IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Miguel Andres
6/8/2023

Outline

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

Executive Summary

- Summary of methodologies
- Exploratory Data Analysis
  - To analyzed the success rate of first stage landings base on historical data to identify patterns and factors that contribute to successful landings
- Interactive map with Folium and Plotly Dash
  - To analyze for optimum launch site selection, allowed us to assess various factors such as geographical locations, infrastructure, and logistical considerations
- Predictive Analysis
  - Hyperparameter for SVM, Classification Trees, and Logistic Regression, we aimed to predict the likelihood of first stage reuse
- Summary of all results
- Our exploratory analysis revealed key insights into the success rate of first stage landings, with Site KSC LC-39A achieving the highest success rate of 76.9%. Furthermore, our interactive map highlighted optimal launch site options based on various factors.
- The predictive analysis using SVM, Classification Trees, and Logistic Regression models demonstrated promising results in determining the likelihood of first stage reuse. The models achieved accuracy rates ranging from 83.33% to 88.89% on the test data, indicating their ability to predict successful first stage landings. These findings suggest that these models can be valuable tools in assessing the potential reuse of the first stage, providing insights into cost-saving opportunities for SpaceX

Introduction

- The space travel industry is experiencing rapid growth and presents numerous opportunities for innovation and economic prosperity. As we envision a future where space travel becomes more accessible and commonplace, it is essential for us to position ourselves at the forefront of this transformative industry. By providing the rockets needed for launches, we can tap into the immense potential and lucrative market that space travel offers.
- To be successful in this competitive landscape, it is crucial to address the cost-effectiveness of our rockets. One key aspect that significantly impacts costs is the reusability of the first stage rockets. If we can safely land and reuse these rockets for future launches, we can achieve substantial cost savings, allowing us to compete effectively against established players like SpaceX and other competitors.
- The objective of this analysis is to determine the likelihood of the first stage rockets landing safely back on Earth, making them available for reuse in subsequent launches. By leveraging data and employing advanced analytical techniques, we aim to gain insights into the factors influencing launch success and predict the probability of successful first stage recovery.
- Through this analysis, we seek to uncover the critical factors that contribute to successful first stage reuse. Armed with this knowledge, we can make informed decisions regarding launch strategies, operational efficiency, and cost optimization. Ultimately, our goal is to establish ourselves as a reliable and cost-effective provider of rockets, positioning us as a major player in the exciting and rapidly expanding space travel industry.
- By addressing the challenges of cost-effectiveness and first stage reuse, we can seize the opportunities presented by the space travel sector and shape the future of human exploration and commercial ventures beyond Earth's boundaries.

Section 1

Methodology

Methodology

Executive Summary

- Data source:
  - wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
  - https://api.spacexdata.com/v4/launches/past
- Perform data wrangling
  - For the SpaceX API we isolated the Falcon 9 launches and replace an missing values with mean value.
  - Wikipedia was a simple process of converting the data into a working DataFrame.
  - Finally we created a column for mission outcome for orbit and landing outcome
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Using SQL to call the data we clean and use it to gain insight with bar and scatter graphs to see the site with high success launch rates by comparing flight number vs launch site vs orbit vs payload mass, concluding that site KSC LC-39A had 77% success and that the payload mass was the only factor to determine the outcome.

Continue on to next slide 6

Methodology

Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash
- Created a map to analysis
- To select a optimum site for launches considering logistic transportation.
- Plotly Dash to interactively see payload mass for each launch site and the success rate.
- Perform predictive analysis using classification models
- Predictive Analysis:
- Built, tuned, and evaluated classification models to predict the likelihood of first stage rocket reuse.
- Applied various classification models and evaluated their performance.

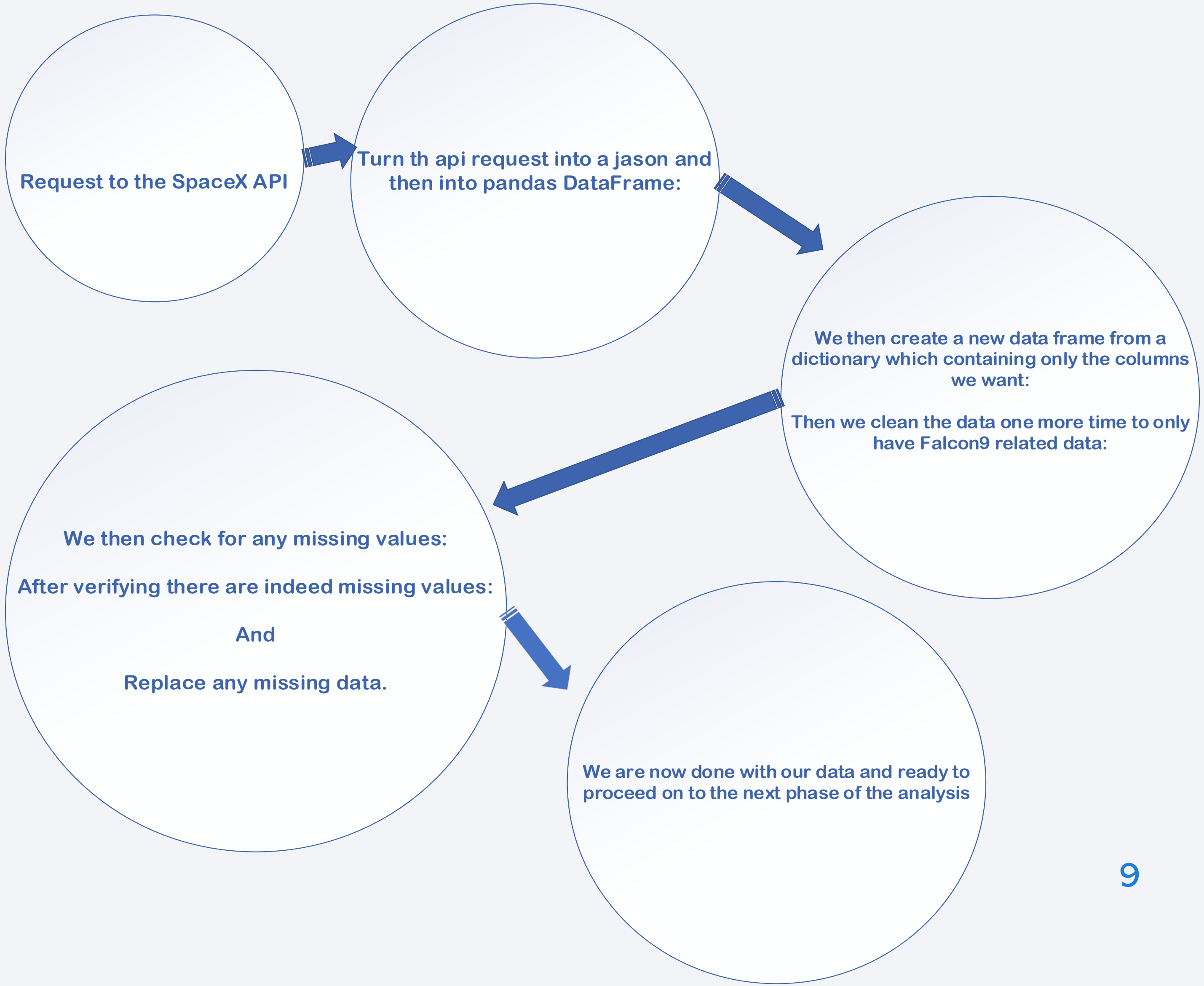
Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

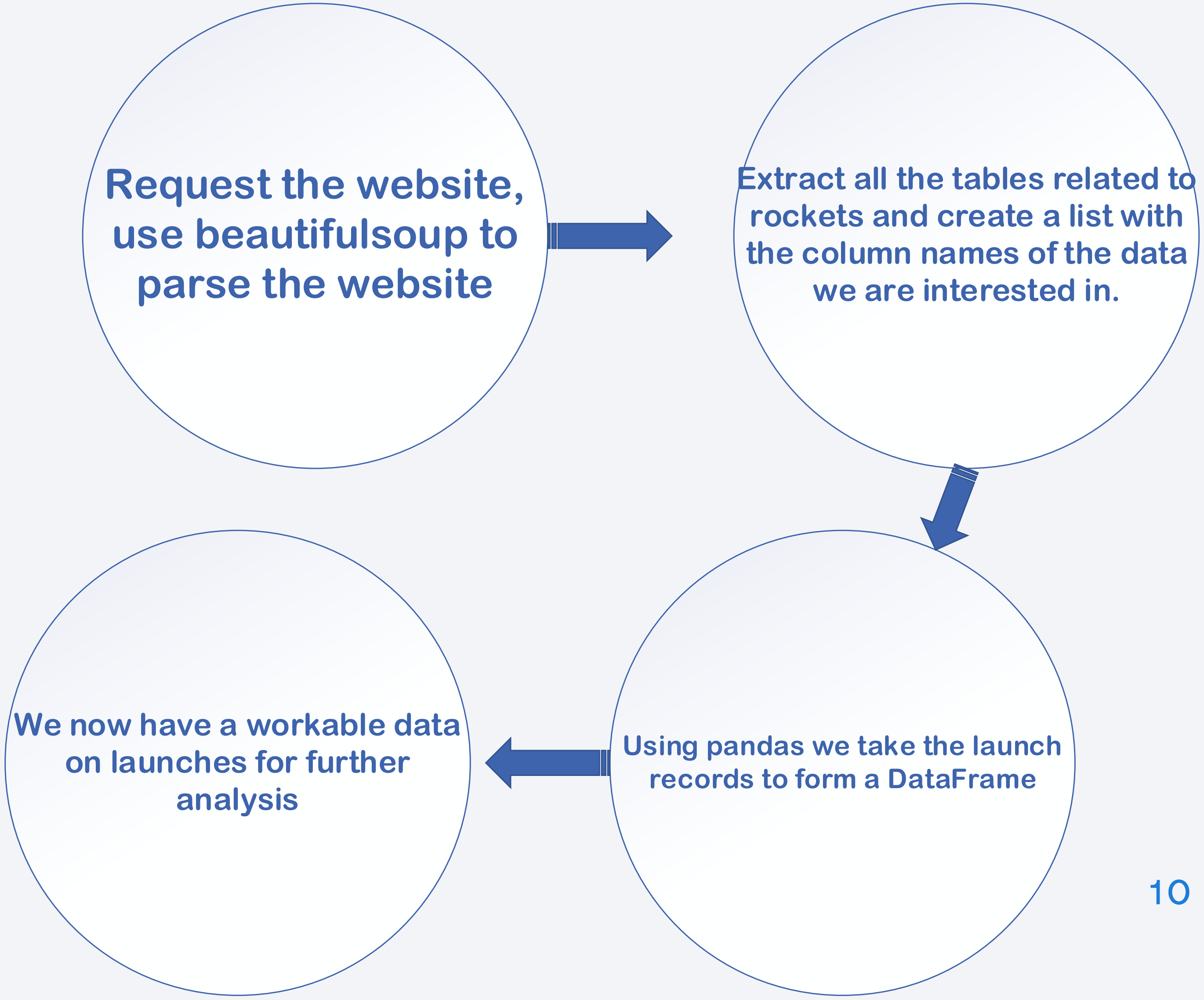
- static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
- jd = re.json()
- data = pd.json_normalize(jd)
- df = pd.DataFrame(launch_dict)
- data_falcon9 = df[df['BoosterVersion'] == 'Falcon 9']
- data_falcon9.isnull().sum()
- # Calculate the mean value of PayloadMass column
- mean_payload = data_falcon9['PayloadMass'].mean()
- # Replace the np.nan values with its mean value
- data_falcon9['PayloadMass'].replace(np.nan, mean_payload)

GitHub URL SpaceX API calls notebook(https://github.com/NightKnightSight/AnalysisReusableRockets/blob/8770afe3github/spacex-data-collection-api.ipynb)



Data Collection - Scraping

- soup = BeautifulSoup(response.content, 'html.parser’)
- # Use the find_all function in the BeautifulSoup object, with element type `table` html_tables = soup.find_all('table’)
- launch_dict= dict.fromkeys(column_names)
- df = pd.DataFrame(launch_dict)
- GitHub URL of the completed web scraping notebook(https://github.com/NightKnightSight/AnalysisReusableRockets/blob/b8396ee4d80e105cbb2d0335ec27c57ef86d708f/jupyter-labs-webscraping.ipynb)



Data Wrangling

- GitHub URL(https://github.com/NightKnightSight/AnalysisReusableRockets/blob/9895c25c055ee40833f75459d8299752bd8237ce/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb)

Using the data we have collected from the two previous slide into a dataframe to preform Exploratory Data Analysis

We divide the total launches to there respective launch site: df['LaunchSite'].value_counts()

Next we calculate the occurrence of each orbit: df['Orbit'].value_counts()

Next we calculate the mission outcome for each orbit into a new dataframe: landing_outcomes = df['Outcome'].value_counts()

Using the landing_outcome dataframe to iterate to find all the bad outcome code number and add into a new set: bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])

Finally we add new column to the dataframe that replaces all bad outcomes as Zero and successful landings with 1 to calculate the success rate for falcon 9 at 66.66%

EDA with Data Visualization

- **Scatter plot**: We created scatter plots to examine the relationship between flight numbers and launch sites, as well as payload mass and launch sites. These plots help us understand if there is any correlation between these variables and the success of each launch. By visualizing the data points on a scatter plot, we can observe any patterns or trends that may exist.
- **Scatter plot**: Another scatter plot was used to explore the relationship between orbit type and payload mass. This chart allows us to analyze how different payload masses are associated with different orbit types, and whether this has any influence on launch success.
- **Bar chart**: We employed a bar chart to examine the relationship between the success rate and orbit type. By visually comparing the success rates for different orbit types, we can identify if any particular orbit type exhibits a higher or lower success rate.
- **Line graph**: To assess the improvement in launch success over the years, we utilized a line graph. This graph represents the success rate over time from 2010 to 2020. By plotting the data points and connecting them with a line, we can observe if there has been any significant improvement or variation in launch success during this period.
- GitHub URL for completed EDA with data visualization notebook(https://github.com/NightKnightSight/AnalysisReusableRockets/blob/680ce71354bdedcfee80823c529f8440a294e0a4/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb)

EDA with SQL

- We then %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTBL to see all launch sites
- we wanted to see all sites that start with CCA with %sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5
- To see total payload mass in database: %sql SELECT SUM(Payload_Mass__kg_) AS Total_Payload_Mass FROM SPACEXTBL WHERE Customer = 'NASA (CRS)’
- The average mass carried by Falcon 9 v1.1: %sql SELECT AVG(Payload_Mass__kg_) AS Average_Payload_Mass FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1’
- To get the first successful landing with ground pad: %sql SELECT MIN(Date) AS First_Successful_Landing_Date FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)’
- To get all successful drone ship with mass between 4000 – 6000: %sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND Payload_Mass__kg_ > 4000 AND Payload_Mass__kg_ < 6000
- List the total success and failure mission outcomes: %sql SELECT Mission_Outcome, COUNT(*) AS Total_Outcomes FROM SPACEXTBL GROUP BY Mission_Outcome
- List names of Booster versions that have carried the maximum payload: %sql SELECT Booster_Version FROM SPACEXTBL WHERE Payload_Mass__kg_ = (SELECT MAX(Payload_Mass__kg_) FROM SPACEXTBL)
- failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015 : %sql SELECT CASE substr(Date, 4, 2) WHEN '01' THEN 'January' WHEN '02' THEN 'February' WHEN '03' THEN 'March' WHEN '04' THEN 'April' WHEN '05' THEN 'May' WHEN '06' THEN 'June' WHEN '07' THEN 'July' WHEN '08' THEN 'August' WHEN '09' THEN 'September' WHEN '10' THEN 'October' WHEN '11' THEN 'November' WHEN '12' THEN 'December' END AS Month_Name, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTBL WHERE substr(Date, 7, 4) = '2015' AND Landing_Outcome LIKE '%Failure%' AND Landing_Outcome LIKE '%drone ship%’
- List the total successful landing between 2015 – 2017: %sql SELECT 'Landing_Outcome', COUNT(*) AS Count FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' AND Landing_Outcome LIKE '%Success%' GROUP BY Landing_Outcome ORDER BY Count DESC
- **GitHub URL** https://github.com/NightKnightSight/AnalysisReusableRockets/blob/a74e5612941fba423a55bd660b6e5e927f75ea82/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Marker: Markers clusters can be added to represent the location of the rocket booster launch site on the map. They provide a visual reference point for the exact coordinates of the launch site and successful and fail launches.
- Circle: A circle can be added to represent the safety radius or exclusion zone around the launch site. This helps visually indicate the area that needs to be cleared and secured during the launch.
- Polyline: Polyline objects can be used to mark to calculate near by locations, This can be useful for visualizing the planned or actual path of the rocket during launch.
- Added polyline to the selected coastline for the possibility to recover rockets off the coast after landing on pad off the coast.
- Added polyline to closest railway depict the closest train rail. This object provides a visual representation of the train rail route, which is significant for logistical planning. It helps identify transportation options, connectivity, and accessibility to railways for efficient movement of rockets to the launch site
- Added polyline to the closest city for potential risk to civilians
- Added polyline to the closest highway object shows the route to the nearest major roadway, which is crucial for transportation and logistics. It assists in evaluating road accessibility, travel times, and connectivity to major transportation networks.
- GitHub URL: https://github.com/NightKnightSight/AnalysisReusableRockets/blob/81bf6a34496ec002350243a8b6ae009cb52853ff/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- **Pie Chart**: The "success-pie-chart" displays the total successful launches by site or the success versus failed launches at a specific launch site. The dropdown menu "site-dropdown" allows to choose between viewing data for all sites or a specific launch site.
- **Range Slider** and **Scatter Chart**: The "payload-slider" is a range slider that enables the selection of a payload mass range which in turn will effect the display of the Scatter chart. The "success-payload-scatter-chart" is a scatter plot that shows the relationship between payload mass and launch success, with additional categorization based on the booster version. The scatter plot is updated based on the selected site from the dropdown menu and the chosen payload range from the slider.
- The purpose of adding these plots and interactions is to provide a dashboard for analyzing SpaceX launch records. The pie chart helps understand the overall success distribution by launch site, allowing for a quick comparison between sites. These visualizations enable us to gain insights into launch success patterns, payload distribution, and the impact of different launch sites and payload masses on mission outcomes success rate
- GitHub URL

Predictive Analysis (Classification)

- After performing this process the results are as follows:
- Accuracy on test data - Logistic Regression: 0.8333333333333334
- Accuracy on test data - SVM: 0.8333333333333334
- Accuracy on test data - Decision Tree: 0.8888888888888888
- Accuracy on test data - KNN: 0.8333333333333334
- Decision Tree performs best.
- GitHub Link: https://github.com/NightKnightSight/AnalysisReusableRockets/blob/0103384814516f5b47cd244ba2ba8181c01834ea/S_Part_5.jupyterlite.ipynb

Load in both data set the one we modify to have 1 and 0 and the original unmodified data set

We assign the unmodified set as y numpy dataframe and modify as x transformer:
Y = data['Class'].to_numpy()
X = transform.fit_transform(X)

For each data set we split into two groups one for training the models and one for testing the models:
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
This results into 80 for training and 20 for testing.

logistic regression object and GridSearchCV object is use to train the set into 10 different folds.Then confusion matrix based on the predicted labels (yhat) and the true labels (Y_test). The confusion matrix provides a comprehensive view of the model's performance, showing the number of true positives, true negatives, false positives, and false negative

support vector machine object and a GridSearchCV object. This step explores different combinations of hyperparameters and evaluates the model's performance using cross-validation.
We use a confuse matrix again then visual see the true positives, true negatives, false positives, and false negative

We repeat the process this time for a decision tree classifier object
And use a confusion matrix again.

We do the process one more time this time for a K Nearest Neighbors (KNN) is a non-parametric classification and regression algorithm and use a confusion matrix again

Then we take all the results from each algorithm to find the best best performing one:
best_method = max(lr_accuracy, svm_accuracy, tree_accuracy, knn_accuracy)

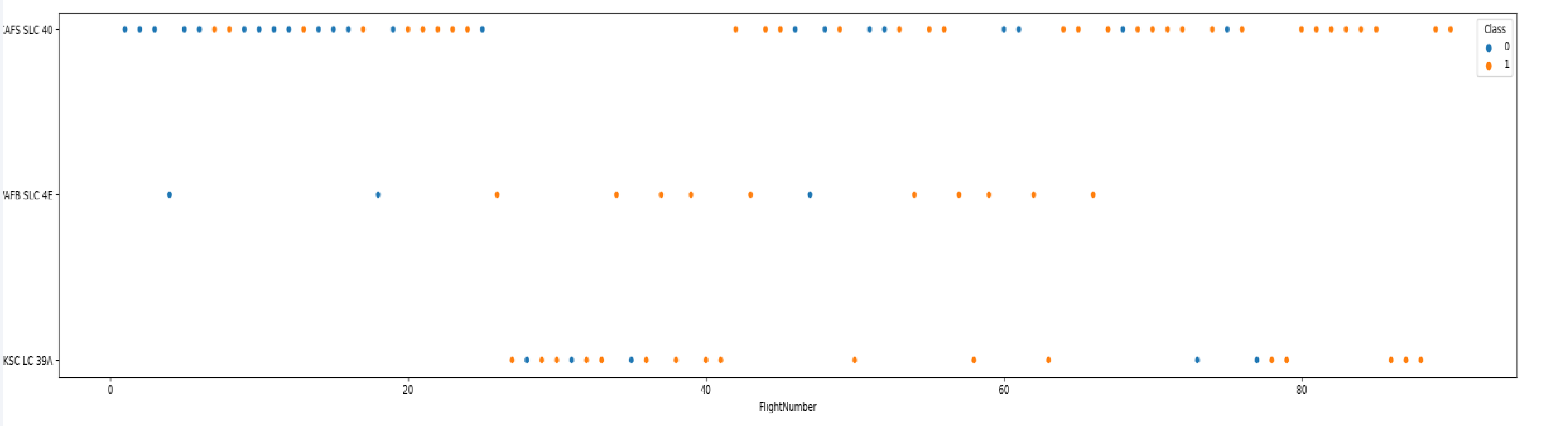
Results

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

Section 2

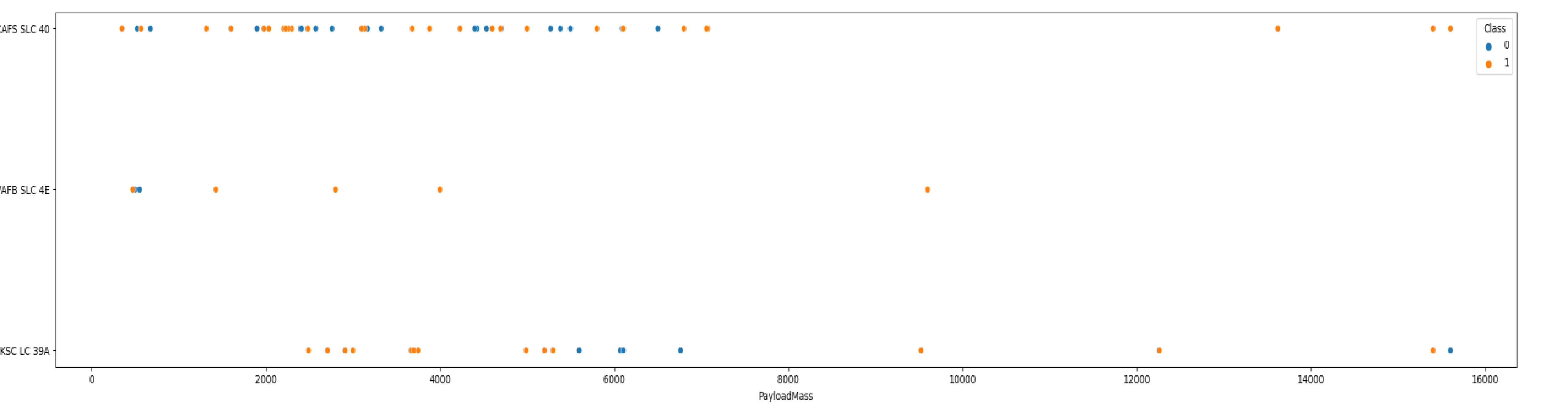
Insights drawn from EDA

Flight Number vs. Launch Site



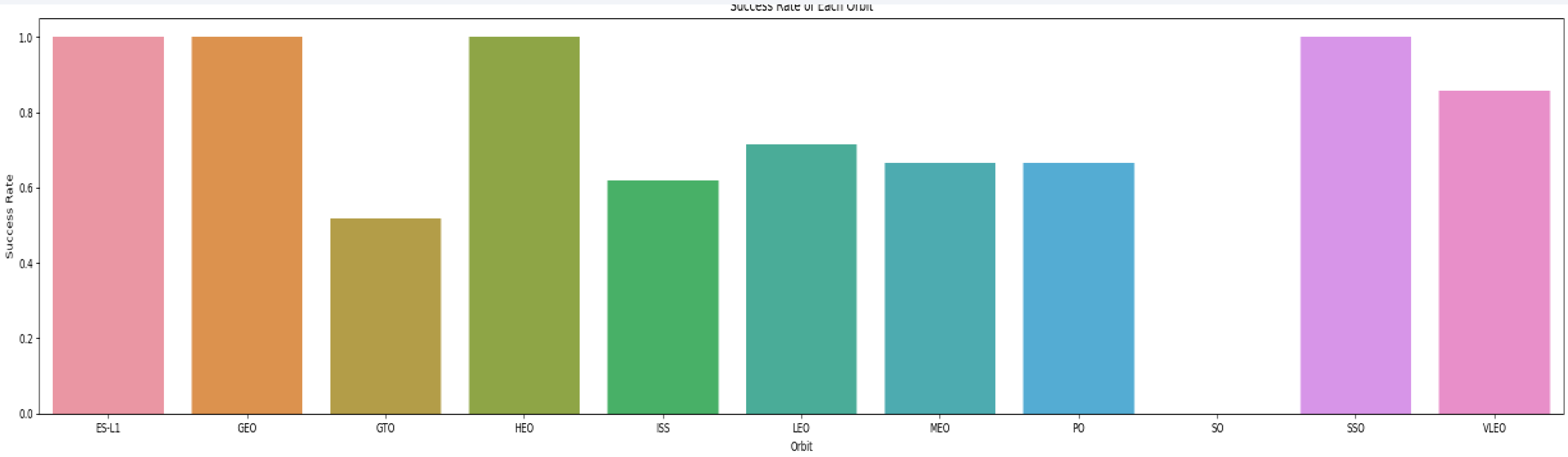
- We can see VAFB has a high success launches, but it also has the least flight. KSC has an impressive launch success and multiple flight test.

Payload vs. Launch Site



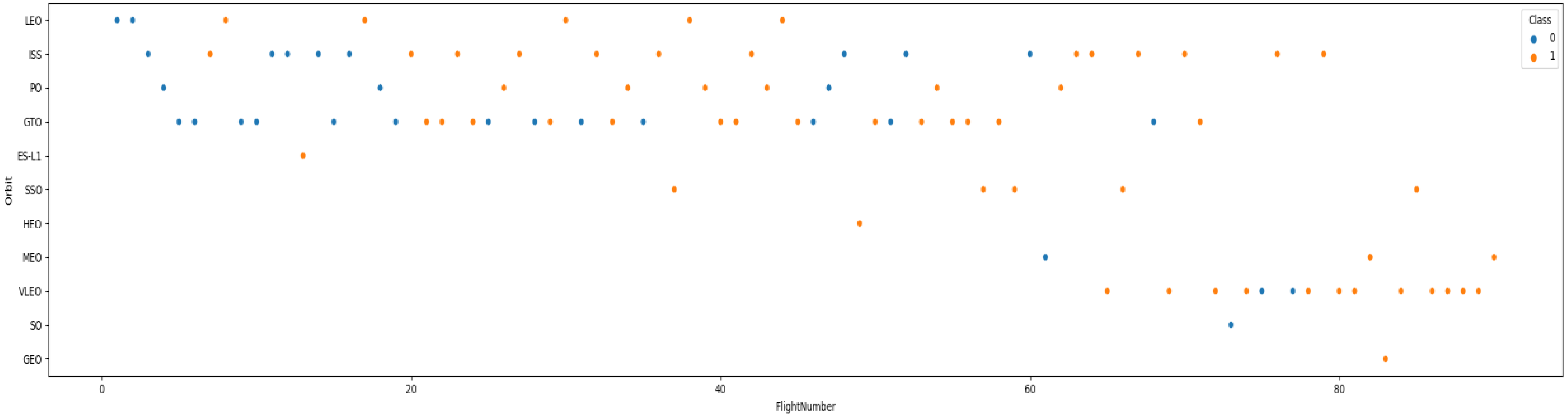
- We can see that launch sites are a non-factor when the payload mass is above 9000 with 7 out of 8 are successful

Success Rate vs. Orbit Type



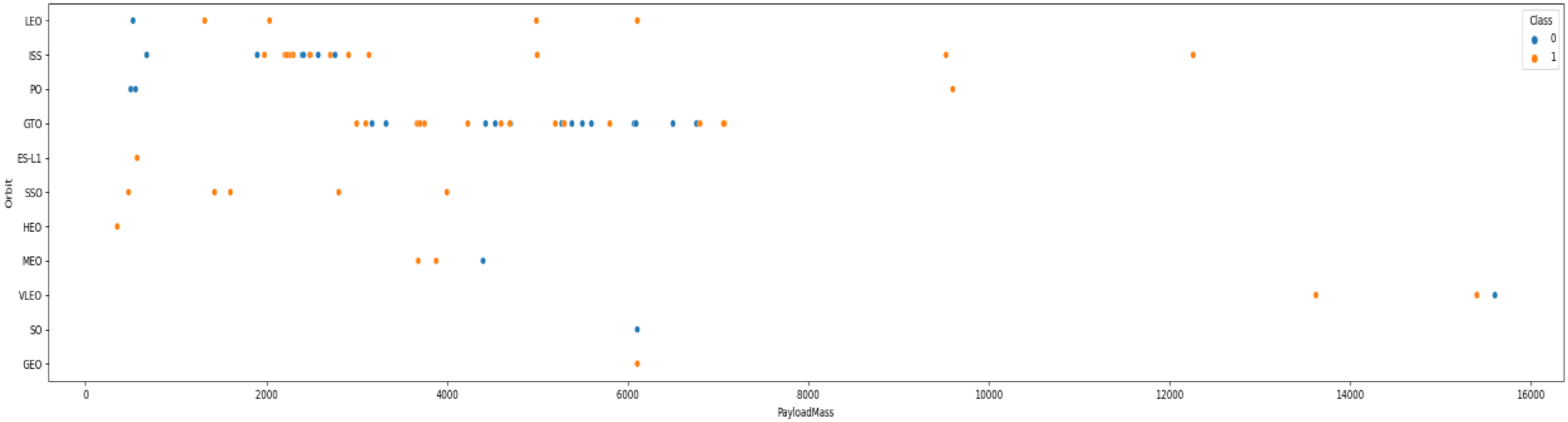
The only thing this tell us is there been no launch for SO orbit other then that orbit does not contribute to the success rate.

Flight Number vs. Orbit Type



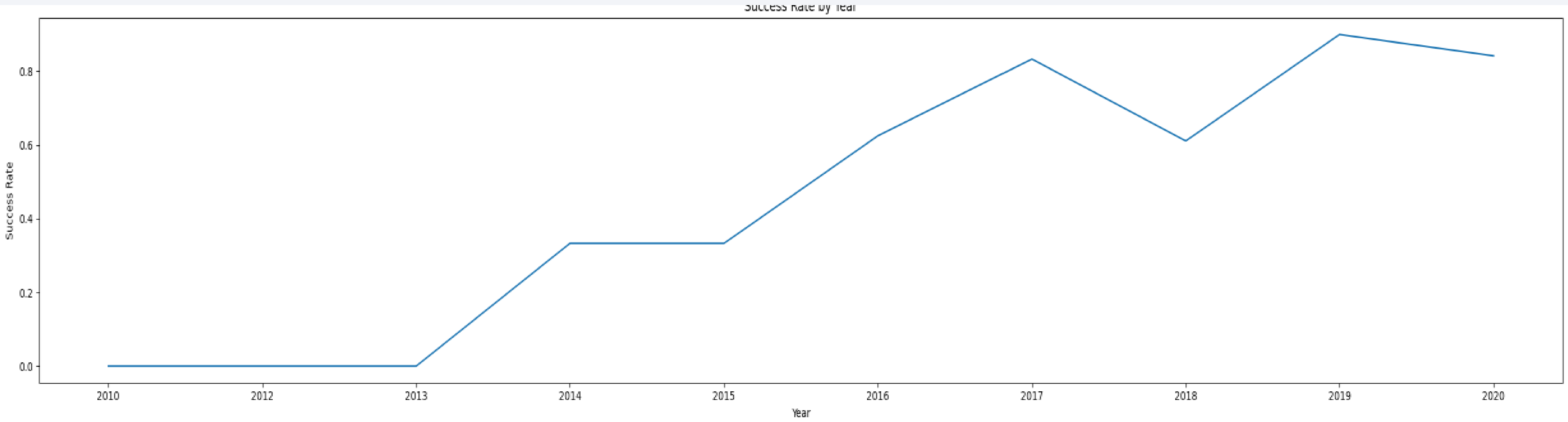
- SSO seems to have the highest success rate.

Payload vs. Orbit Type



- A payload mass over 9000 has a success in the orbits its been tested.

Launch Success Yearly Trend



- From 2013 to 2019 the trend of success rate improves with 2019 to 2020 being a slight decline in success.

All Launch Site Names

Task 1

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

```
n [12]: %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTBL
```

```
 * sqlite:///my_data1.db
Done.
```

ut[12]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

- Use DISTINCT to return only unique results from the database SPACEXTBL

Launch Site Names Begin with 'CCA'

Task 2

Display 5 records where launch sites begin with the string 'CCA'

In [13]:
```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

```
 * sqlite:///my_data1.db
Done.
```

Out[13]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

- To find sites that have the string `CCA` the % means to return it if it has any combination of the given string and use LIMIT to set how much results we want return.

Total Payload Mass

Task 3

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

[14]:
```
%sql SELECT SUM(Payload_Mass__kg_) AS Total_Payload_Mass FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

[14]:

Total_Payload_Mass
45596.0

- Using the sum() function to add the total payload mass where the customer was NASA (CRS)

Average Payload Mass by F9 v1.1

Task 4

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

```
[15]: %sql SELECT AVG(Payload_Mass__kg_) AS Average_Payload_Mass FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1'
```

```
 * sqlite:///my_data1.db
Done.
```

[15]:

Average_Payload_Mass
2928.4

- Using the avg() functioin to calculate the average payload mass for F9 v1.1

First Successful Ground Landing Date

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
18]: %sql SELECT MIN(Date) AS First_Successful_Landing_Date FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'
```

```
 * sqlite:///my_data1.db
Done.
```

18]:

First_Successful_Landing_Date
01/08/2018

- Using the min() function to find the lowest date for a successful ground pad

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

2]:
```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND Payload_Mass__kg_ > 4000 AND Payloa
```

```
 * sqlite:///my_data1.db
Done.
```

2]:

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

- This list all boosters that has success drone ship landing with a payload between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

[20]:
```
%sql SELECT Mission_Outcome, COUNT(*) AS Total_Outcomes FROM SPACEXTBL GROUP BY Mission_Outcome
```

```
 * sqlite:///my_data1.db
Done.
```

[20]:

Mission_Outcome	Total_Outcomes
None	898
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- This uses the count() function and the GROUP to add up every unique element in Mission_outcome

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery ¶

1]:
```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE Payload_Mass__kg_ = (SELECT MAX(Payload_Mass__kg_) FROM SPACEXTBL)
```

```
 * sqlite:///my_data1.db
Done.
```

1]:

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

- This list booster versions that have carried the maximum payload mass

2015 Launch Records

```
[46]: %sql SELECT CASE substr(Date, 4, 2) WHEN '01' THEN 'January' WHEN '02' THEN 'February' WHEN '03' THEN 'March' WHEN '04' THEN 'Apr
```

```
 * sqlite:///my_data1.db
Done.
```

t[46]:

Month_Name	Landing_Outcome	Booster_Version	Launch_Site
October	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- This show that OCT and APRAIL of 2015 had fail drone ship landing

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

Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%sql SELECT 'Landing_Outcome', COUNT(*) AS Count FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' AND Landing_Outc
```

```
 * sqlite:///my_data1.db
Done.
```

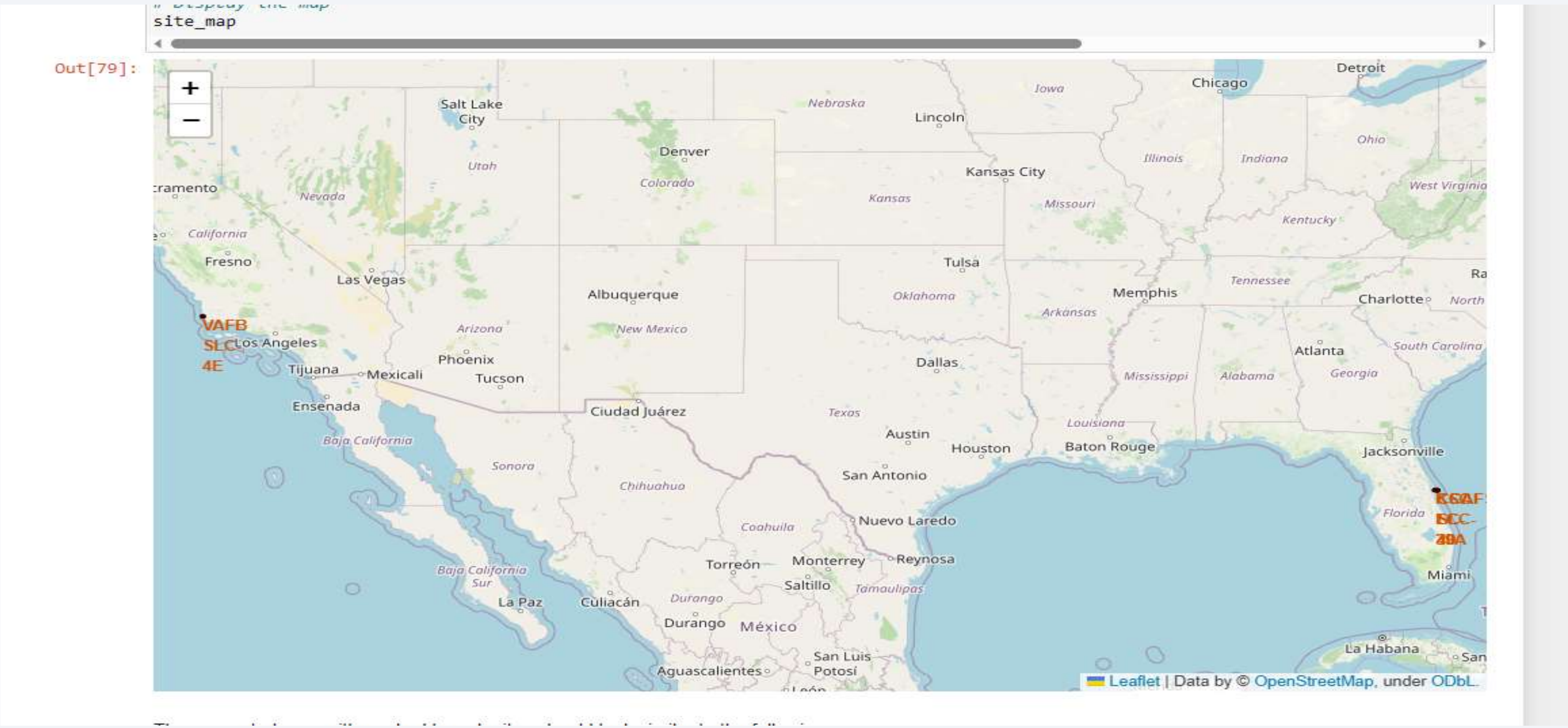
Landing_Outcome	Count

Couldn't get it to work

Section 3

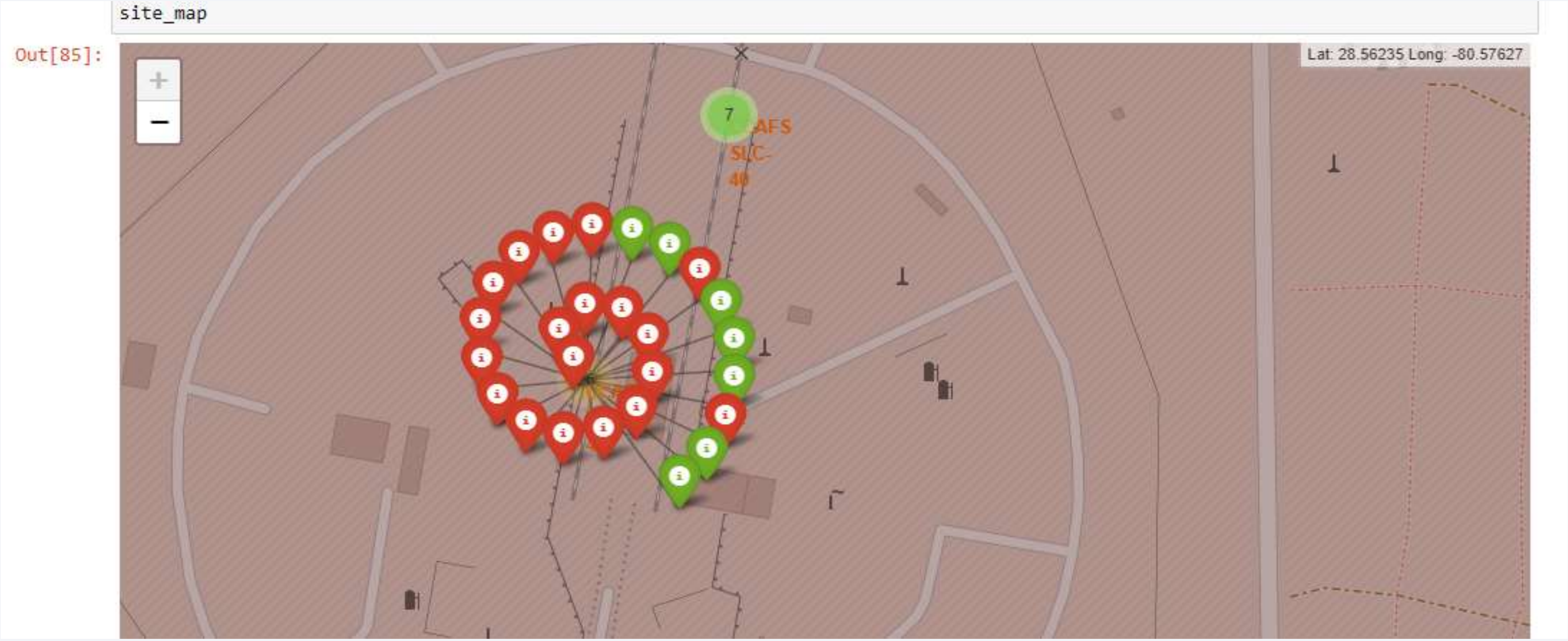
Launch Sites Proximities Analysis

Mark All Launch Sites on a Map



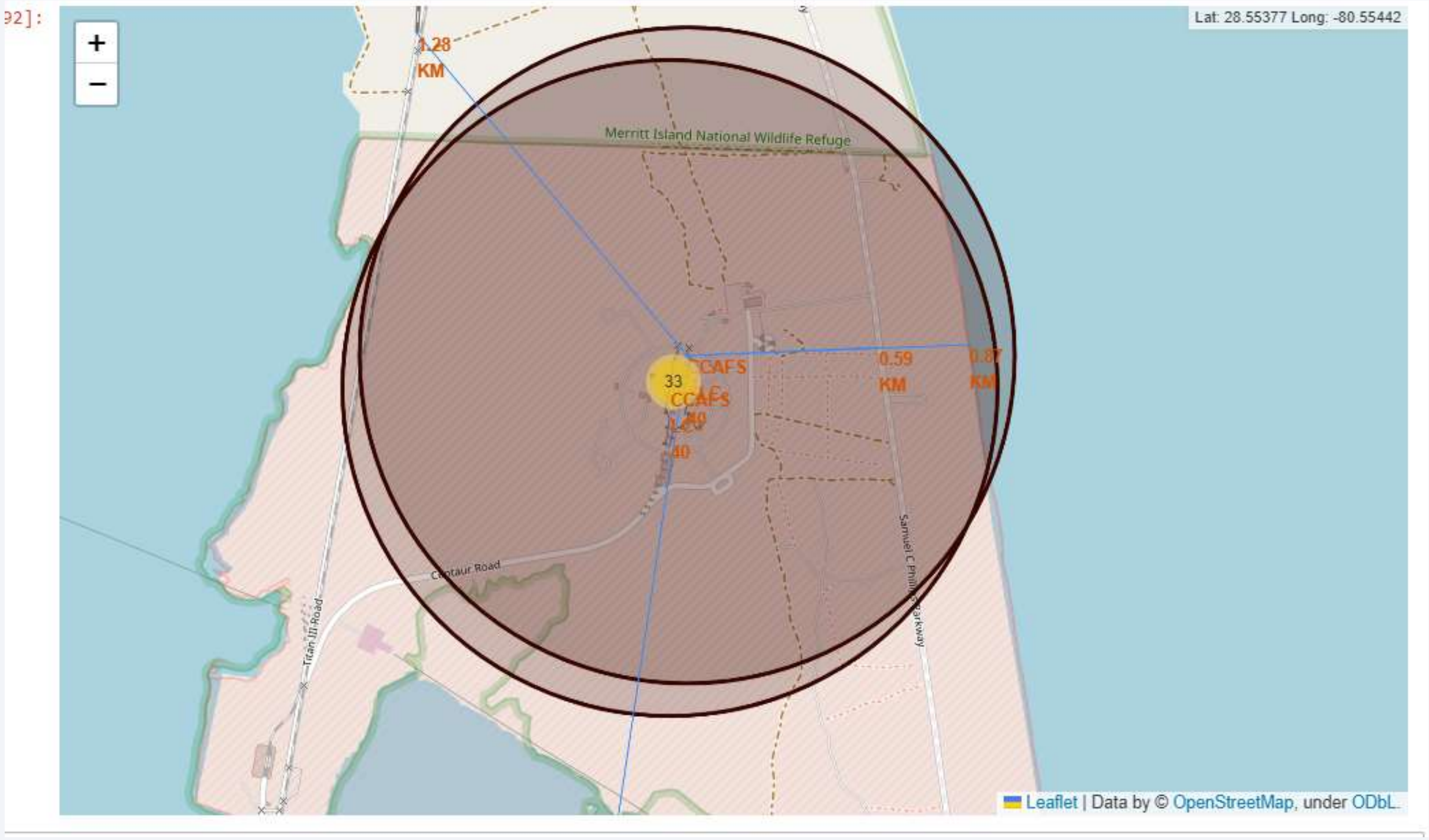
- This shows all launch sites we have been analyzing. We can see that California and Florida are the two states for launch sites

Mark the success/failed launches for each site



- We can see every site success and failure of a launch site by clicking on it

<Folium Map Screenshot 3>

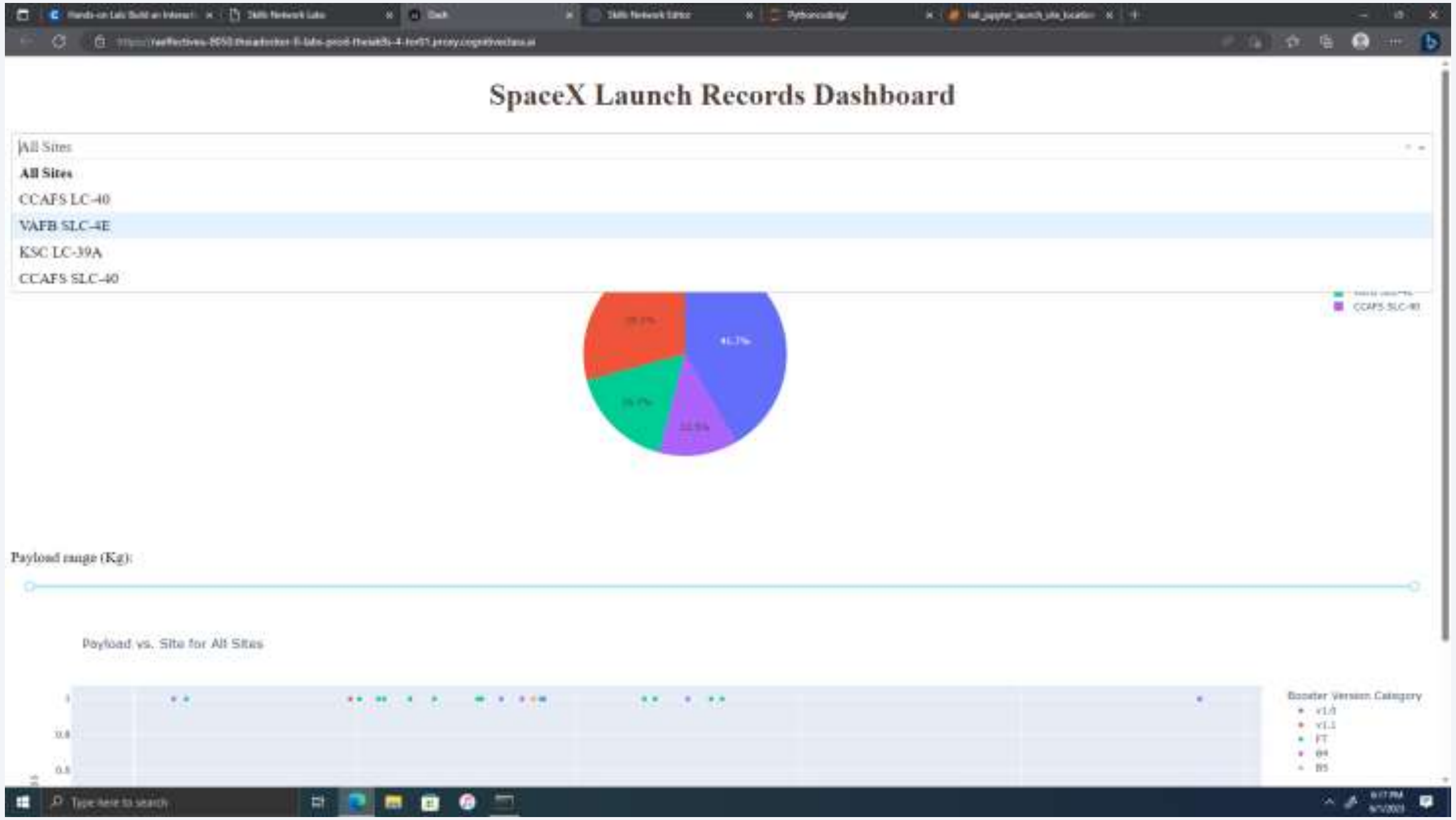


- Displays the closest coast at 0.87km, highway at 0.59km, railroad at 1.28 and the closest city 18.26 being Cape Canaveral

Section 4

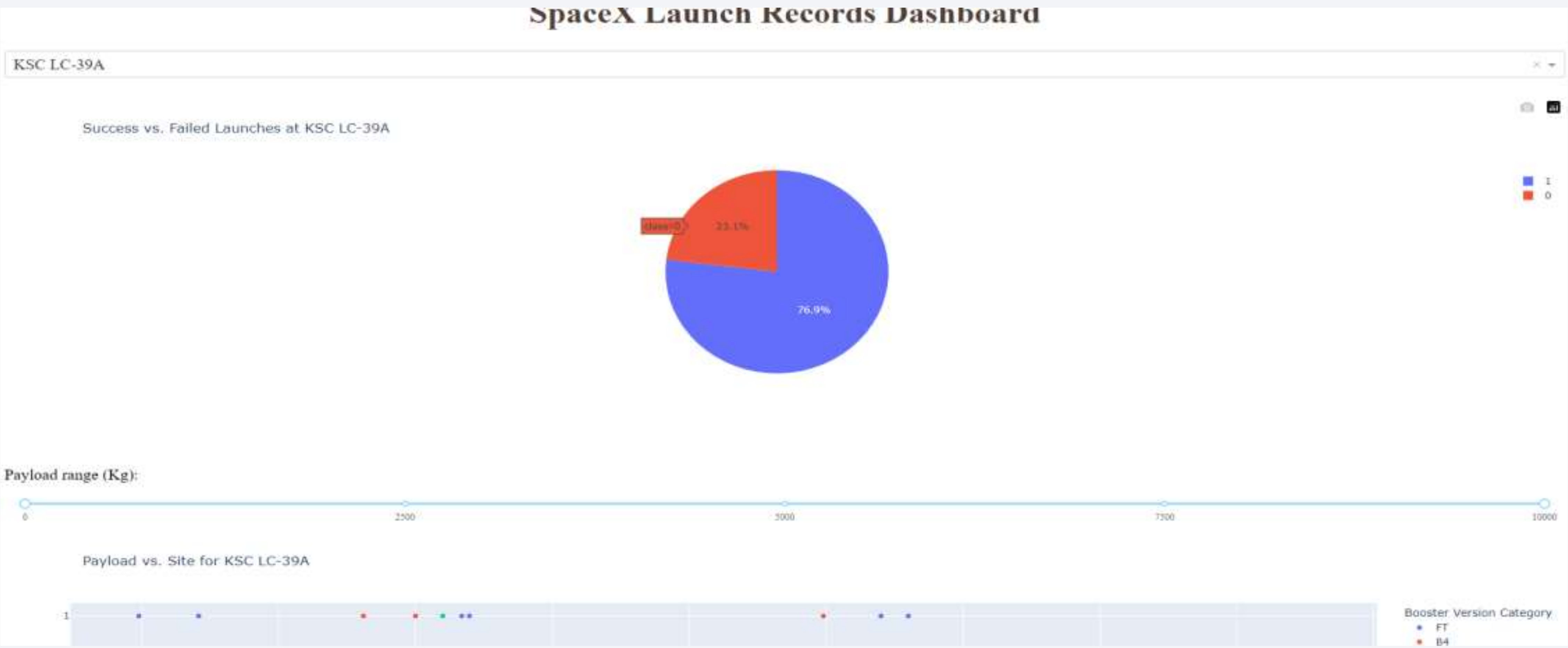
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



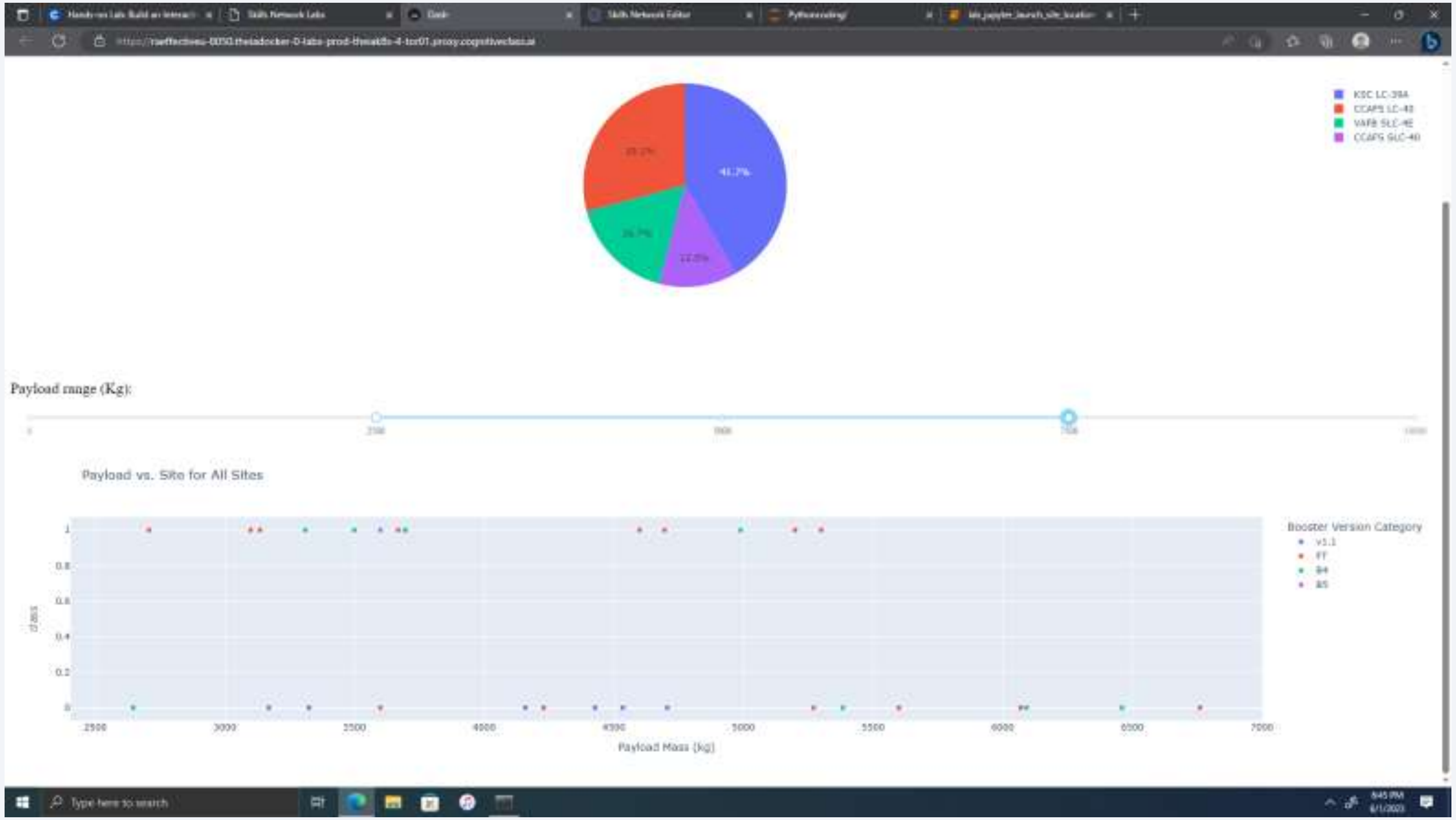
The pie chart show that combine success rate for all sites KSC LC-39A at 41.7% account for the highest

<Dashboard Screenshot 2>



- KSC LC-39A has the highest success at 76.9%

<Dashboard Screenshot 3>



- We can see a payload mass between5500 and 7500 has a Zero success rate and that FT and B4 perform the best between a payload mass of 2500and 5500

Section 5

Predictive Analysis (Classification)

Classification Accuracy

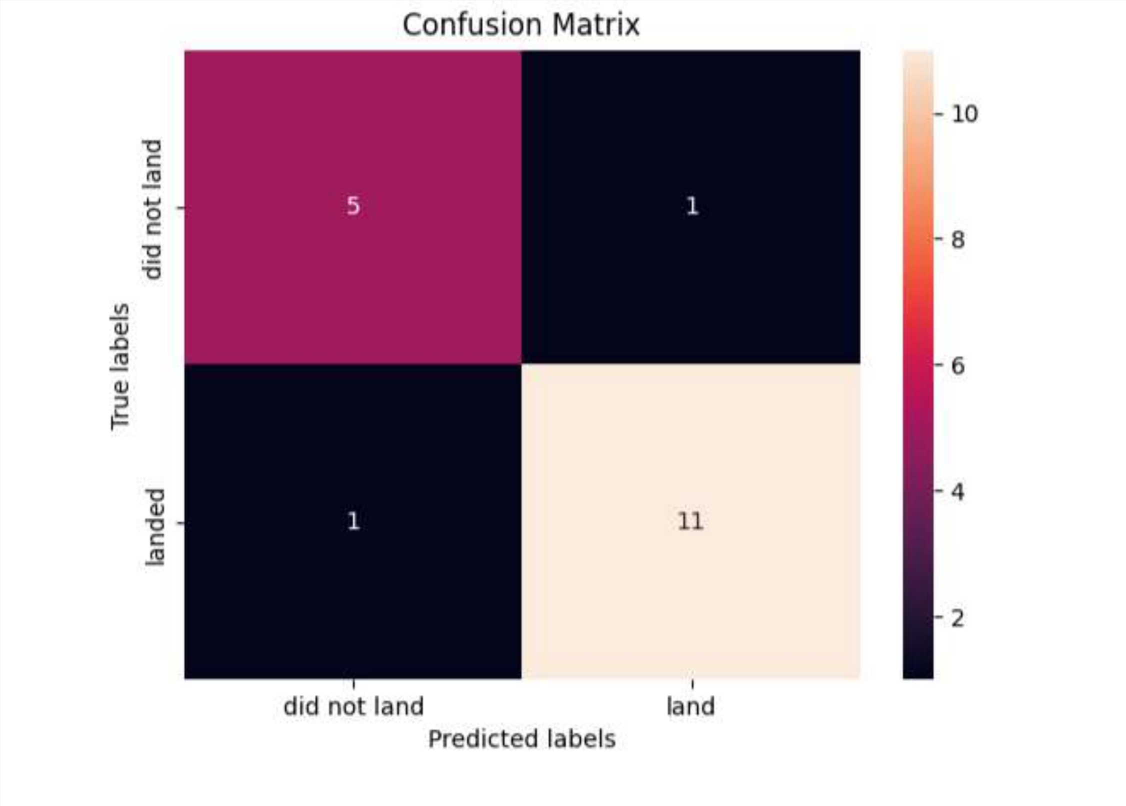
Find the method performs best:

In [35]:
```
# Calculate accuracy scores
lr_accuracy = logreg_cv.score(X_test, Y_test)
svm_accuracy = svm_cv.score(X_test, Y_test)
tree_accuracy = tree_cv.score(X_test, Y_test)
knn_accuracy = knn_cv.score(X_test, Y_test)

# Print accuracy scores
print("Accuracy on test data - Logistic Regression: ", lr_accuracy)
print("Accuracy on test data - SVM: ", svm_accuracy)
print("Accuracy on test data - Decision Tree: ", tree_accuracy)
print("Accuracy on test data - KNN: ", knn_accuracy)

# Find the method with the highest accuracy
best_method = max(lr_accuracy, svm_accuracy, tree_accuracy, knn_accuracy)

if best_method == lr_accuracy:
    print("Logistic Regression performs best.")
elif best_method == svm_accuracy:
    print("SVM performs best.")
elif best_method == tree_accuracy:
    print("Decision Tree performs best.")
else:
    print("KNN performs best.")
```

```
Accuracy on test data - Logistic Regression:  0.8333333333333334
Accuracy on test data - SVM:  0.8333333333333334
Accuracy on test data - Decision Tree:  0.8888888888888888
Accuracy on test data - KNN:  0.8333333333333334
Decision Tree performs best.
```

- The Decision Tree model has the highest classification accuracy at 83.33%

Confusion Matrix



- It shows we have 5 true negatives and 2 false positive and 11 true positives this shows that our model is accurate with only 2 false positive

Conclusions

- Falcon 9 version FT is the best booster
- With KSC LC-39A being the best site to launch at 76.9%
- Our payload mass should be between 2500 - 5500
- With the success rate of predictability reusable rocket at 88.33%

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!