rocket launch cost analysis is done with 4 methods, each model's accuracy and advantages are shown in notebooks.
- Models used in the project are as follows:
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree
  - K-nearest Neighbors
- Data collected from SpaceX ('rival comp.') API and webpages (web scraping)
- Collected data is processed using popular python libraries such as pandas
- Insights about data are drawn using different approaches using SQL, Python

Introduction

- Project is about a hypothetical company that need to compete with other companies in the business and our purpose is to estimate the cost of our company
- The sources used in the project are gathered from the course flow and edited by the author
- The main purpose of the project is to mimic a data scientist to gain experience on a project basis
- Parts of the projects are designed to touch every subject of the field
- As a capstone project, we analyzed the data from various points of view
- The result was experience with the most popular methods and tools of data science

Section 1

Methodology

Methodology

Executive Summary

- Data collection:
  - Web scraping / crawling
  - SpaceX API
- Perform data wrangling
  - Label determination for training 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
  - Four different models are used

Data Collection

- Data was gathered from two open sources:
  - SpaceX API
  - Wikipedia
- Python library Beautiful Soup was used to get data from Wikipedia page, List of Falcon 9 and Falcon Heavy Launches
- For API, requests library is used to simply gather data in data frame format

Data Collection – SpaceX API

- Requested and parsed data from API using requests.get

- Filtered data to contain only Falcon 9 related information

- Replaced empty values with the mean of the column

- For more detailed version:

https://github.com/fuatcanakgun/capstone-project/blob/main/api_data_collection.ipynb

Data Collection - Scraping

- Requested data from Wikipedia page mentioned (containing launch list) using requests.get
- Extracted data from HTML Table element
- Parsed data to display in data frame format using pandas.json_normalize
- For more detailed version:

https://github.com/fuatcanakgun/capstone-project/blob/main/scraping_data_collection.ipynb

Data Wrangling

- Data was processed by calculating the number of launches on each launch site, orbit and mission outcome
- Custom created CLASS column is used to store the outcome of each launch
- Used 0 and 1 as Fail and Success of the mission outcome
- Failed missions include: not attempted, unable to be attempted, ship landing failure, ocean landing failure, pad landing failure
- Succeed missions include: ship landing successful, pad landing successful, ocean landing succesful
- For more detailed version:

https://github.com/fuatcanakgun/capstone-project/blob/main/data_wrangling.ipynb

EDA with Data Visualization

- Using pandas library, first created a data frame object to begin analyzing the data
- Two important and powerful visualization libraries were used to exploit insights about our dataset: matplotlib and seaborn
  - Flight number vs [Payload, Site, Orbit]
  - Payload vs [Orbit, Site]
  - Success rate vs [Year, Orbit]

    are all of the graphs and figures that are drawn during this part of visualization
- For more detailed version:

https://github.com/fuatcanakgun/capstone-project/blob/main/exploratory_data_analysis_jupyter.ipynb

EDA with SQL

- Using IBM DB2 data loaded as a csv file into the database and formed tables
- By the help of sqlalchemy library, gathering SQL queries within python notebook became possible (see notebook URL for more)
- Some ran SQL queries in order to get tables containing information about the data are as following
  - Launch Sites
  - Payload Masses
  - Booster Versions
  - Mission Outcomes
- For more detailed version:

https://github.com/fuatcanakgun/capstone-project/blob/main/exploratory_data_analysis_sql.ipynb

Build an Interactive Map with Folium

- Using Python library Folium locational visualization was performed
- All launched sites were marked on the interactive map
- The outcome information was integrated to the map as pins
- The distances between launch sites and its proximities were also calculated as detailed information
  - Railways
  - Highways
  - Coastlines
  - Cities
- For more detailed version:

https://github.com/fuatcanakgun/capstone-project/blob/main/launchsite_analysis_folium.ipynb

Build a Dashboard with Plotly Dash

- Plotly Dash is a framework to create browser-based applications and its functionality is very impressive
- As a first part using a dropdown menu, one can filter launch sites and see related (interactively) pie charts which show success rates
- In the second part of the app, scatter figures are shown to display, again the success rates but with an option to limit the data by payload mass
- The scatter plot also creates a 3rd dimension with colors to identify different booster version categories
- For more detailed version

https://github.com/fuatcanakgun/capstone-project/blob/main/dashboard.py

Predictive Analysis (Classification)

- The libraries used in this part of the project are: Pandas, NumPy, Matplotlib, Seaborn, Sklearn
- Using collected data as a data frame, standardization of data is done
- To validate our accuracy, data is plit into two parts: test & train
- As mentioned, logistic regression, support vector machine, decision tree classifier, k-nearest neighbors classifier methods are used, and four models are compared
- For more detailed version

https://github.com/fuatcanakgun/capstone-project/blob/main/machine_learning_prediction.ipynb

Results

WHAT IS AHEAD?

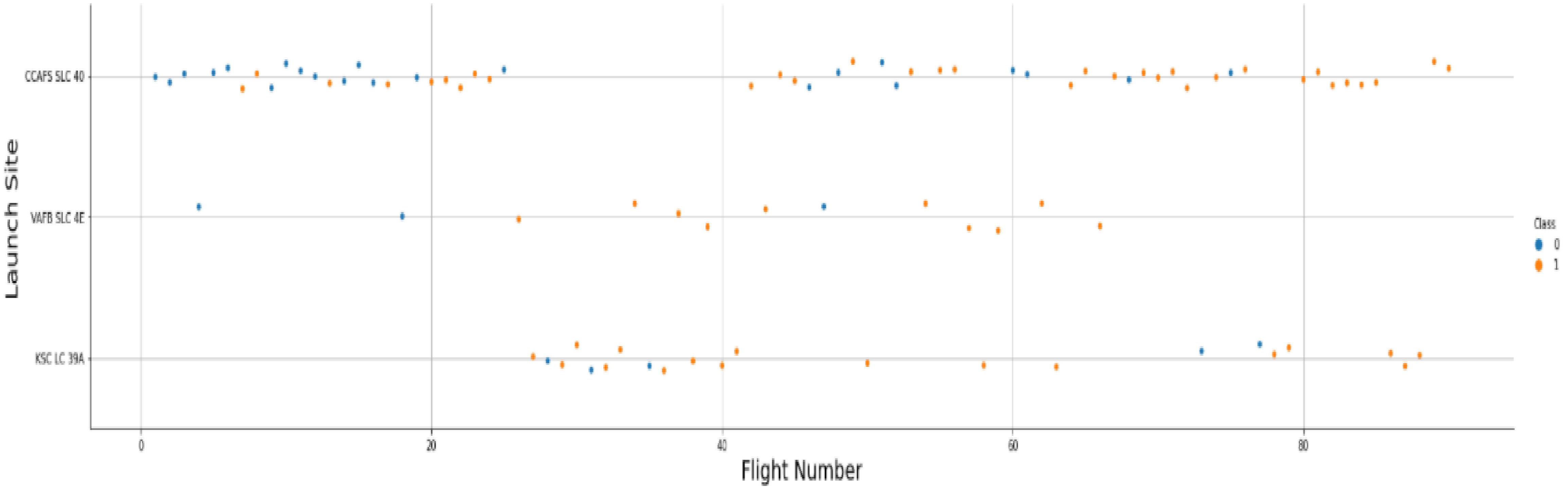
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

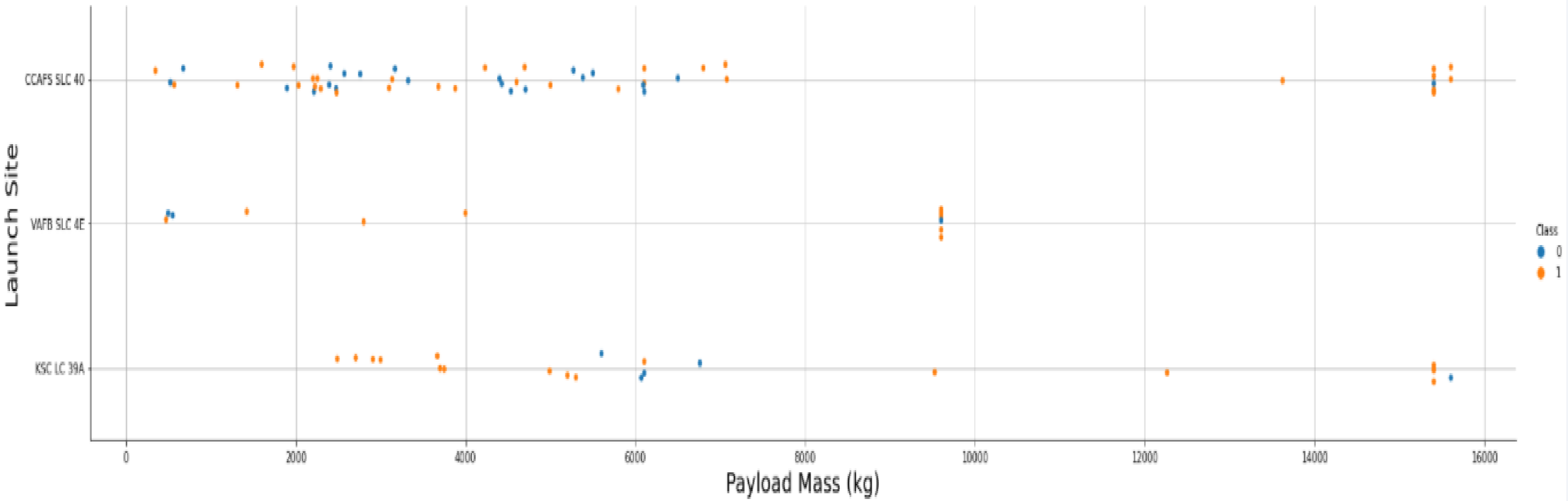
Flight Number vs. Launch Site

```
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.grid()
plt.show()
```



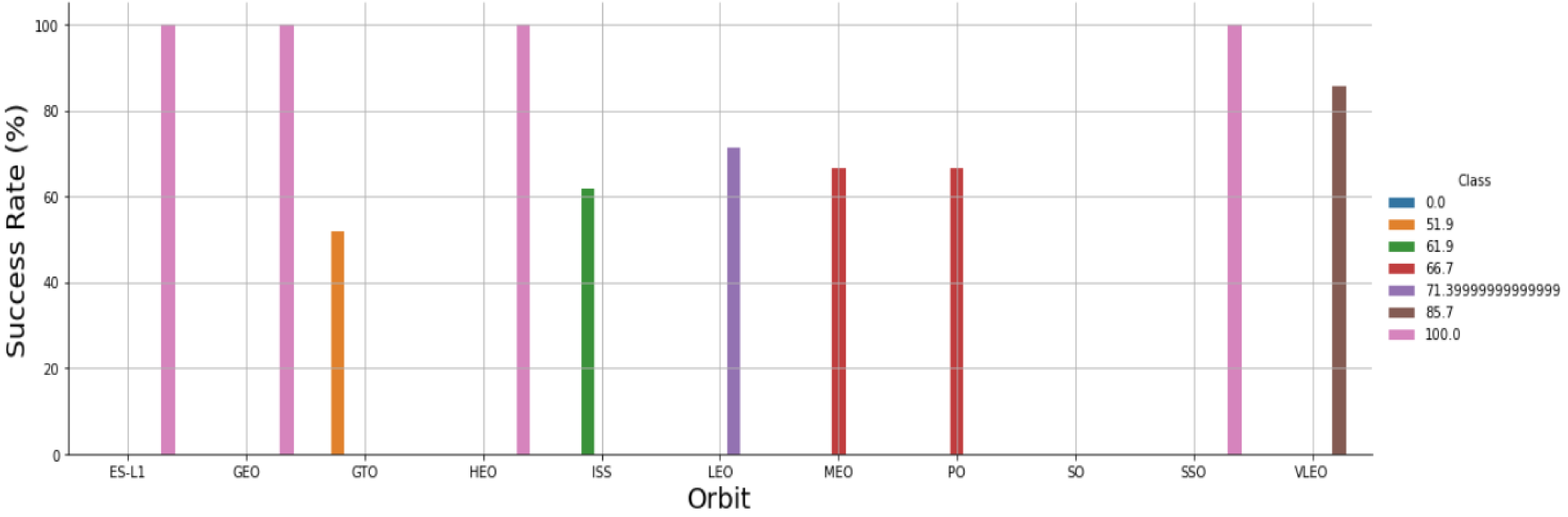
Payload vs. Launch Site

```
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.grid()
plt.show()
```



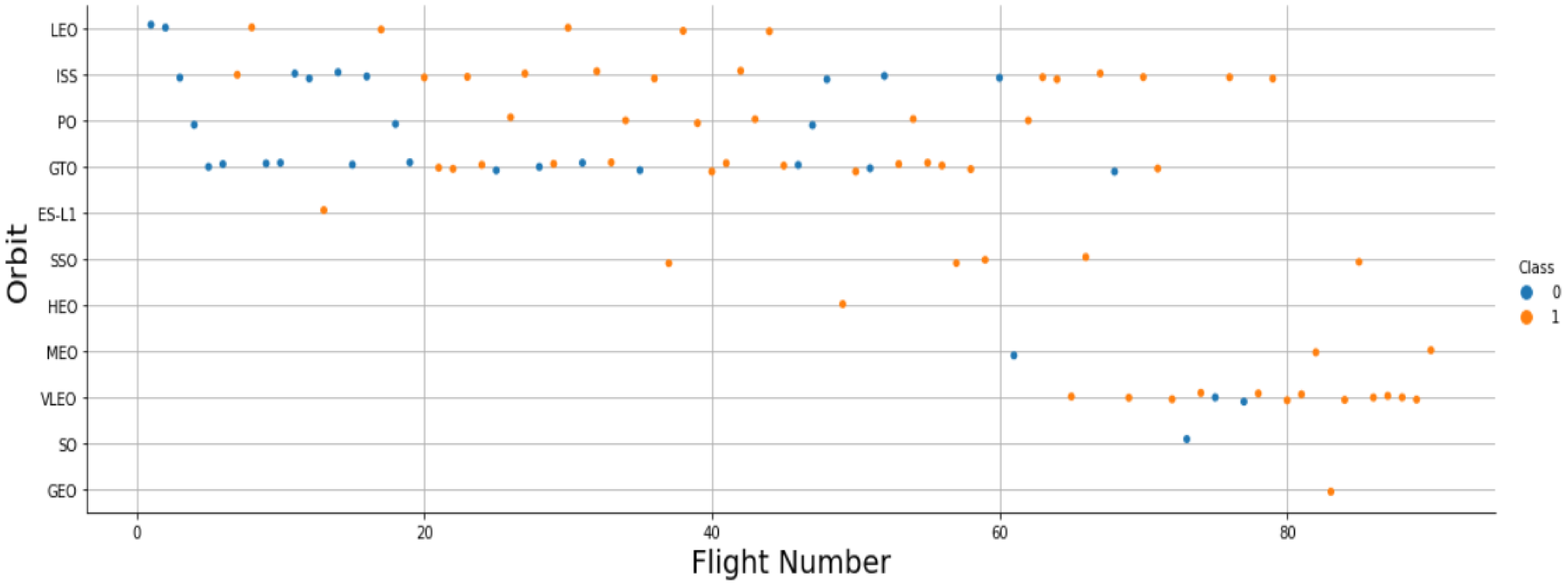
Success Rate vs. Orbit Type

```
orbit_success = df.groupby('Orbit').mean().round(3)*100
orbit_success.reset_index(inplace=True)
sns.catplot(x="Orbit",y="Class",data=orbit_success,hue='Class', kind='bar', aspect=3)
plt.xlabel("Orbit",fontsize=20)
plt.ylabel("Success Rate (%)",fontsize=20)
plt.grid()
plt.show()
```



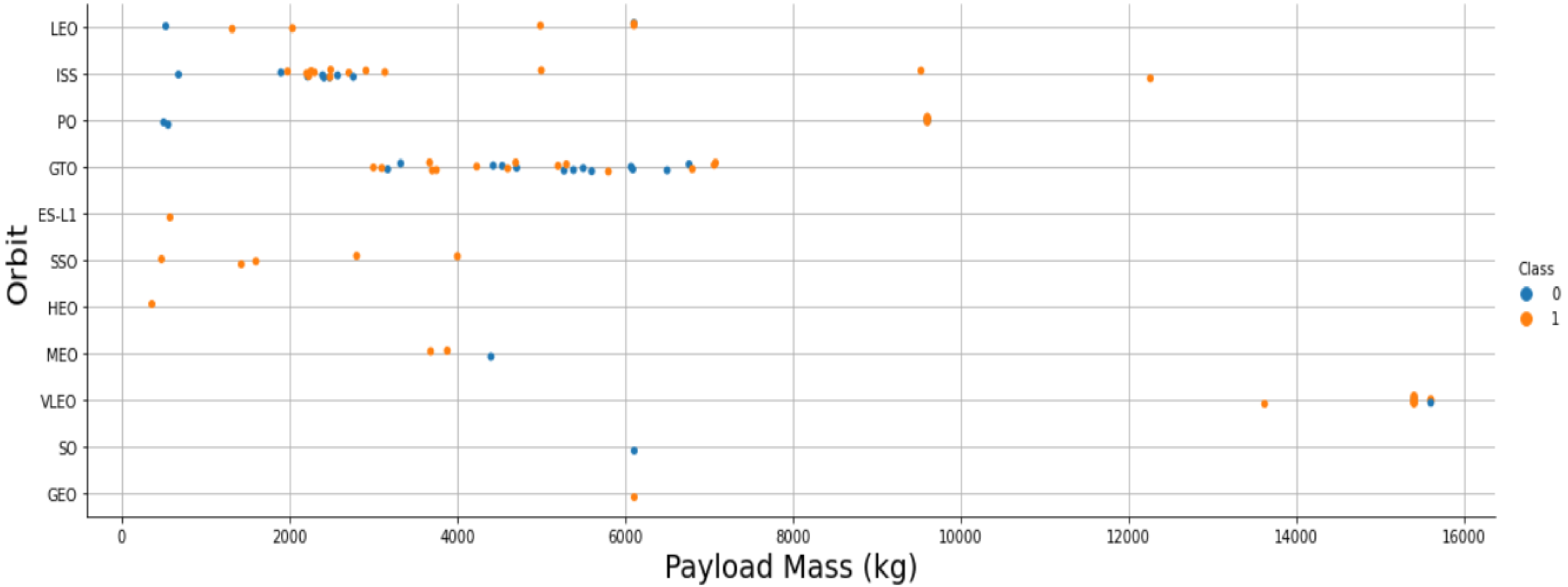
Flight Number vs. Orbit Type

```
sns.catplot(x="FlightNumber",y="Orbit",data=df,hue='Class', aspect=3)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.grid()
plt.show()
```



Payload vs. Orbit Type

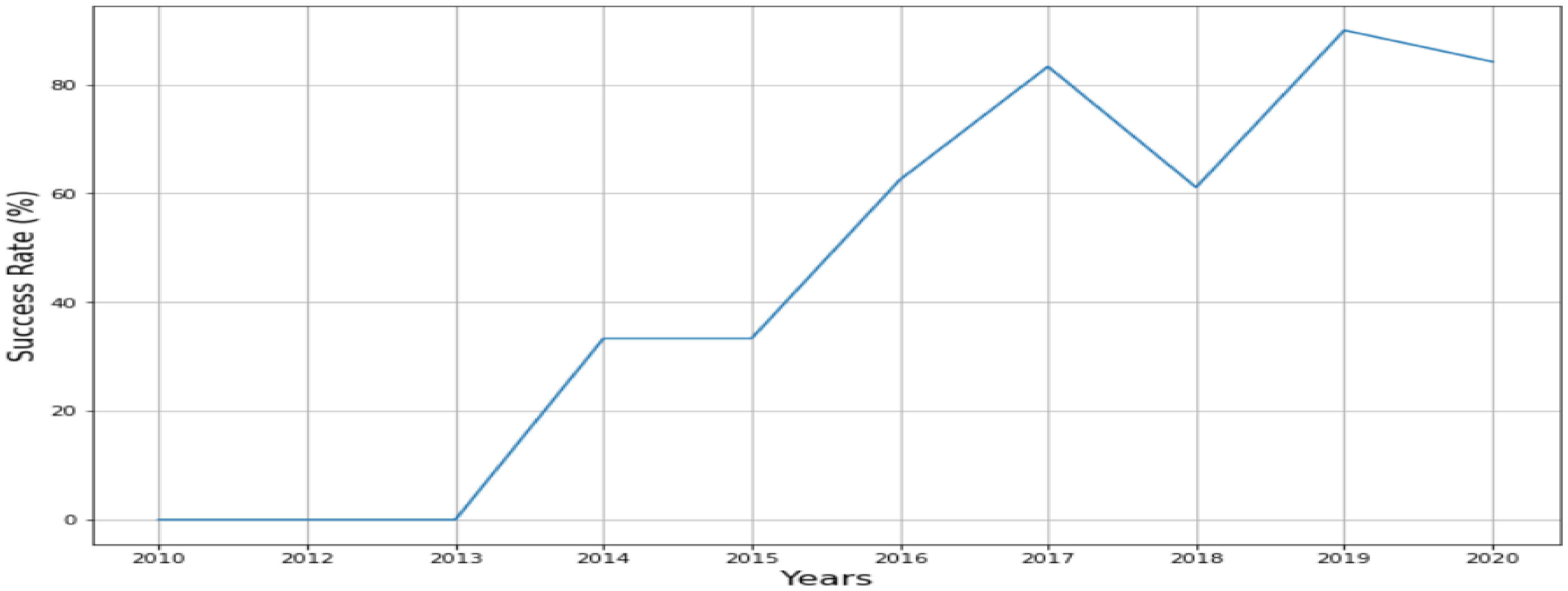
```
sns.catplot(x="PayloadMass",y="Orbit",data=df,hue='Class', aspect=3)
plt.xlabel("Payload Mass (kg)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.grid()
plt.show()
```



Launch Success Yearly Trend

```
x = average_by_year["Year"]
y = average_by_year["Class"]
plt.figure(figsize=(12,8))
plt.xlabel('Years', {'size':16})
plt.ylabel('Success Rate (%)', {'size':16})
plt.grid()
plt.plot(x,y)
```

```
[<matplotlib.lines.Line2D at 0x1eb354a4160>]
```



All Launch Site Names

```
# The names of the unique launch sites

%sql SELECT UNIQUE(LAUNCH_SITE) FROM SPACEX;
```

```
Done.
```

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

Launch Site Names Begin with 'CCA'

```
# The 5 records where launch sites begin with the string 'CCA'

%sql SELECT LAUNCH_SITE FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://wgm87184:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
```

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

Total Payload Mass

```
# The total payload mass carried by boosters launched by NASA (CRS)

%sql SELECT CUSTOMER, PAYLOAD_MASS__KG_ FROM SPACEX WHERE CUSTOMER = 'NASA (CRS)';
```

```
 * ibm_db_sa://wgm87184:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg
Done.
```

customer	payload_mass__kg_
NASA (CRS)	500
NASA (CRS)	677
NASA (CRS)	2296
NASA (CRS)	2216
NASA (CRS)	2395
NASA (CRS)	1898
NASA (CRS)	1952
NASA (CRS)	3136
NASA (CRS)	2257
NASA (CRS)	2490
NASA (CRS)	2708
NASA (CRS)	3310
NASA (CRS)	2205
NASA (CRS)	2647
NASA (CRS)	2697

Average Payload Mass by F9 v1.1

```
#Average payload mass carried by booster version F9 v1.1

%sql SELECT AVG(PAYLOAD_MASS__KG_) AS average_payload_mass FROM SPACEX WHERE BOOSTER_VERSION LIKE 'F9 v1%';
```

```
 * ibm_db_sa://wgm87184:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
```

average_payload_mass
1986

First Successful Ground Landing Date

```
#The date when the first successful landing outcome in ground pad was acheived

%sql SELECT MIN(DATE) AS date_of_the_first_successful_landing_outcome_in_ground_pad FROM SPACEX WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://wgm87184:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
```

date_of_the_first_successful_landing_outcome_in_ground_pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
#The names of the boosters which have success in drone ship and have payload mass between 4000-6000

%sql SELECT BOOSTER_VERSION, landing__outcome, payload_mass__kg_ FROM SPACEX WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ > 40
```

```
 * ibm_db_sa://wgm87184:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
```

booster_version	landing__outcome	payload_mass__kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

```
# The total number of successful and failure mission outcomes

%sql SELECT MISSION_OUTCOME, COUNT(*) AS CNT FROM SPACEX GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://wgm87184:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
```

mission_outcome	cnt
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

```
#The names of the booster_versions which have carried the maximum payload mass

%sql SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_ FROM SPACEX WHERE PAYLOAD_MASS__KG_ =  (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

```
 * ibm_db_sa://wgm87184:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
```

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

```
]:  #The failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

    %sql SELECT DATE, LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE LANDING__OUTCOME =  'Failure (drone ship)' AND DATE LIKE '2015%';

 * ibm_db_sa://wgm87184:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
]:
```

DATE	landing__outcome	booster_version	launch_site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
#The count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20

%sql SELECT LANDING__OUTCOME, COUNT(*) AS CNT FROM SPACEX WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY CNT DES
```

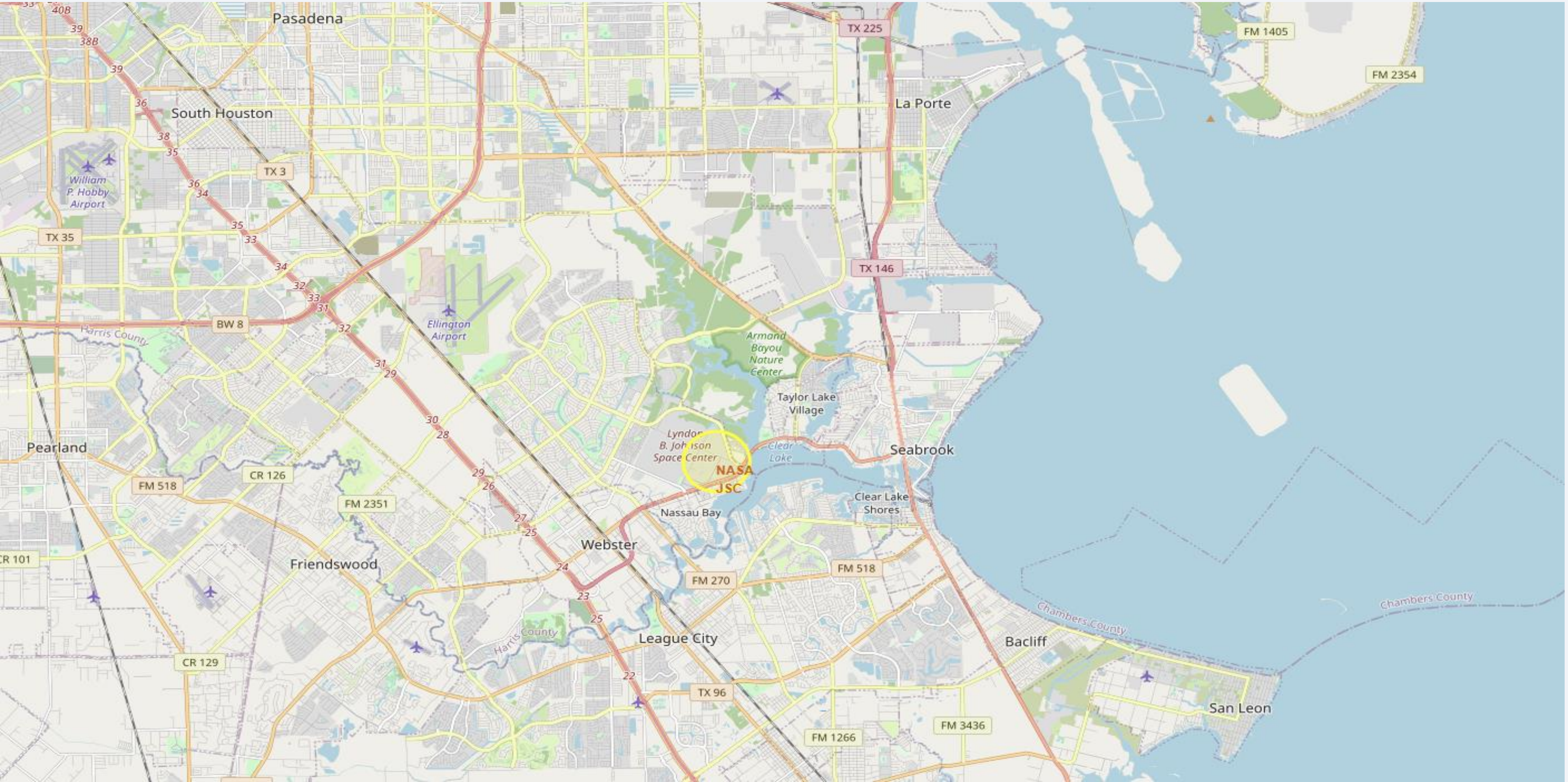
```
 * ibm_db_sa://wgm87184:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
```

landing__outcome	cnt
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

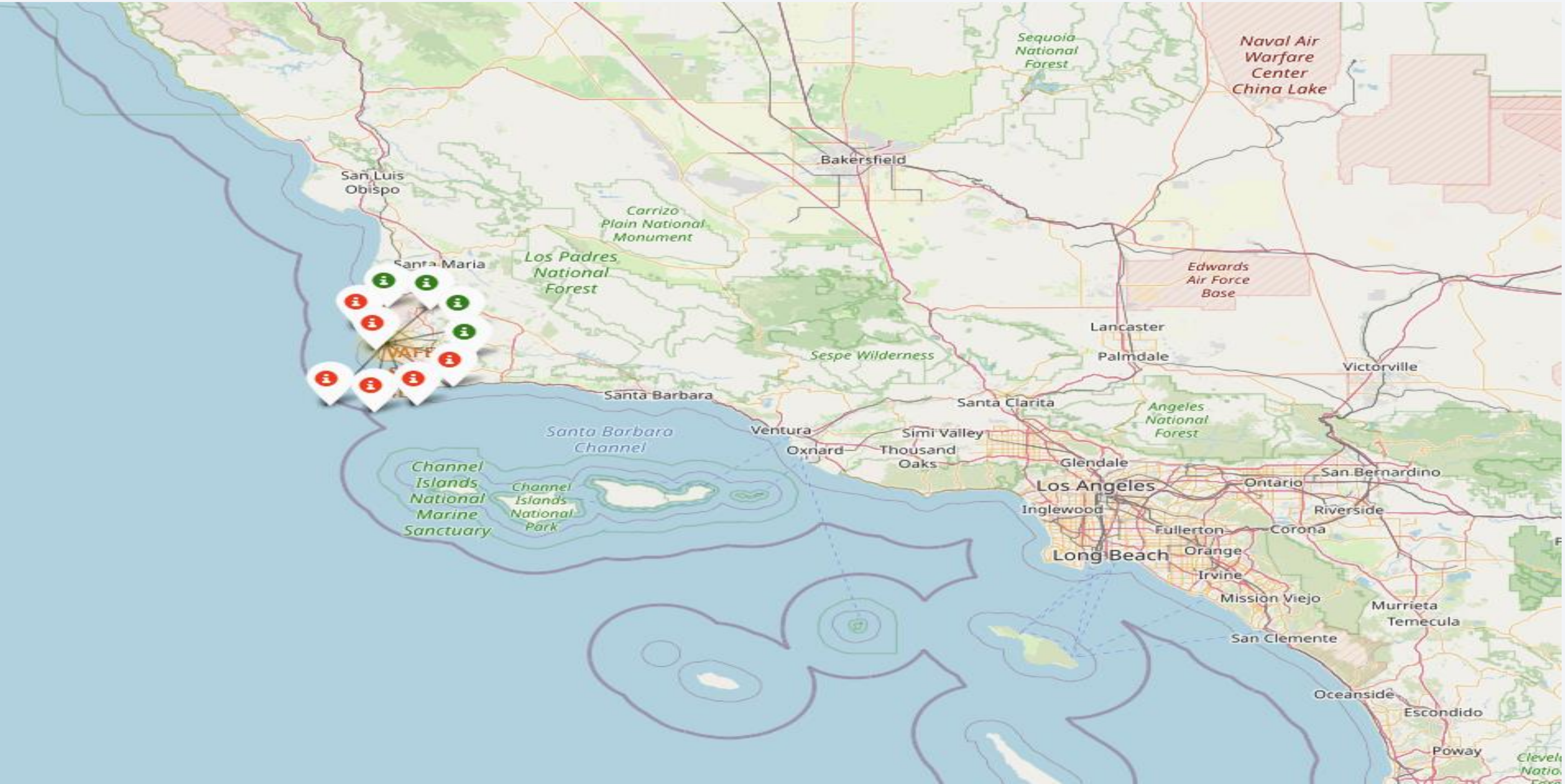
Section 3

Launch Sites Proximities Analysis

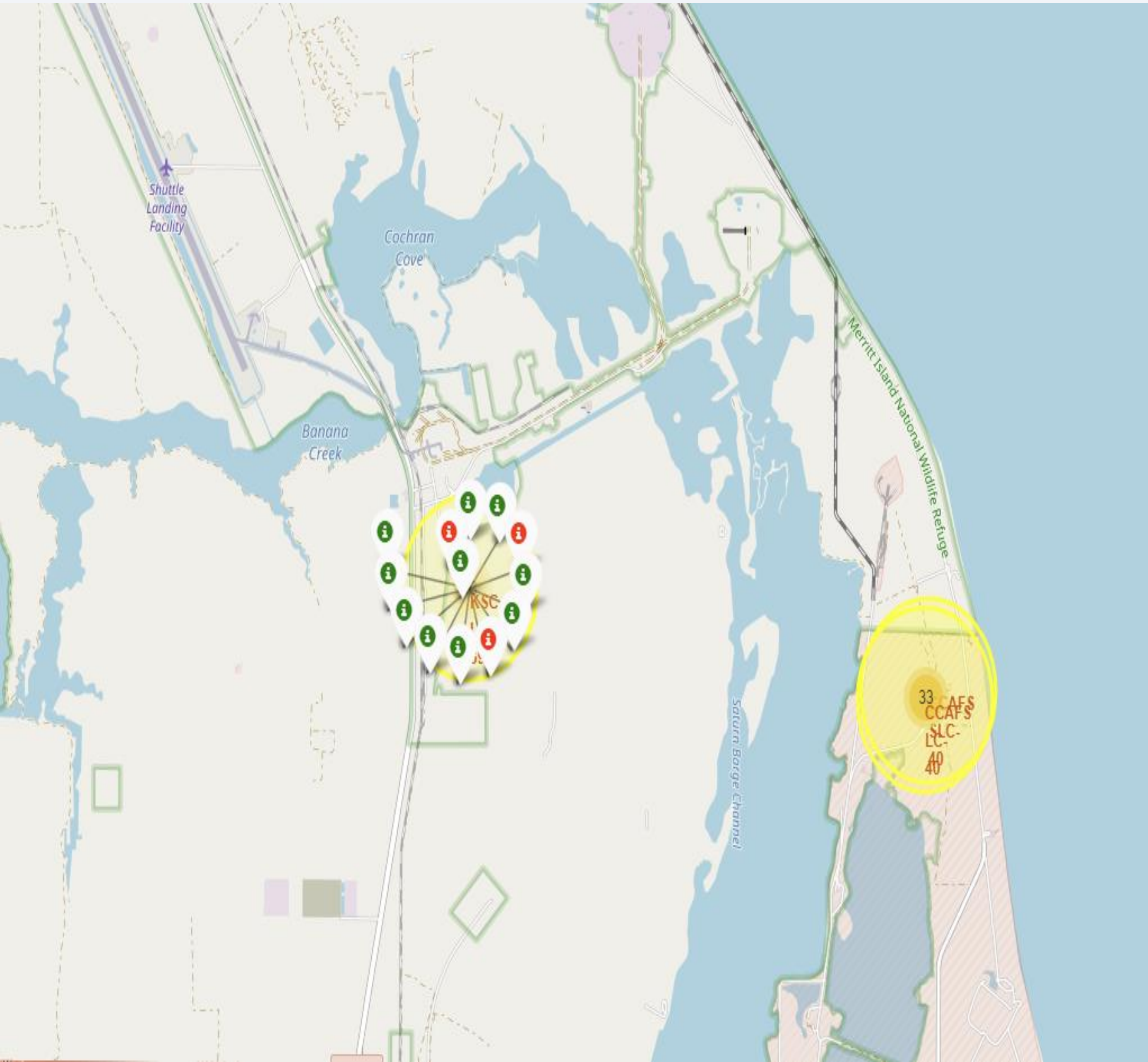
NASA Johnson Space Center

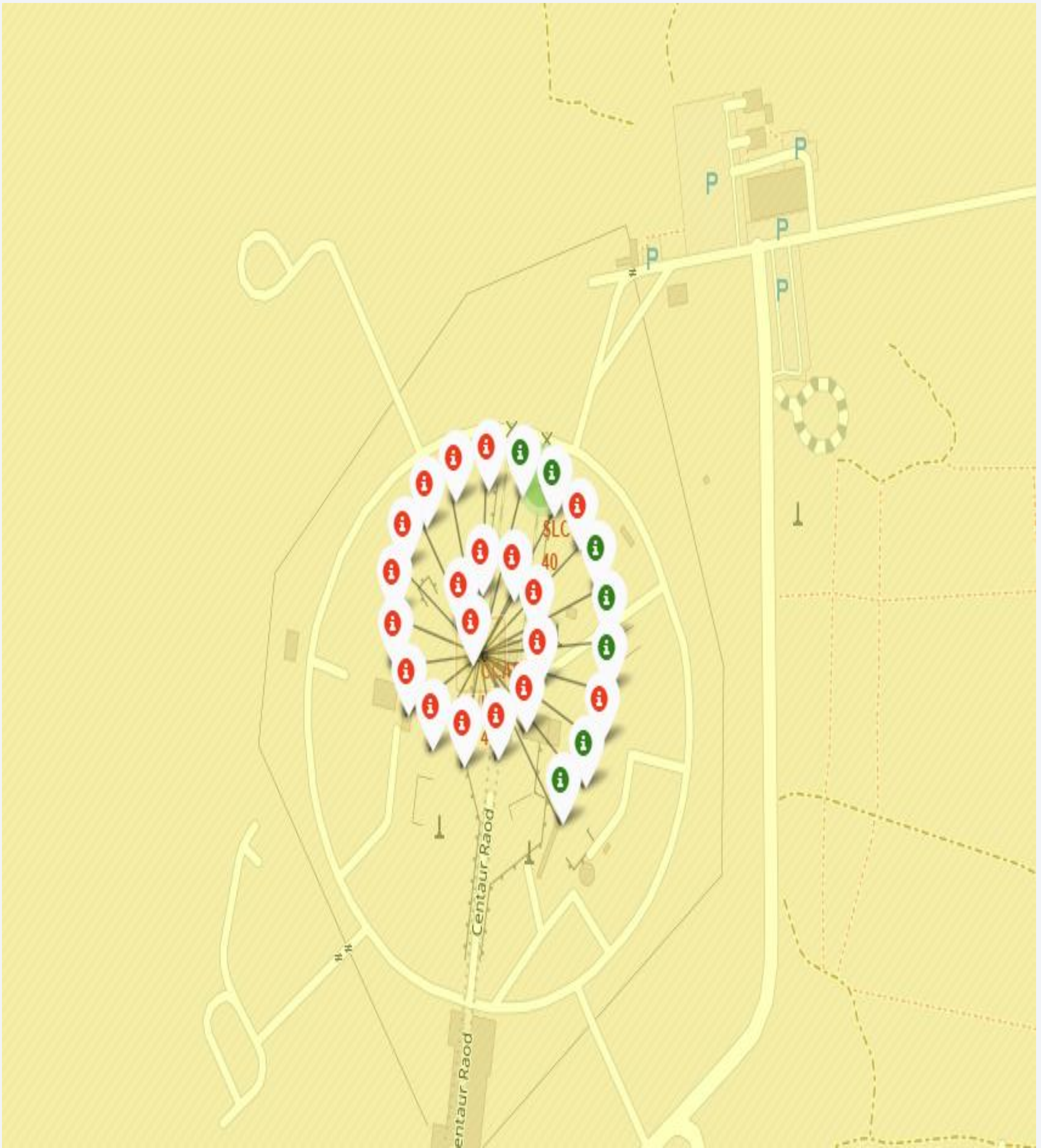


Launch Sites (west side)



Launch Sites (east side)

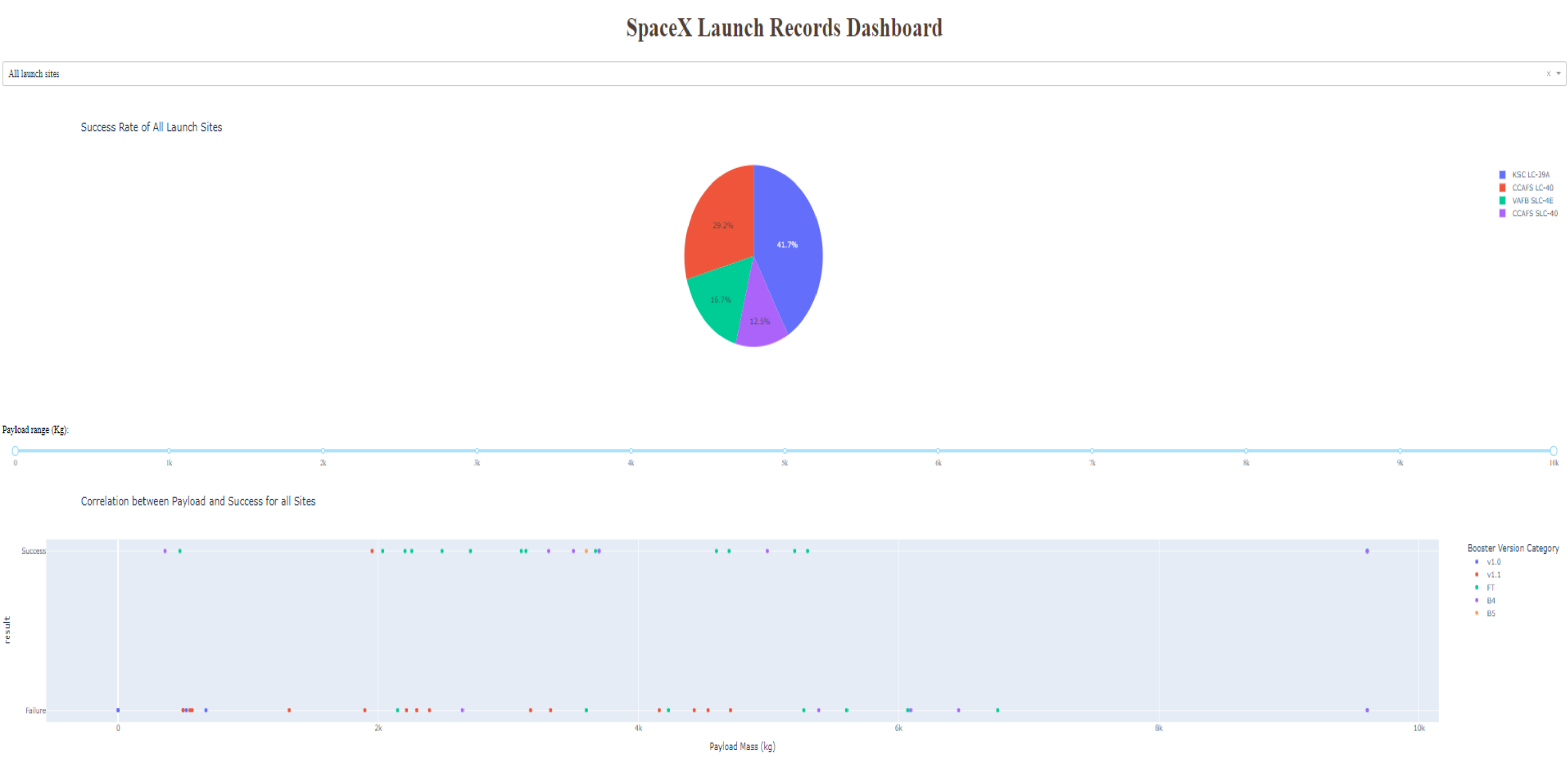




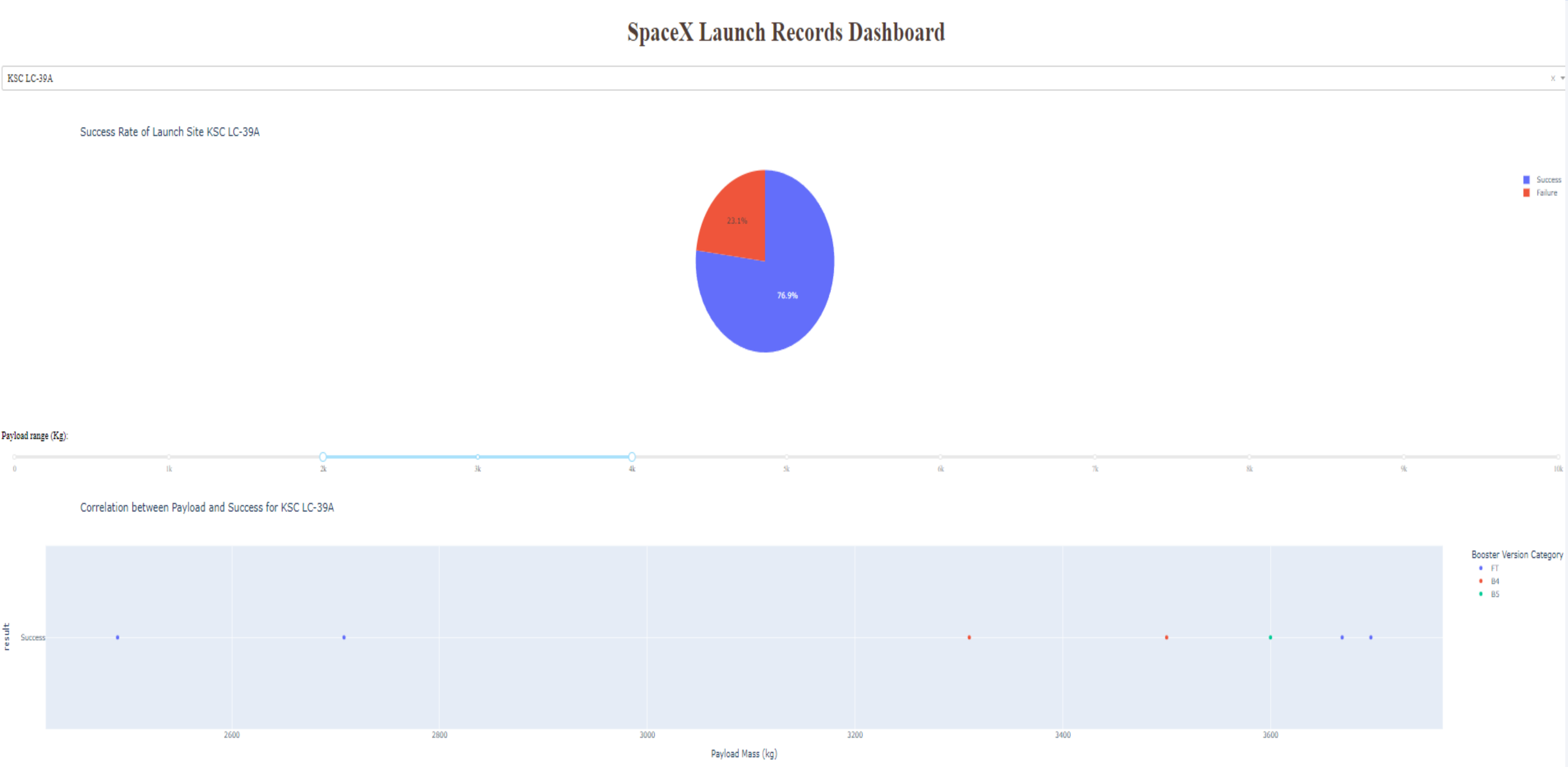
Section 4

Build a Dashboard with Plotly Dash

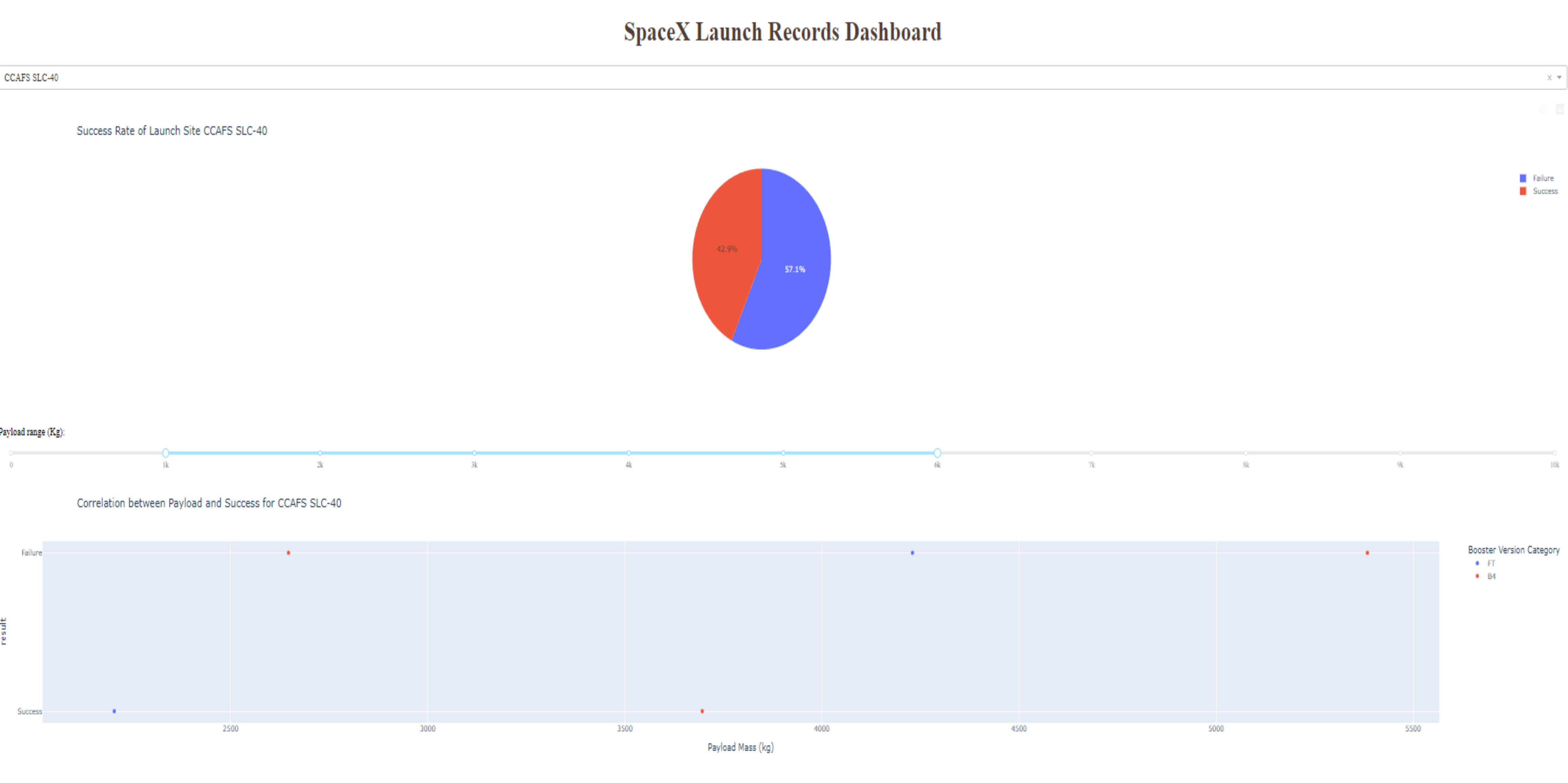
Full App View with selection combination 1



Full App View with selection combination 2



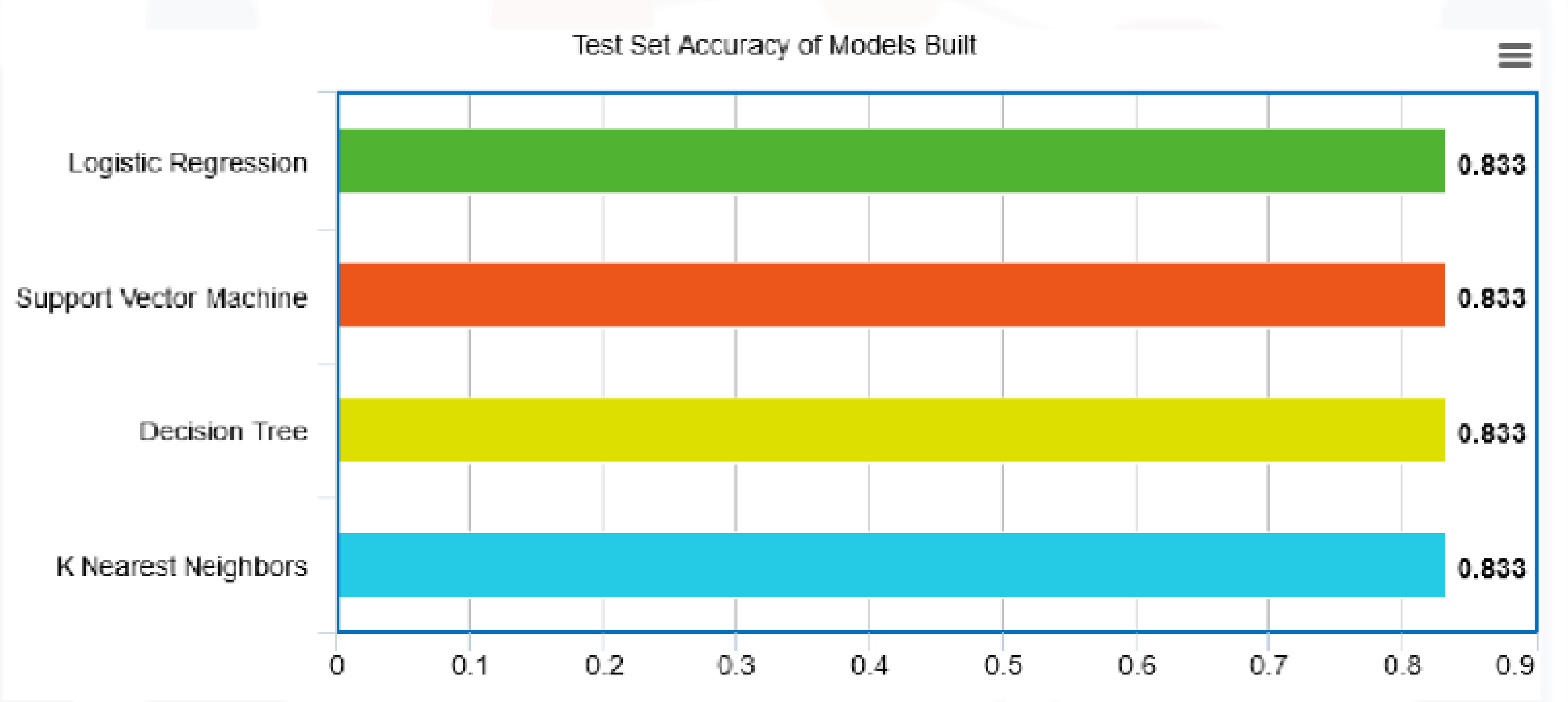
Full App View with selection combination 3



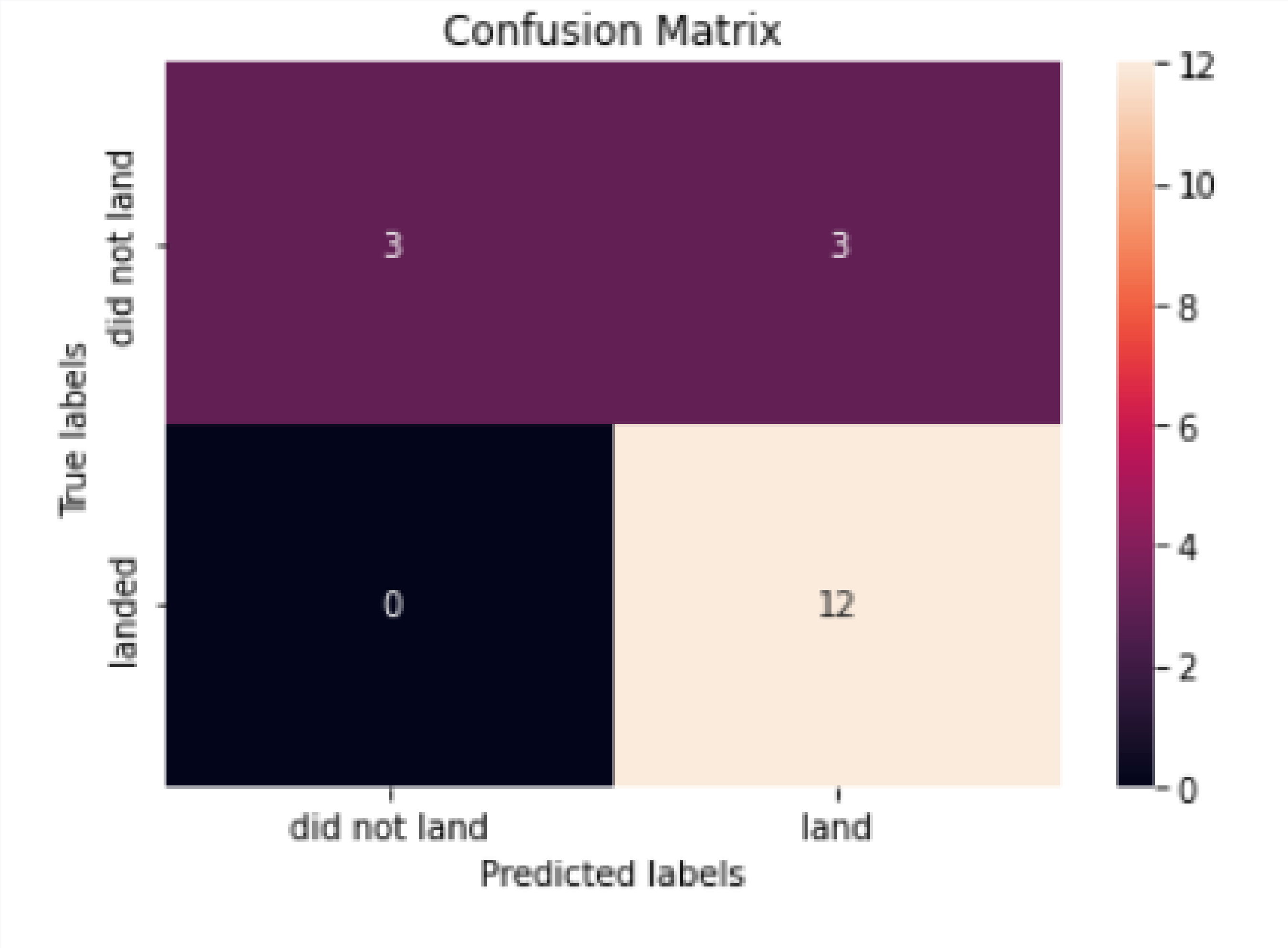
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Conclusions

- With around 83% accuracy first stage of SpaceY launches can be predicted
- From our four models, three of them gave results almost identical
- Our data and methods are enough and clear to make specific predictions and our accuracy and forecast results are very good
- Further analysis can be done using different models and more wide historical data to better predict our cost

Thank you!