IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Augusto Oliveira Silva

July 1st, 2024

Outline

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

Executive Summary

- Section 1 – Methodology
- Data Collection
  - Data Collection API
  - Data Collection Scraping
  - Data Wrangling
  - EDA with Data Visualization
  - EDA with SQL
  - Build a Interactive Map
  - Build a Dashboard
  - Predictive Analysis (Classification)
- Section 2 - Insights drawn from EDA
- Section 3 - Lauch Sites Proximities Analysis
- Section 4 - Dashboard (Plotly Dash)
- Section 5 - Predictive Analysis (Classification)

Introduction

There is a rocket company called Space Y that would like to compete with SpaceX founded by billionaire industrialist Allon Musk. The objective of this project is to determine the price of each release. This was gathering information about Space X and creating dashboards for his team. Additionally, the project determines whether SpaceX will reuse the first stage. Instead of using rocket science to determine whether the first stage will land successfully, a machine learning model was trained and used public information to predict whether SpaceX will reuse the first stage.

- Problems to find answers:
  - Determine whether the first stage will land successfully;
  - Determine the price of each release.

Section 1

Methodology

Methodology

Executive Summary

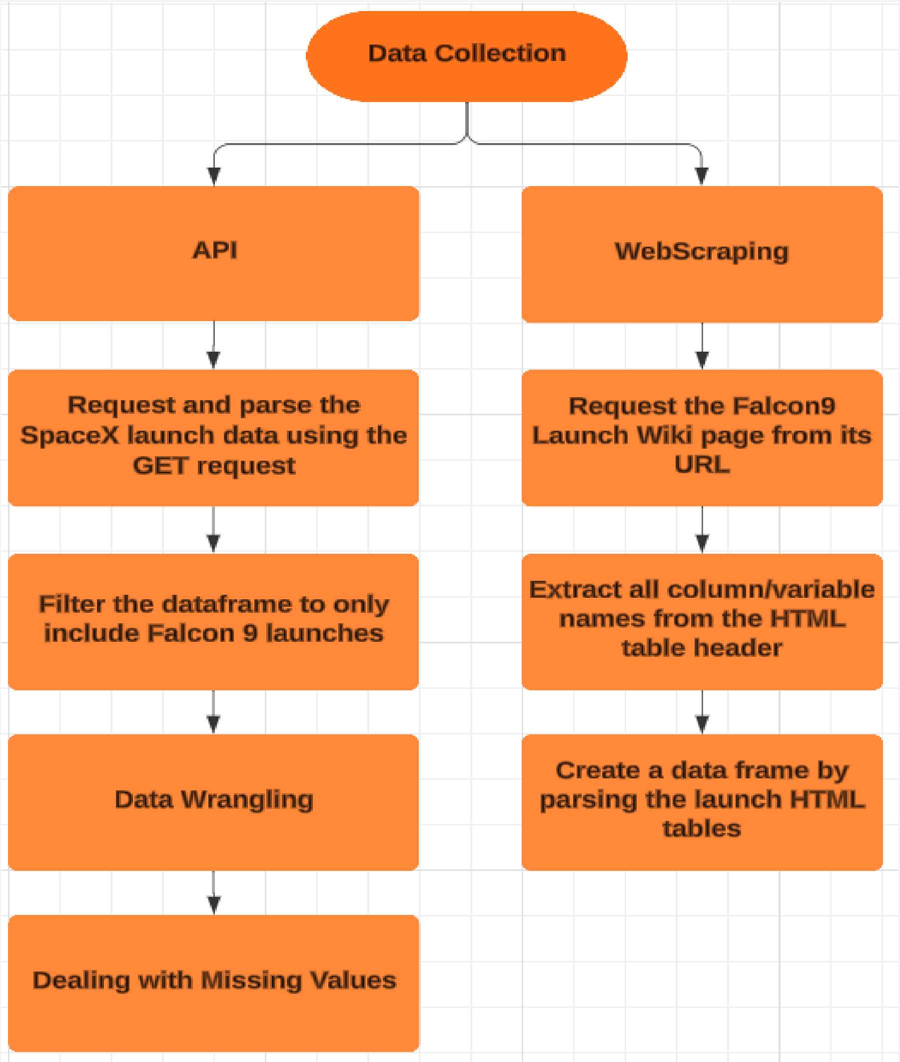
- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

I used two ways to collect the data.

1 - The first using an API requesting and analyzing SpaceX launch data using the GET request with JSON request and then specifying the desired columns of the data frame. Finally, I built the dataset using the data we obtained. We combine the columns into a dictionary;

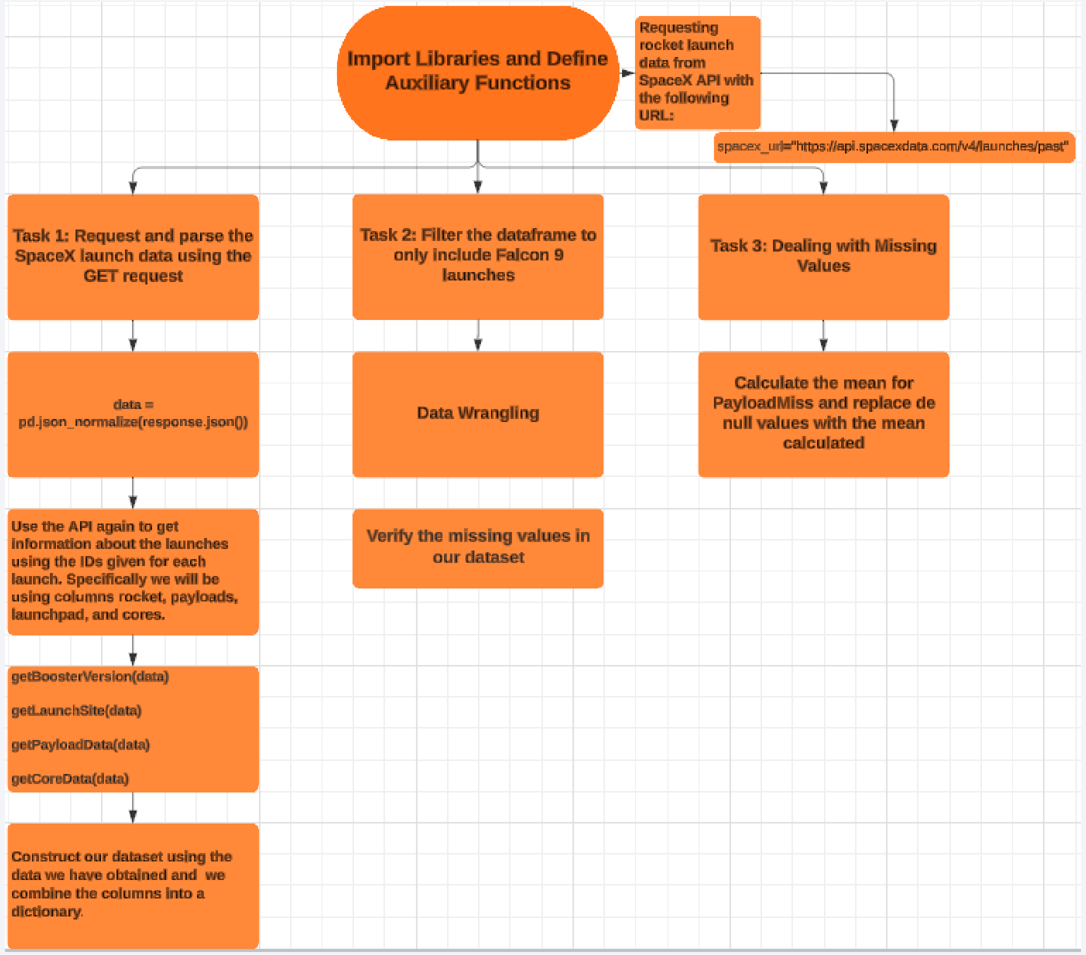
2- The second way was through webscraping with the release records with BeautifulSoup. After that, I requested the Wiki page with URL, extracted the column and variable names and created a data frame.



Data Collection – SpaceX API

- Import Libraries;
- Request and parse the data;
- Filter the dataframe;
- Dealing with missing values
- GitHub URL:

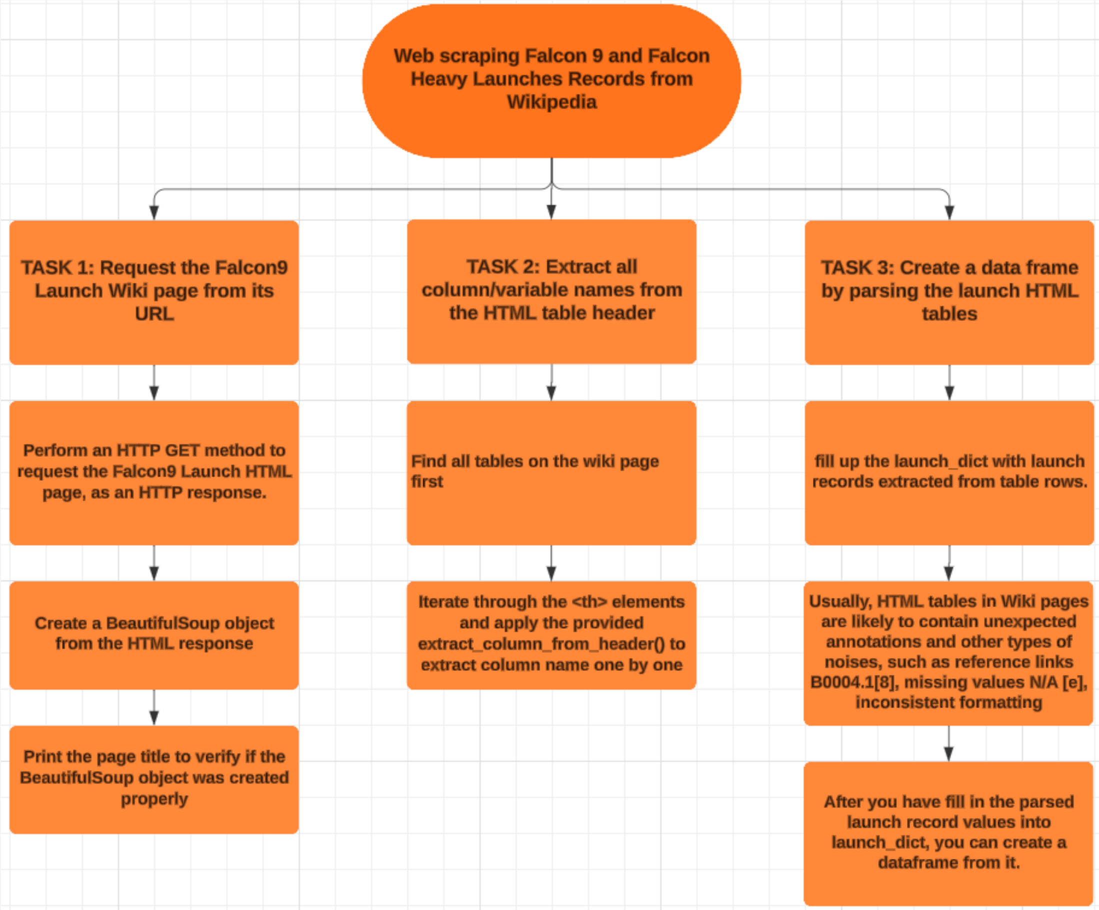
https://github.com/gut0oliveira/Applied-Data-Science-Capstone/blob/main/Data%20Collection/data-collection-api.ipynb



Data Collection - Scraping

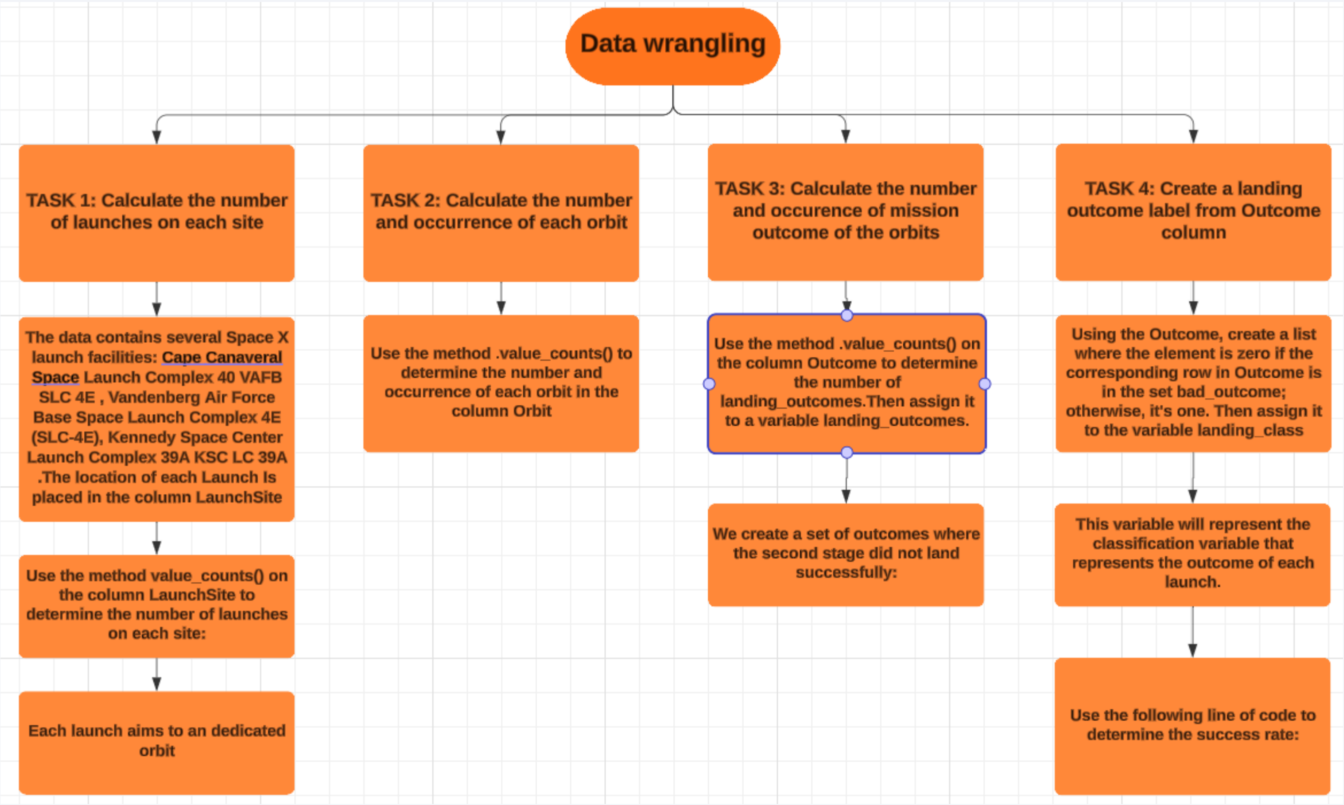
- Request the Wiki page;
- Extract all column/variable names;
- Create a dataframe;
- GitHub URL:

https://github.com/gut0oliveira/Applied-Data-Science-Capstone/blob/main/Data%20Collection/webscraping.ipynb



Data Wrangling

- Calculate the number of launches on each site;
- Calculate the number and occurrence of each orbit;
- Calculate the number and occurence of mission outcome of the orbits;
- Create a landing outcome label from Outcome column;
- GitHub URL: https://github.com/gut0oliveira/Applied-Data-Science-Capstone/blob/main/Data%20Wrangling/data-wrangling.ipynb



EDA with Data Visualization

Several graphs were used, including:

- Scatter Plot: Mainly used to observe relationships between different columns in the dataset;
- Bar Chart: Used to observe the relationship between orbit type and success rate in each specific orbit;
- Category Plot: Used to observe the relationship between payload mass for each orbit;
- Line Plot: Used to observe the variation in success rate over the years.
- GitHub URL: https://github.com/gut0oliveira/Applied-Data-Science-Capstone/blob/main/Exploratory%20Data%20Analysis%20(EDA)/eda-pandas-matplotlib.ipynb

EDA with SQL

- Database Setup
  - %sql sqlite:///my_data1.db
  - Created necessary tables and defined primary keys
- Display the names of the unique launch sites
- Display 5 records where launch sites begin with the string 'CCA'
- 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 when the first succesful landing outcome in ground pad was acheived.
- 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 carried the maximum payload mass
- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- GitHub URL: https://github.com/gut0oliveira/Applied-Data-Science-Capstone/blob/main/Exploratory%20Data%20Analysis%20(EDA)/eda-sql.ipynb

Build an Interactive Map with Folium

I added markers and circular objects that showed the exact launch locations. I also added green and red markers, where the green markers represented successful launches, while the red markers represented unsuccessful launches. Finally, I also added lines that showed and calculated the distance between the launch points and the sea coast, etc. These objects were added because they are frequently used for visual analysis. Plotting maps is extremely important when it comes to data, especially in the case of rockets and a large company.

- GitHub URL: https://github.com/gut0oliveira/Applied-Data-Science-Capstone/blob/main/Visual%20Analysis%20and%20Dashboards/launch_site_location.ipynb

Build a Dashboard with Plotly Dash

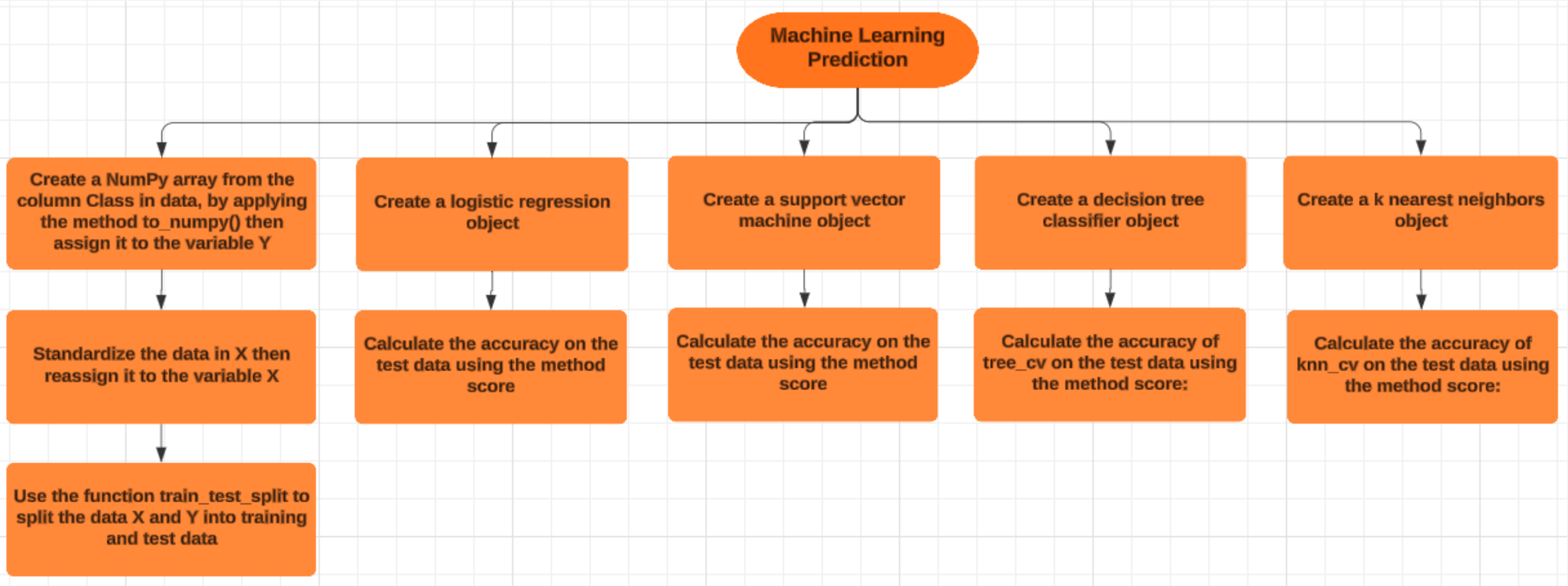
Pie charts and a scatter chart have been added to the dashboard. Pie chart was added and showed the percentage of successful launches of all engine types or showed just one engine chosen by the user from the dropdown menu. The scatterplot showed the percentage of successful launches, but with an additional payload mass range that the user could modify this range according to the analysis they would like to do.

- GitHub URL: https://github.com/gut0oliveira/Applied-Data-Science-Capstone/blob/main/Visual%20Analysis%20and%20Dashboards/spacex_dash_app.py

Predictive Analysis (Classification)

- Creating Logistic Regression Calculating accuracy
- SVM creation Calculating accuracy
- Decision tree creation Calculating accuracy
- KNN Creation Calculating accuracy
- Calculating the best result between models

- GitHub URL: https://github.com/gut0oliveira/Applied-Data-Science-Capstone/blob/main/Machine%20Learning%20Prediction/SpaceX_Machine-Learning-Prediction.ipynb

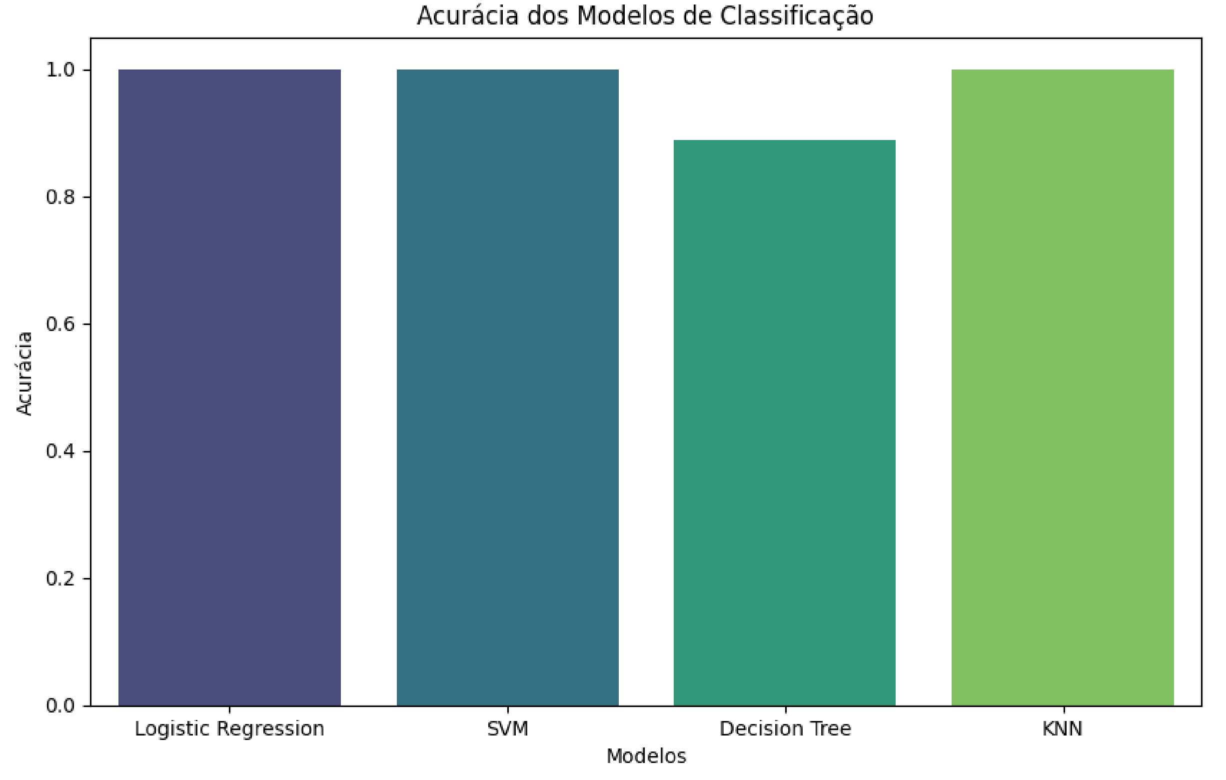


Results

Exploratory data analysis results

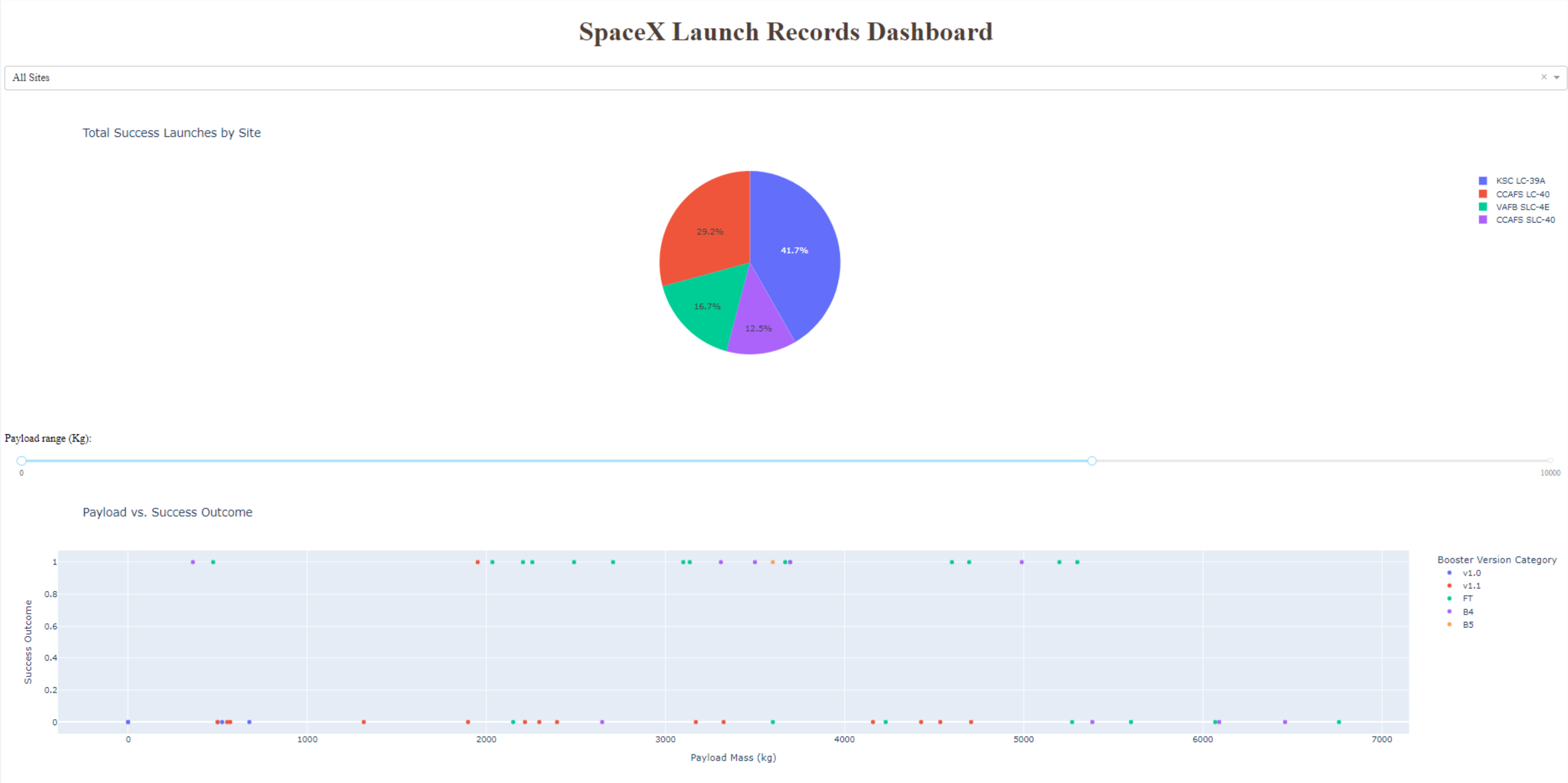
```
   FlightNumber  Date BoosterVersion  PayloadMass      Outcome  Flights  \
0           1.0  2010       Falcon 9  6104.959412    None None      1.0
1           2.0  2012       Falcon 9   525.000000    None None      1.0
2           3.0  2013       Falcon 9   677.000000    None None      1.0
3           4.0  2013       Falcon 9   500.000000  False Ocean      1.0
4           5.0  2013       Falcon 9  3170.000000    None None      1.0

   GridFins  Reused  Legs  Block  ...  Serial_B1048  Serial_B1049  \
0       0.0     0.0   0.0    1.0  ...         False         False
1       0.0     0.0   0.0    1.0  ...         False         False
2       0.0     0.0   0.0    1.0  ...         False         False
3       0.0     0.0   0.0    1.0  ...         False         False
4       0.0     0.0   0.0    1.0  ...         False         False

   Serial_B1050  Serial_B1051  Serial_B1054  Serial_B1056  Serial_B1058  \
0         False         False         False         False         False
1         False         False         False         False         False
2         False         False         False         False         False
3         False         False         False         False         False
4         False         False         False         False         False

   Serial_B1059  Serial_B1060  Serial_B1062
0         False         False         False
1         False         False         False
2         False         False         False
3         False         False         False
4         False         False         False

[5 rows x 86 columns]
```

Predictive analysis results



Results

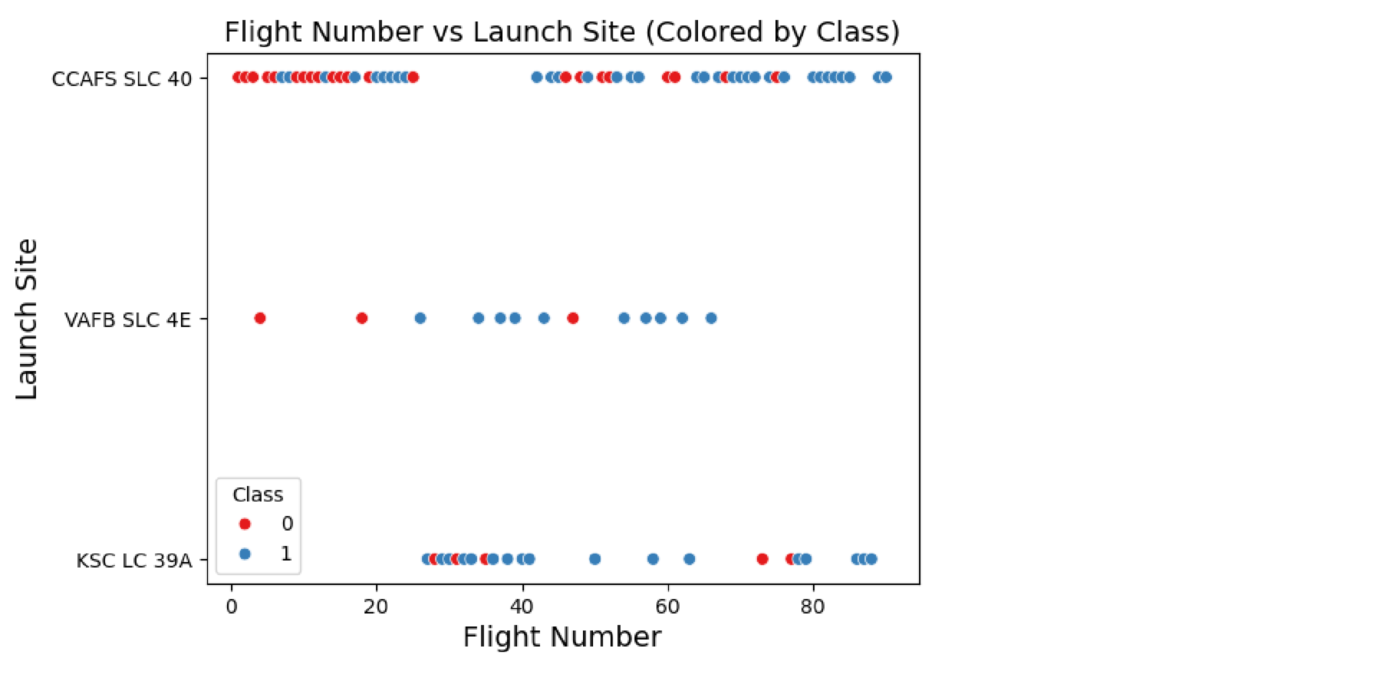
Interactive analytics demo in screenshots

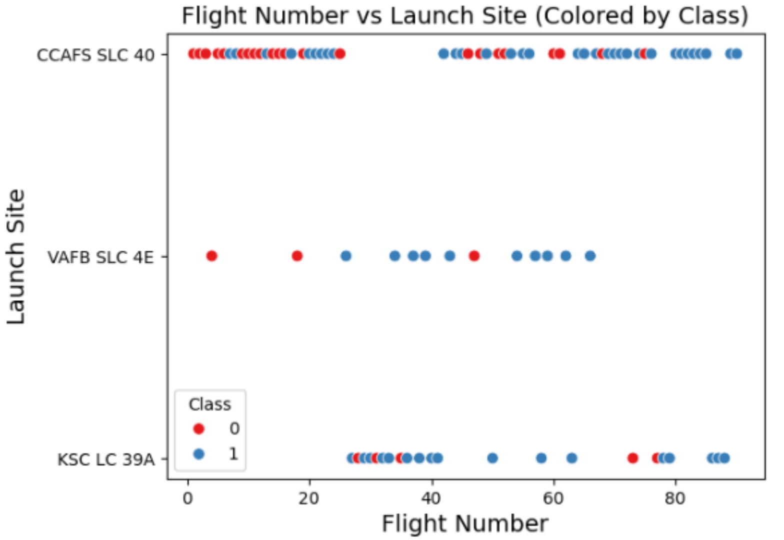


Section 2

Insights drawn from EDA

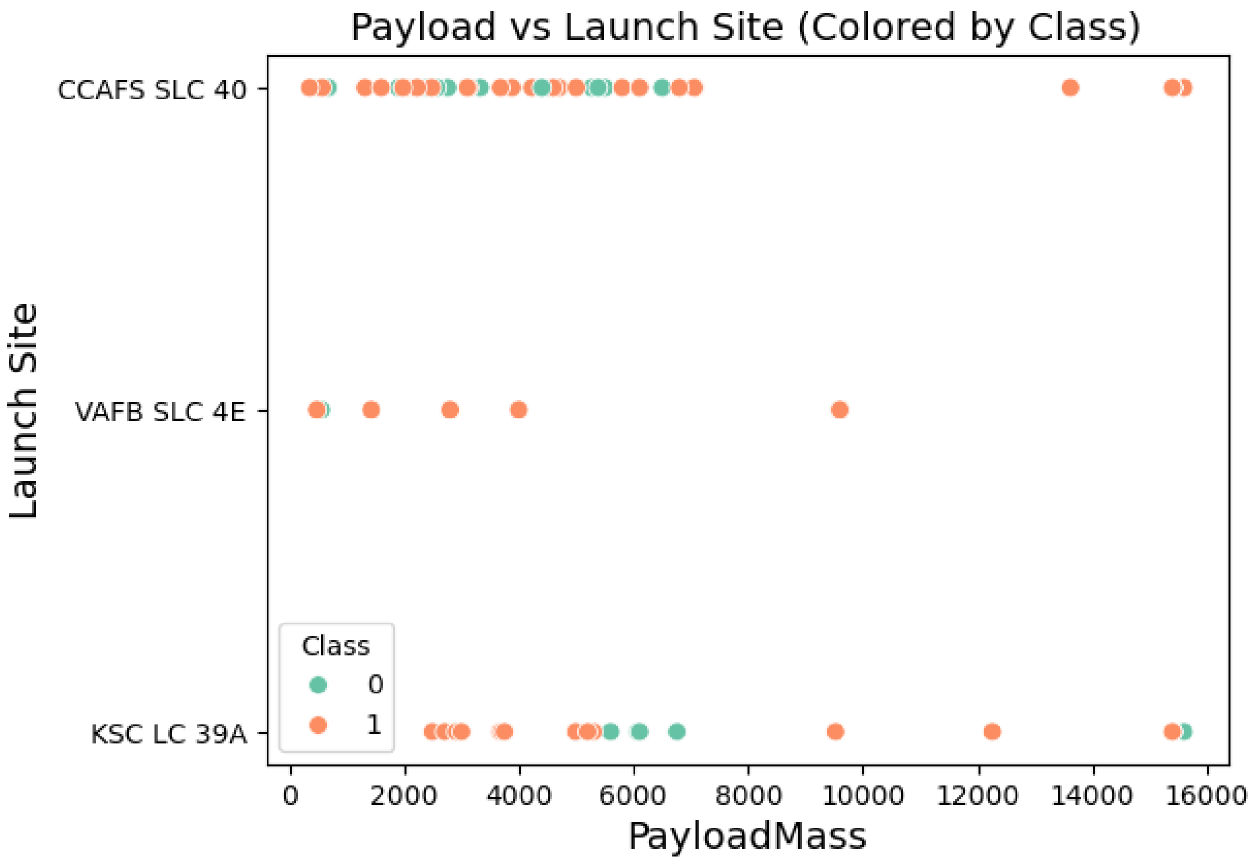
Flight Number vs. Launch Site





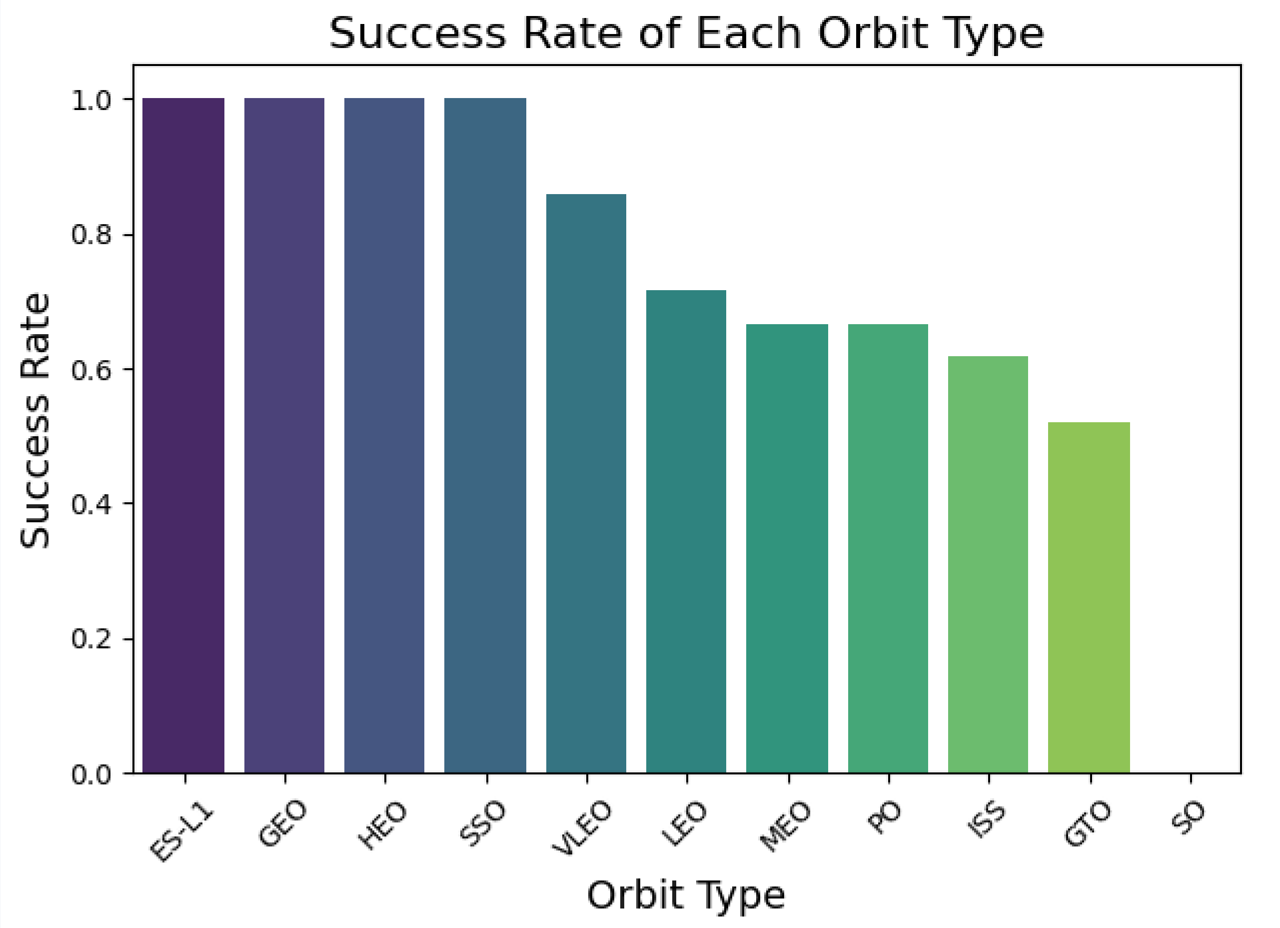

```
Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.
Flight Number Grouping:
Clustering of points within certain flight number ranges suggests preferred launch sites for specific flight numbers.

Temporal Trends:
Increasing flight numbers over time may show improved success rates at specific launch sites.
Indicates whether the space company is improving operations over time.

Risk Analysis:
Launch sites with high failure concentrations can be identified as higher risk, informing future strategies and operations.

Operational Efficiency:
Launch sites with high success rates can be deemed more efficient or favorable regarding launch conditions and infrastructure.

Correlation Analysis:
Analysis may reveal significant correlations between launch sites and flight success, identifying factors influencing launch success.
```

Payload vs. Launch Site



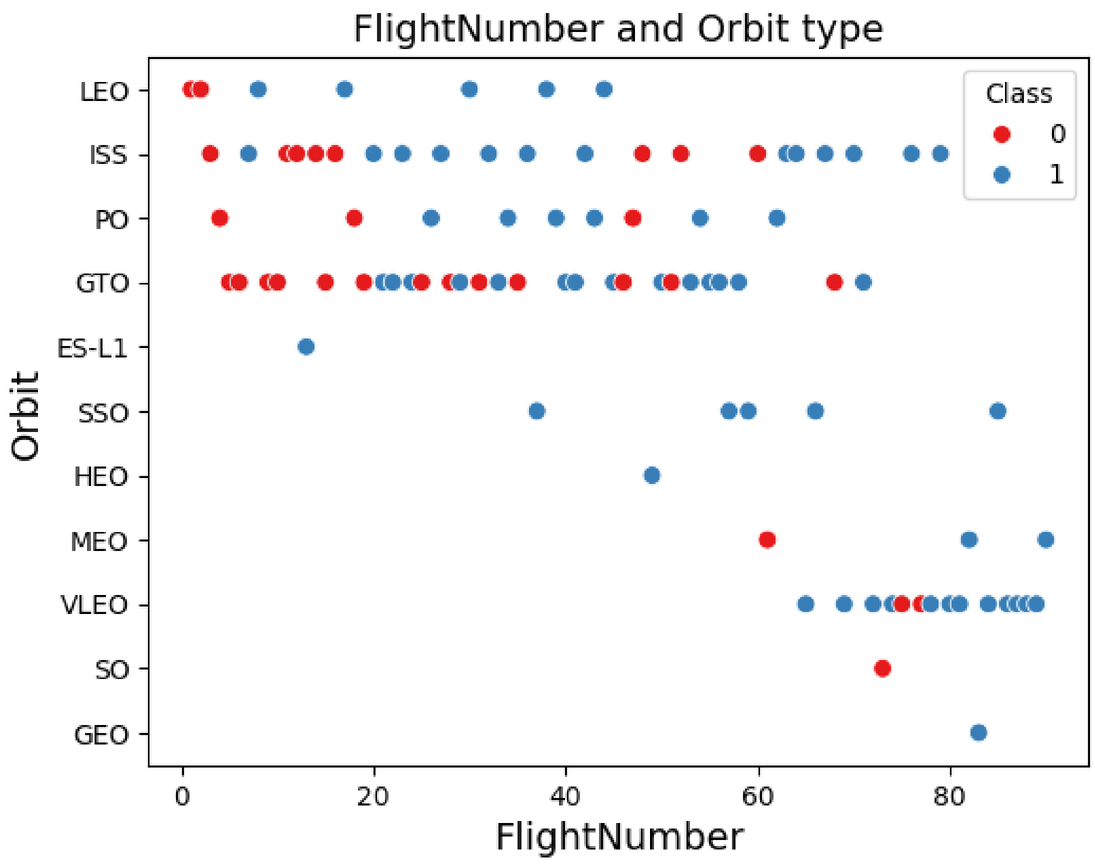
Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type



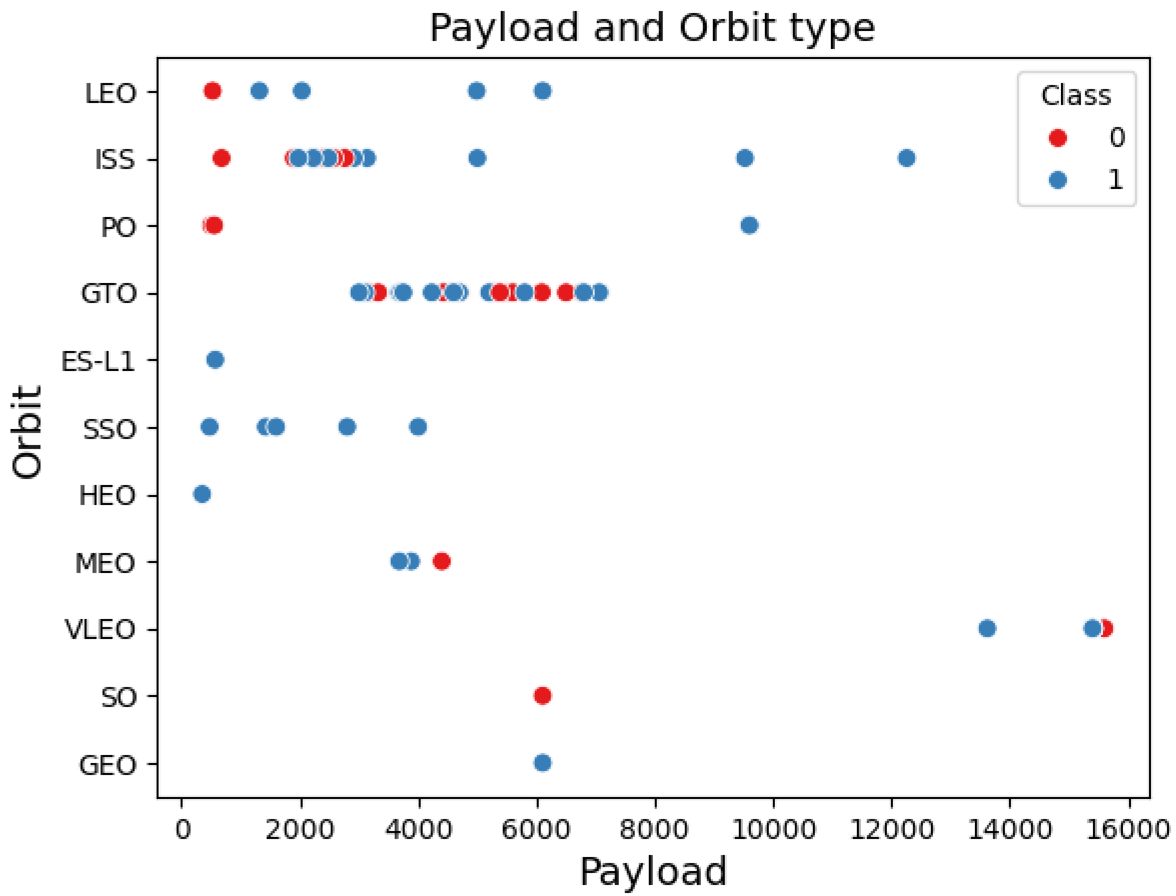
As we can see from the bar chart, the ES-L1, GEO, HEO, and SSO orbits achieved 100% success, while the other orbits did not reach 1.0. We can also observe that the SO orbit did not achieve any success, remaining at 0.

Flight Number vs. Orbit Type



You should see that 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.

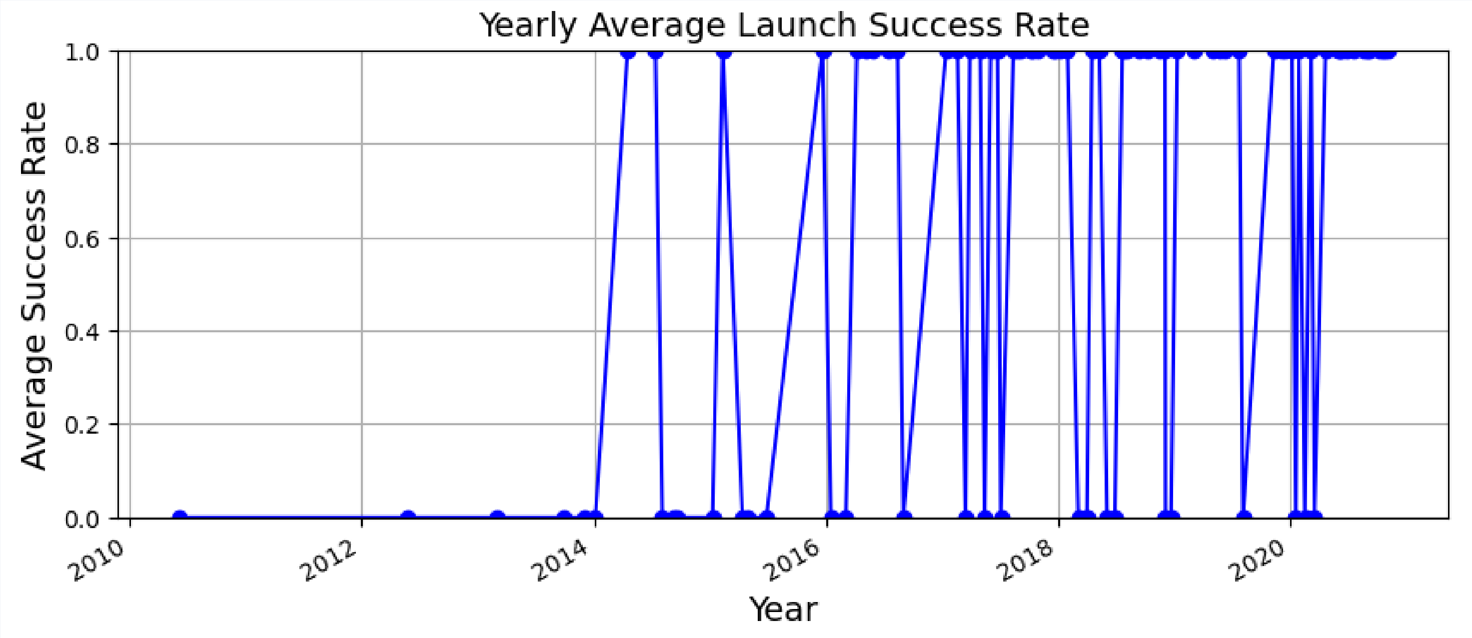
Payload vs. Orbit Type



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

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



The graph shows the percentage of success over the years. As we can see, the average rose to 100% from 2014 onwards and since then there have been falls and rises. I made a mistake in the measurement but I tried to do it the best I could.

All Launch Site Names

```
unique_launch_sites = df['LaunchSite'].unique()
print(unique_launch_sites)
```

```
['CCAFS SLC 40' 'VAFB SLC 4E' 'KSC LC 39A']
```

df['LaunchSite'].unique(): This function returns an array with the unique values from the 'LaunchSite' column of the dataframe. It lists all the unique launch site names.

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
```

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

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

SELECT DISTINCT "Launch_Site": This function returns an table with the unique values from the 'LaunchSite' column of the dataframe. It lists all the unique launch site names.

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5
```

%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5: This function returns the first 5 values from the "Launch_Site" column of the data frame that begins with the letters 'CCA'.

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE
```

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

sum(PAYLOAD_MASS__KG_)
619967

%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE: This function returns an table with the total payload mass from the 'PAYLOAD_MASS__KG_' column of the data frame.

Average Payload Mass by F9 v1.1

```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1': This code displays the average payload mass carried by the F9 v1.1 booster version.

First Successful Ground Landing Date

```
%sql select min(Date) from SPACEXTABLE where Landing_Outcome like "Success%"
```

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

min(Date)
2015-12-22

%sql select min(Date) from SPACEXTABLE where Landing_Outcome like "Success%": This code displays the date when the first succesful landing outcome in ground pad was achieved.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select Booster_Version from SPACEXTABLE where Landing_Outcome like '%drone ship%' and PAYLOAD_MASS__KG_ between 4000 and 6000
```

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

Booster_Version
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

%sql select Booster_Version from SPACEXTABLE where Landing_Outcome like '%drone ship%' and PAYLOAD_MASS__KG_ between 4000 and 6000: This code displays the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.

Total Number of Successful and Failure Mission Outcomes

```
%sql select count(Mission_Outcome) from SPACEXTABLE
```

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

count(Mission_Outcome)
101

%sql select count(Mission_Outcome) from SPACEXTABLE: This code displays the total number of successful and failure mission outcomes.

Boosters Carried Maximum Payload

```
%sql select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

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

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

%sql select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE): This code displays the names of the booster_versions which have carried the maximum payload mass.

2015 Launch Records

```
%sql select substr(Date, 6, 2) as month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE substr("Date", 0, 5) = '2015' AND "Landing_Outcome" = 'Failure (drone ship)'
```

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

month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

%sql select substr(Date, 6, 2) as month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE substr("Date", 0, 5) = '2015' AND "Landing_Outcome" = 'Failure (drone ship)': This code displays the names of the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

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

```
%sql SELECT Count(Landing_Outcome) FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' AND Landing_Outcome = 'Failure (drone ship)' or 'Success (ground pad)' ORDER BY La
```

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

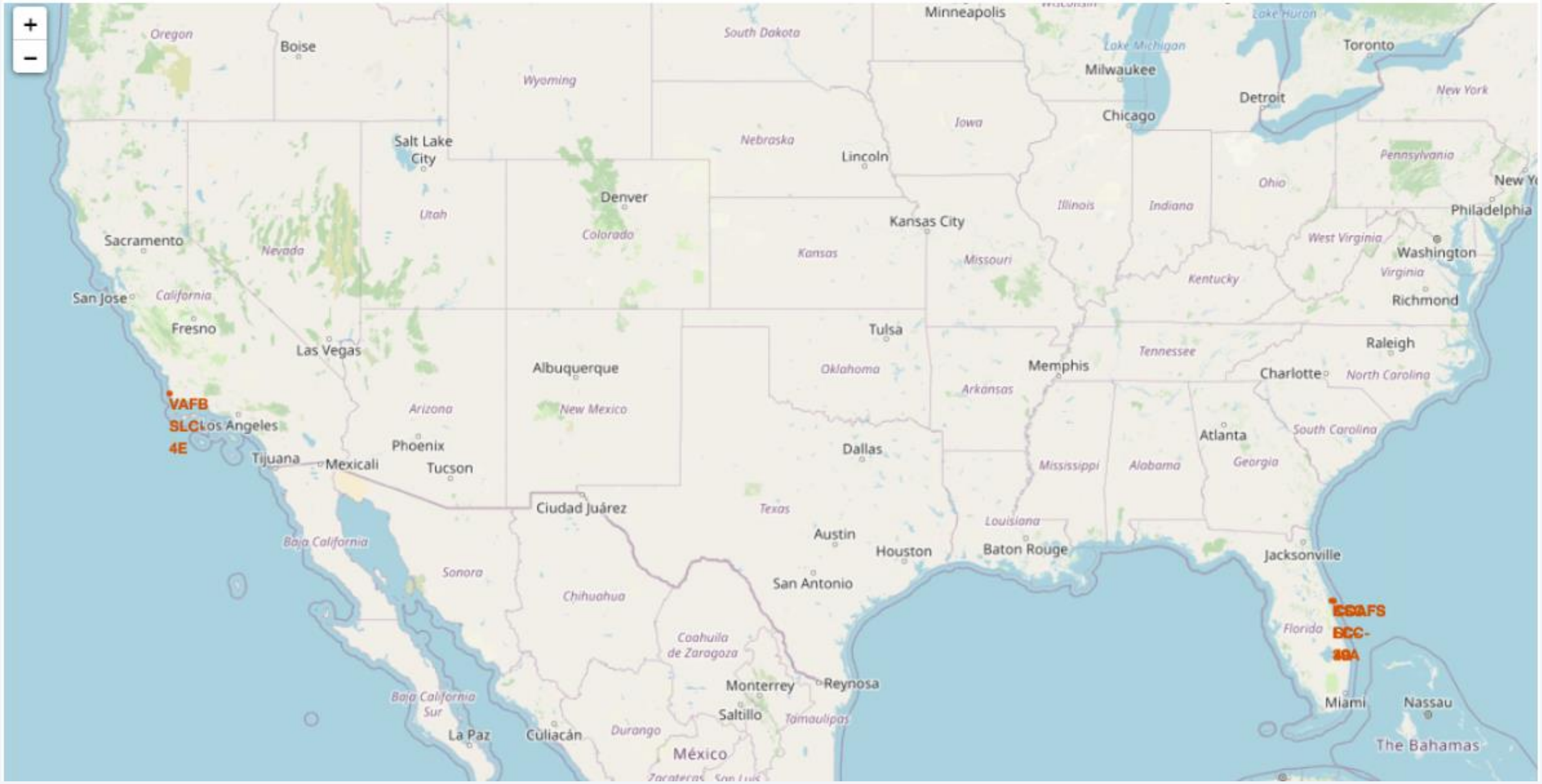
Count(Landing_Outcome)
5

%sql SELECT Count(Landing_Outcome) FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' AND Landing_Outcome = 'Failure (drone ship)' or 'Success (ground pad)' ORDER BY Landing_Outcome DESC: This code Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

Section 3

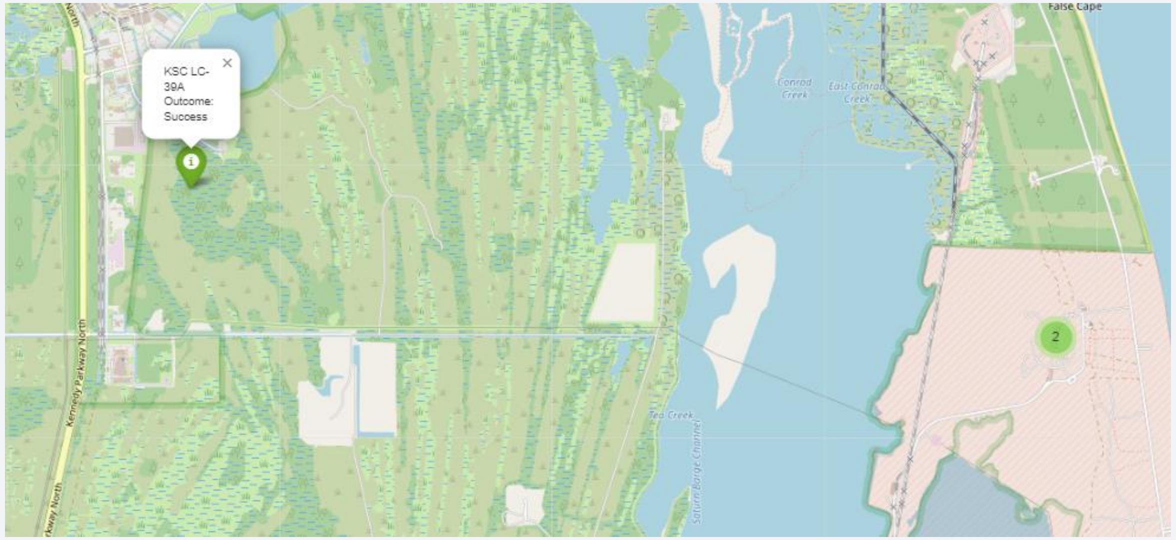
Launch Sites Proximities Analysis

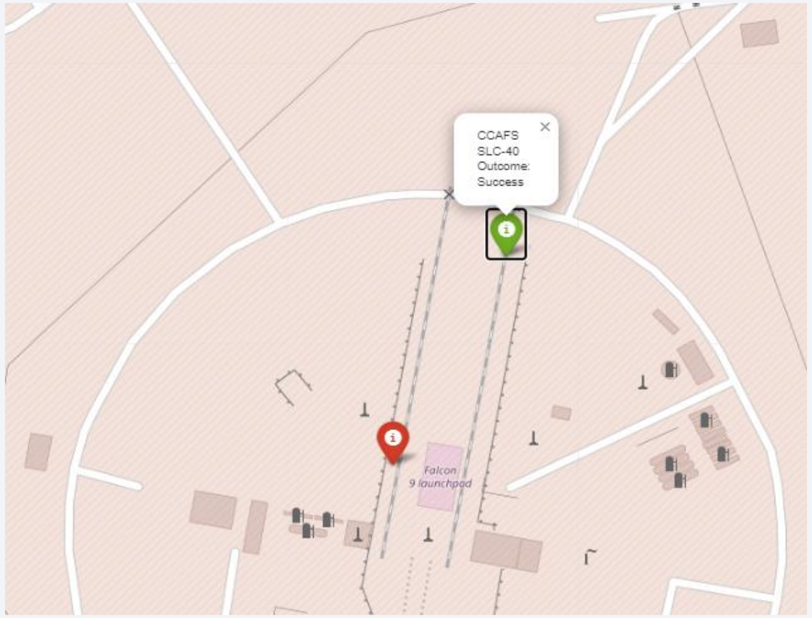
All launch sites' location markers on a global map



On this map we can see all the launch points. The dots are in orange.nt elements and findings on the screenshot.

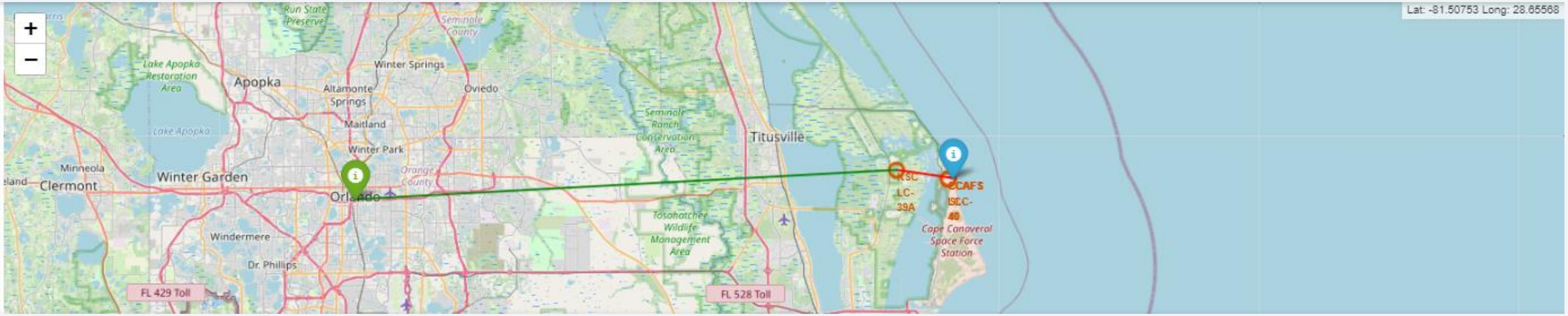
Results of launches marked by colors





These images show the results of launches marked by colors (red and green)

Selected launch site to its proximities

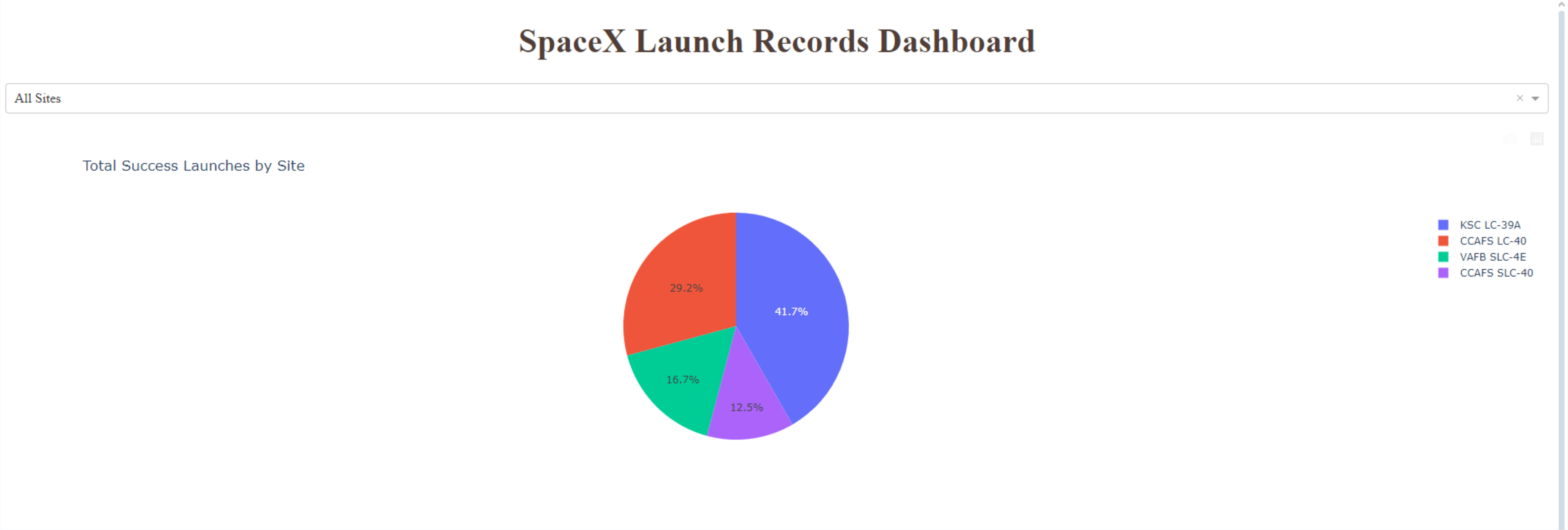


On this map we can see the launch points marked in orange. We were also able to see the marking between a launch point and the city of Orlando. By clicking on the Orlando city marker, you can see the distance between the launch point and the city.

Section 4

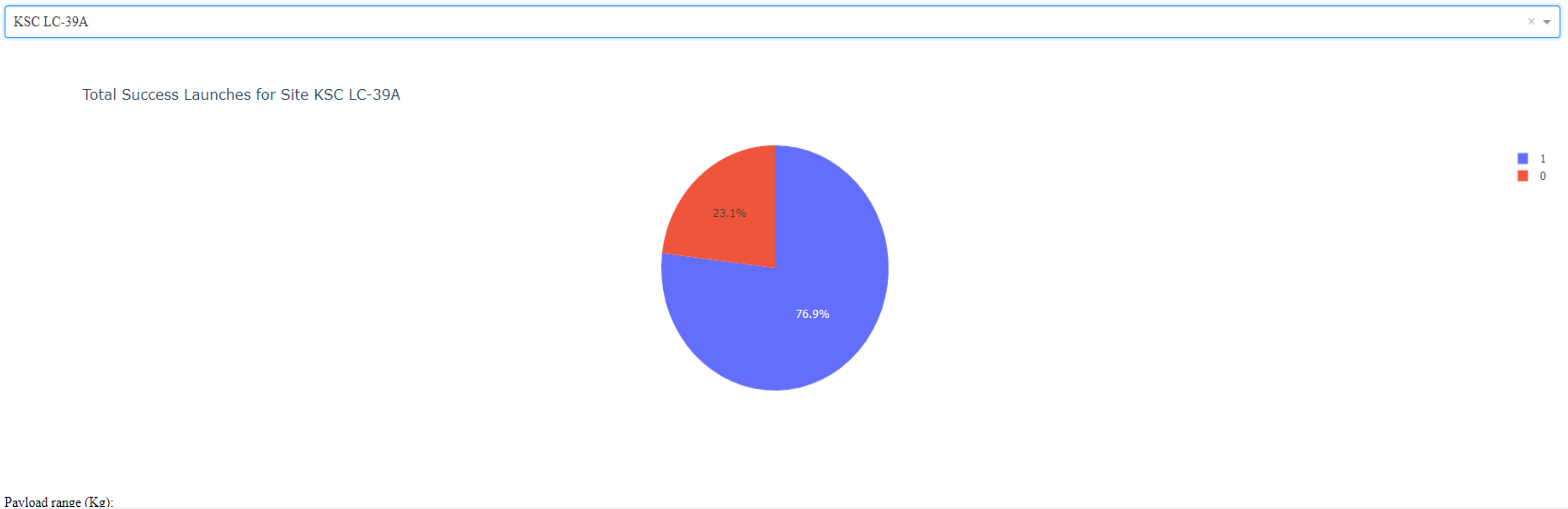
Build a Dashboard with Plotly Dash

Launch Success for all Sites



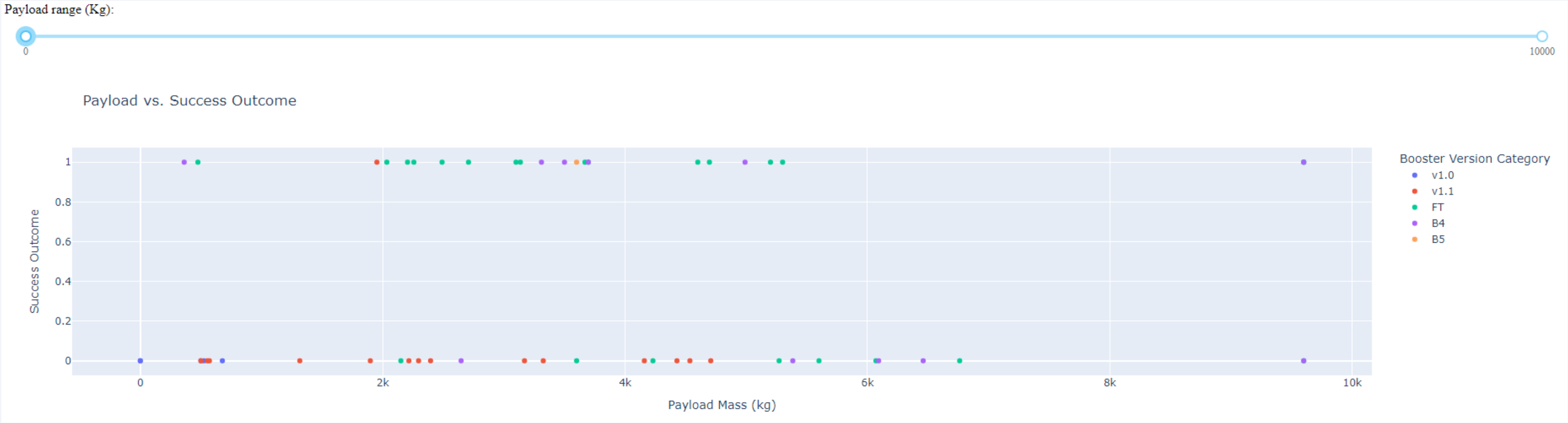
In this image, we can see the percentage of successful launches for all available engine types and we can see that the "KSC LC-39A" local is the local with the highest percentage of successful launches.

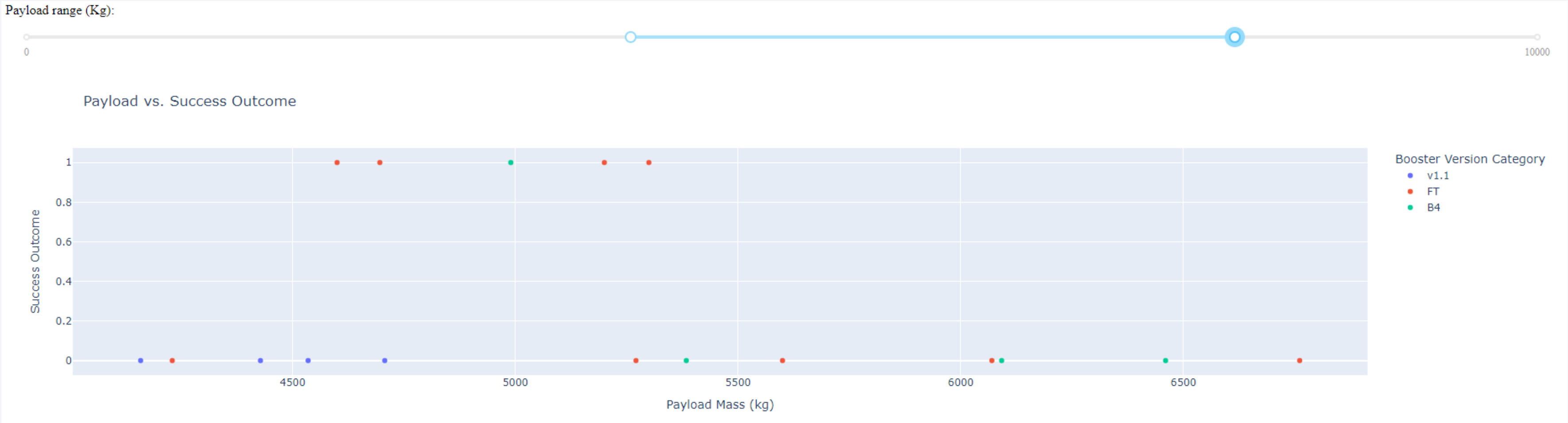
Highest Launch Success Ratio



In this image we see specifically the engine with the highest percentage of success in its launches. This local is the KSC LC-39A.

Payload vs. Launch Outcome in Different Ranges



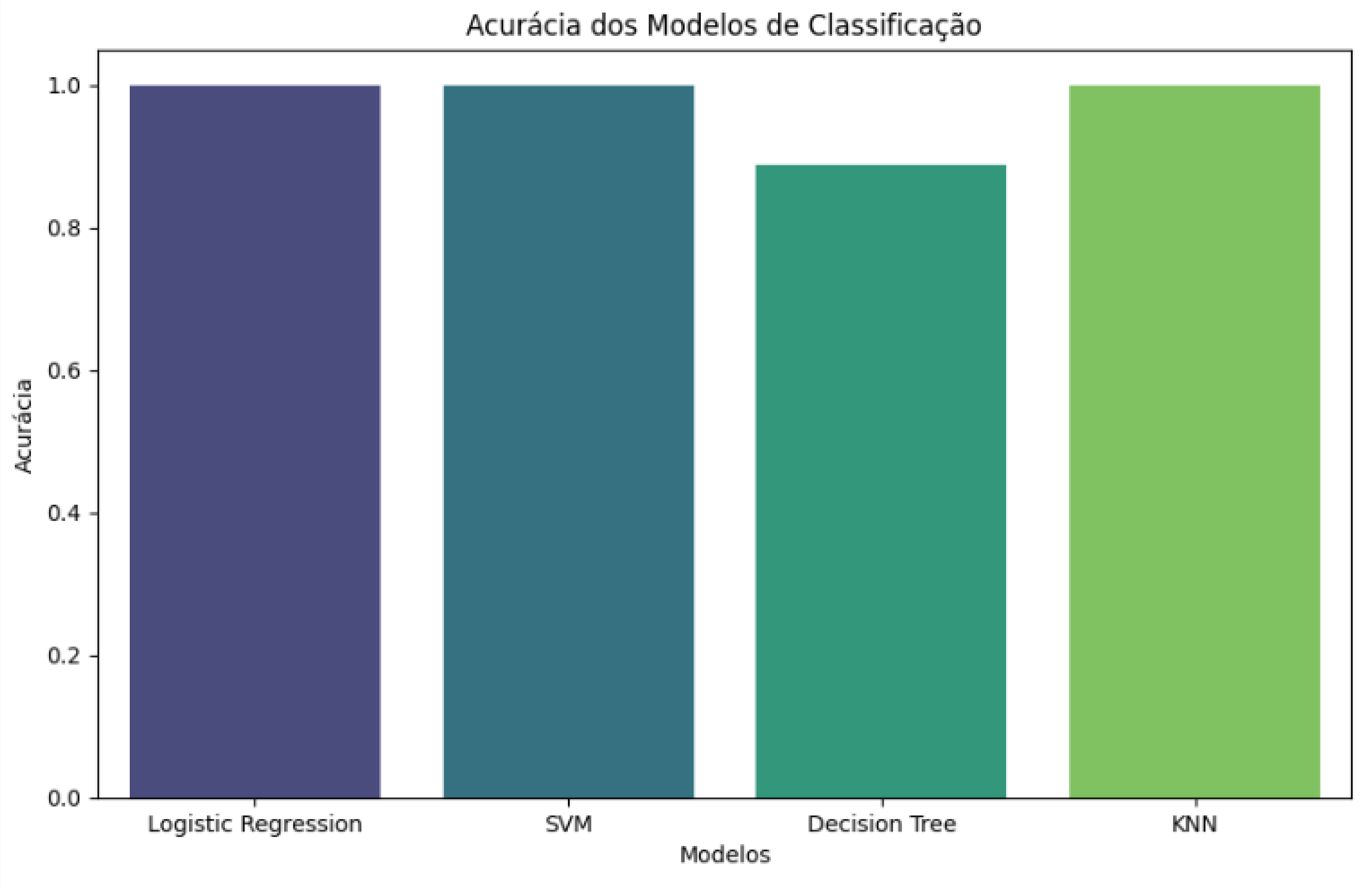


These images show the scatter plot at different payload ranges and we can also see which version of the engine achieved the highest percentage of success. In this case, by analyzing these two graphs, we can see that the "FT" version stood above all the others.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

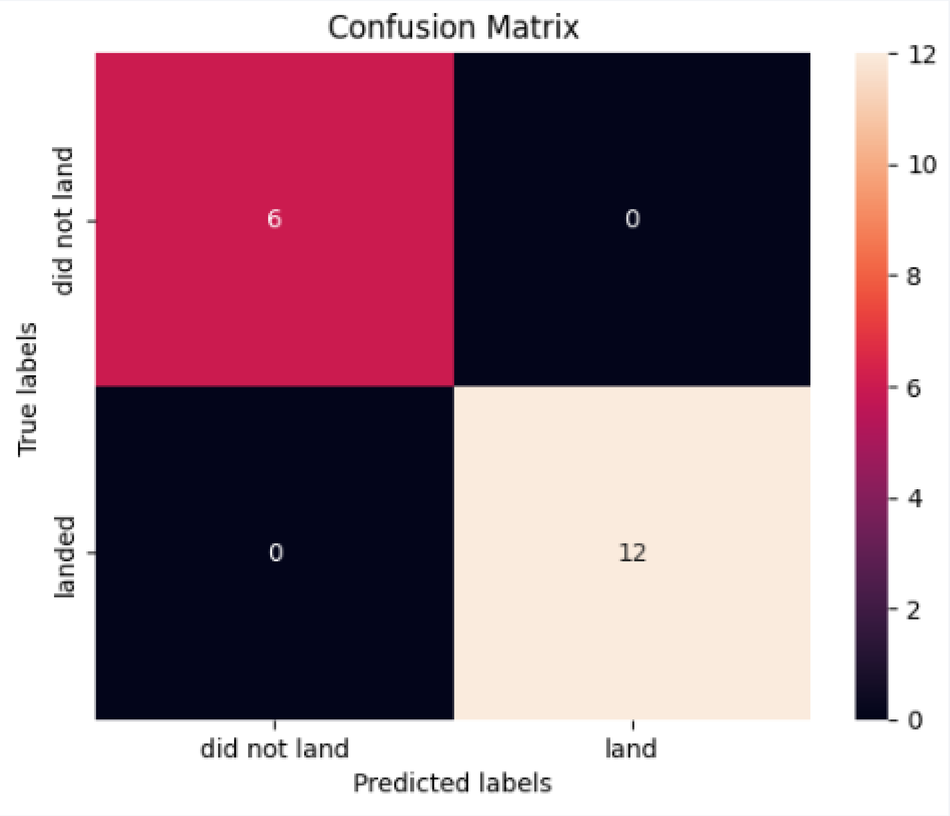


The model with the highest classification accuracy, according to the analysis, was the Logistic Regression model, but other models, such as SVM and KNN, reached very similar accuracy, 1.0 and 0.985 respectively.

Confusion Matrix

Best performing model:

As we can see in the confusion matrix, the model scores 12 points precisely in the "landed/landing" column and row, which shows us that the model predicted correctly and obtained the best performance in this analysis and prediction of the data.



Conclusions

- The best model is Logistic Regression with an accuracy of 1.00;
- The "KSC LC-39A" local is the local with the highest percentage of successful launches;
- The percentage of successful launches is greater than the percentage of failures;
- The percentage of successful launches is greater than the percentage of failures;
- The decision tree model had the worst performance among all models tested;
- The SO orbit was the one with the worst success rate, reaching 0.0%.

Appendix

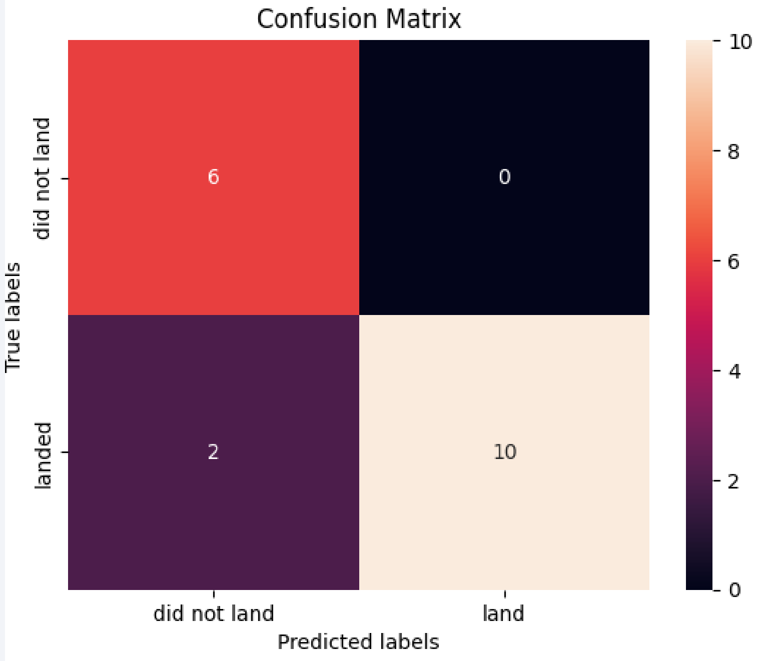
Calculate the accuracy of tree_cv on the test data using the method `score`:

```
tree_accuracy = tree_cv.score(X_test,Y_test)
tree_accuracy
```

```
0.8888888888888888
```

We can plot the confusion matrix

```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
plt.show()
```



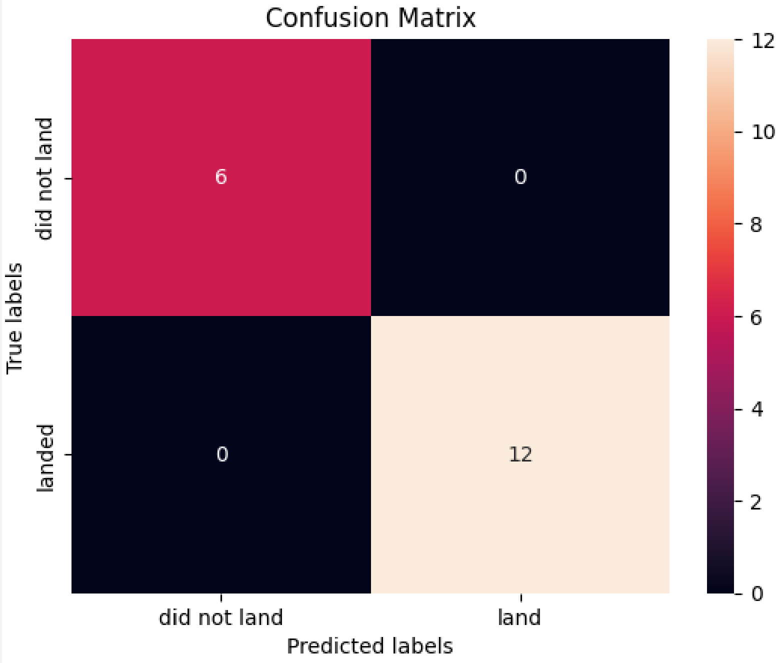
Calculate the accuracy on the test data using the method `score`:

```
logreg_accuracy = logreg_cv.score(X_train, Y_train)
logreg_accuracy
```

```
1.0
```

Lets look at the confusion matrix:

```
yhat = logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
plt.show()
```



Thank you!