IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

John Ales
2022-03-31

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

- Gathered Data using the SpaceX API and web scraping from Wikipedia

- Perform data wrangling

- One-hot encoding was used to represent success or failure

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

    - - Used different models to predict landing success

Data Collection

Data collection was done using:

Get request to the SpaceX API

Decoded the response content using .json() function call and turn it into a pandas dataframe with .json_normalize()

Cleaned the data of missing values

Web scraping from Wikipedia for Falcon 9 launch data using BeautifulSoup

Take the HTML table and convert it to a pandas dataframe for analysis

Data Collection – SpaceX API

one Get request to receive .json file through the Space X API

two Converted .json to a pd dataframe, then extracted Falcon 9 data

three Cleaned the data by replacing missing values, with the mean

https://github.com/JohnAles/IBM-Data-Science-Capstone-SpaceX/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

one Use beautifulsoup to scrape data from Falcon 9 wikipedia page

two Extract the variables, or column names

three Create a dataframe out of the HTML table

https://github.com/JohnAles/IBM-Data-Science-Capstone-SpaceX/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

one Exploratory Data Analysis, determined the training labels

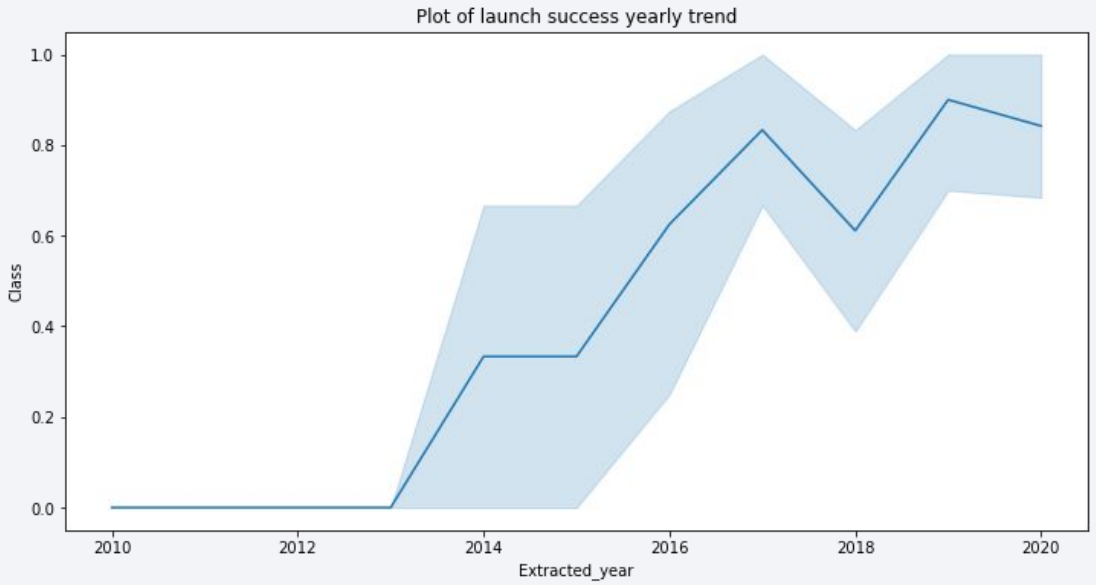
two Summarized the number of launches at each site, number, and occurrence of each orbit

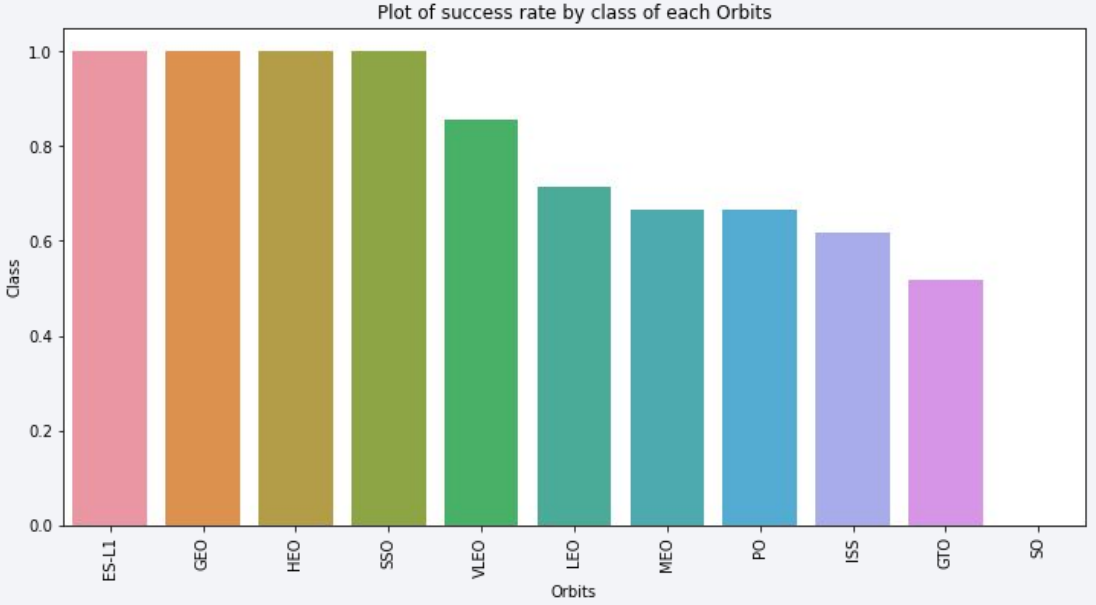
three Created landing outcome labels from outcome column

https://github.com/JohnAles/IBM-Data-Science-Capstone-SpaceX/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Visualized the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.





https://github.com/JohnAles/IBM-Data-Science-Capstone-SpaceX/blob/main/jupyter-labs-eda-dataviz.ipynb

Build an Interactive Map with Folium

Launch sites marked, and added map objects such as markers to mark the success or failure of launches for each site on the map.

Assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.

Color-labeled marker clusters used to identify which sites have a higher success rate.

Calculated the distances between a launch site to its proximities, and covered whether launch sites are certain distance from cities.

https://github.com/JohnAles/IBM-Data-Science-Capstone-SpaceX/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Interactive dashboard with Plotly dash

Plotted pie charts showing the total launches by site location

Plotted scatter graph showing the correlation between Outcome and Payload by the particular booster iteration

https://github.com/JohnAles/IBM-Data-Science-Capstone-SpaceX/blob/main/dash_app.py

Predictive Analysis (Classification)

one — load the data using numpy and pandas, transform the data, split the data into training and testing

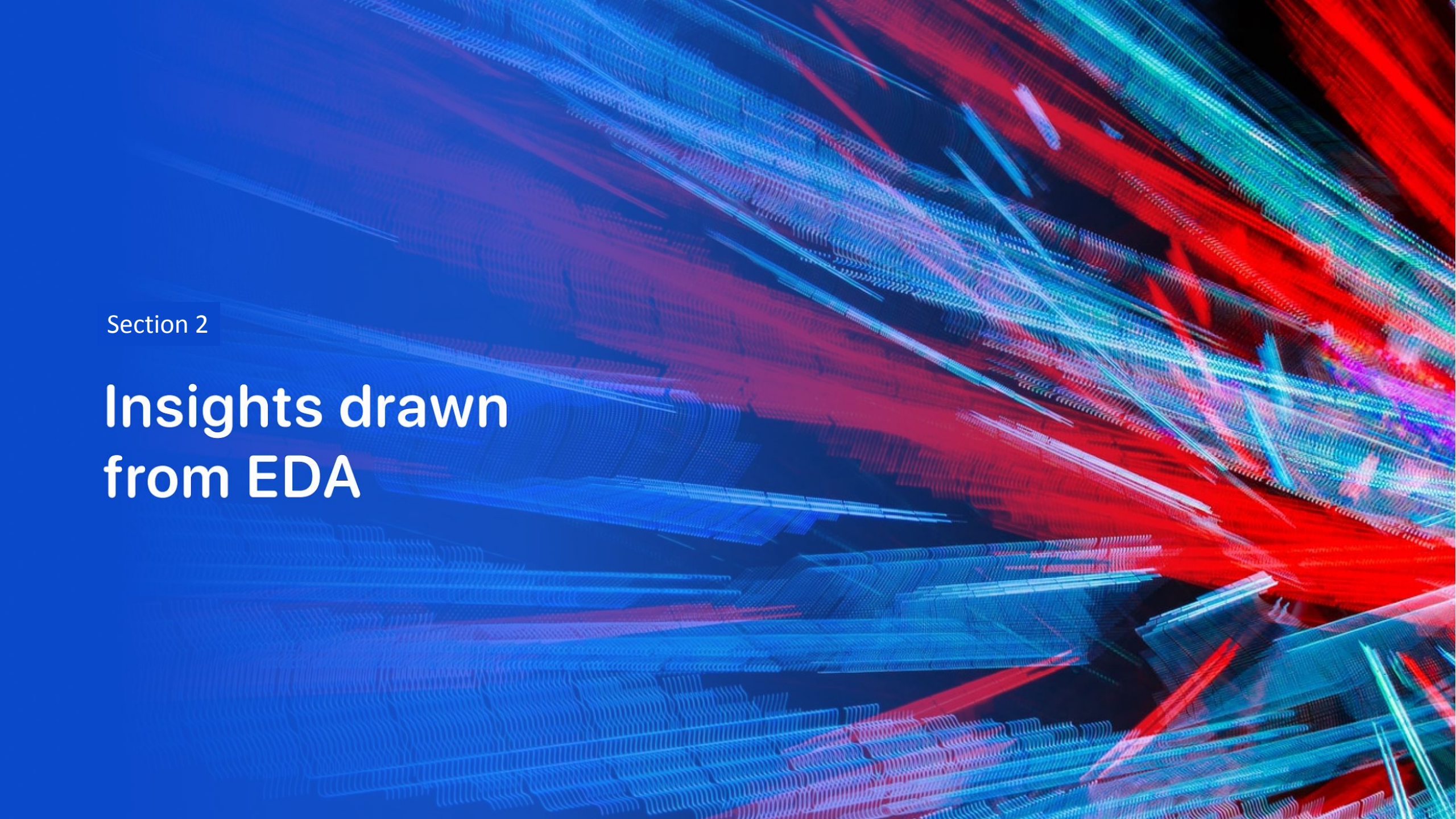
two — split the data into training and testing

three — build and fit different machine learning models

four — Test for accuracy of the model

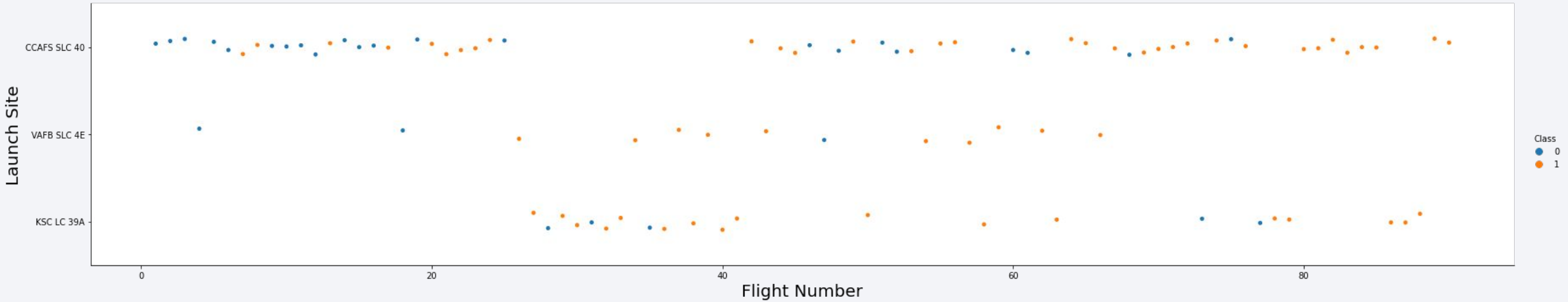
five — Algorithms used: logistic regression, support vector, decision tree, k nearest neighbor

https://github.com/JohnAles/IBM-Data-Science-Capstone-SpaceX/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb


Section 2
Insights drawn from EDA

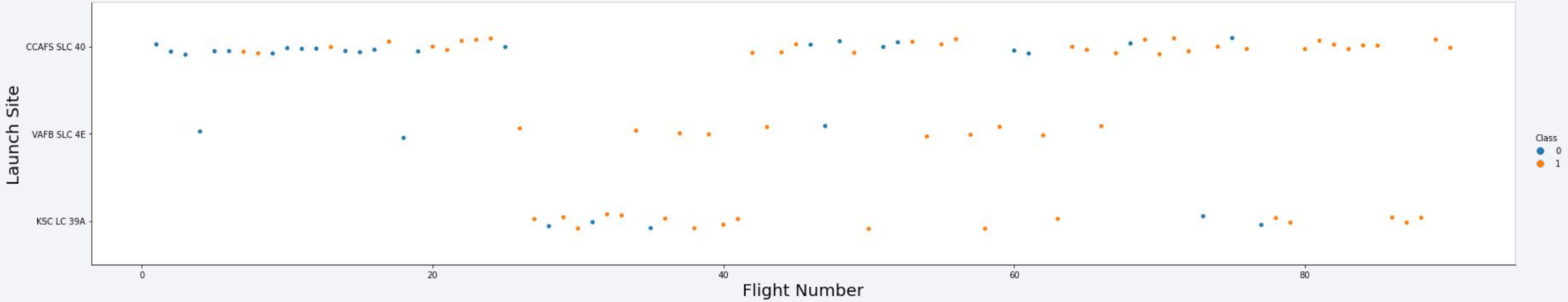
Flight Number vs. Launch Site

- CCAF5 is the oldest and most used
- VAFB less frequently used
- KSC is the newest site



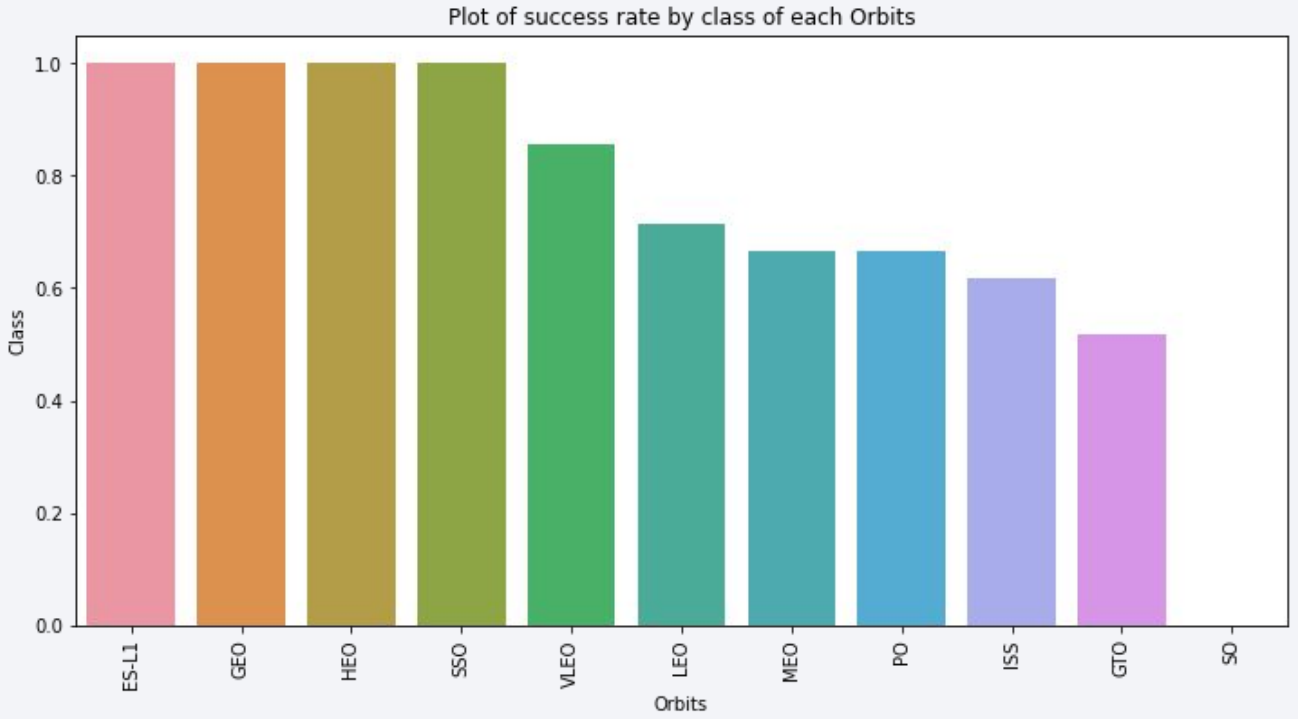
Payload vs. Launch Site

There was a higher success rate for rockets carrying higher payloads at the CCAFS sit.



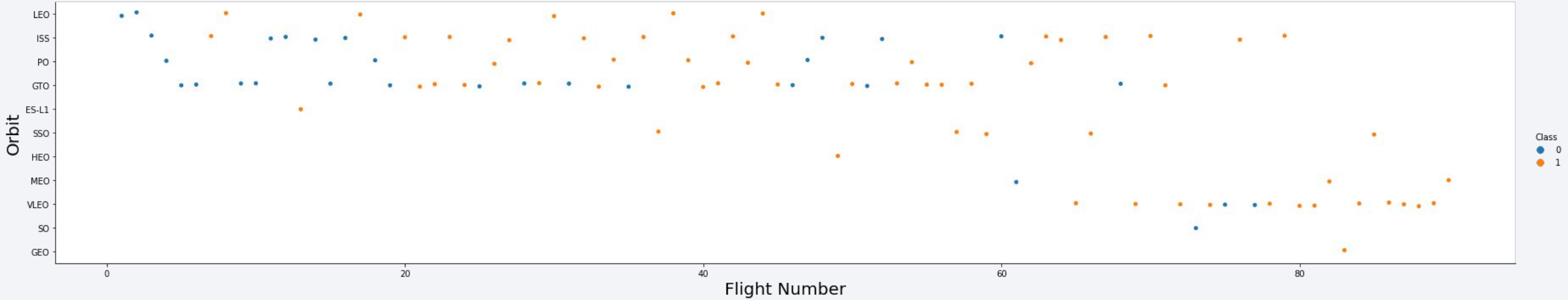
Success Rate vs. Orbit Type

ES-L1, GEO, HEO, SSO, VLEO had the highest success rates.



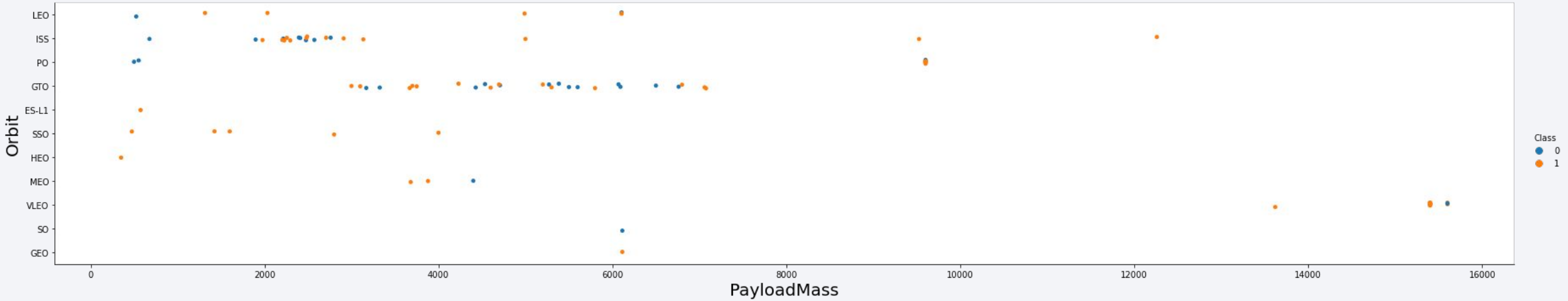
Flight Number vs. Orbit Type

Observed here is that in the LEO orbit success is related to the number of flights, whereas in the GTO orbit, there isn't a correlation between flight number and orbit.



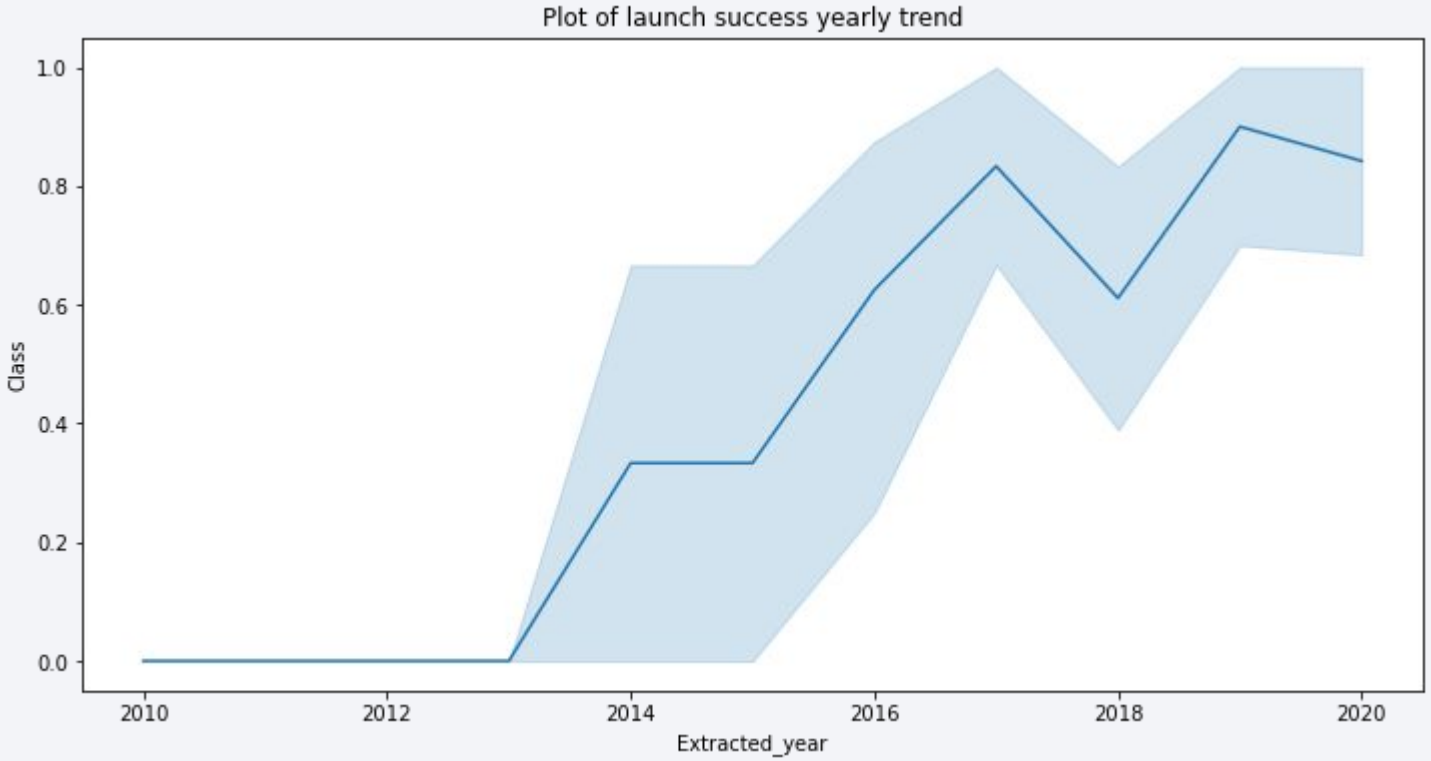
Payload vs. Orbit Type

For PO, LEO and ISS orbits with heavy payloads had higher rates of successful landings



Launch Success Yearly Trend

Success rate since 2013 kept on increasing till 2020.



All Launch Site Names

```
# Select relevant sub-columns: `Launch Site`, `Lat(Latitude)`, `Long(Longitude)`, `class`
spacex_df = spacex_df[['Launch Site', 'Lat', 'Long', 'class']]
launch_sites_df = spacex_df.groupby(['Launch Site'], as_index=False).first()
launch_sites_df = launch_sites_df[['Launch Site', 'Lat', 'Long']]
launch_sites_df
```

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

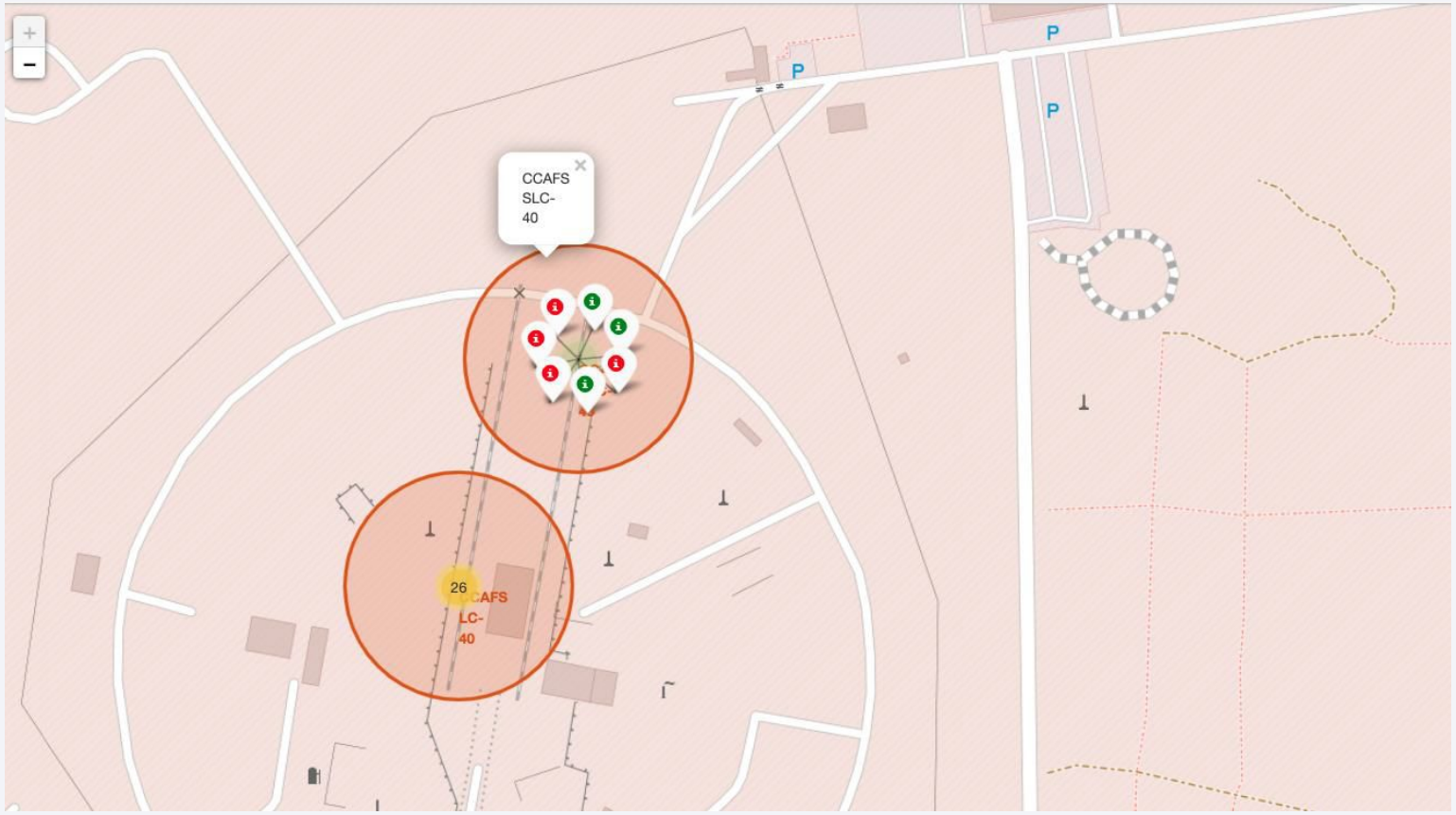

Section 3
Launch Sites
Proximities Analysis

Launch Sites

Launch sites are located in California and Florida

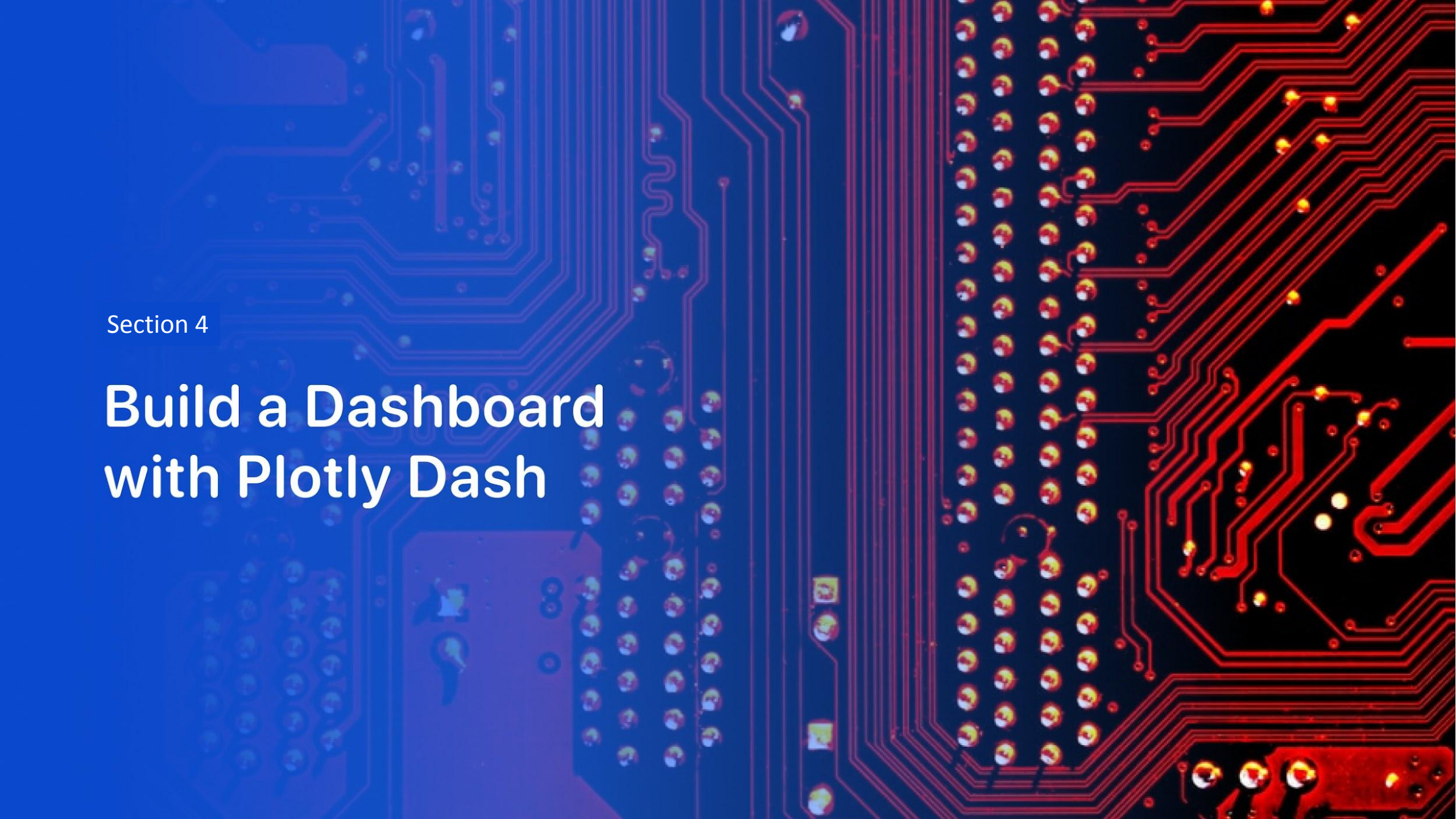
Launch success and failure

Red markers indicate failed launch, green indicates success



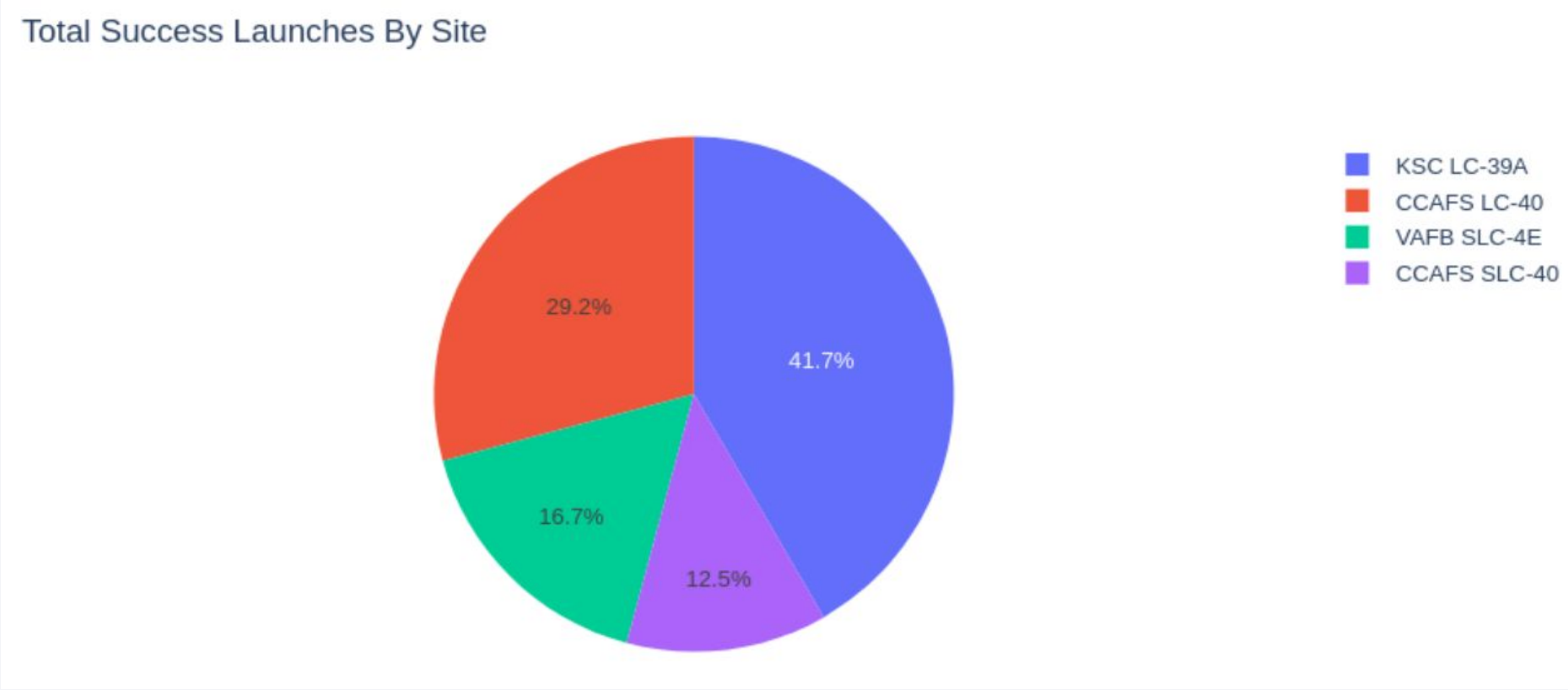
Site proximities

Launch sites are close to the coast line, far from population centers


Section 4
Build a Dashboard with Plotly Dash

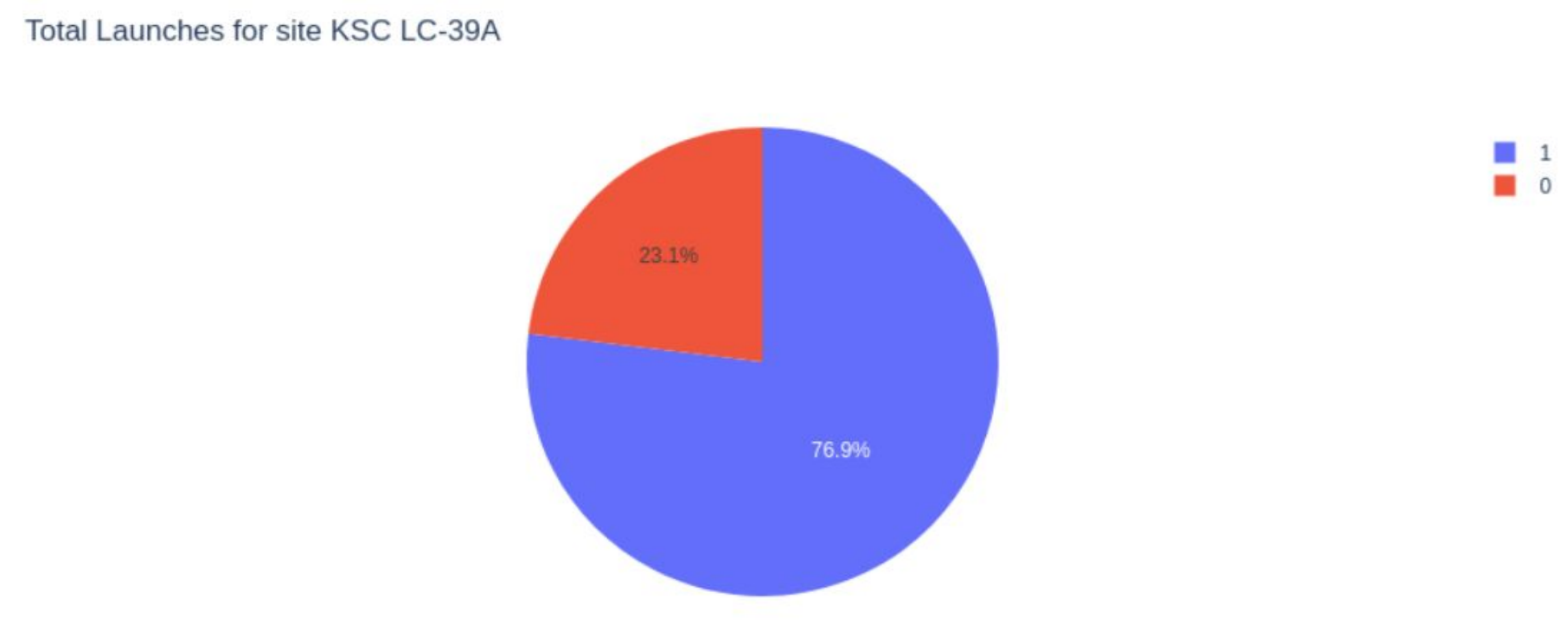
Success rate by site

KSC is the site with the highest success rate



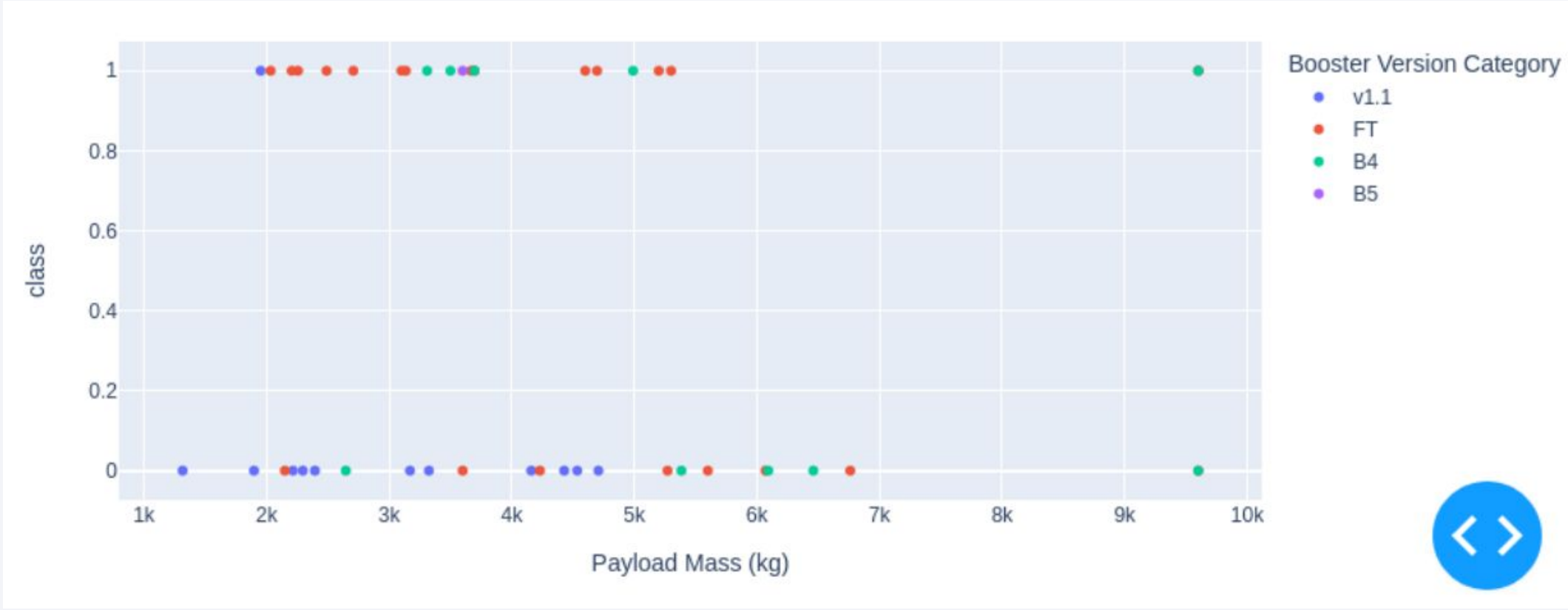
Success rate of Launches at KSC

Over ¾ of launches are a success at KSC



<Dashboard Screenshot 3>

Launches that use FT boosters and have a payload less than 6,000 kg have the highest success rate.



Section 5

Predictive Analysis (Classification)

Classification Accuracy

Decision Tree Classifier had the highest level of accuracy

```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

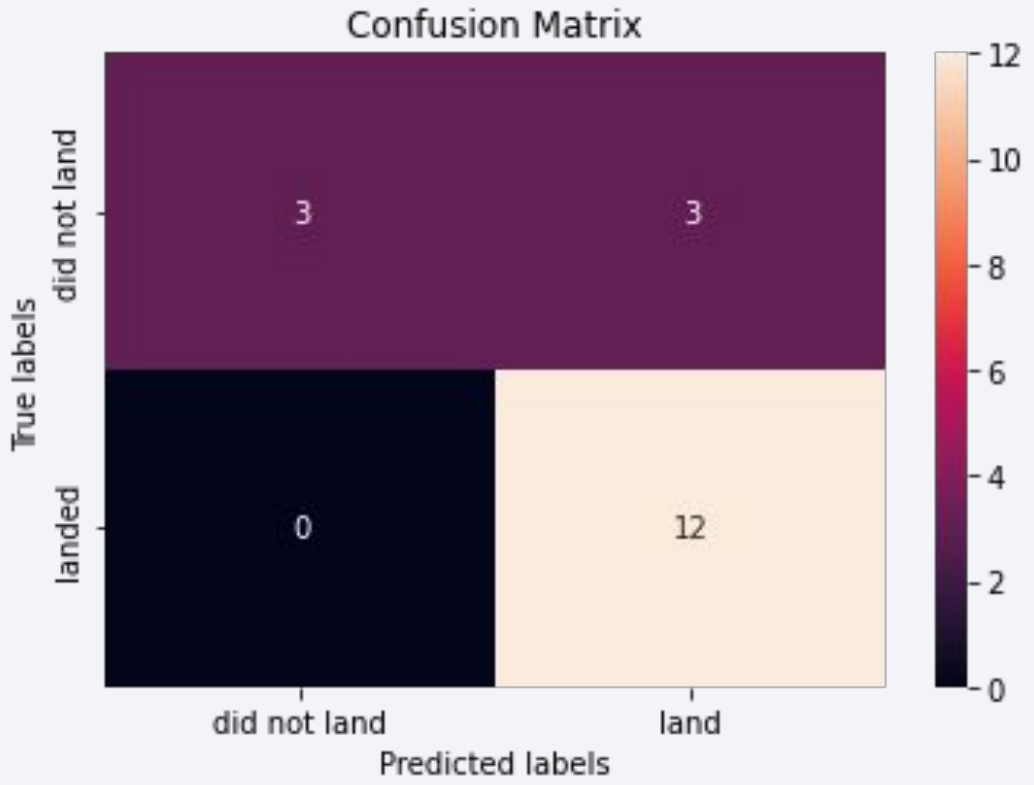
```
grid_search = GridSearchCV(tree, parameters, cv=10)
tree_cv = grid_search.fit(X_train, Y_train)
```

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split':
2, 'splitter': 'best'}
accuracy : 0.8857142857142858
```

Confusion Matrix

The confusion matrix for the decision tree shows that it can successfully distinguish and predict between the different classes, also it shows a solid majority being true positives.



Conclusions

- The launch site with the highest success rate is KSC LC 39A
- Launches above 7,000kg have a higher success rate
- ES-L1, GEO, HEO, SSO, VLEO Orbits have the highest success rate
- Successful landing outcomes seem to improve over time, according the evolution of processes and rockets, launch success rate started to increase in 2013

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!