IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Raphael Nnaemelum Ogbodo
November 18, 2025
University of Iowa.

Outline

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

Executive Summary

- **Summary of methodologies**
  - Collected and cleaned SpaceX launch data from REST API and web sources
  - Performed exploratory data analysis to identify trends in payload, launch site, orbit, and booster version.
  - Built ML models (Logistic Regression, KNN, SVM, Decision Tree) and tuned them with GridSearchCV

- **Summary of all results**
  - Landing success strongly depends on booster version, payload mass, and launch site
  - Decision Tree (tuned) showed the highest prediction accuracy
  - KSC LC-39A exhibited the highest landing success rates
  - ML models can reliably predict Falcon 9 landing outcomes, supporting cost-efficient mission planning

Introduction

- **Project Background and Context**
  - SpaceX aims to reduce launch costs by successfully landing and reusing the Falcon 9 first stage.
  - Understanding which factors influence landing success helps optimize operations and support mission planning.
  - The dataset includes launch records, booster details, payload mass, orbit types, and landing outcomes.

- **Problems Statements**
  - What factors most strongly determine whether a Falcon 9 booster will land successfully?
  - Can machine learning models accurately predict landing success?
  - How do payload mass, launch site, orbit type, and booster version influence landing outcomes?

Section 1

Methodology

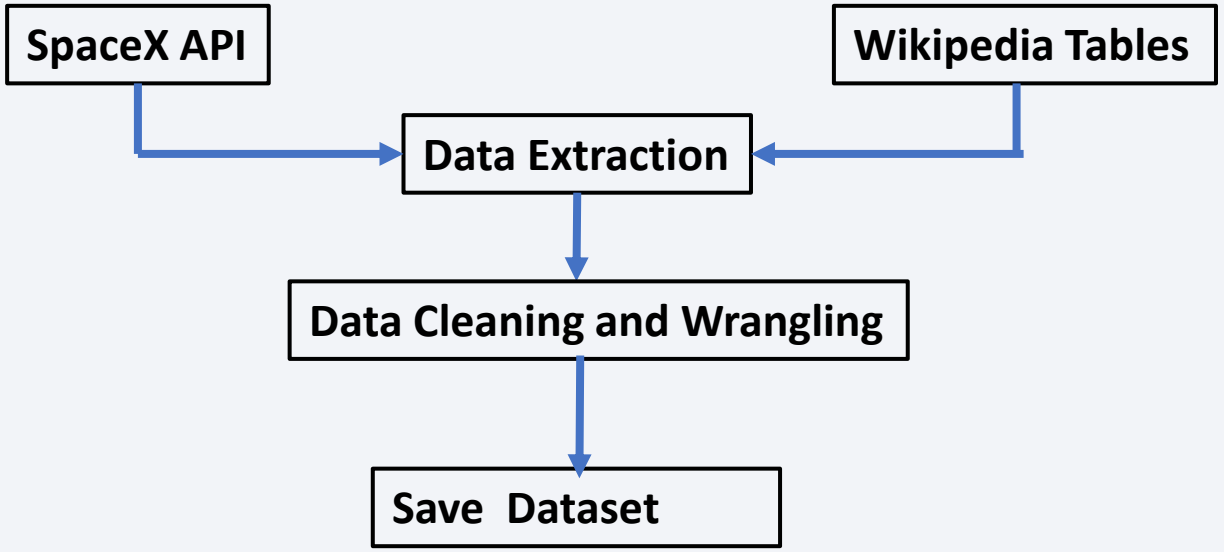
Methodology

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

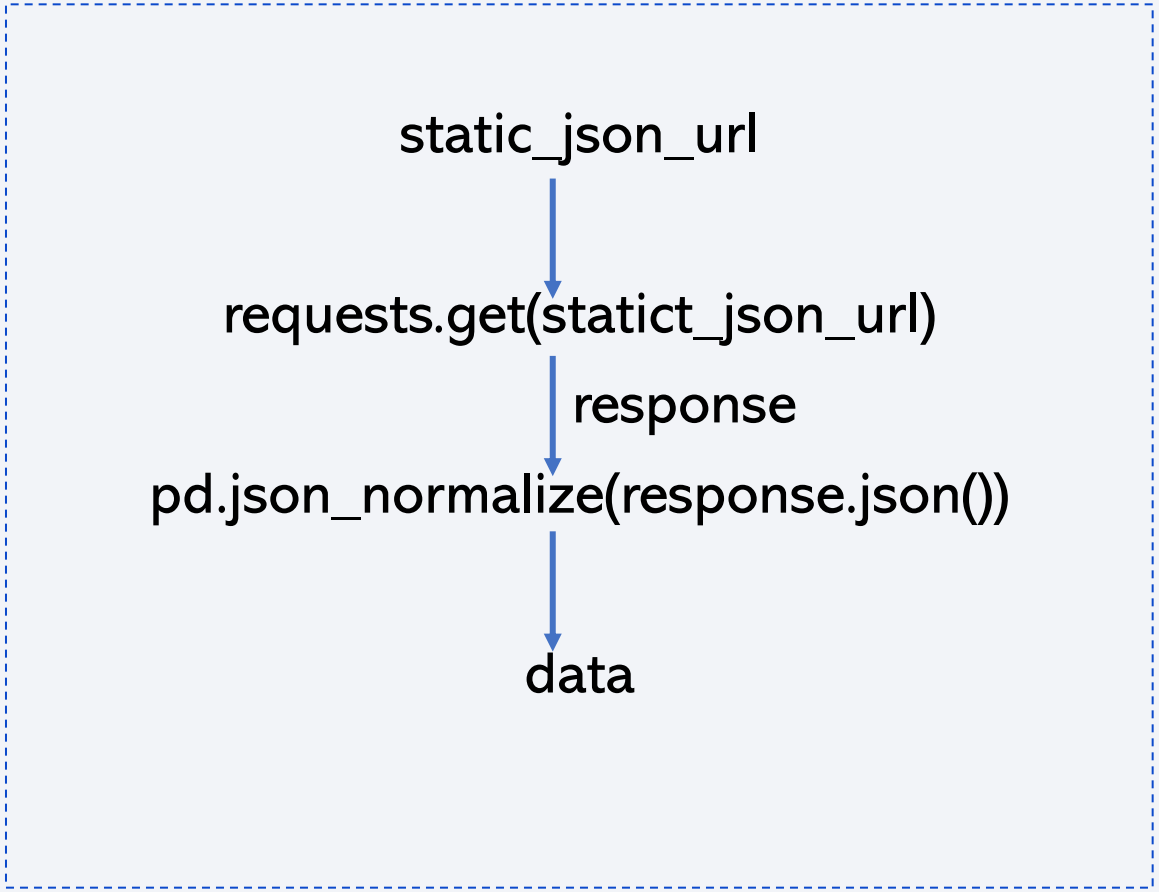
Data Collection

- **How the Data Was Collected**
  - Retrieved SpaceX launch records using the SpaceX REST API
  - Scraped historical data from Wikipedia launch tables
  - Loaded datasets into Pandas DataFrames
  - Cleaned and merged datasets (handled missing values, standardized columns)
  - Prepared final dataset for EDA and Machine Learning



Data Collection – SpaceX API

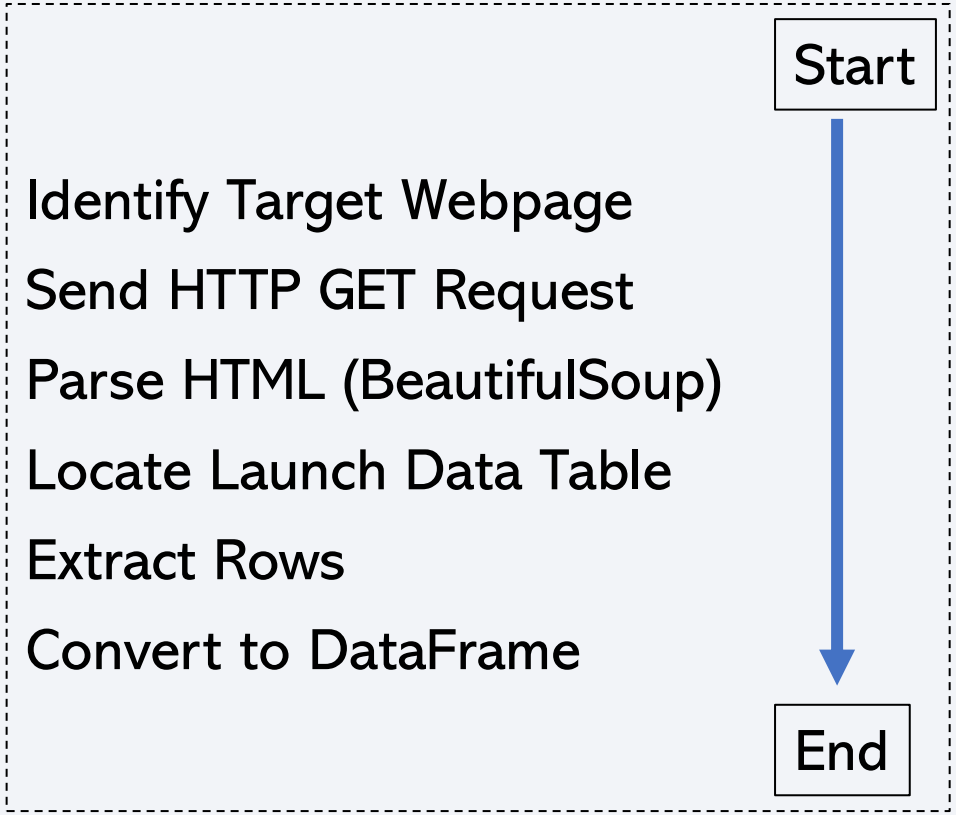
- SpaceX REST API calls was made using a statistic response object (static_json_url).
- Response content (Json) of the Get request method used above was turned into a dataframe using .json_normalize() method in pandas.



GitHub URL:
https://github.com/Raphaelogbodo/Data-Science-Course/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

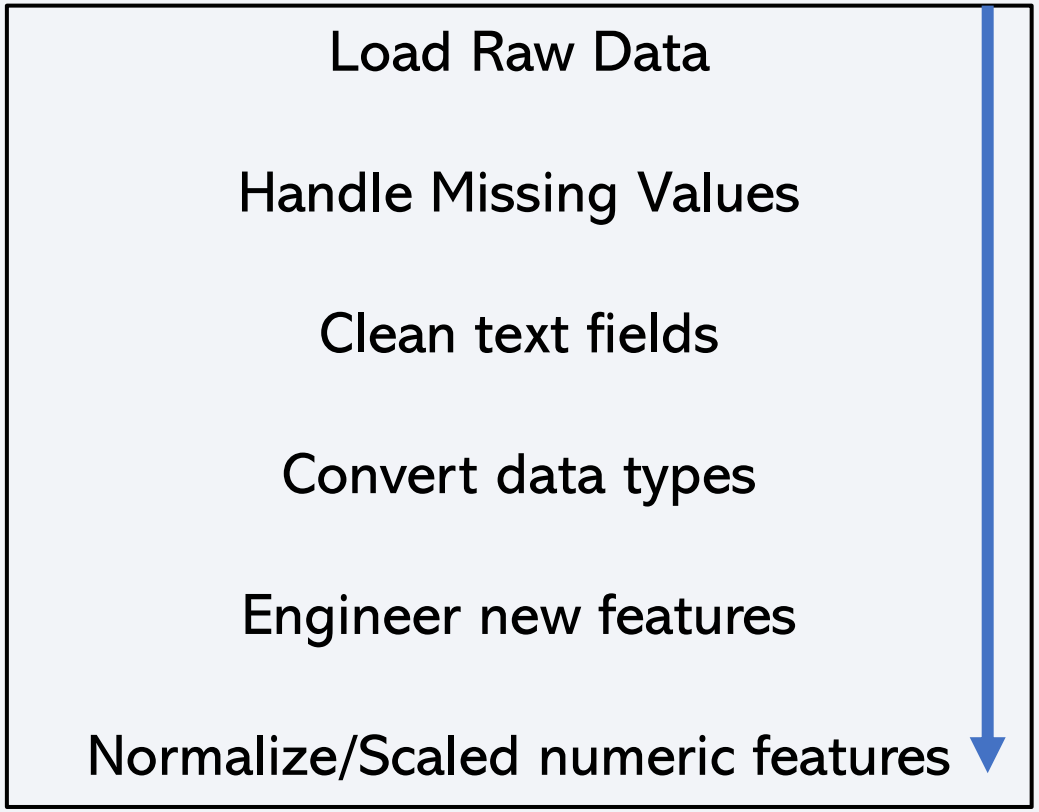
Data Collection - Scraping

- **Key steps and Phrases**
  - Identified target table on Wikipedia: Falcon 9 and Falcon Heavy Launches
  - Used Requests to fetch HTML content and Parsed HTML using BeautifulSoup
  - Located and extracted launch data tables
  - Converted table rows into a structured pandas DataFrame
  - Cleaned text fields (dates, booster versions, payload mass, landing outcomes
  - Stored processed data for EDA and Machine Learning
- GitHub URL: https://github.com/Raphaelogbodo/Data-Science-Course/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- Key steps:
  - Loaded raw data from API and web scraping into Pandas
  - Handled missing values
  - Cleaned text fields (booster version, landing outcome, orbit)
  - Converted data types
  - Engineered new features (class label for landing success, one-hot encoding for categorical variables)
  - Normalized/Scaled numeric features for ML modeling



GitHub URL: https://github.com/Raphaelogbodo/Data-Science-Course/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Charts used EDA

- Categorical plots were used to find relationship between/among variables below:
  - LaunchSite vs FlightNumber (Class as hue)
  - LaunchSite vs PayloadMass (Class as hue)
  - Orbit vs FlightNumber (Class as hue)
  - Orbit vs PayloadMass (Class as hue)
- Bar Chart to visualize relationship between success rate of each orbit type
- Line plot to visualize launch success yearly trend

- GitHub URL: https://github.com/Raphaelogbodo/Data-Science-Course/blob/main/edadataviz.ipynb

EDA with SQL

- **Bullet summary of the SQL queries performed**
  - Identified all unique launch sites and previewed sample launches from key sites.
  - Analyzed payloads by customer and booster version, including average payload mass for F9 v1.1 and total NASA (CRS) payloads
  - Examined landing outcomes: first successful landing, successful drone landings for 4000–6000 kg payloads, and overall counts of successes and failures.
  - Investigated extreme payloads, highlighting the booster used for the heaviest launch.
  - Explored temporal trends and failures: monthly drone landing failures in 2015 and overall landing outcome trends between 2010–2017.
- GitHub URL: https://github.com/Raphaelogbodo/Data-Science-Course/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Summary off the interactive map
  - Created a base Folium map centered on SpaceX launch sites to provide geographic context.
  - Added markers for each launch site to indicate exact locations and show popups with site name, total launches, and success rate.
  - Added circle markers to represent payload mass or launch success, using size and color to visualize differences across sites.
  - Included popups and tooltips for interactive exploration, displaying detailed information like booster version, date, and landing outcome.
  - Used color coding to distinguish between successful and failed launches for quick visual interpretation.
  - Added lines or polylines to illustrate launch trajectories or connections between sites, highlighting spatial patterns

- GitHub URL: https://github.com/Raphaelogbodo/Data-Science-Course/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

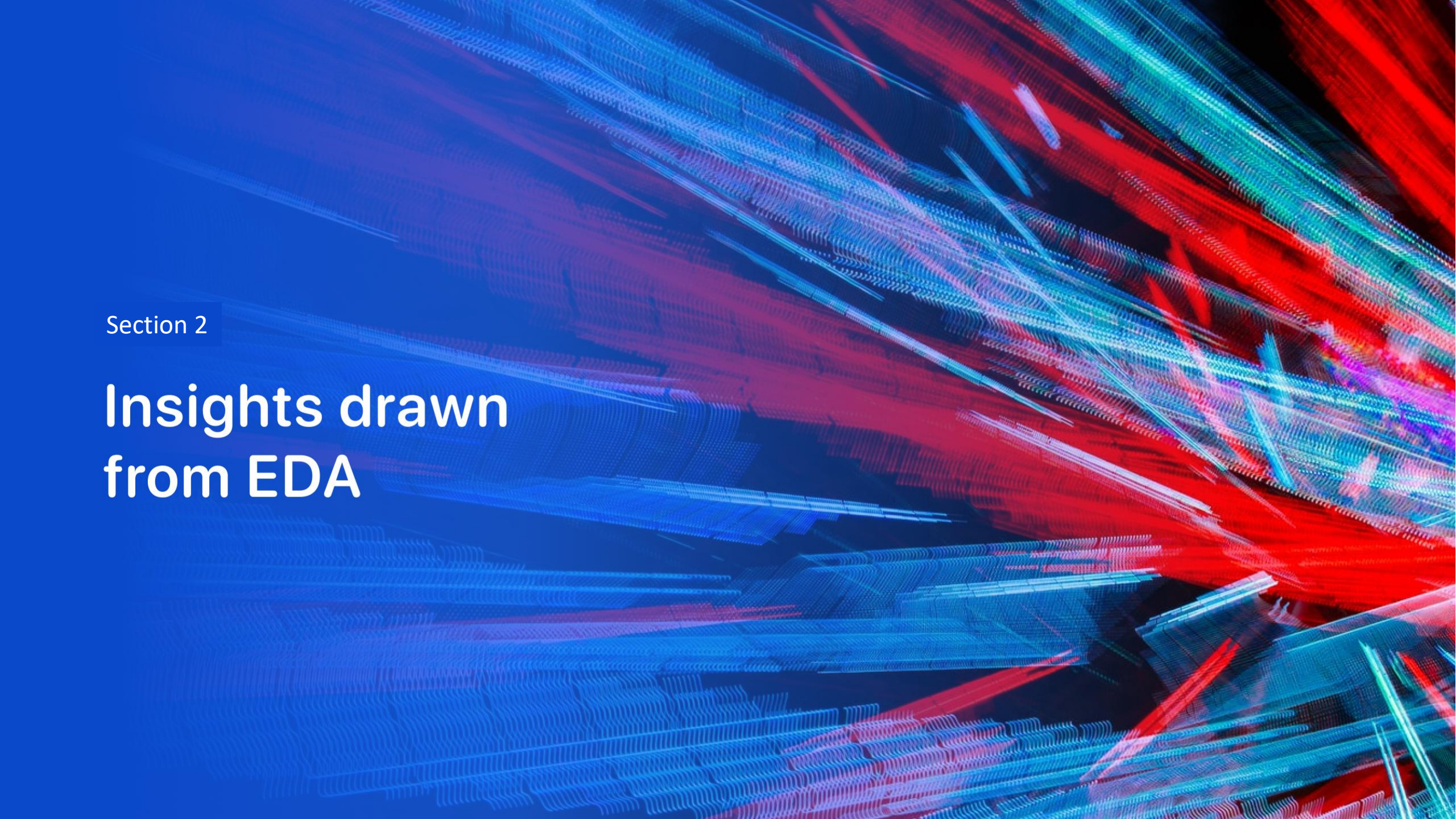
- Dashboard plots/graphs
  - Scatter plots
    - Used to show the relationships between payload mass and success rate
  - Pie chart
    - Used to show the proportion of successful launches at each site if all sites are considered
    - Shows proportion of success and failure of launches if individual site is selected
- Dropdown is used for site selection
- RangeSlider is used for payload mass range selection
- GitHub URL: https://github.com/Raphaelogbodo/Data-Science-Course/blob/main/spacex-dash-app.py

Predictive Analysis (Classification)

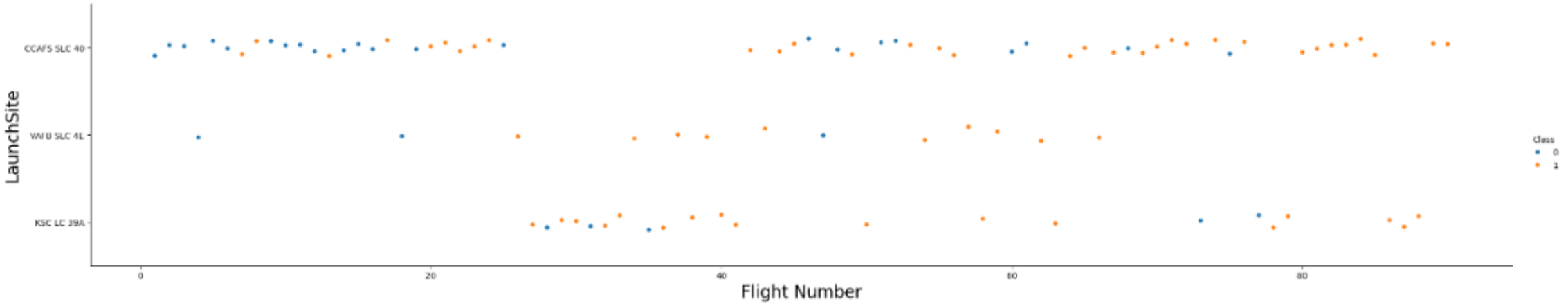
- Models used:
  - Logistic Regression, DecisionTree Classifier, KNN, and SVM
- GridSearchCV to find the best hyperparameters.
- Trained with optimal parameters identified by GridSearch.
- Evaluated models using accuracy, precision, recall, F1-score, and confusion matrix.
- Compared all results and selected the best-performing classifier as the final model
- GitHub URL: https://github.com/Raphaelogbodo/Data-Science-Course/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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


Section 2
Insights drawn from EDA

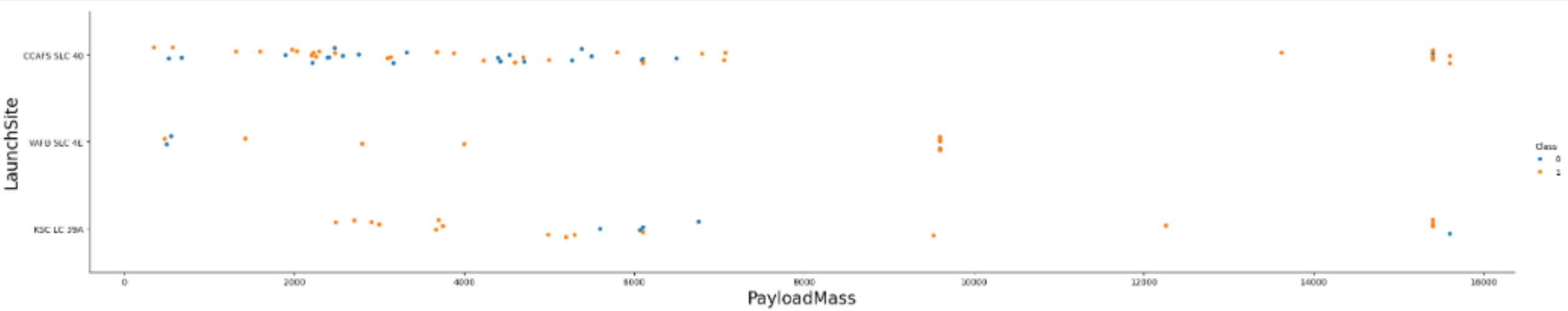
Flight Number vs. Launch Site



Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

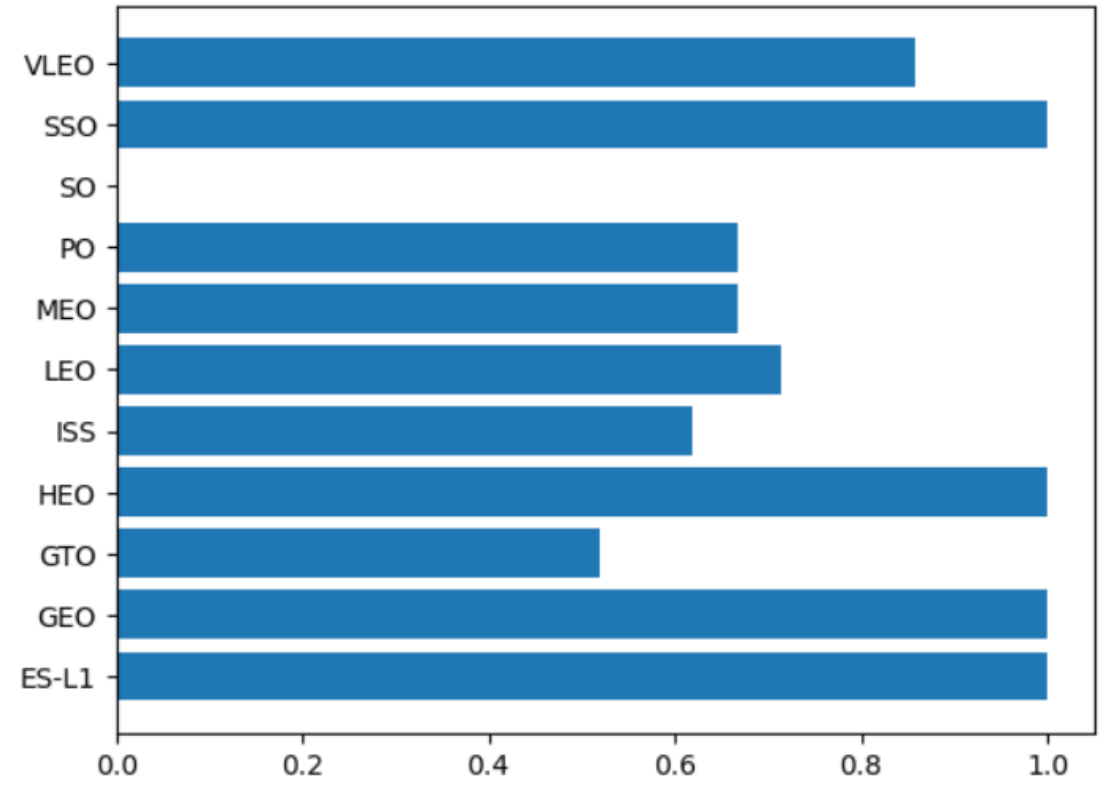
- There is a higher chance for the first stage to fail at launch site CCAFS SLC 40 compared to the other two sites.
- There is a noticeable increase in the chance of fisrt stage succes as at all the sites as flight number increases

Payload vs. Launch Site



Now if you observe Payload Mass 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).

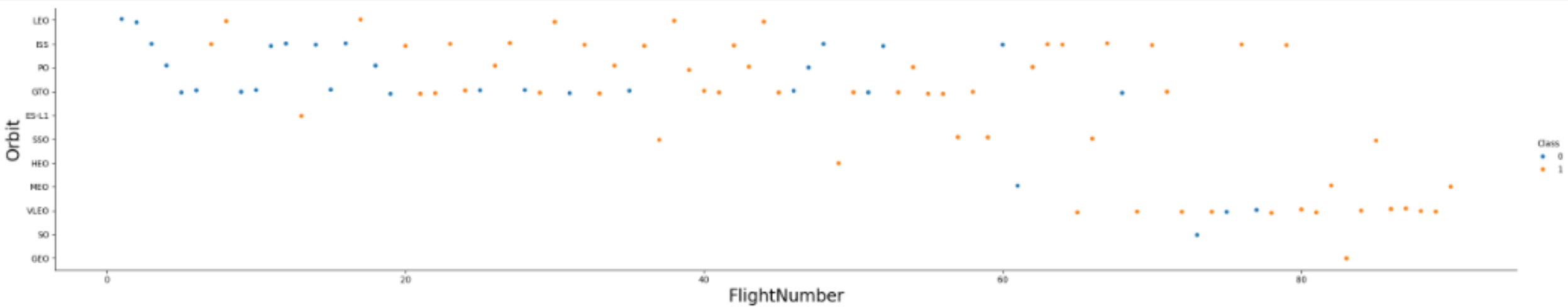
Success Rate vs. Orbit Type



Analyze the plotted bar chart to identify which orbits have the highest success rates.

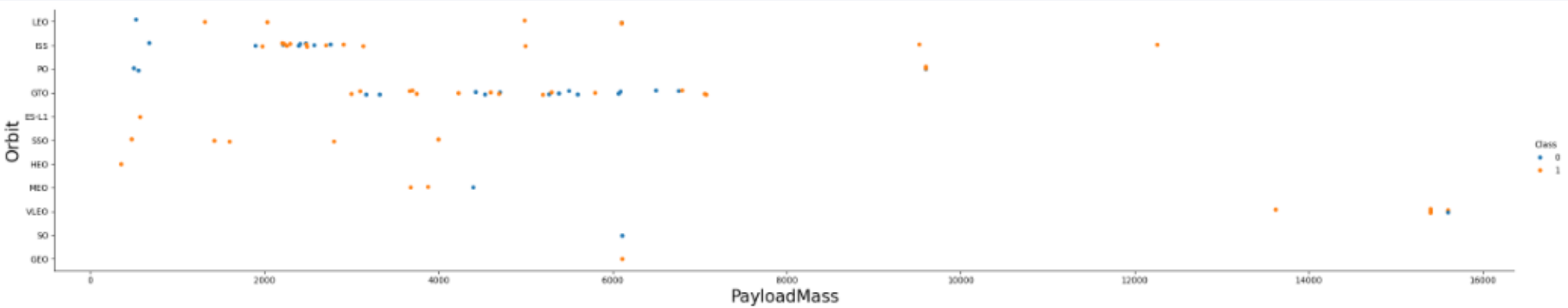
- SSO, HEO, GEO, ES-L1 orbits has similar success rate of 100%.
- VLEO has about 85% succee rate

Flight Number vs. Orbit Type



You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

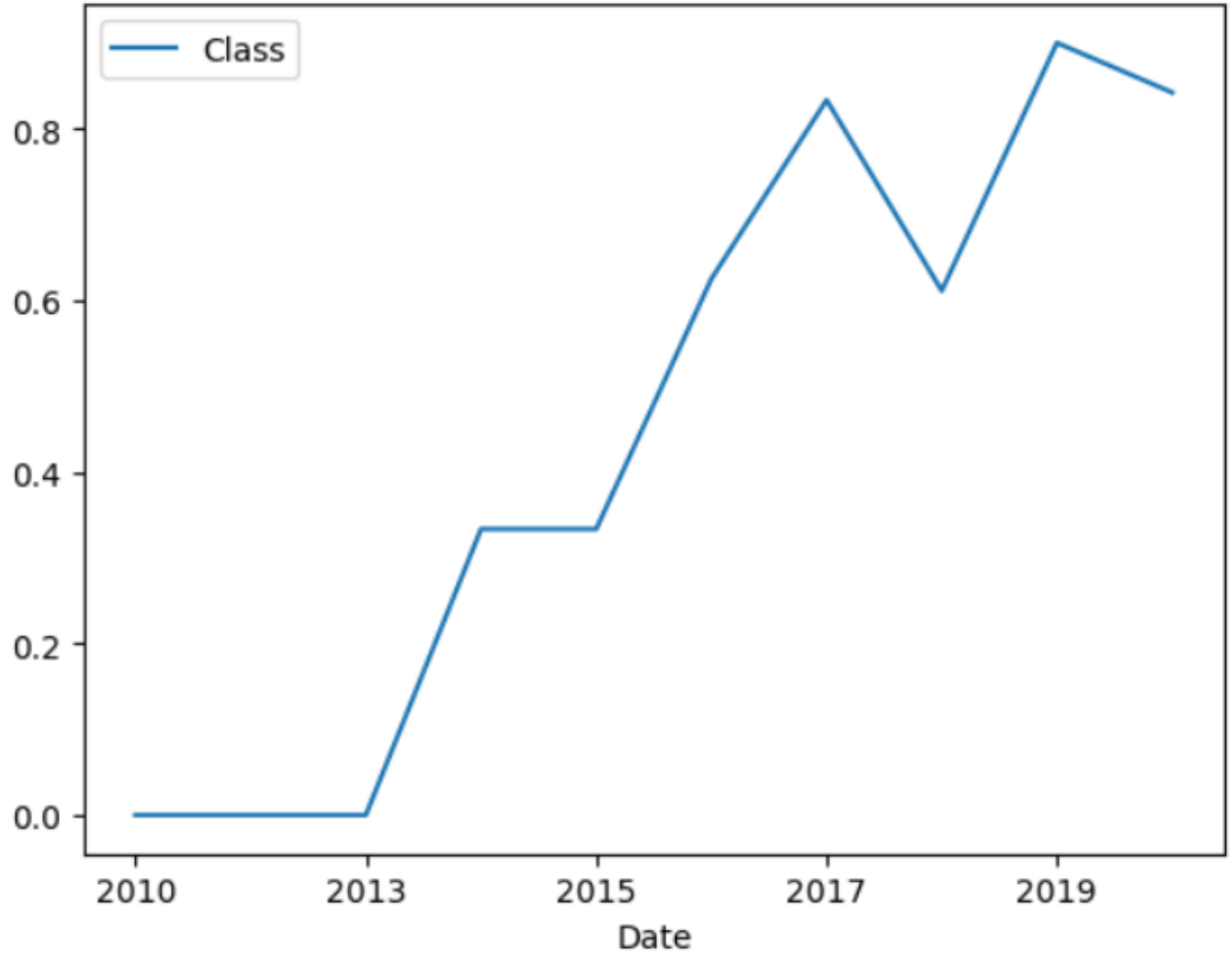
Payload vs. Orbit Type



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

However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

- Querry:

%sql select distinct(Launch_Site) from SPACEXTABLE;

- Result

CCAFS LC-40

VAFB SLC-4E

KSC LC-39A

CCAFS SLC-40

- The above result shows from the table SPACEXTABLE the unique sites as listed in the “Launch Site” column

Launch Site Names Begin with 'CCA'

- Querry:

%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5;

- Results:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- The above result shows the first 5 rows from table “SPACEXTABLE” where column “Launch Site” entries begins with “CCA”

Total Payload Mass

- Query:

%sql select sum(PAYLOAD_MASS__KG_) as Tot_payload_mass from SPACEXTABLE where Customer = 'NASA (CRS)';

- Results:

Tot_payload_mass
45596

- The above result shows the total payload mass in Kg from the “SOACEXTABLE” records where the “Customer” column is “NASA (CRS)”.

Average Payload Mass by F9 v1.1

- Querry:

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

- Results:

avg_payload_mass
2928.4

- The above result shows the average payload mass in Kg of all records in “SPACEXTABLE” where “Booster_Version” column is “v1.1” for only F9 launchers.

First Successful Ground Landing Date

- Querry:

%sql select min(Date) as first_success from SPACEXTABLE where Landing_Outcome like 'Success%';

- Result:

first_success
2015-12-22

- The result shows the date for the first successful landing outcome of the launches.

Successful Drone Ship Landing with Payload between 4000 and 6000

- Query:

```
%sql select Booster_Version from SPACEXTABLE where Landing_Outcome like 'Success (drone%' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000;
```

- Result:

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

- The result above shows boosters that successfully landed on drone ship for a subset of “SPACEXTABLE” records where “PayloadMass” values are greater than 4000 and less than 6000 Kg.

Total Number of Successful and Failure Mission Outcomes

- Query:

%sql select count(*) as total_success_failure from SPACEXTABLE where Landing_Outcome like 'Success%' or Landing_Outcome like 'Failure%';

- Result:

total_success_failure
71

- The result above shows the combined total of both “Success” and “Failure” landing outcomes as listed in column “Landing_Outcome”.

Boosters Carried Maximum Payload

- Query:

%sql select Booster_Version as BV from SPACEXTABLE where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTABLE);

- Result:

BV
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

- The result above show list of booster versions from “SPACEXTABLE” where payload mass is equal to the maximum mass in “PAYLOAD_MASS__KG_” column. This is an example of a nested query.

2015 Launch Records

- Query:

%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 LIKE '%Failure (drone%' ORDER BY Month;

- Result:

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

- The result above shows the failed landing outcomes in 2015 and result ordered by month.

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

- Query:

%sql SELECT Landing_Outcome, COUNT(*) AS outcome_count FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY outcome_count DESC;

- Result:

Landing_Outcome	outcome_count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- The result above shows the order of landing outcome counts between the years 2010-06-04 and 2017-03-20 with “No attempt” the highest and “Precluded (done ship)” the least.

Section 3

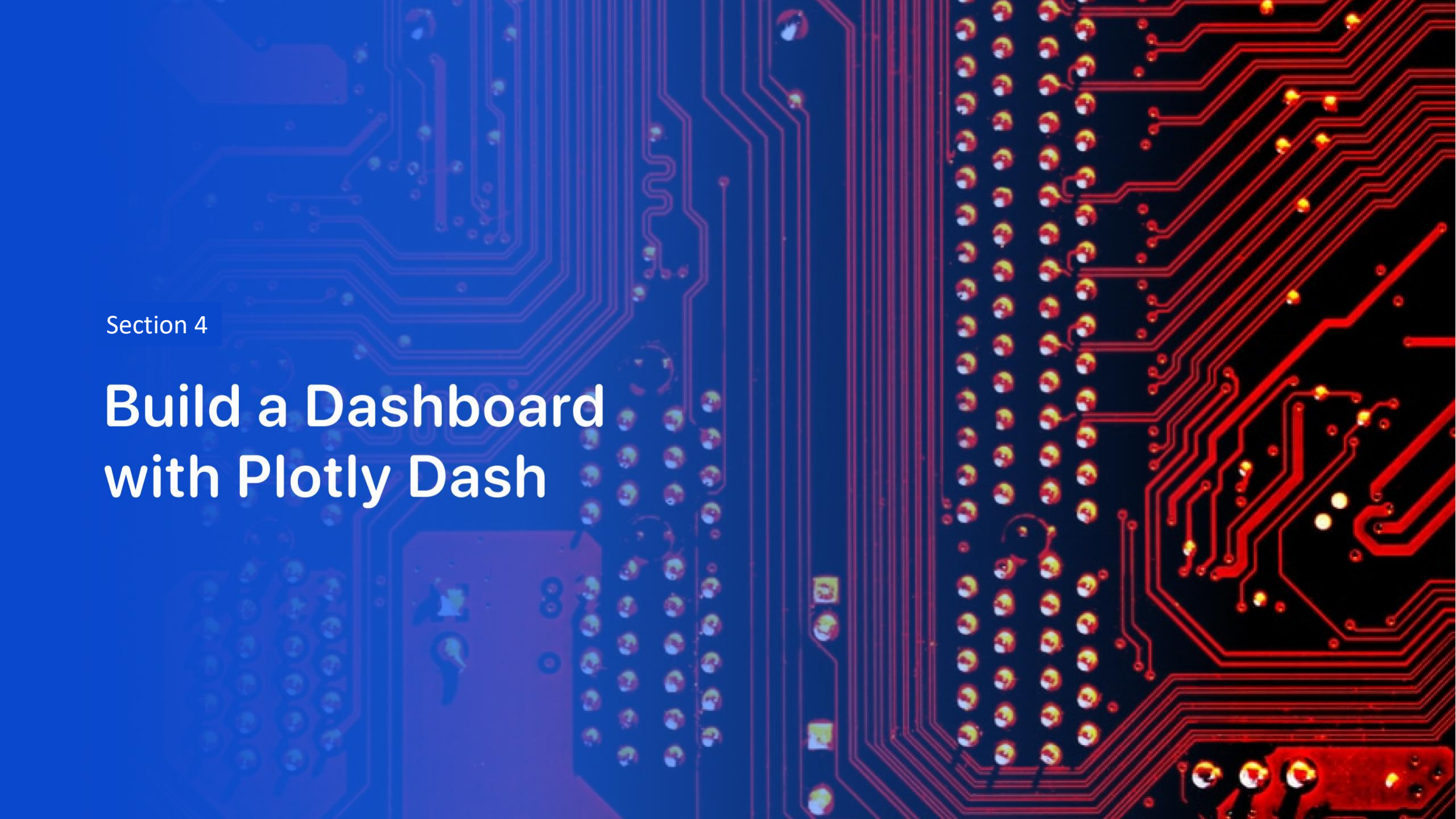
Launch Sites Proximities Analysis

Map of Launch Sites Locations

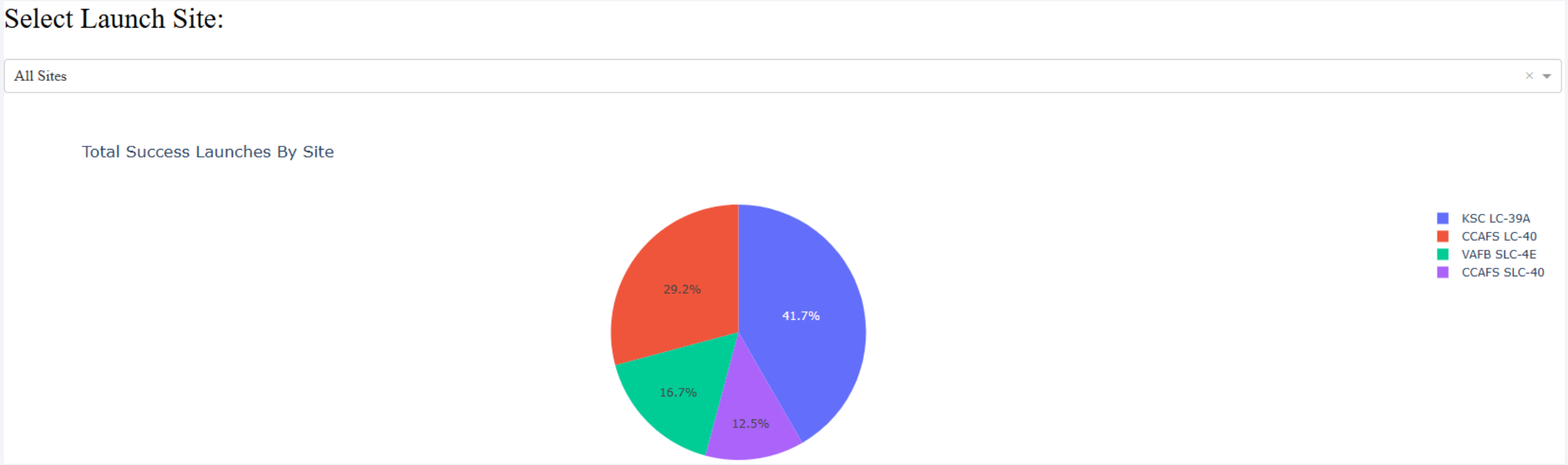
- This map shows the launch site locations depicted with dark circles.

Launch Sites and Outcomes Cluster Map

- This map shows both the location of the launch sites and the number of success/failed launches for each side.

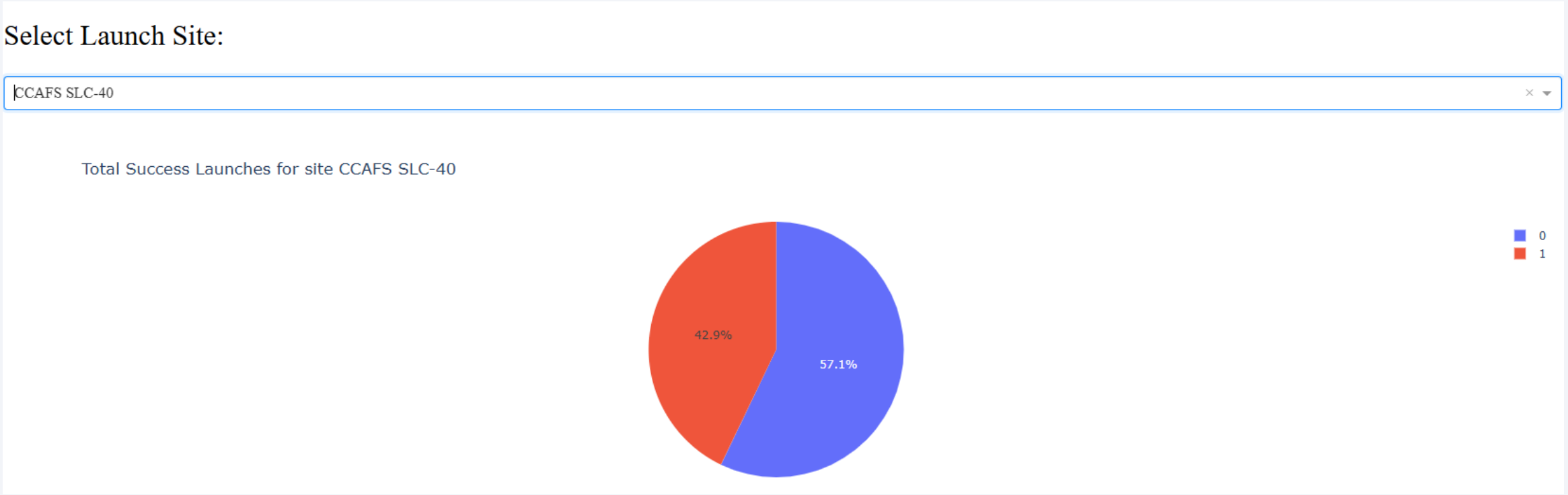

Section 4
Build a Dashboard with Plotly Dash

Launch Site Success Proportion



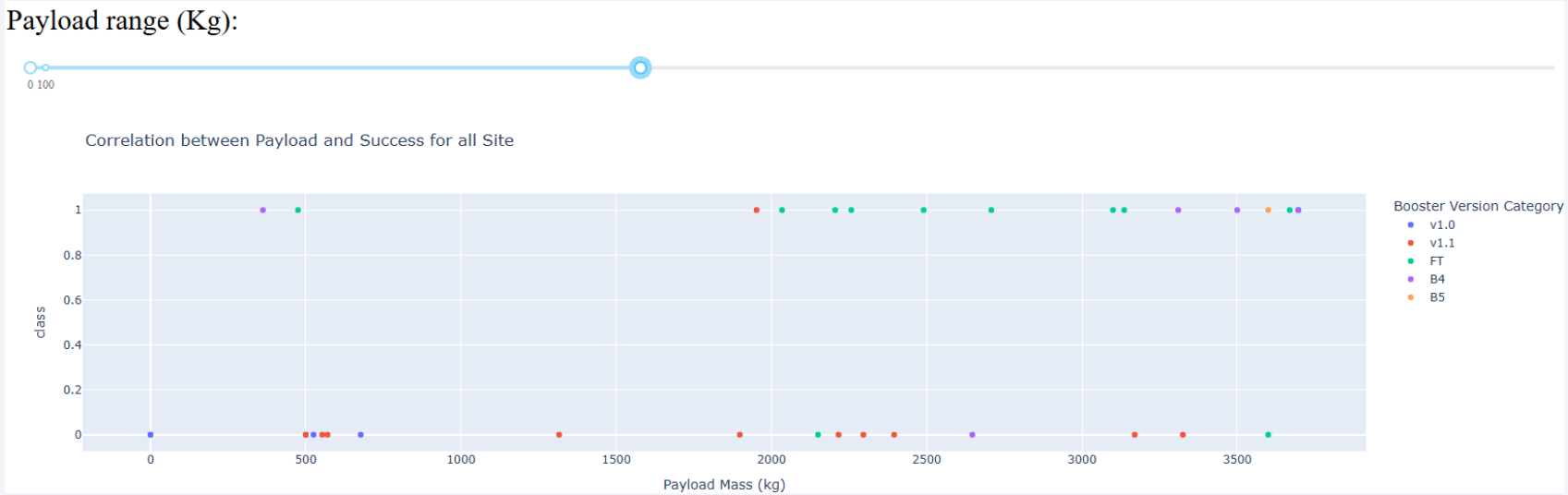
- This pie chart shows the proportion of successful launches for each site.

Success/Failure Proportion



- Proportion of success (1) and failure (0) for CCAFS SLC-40 with the highest success ratio.

Payload Mass Ranges and Success

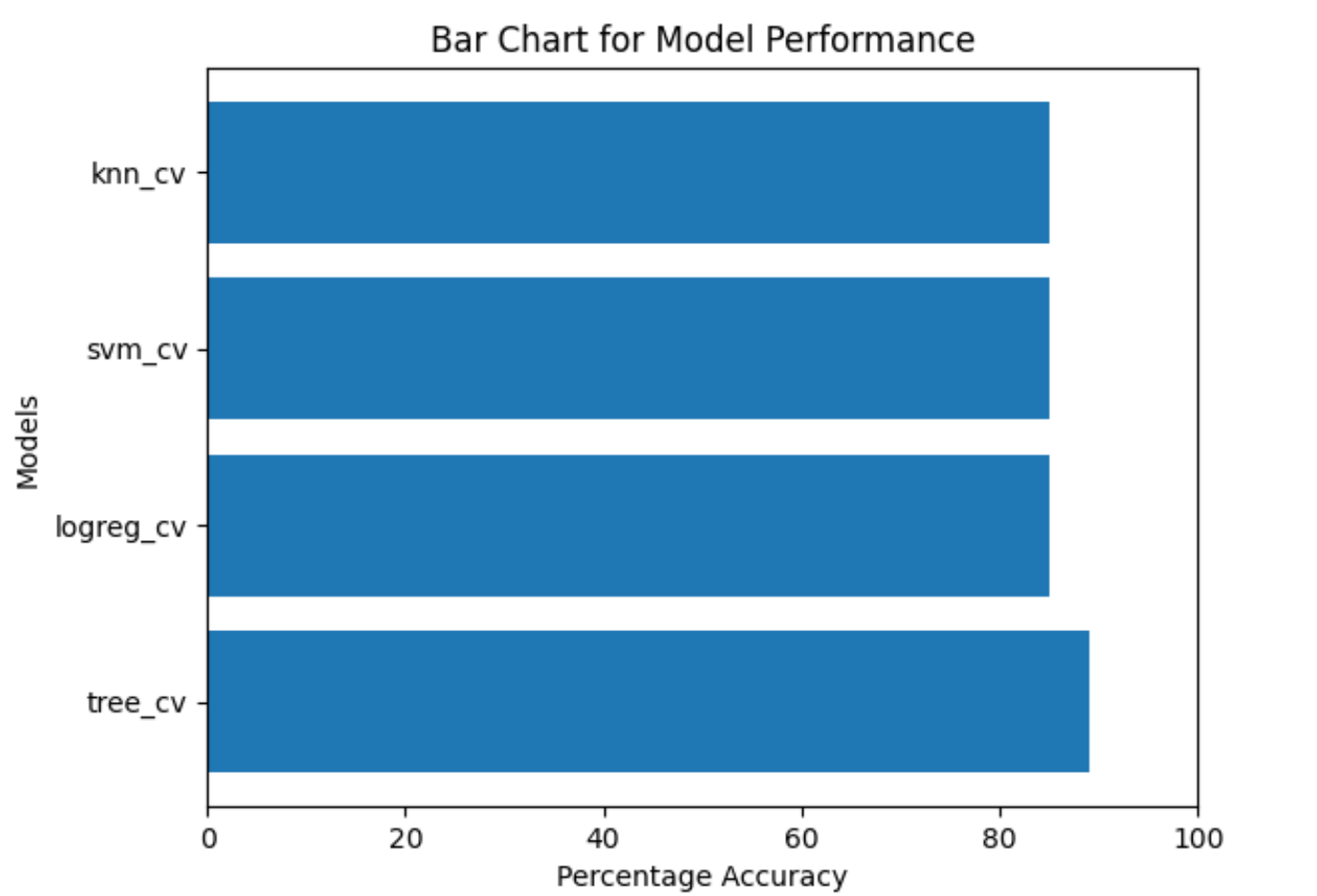


- The booster version FT (green) records more success for almost all ranges.

Section 5

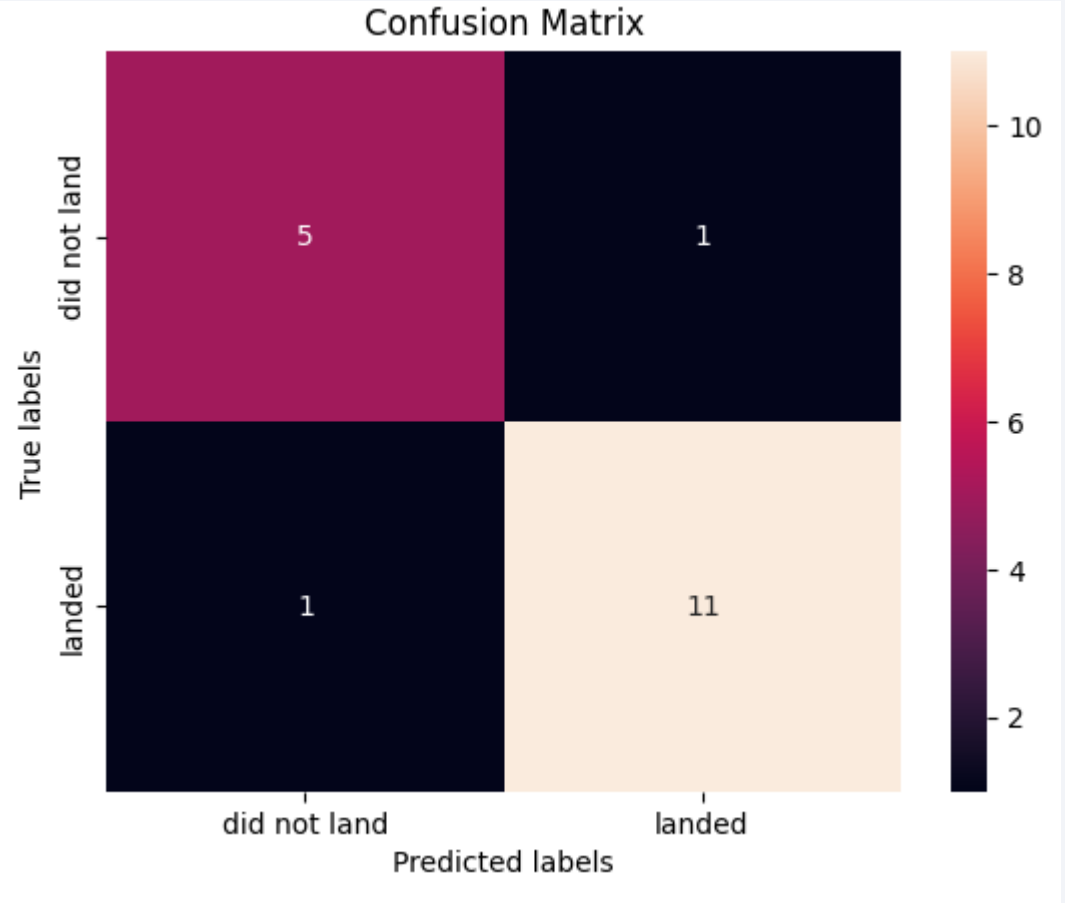
Predictive Analysis (Classification)

Classification Accuracy



- The Decision Tree Classifier model (tree_cv) is the best model with accuracy of 89%

Confusion Matrix



- It correctly predicted 11 (TP) and 5 (TN) while incorrectly predicting 1 for FP and FN.
- This is much better performance compared to the other models considered.

Conclusions

- The exploratory data analysis identified distinct patterns in launch frequency, payload characteristics, and mission outcomes across different sites and booster versions, offering a comprehensive understanding of SpaceX operational trends.

- The development of an interactive Folium map provided a clear geographic representation of launch locations and performance variations, enhancing the interpretability of spatial factors influencing mission success.

- The classification modelling demonstrated that launch outcomes can be reliably predicted using mission-related features, with performance substantially improved through systematic hyperparameter optimization using GridSearchCV.

- The final selected model exhibited strong predictive capability, indicating its potential usefulness for supporting future mission planning and informed decision-making within similar aerospace operational contexts.

Appendix

TASK 12

```
models = [logreg_cv , svm_cv, tree_cv, knn_cv]
labels = ['logreg_cv' , 'svm_cv', 'tree_cv', 'knn_cv']
model_accuracy_score = {'Models':[], 'Accuracy':[]}

for i, model in enumerate(models):
    model_accuracy_score['Models'].append(labels[i])
    model_accuracy_score['Accuracy'].append(np.round(model.score(X_test, Y_test), 3))

df_score = pd.DataFrame(model_accuracy_score).sort_values(by='Accuracy', ascending=False)

plt.barh(df_score.Models, df_score.Accuracy*100)
plt.title('Bar Chart for Model Performance')
plt.ylabel('Models')
plt.xlabel('Percentage Accuracy')
plt.xlim(0, 100)
plt.show()
```


The above code snippet was used to calculate and plot the accuracy of the models. I thought it is cool to share.

Thank you!