IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Arunava Kumar Chakraborty
04.10.2021

Outline

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

Executive Summary

- In this project, the SpaceX Launch data has been collected to perform the different exploratory data analyses and predictive experiments presented in section 1. The data pre-processing has been performed to prepare the data for further analysis. During the EDA, SQL and several visualization tools have been used to extract different information from the dataset. The predictive analysis has been performed using Machine Learning models to predict the future trend from the dataset.
- The results from the EDA have been presented in section 2. The different interactive visualization techniques and building a dashboard have been presented in section 4 and section 5 respectively. Section 6 represents the results of Predictive analysis on the dataset.

Introduction

- In the past few years, SpaceX is the most successful in designing, manufacturing and launching rockets and spacecraft. Since the company can reuse the first stage of rockets they claimed lower costs than other companies. A new company SpaceY wants to analyze the historical data of SpaceX so that they will be able to understand the advantages and disadvantages of using the first stage in terms of success along with fail rate and conduct cost-efficient launches.
- In this report, I will try to find the solutions for the following identified problems which will help SpaceY further for the detailed analysis.
  - The evaluation of the cost for each launch.
  - Visualizations of the collected information from the data.
  - Determine the overall success rate and the failure rate for each launch.
  - Predict the chance to reuse the first stage based on the Machine Learning approach.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data has been collected using the SpaceX API and Web Scraping methods.
- Perform data wrangling
  - The data wrangling has been performed to prepare the dataset for further experiments.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Different Machine Learning models have been used for predictive analysis.

Data Collection

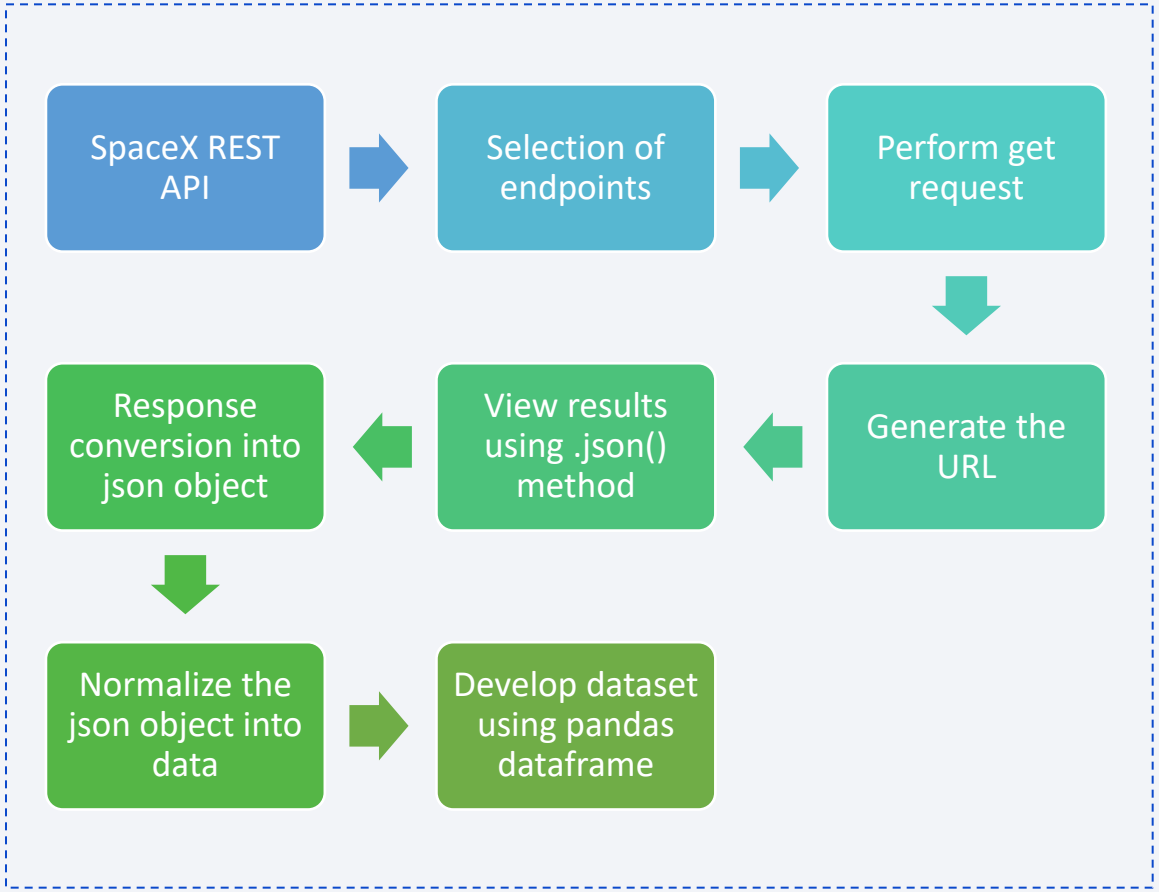
- The overall data collection has been conducted in two phases using SpaceX Rest API and Web Scraping methodology.
- **SpaceX Rest API**

  The SpaceX launch data have been collected using the SpaceX Rest API. The collected data consists of rocket information, delivered payloads, launch and landing specifications and landing outcomes.
- **Web Scraping**

  The Web Scraping technique has been used to collect the historical launches records of Falcon 9 and Falcon Heavy launches from Wikipedia. The BeautifulSoup Python package is required for implementing this phase.

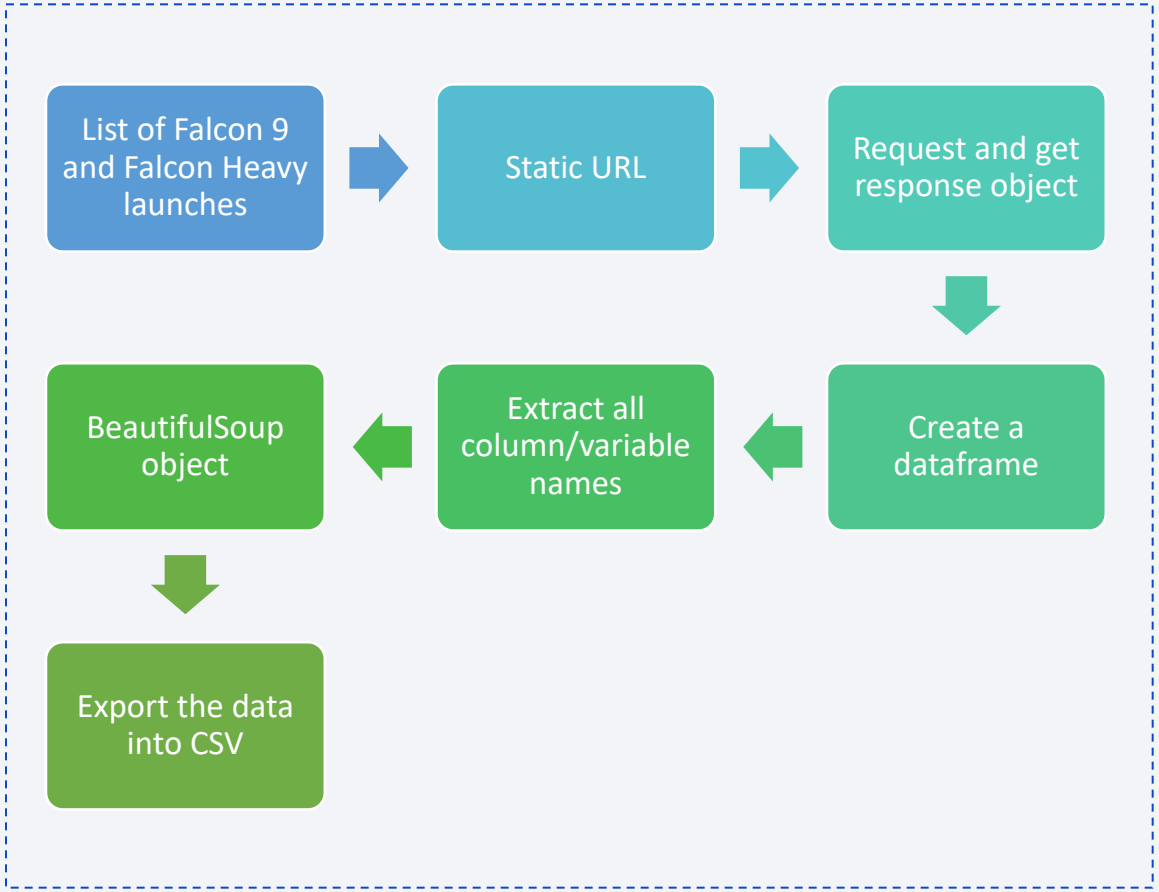
Data Collection – SpaceX API

- The data collection process using API has been presented using the following steps.
  - The URL (https://api.spacexdata.com/v4/launches/past) has been used to target a specific endpoint of the API to get past launch data.
  - The launch data have been fetched using the get request and .json() method will be used to view the result from the request. The response will be converted into a list of a JSON objects.
  - Using the pandas dataframe the database have been developed from the data.
- https://github.com/ArunavaKumar/IBM-Data-Science/blob/main/Applied%20Data%20Science%20Capstone/Week%201/01%20Data%20Collection%20API.ipynb



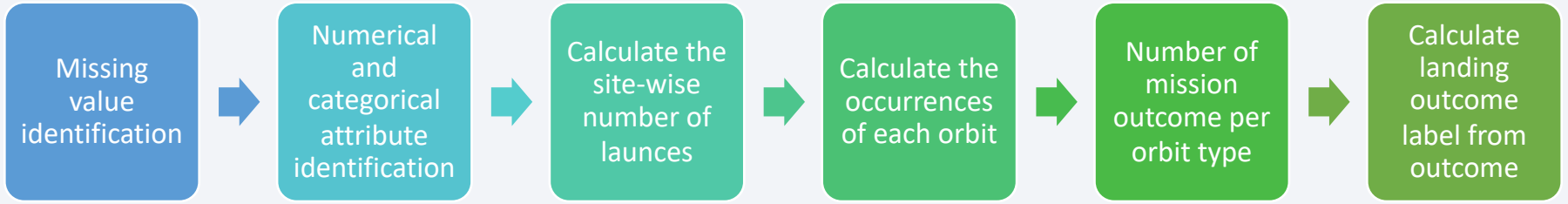
Data Collection - Scraping

- Here, the steps for data collection using web scraping have been presented.
  - The static URL for the Wikipage is https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
  - Request the HTML page to get response.
  - BeautifulSoup object has been used to extract information from the HTML tables.
  - Using the pandas dataframe the extracted data has been converted into CSV file.
- https://github.com/ArunavaKumar/IBM-Data-Science/blob/main/Applied%20Data%20Science%20Capstone/Week%201/02%20Data%20Collection%20with%20Web%20Scraping.ipynb



Data Wrangling

- The SpaceX dataset has been loaded into pandas dataframe. The attributes in the dataset are – FlightNumber, Date, BoosterVersion , PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude and Latitude.
- Here, the flowchart of Data Wrangling process has been presented.



- https://github.com/ArunavaKumar/IBM-Data-Science/blob/main/Applied%20Data%20Science%20Capstone/Week%201/03%20Data%20Wrangling.ipynb

EDA with Data Visualization

- Two catplots have been presented to visualize how the FlightNumber vs. PayloadMass and FlightNumber vs. LaunchSite relationships affect the launch outcome class.
- Two scatterplots represent PayloadMass vs. LaunchSite relationship taking hue as the class and Flights variables.
- The barplot presents the success rate of each orbit.
- The FlightNumber vs. Orbit and PayloadMass vs. Orbit relationships have been presented by two scatterplots.
- https://github.com/ArunavaKumar/IBM-Data-Science/blob/main/Applied%20Data%20Science%20Capstone/Week%202/02%20EDA%20with%20Visualization%20lab.ipynb

EDA with SQL

- Display the names of the unique launch sites in the space mission. List the 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS). Display the average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved. 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 failed landing_outcomes in drone ships, their booster versions, and launch site names for the year 2015. Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the dates 2010-06-04 and 2017-03-20, in descending order.
- https://github.com/ArunavaKumar/IBM-Data-Science/blob/main/Applied%20Data%20Science%20Capstone/Week%202/01%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- The circle object has been created for each launch site based on its coordinate (Lat, Long) values. The popup label of the object is set to Launch site names.
- The marker object has been created to plot markers on the map for each launch site based on the coordinates.
- The marker color has been selected as green if the launch was successful otherwise the color has been selected as red using marker_cluster object.
- The Polylines has been used to draw the lines between the launch site and the selected coordinates for the nearest railway point.
- https://github.com/ArunavaKumar/IBM-Data-Science/blob/main/Applied%20Data%20Science%20Capstone/Week%203/01%20Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

- The Drop-down Input Component, Range Slider, multiple callback functions have been added to prepare the dashboard. The dashboard contains the pie chart and the scatter plot visualizations.
- The pie chart has been used to visualize the success rate ratio for all the sites and the success along with failure rate ratios for an individual site. The scatter plot has been used to visualize the correlation between the Payload and success for all the sites.
- https://github.com/ArunavaKumar/IBM-Data-Science/blob/main/Applied%20Data%20Science%20Capstone/Week%203/02%20spacex_dash_app.py

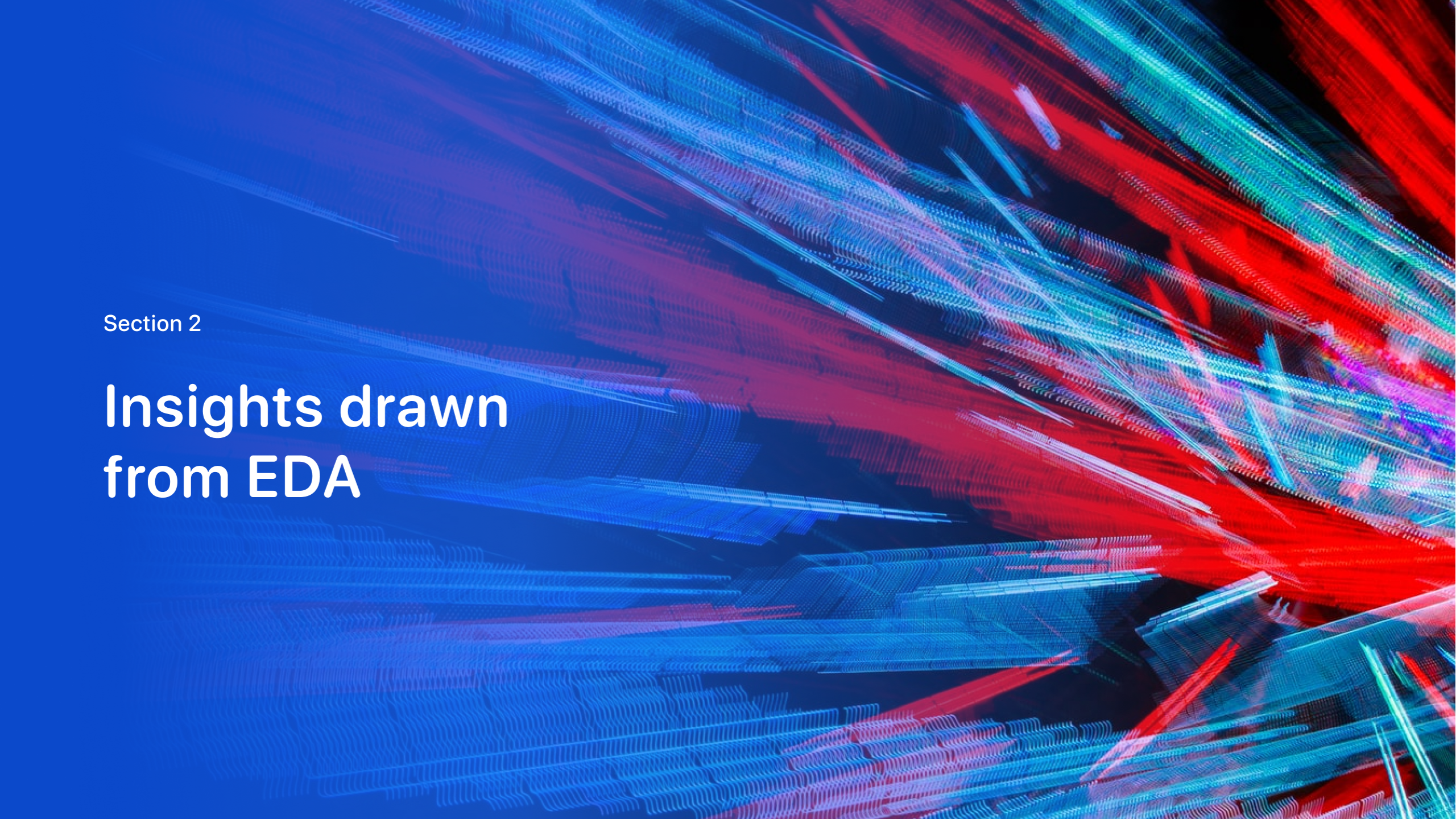
Predictive Analysis (Classification)

- The dataset has been analyzed to predict the landing outcome class ("1" for success and "0" for failure) of the launches. The Logistic Regression, SVM, Decision Tree and KNN models have been trained and tested based on the training and testing data. The GridSearchCV has been used to find the best parameter values for each of the models to improve the prediction accuracy.
- The flowchart presents the step by step model development process.

- https://github.com/ArunavaKumar/IBM-Data-Science/blob/main/Applied%20Data%20Science%20Capstone/Week%204/Machine%20Learning%20Prediction%20lab.ipynb

Results

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

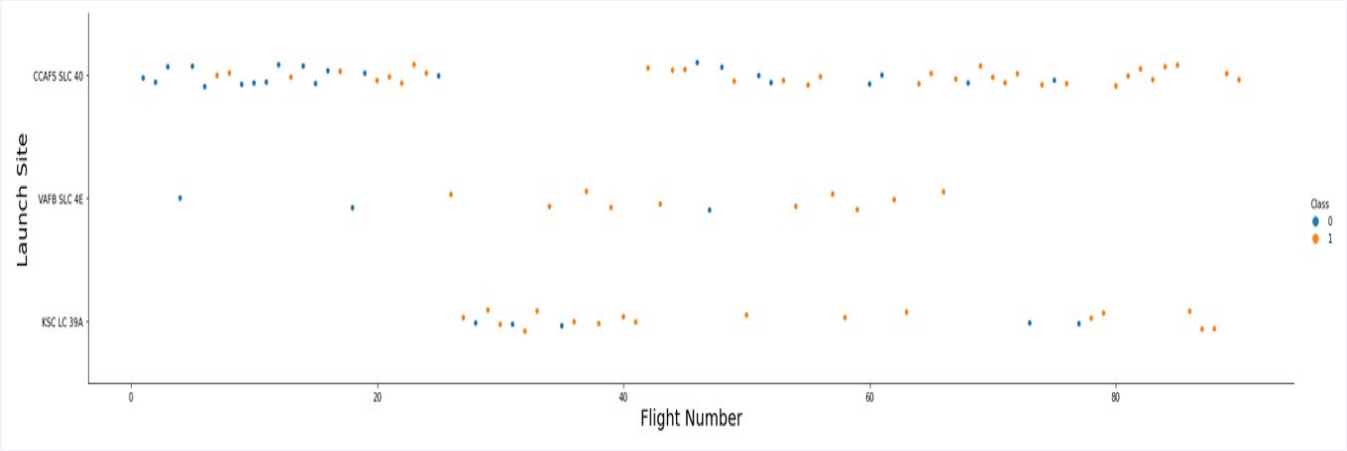


Section 2

Insights drawn from EDA

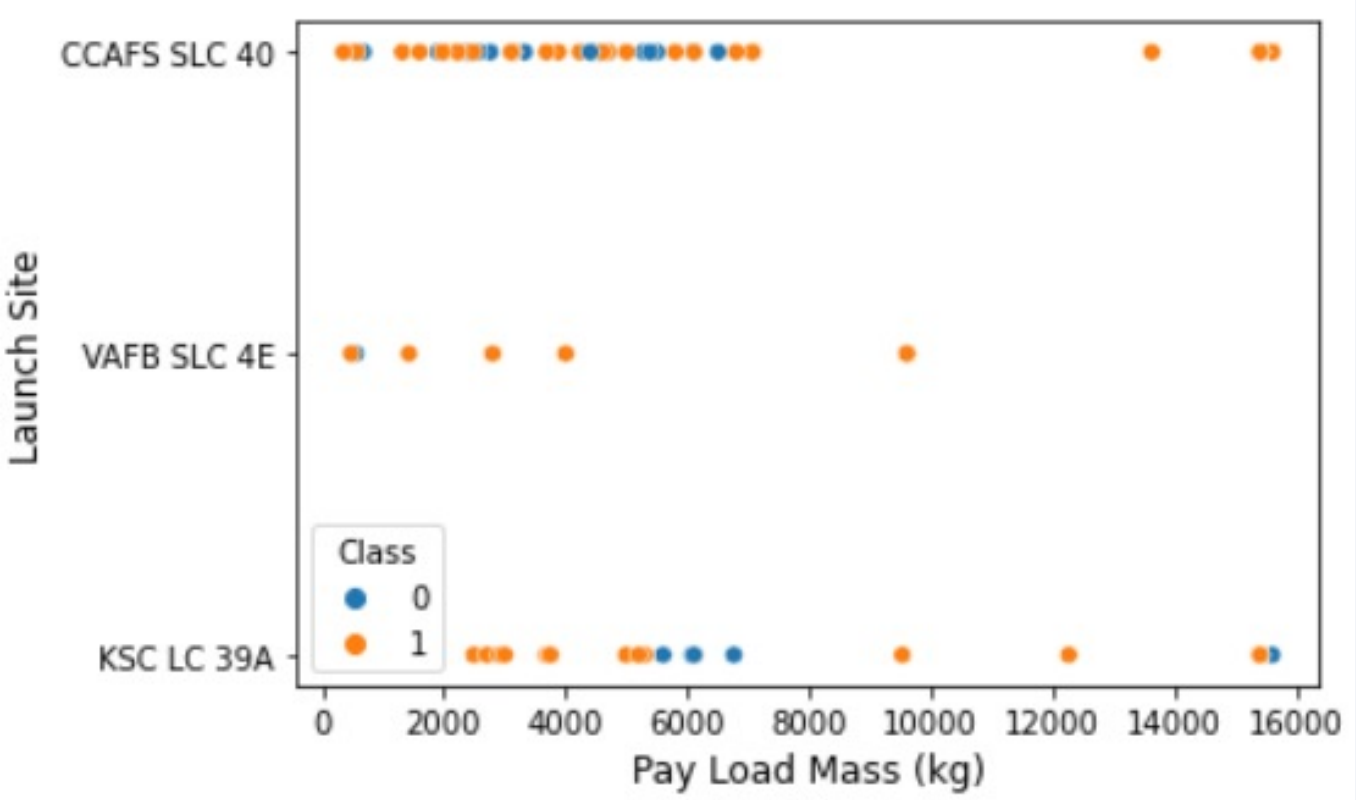
Flight Number vs. Launch Site

- The scatter plot (catplot) visualizes how the relation between Flight Number and Launch Site affects the landing outcome class.
- The class = 0 defines the failure for landing outcome
- Again the class = 1 define the success for landing outcome.



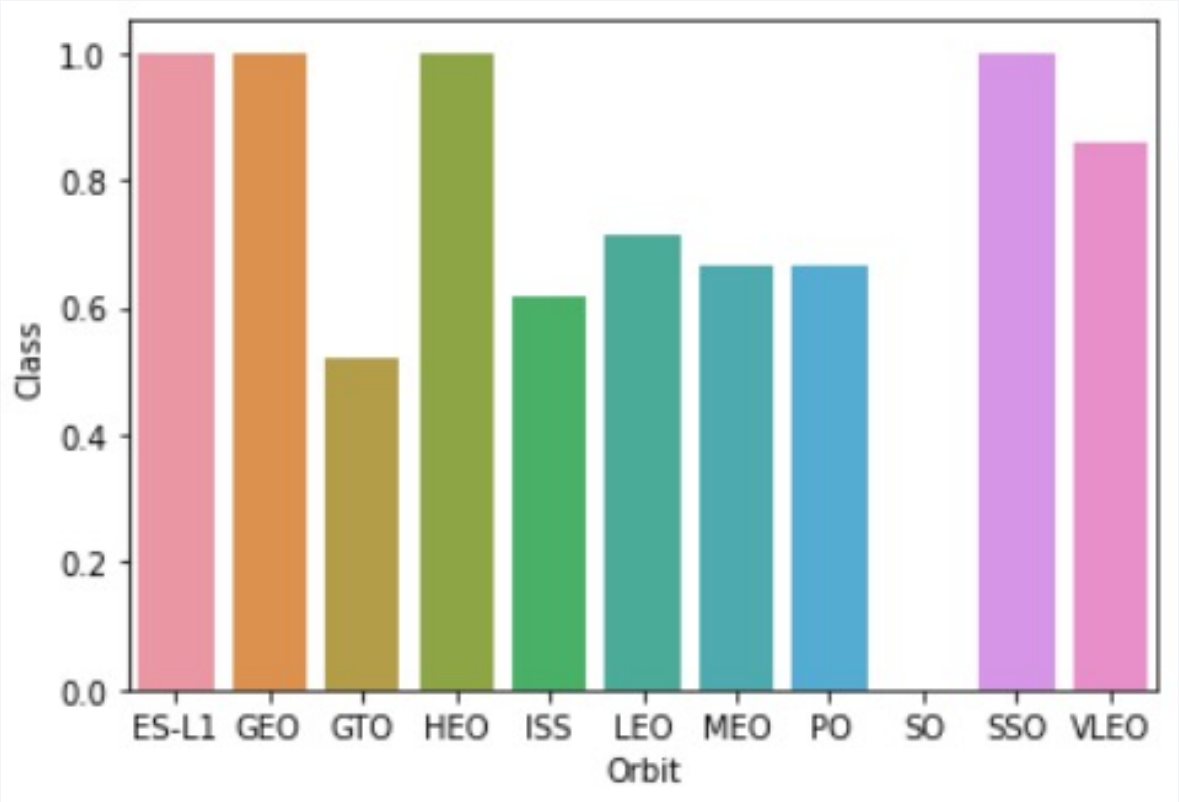
Payload vs. Launch Site

- The scatter plot visualizes how the relation between Launch Site and Payload affects the landing outcome class.
- The class = 0 defines the failure for landing outcome
- Again, the class = 1 define the success for landing outcome.



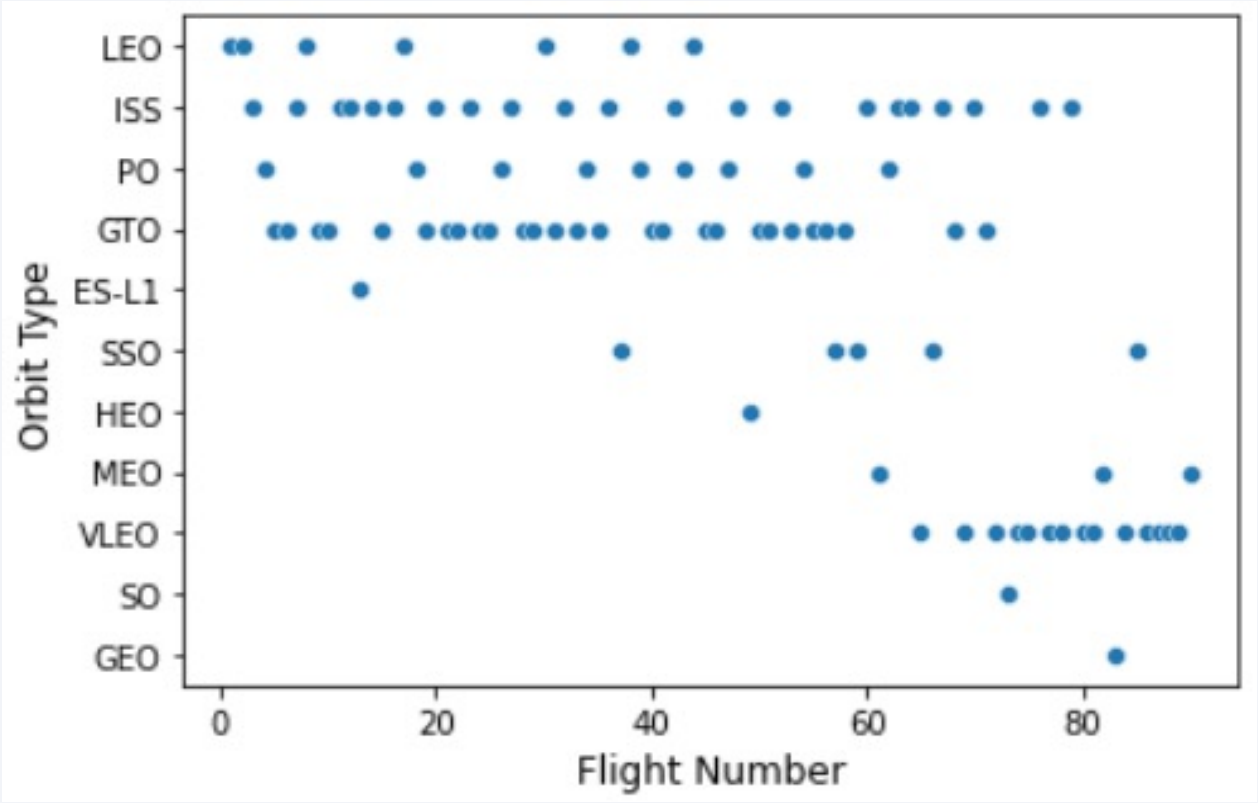
Success Rate vs. Orbit Type

- The bar chart presents the success rate for each orbit type.
- The Y-axis denotes the landing outcome class varying between 1 (Success) and 0 (Failure).
- The scatter plot helps to identify the orbits having high success rates.



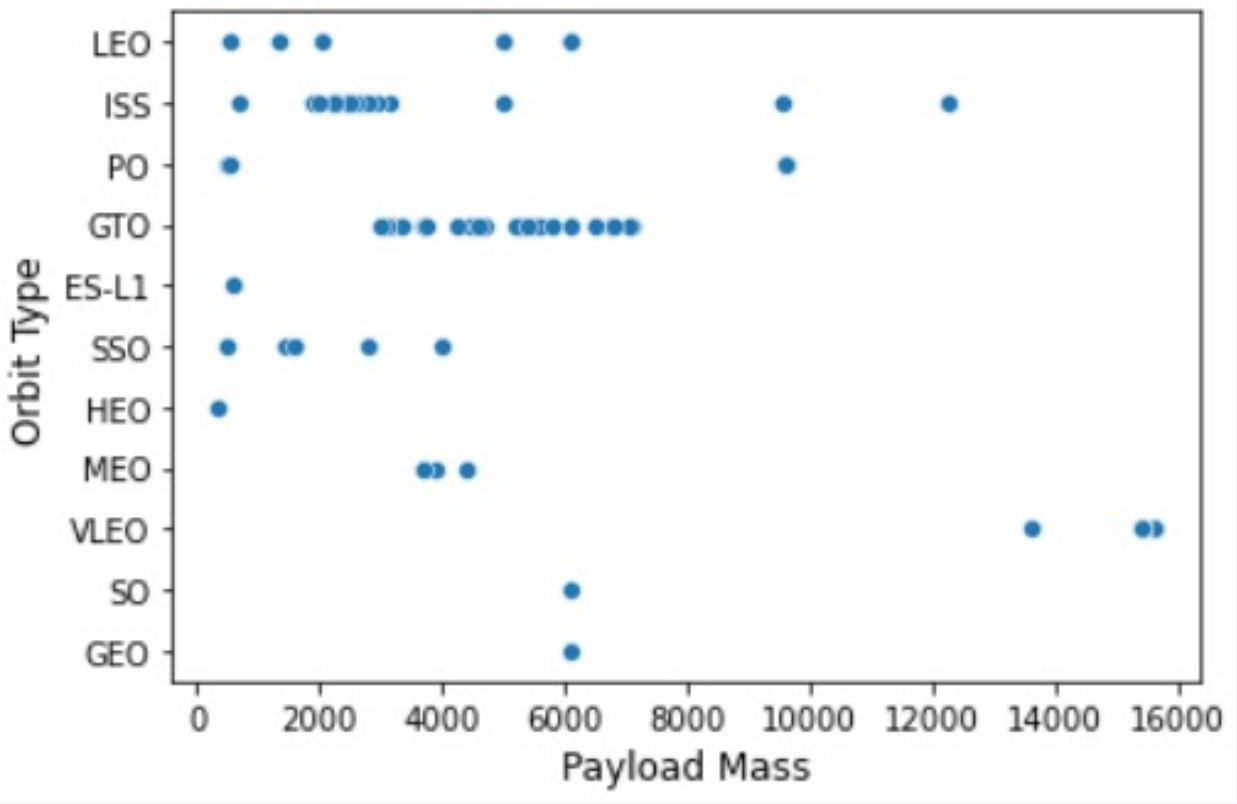
Flight Number vs. Orbit Type

- The scatter plot represents the Flight number vs. Orbit type relationship.
- The X-axis denotes the Flight Numbers. The Orbit Types have been denoted by the Y-axis.
- In the LEO orbit the Success appears related to the number of flights.



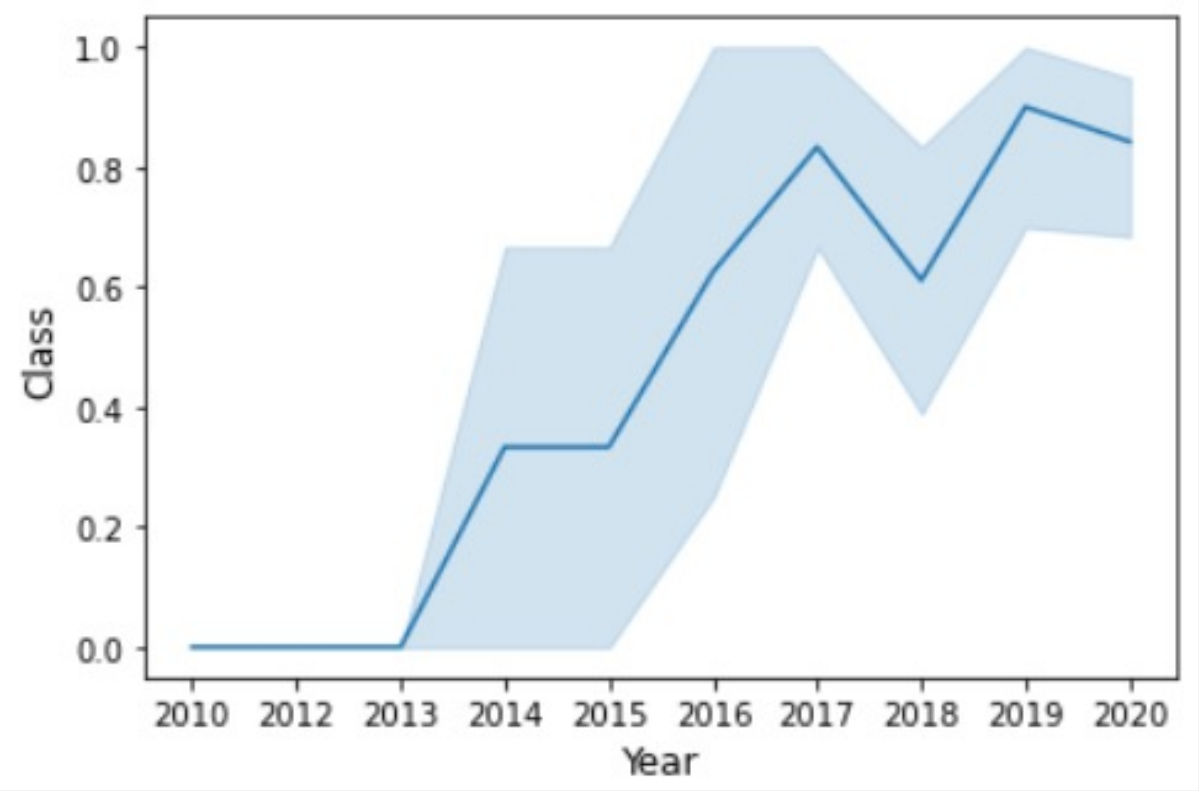
Payload vs. Orbit Type

- The Scatter has been presented the relationship between PayloadMass and Orbit variable.
- The X-axis defines the PayloadMass and the Y-axis defines the Orbit Type.
- Heavy payloads have a negative influence on GTO orbits and a positive on GTO and Polar LEO (ISS) orbits.



Launch Success Yearly Trend

- The line chart represents the yearly trend of launch success.
- The X-axis denotes the Year whereas the Y-axis denotes the landing outcome class.
- It can be observed that the launch success rate since 2013 kept increasing till 2020.



All Launch Site Names

- The SQL query has been used to extract the names of the unique launch sites in the space mission.
- The SELECT statement with DISTINCT keyword has been used. The query extracted four unique Launch Sites from the table.

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

Launch Site Names Begin with 'CCA'

- The SQL query has been used to extract 5 records where launch sites begin with the string 'CCA'.
- The SELECT statement along with the WHERE condition has been used. The LIKE and LIMIT keywords have been used for string matching and record limiting purposes respectively.

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

Total Payload Mass

- The SQL query has been used to extract the total payload mass carried by boosters launched by NASA (CRS).
- The SELECT statement along with the WHERE condition has been used. The SUM() function has been used to calculate the total payload mass. The LIKE keyword has been used for string matching purposes.
- The query presents the total payload mass from the SpaceX dataset as the output.

total_payload_mass
107010

Average Payload Mass by F9 v1.1

- The SQL query has been used to extract average payload mass carried by booster version F9 v1.1.
- The SELECT statement along with the WHERE condition has been used. The AVG() function has been used to calculate the average payload mass. The LIKE keyword has been used for string matching purposes.
- The query displays the average payload mass from the SpaceX dataset as the output.

Average_payload_mass
2534

First Successful Ground Landing Date

- The SQL query has been used to extract the date when the first successful landing outcome in ground pad was achieved.
- The SELECT statement along with the WHERE condition has been used. The MIN() function has been used to find the first date. The = operator has been used to match the string “Success (ground pad)”.
- The query presents the first date for successful landing outcome in ground pad.

First Date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The SQL query has been used to extract the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.
- The SELECT statement along with the WHERE condition has been used. The BETWEEN keyword has been used to set the payload mass range from 4000 to 6000. The query extracted four booster_version based on the given criteria.

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

Total Number of Successful and Failure Mission Outcomes

- The SQL query has been used to extract to calculate the total number of successful and failure mission outcomes.
- The SELECT statement along with the WHERE condition has been used. The COUNT() function has been used to calculate the number of total missions. The LIKE keyword has been used for string matching purposes.

landing__outcome	COUNT
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
Success	38
Success (drone ship)	14
Success (ground pad)	9

Boosters Carried Maximum Payload

- The SQL query has been used to extract the names of the booster which have carried the maximum payload mass.
- The SELECT statement along with the WHERE condition has been used. A subquery has been used in the condition part. The MAX() function has been used to calculate the highest payload mass.

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

2015 Launch Records

- The SQL query has been used to extract the failed landing_outcomes in drone ships, their booster versions, and launch site names for the year 2015.
- The SELECT statement along with the WHERE condition has been used. The LIKE keyword has been used for string matching purposes. The presents the Date, landing_outcome, booster_version and launch_site based on the given criteria.

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 SQL query has been used to rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the dates 2010-06-04 and 2017-03-20, in descending order.
- The SELECT statement along with the WHERE condition has been used. The >= and <= operators are used to set the range of date. The COUNT() function has been used to calculate the number of landing outcomes. The GROUP BY clause is used to get the count based on the landing_outcome.

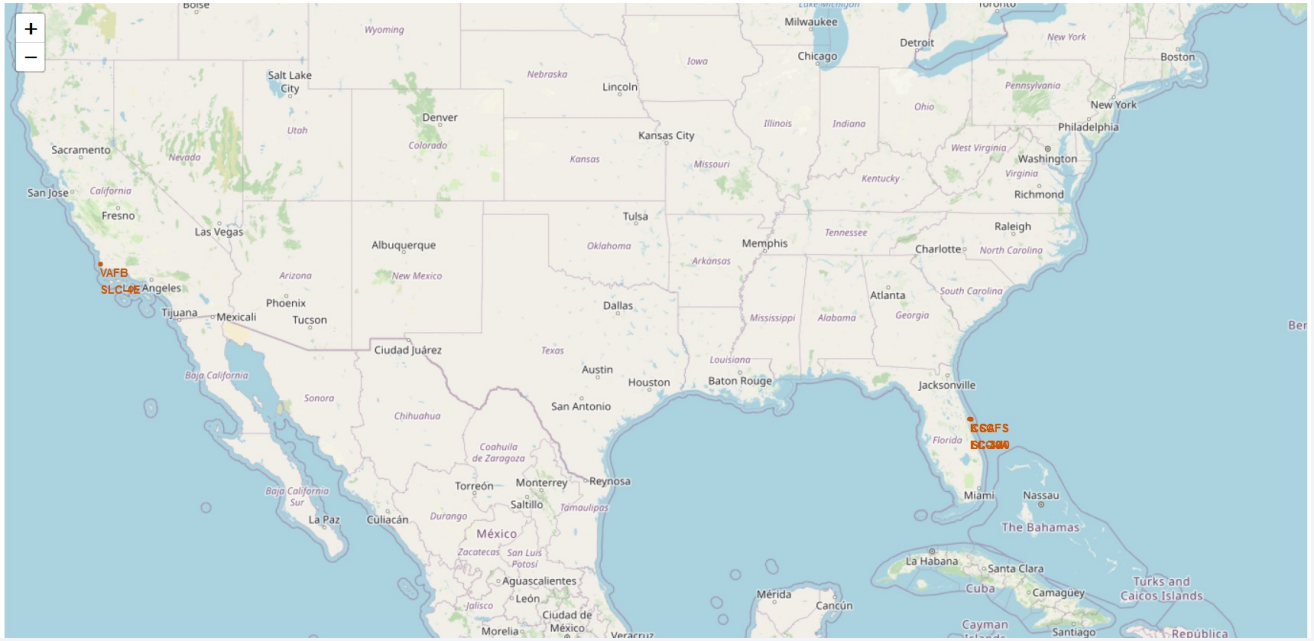
landing__outcome	COUNT
Failure (drone ship)	5
Success (ground pad)	3

Section 4

Launch Sites Proximities Analysis

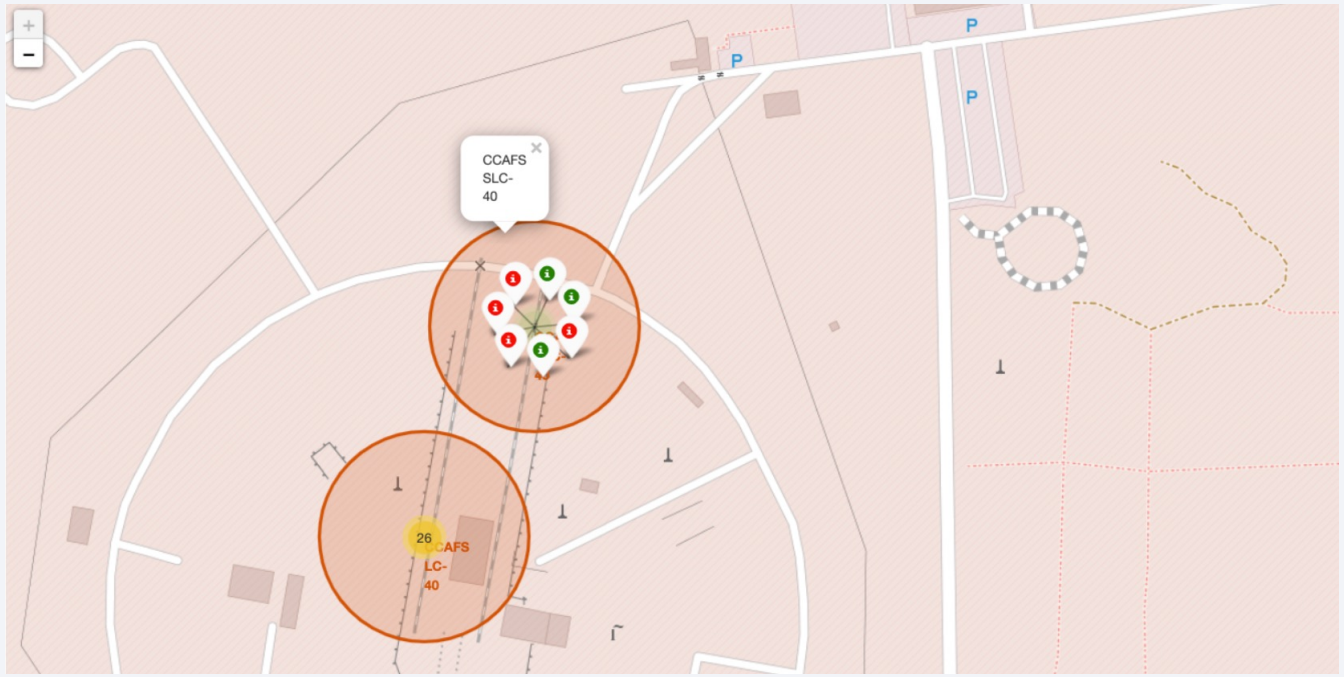
All Launch Sites in Map

- The Folium package has been used to add each site's location on a map using the site's latitude and longitude coordinates from the dataset.
- The map presents the location for each launch site with the red markers. We can explore more sites by zooming on the points as some of the sites are very close to each other.



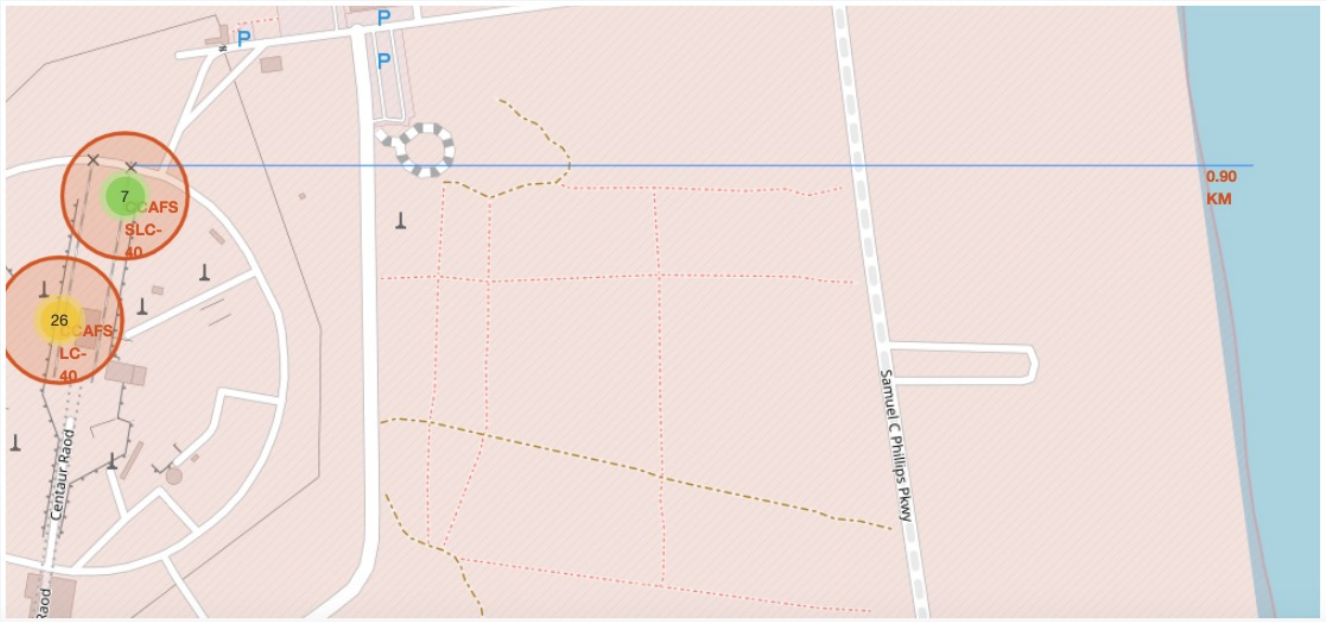
Success/Failed launches by All Launch Sites

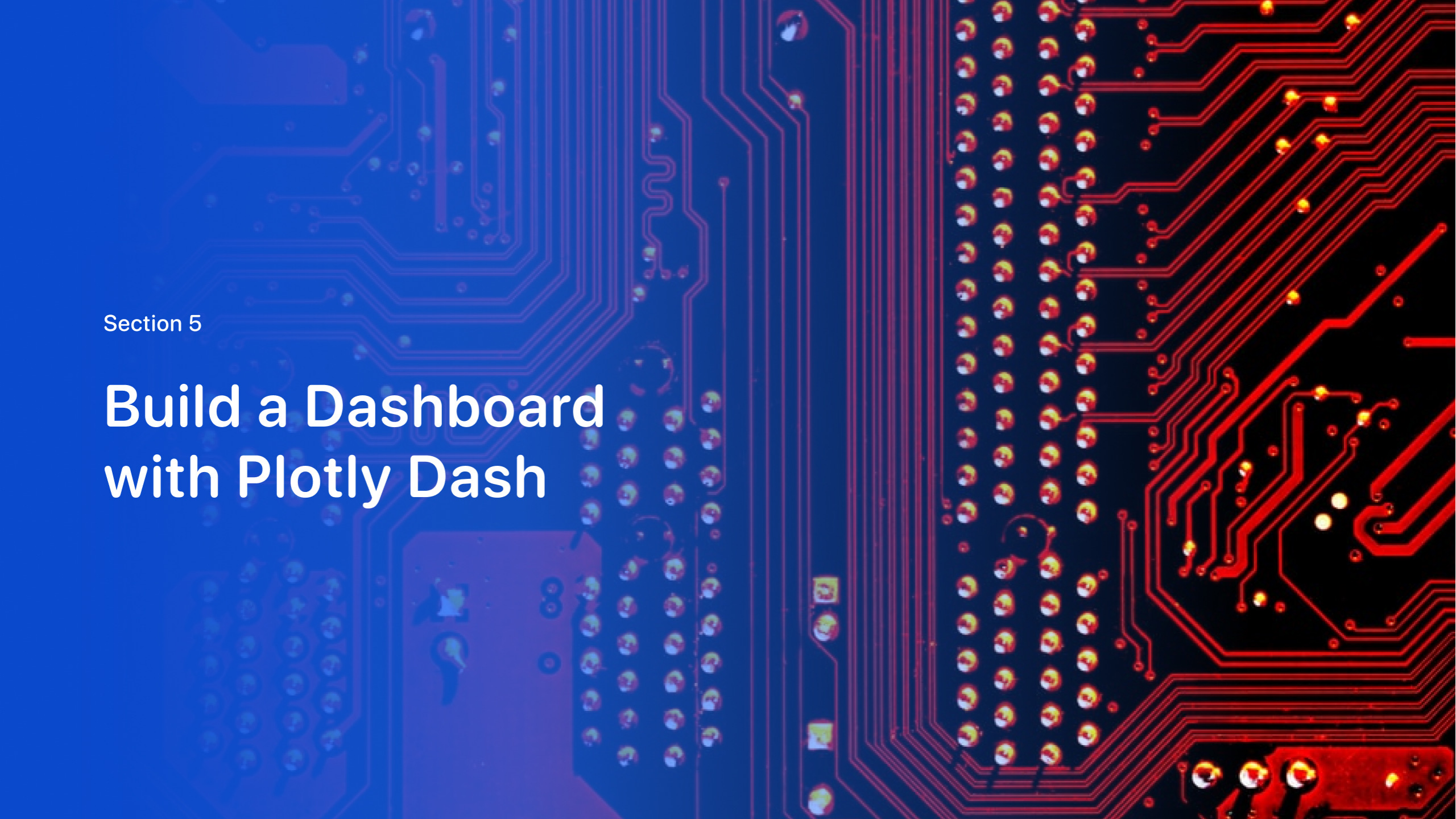
- In this map the success and failed launches have been presented for all the launch sites from the dataset.
- The green markers have been used to present the successful launches whereas the red markers present the failed launches.



Distance between a Launch Site and Railway Point

- This map presents the distance between the selected launch site and the nearest railway point.
- The distance has been measured by a function based on the coordinates of the two points.
- The Polyline object has been used to generate the distance line and further added to the map.



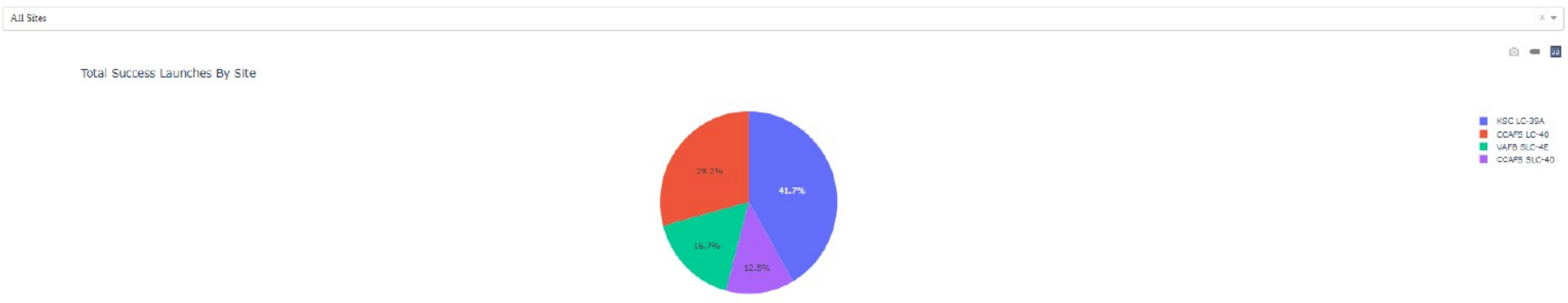


Section 5

Build a Dashboard with Plotly Dash

Success Rate for All Sites

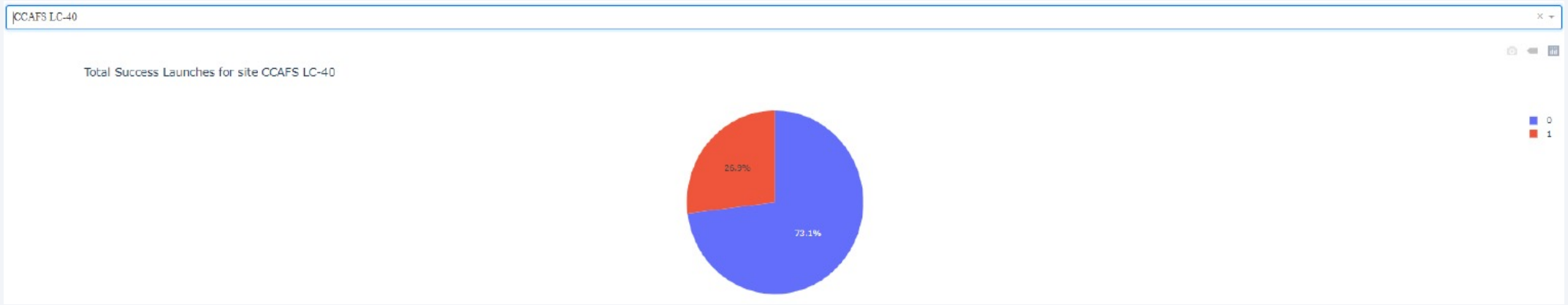
- The pie chart has been used to present the success rate for all the sites. From the visualization, it can be observed that CCAFS LC-40 has the highest 29.2% of success rate.



- In the pie chart, the success rate in terms of percentage has been presented for all the sites so that the launch site having the highest success rate can be identified.

Highest Success Ratio of a Launch Site

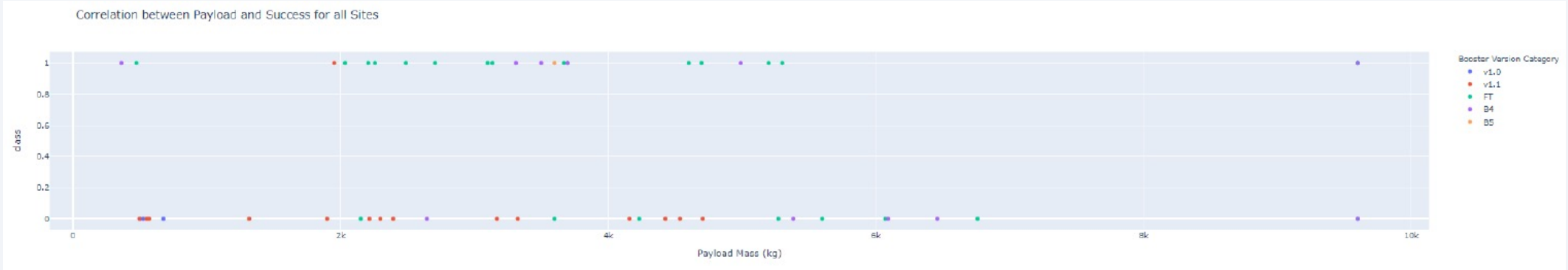
- This pie chart presents the success and failure rate of CCAFS LC-40 launch site which has the highest success rate among all other sites. From the visualization, it has been derived that the site individually achieved a 73.1% of success rate along with a 26.9% failure rate.



- In the pie chart the blue color has been used to present the successful launches from the dataset and the red color has been used to present the failure launches.

Payload vs. Launch Outcome Correlation

- The scatter plot presents the correlation between the payload mass and the launch outcome class (0 or 1) based on the different booster version categories.



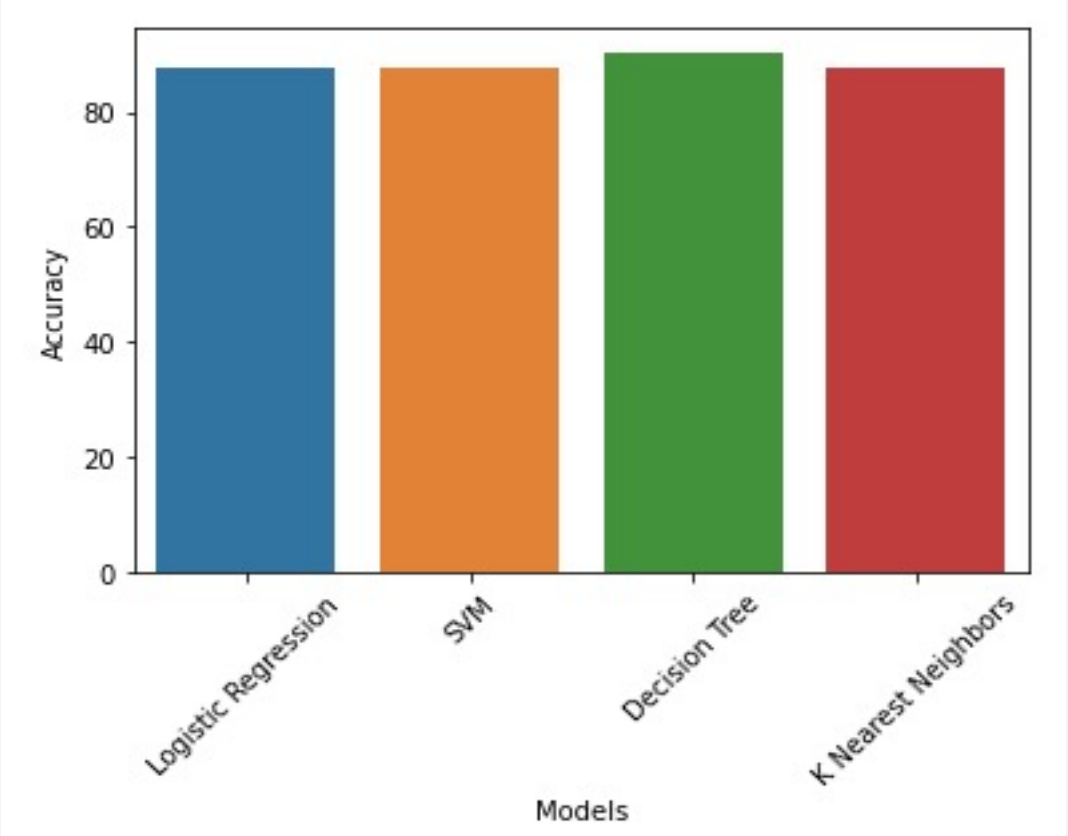
- The range slider has been used to generate the scatter plot to present the correlation. The correlation is higher for the several booster versions e.g. – v1.1, FT, B4 and B5.

Section 6

Predictive Analysis (Classification)

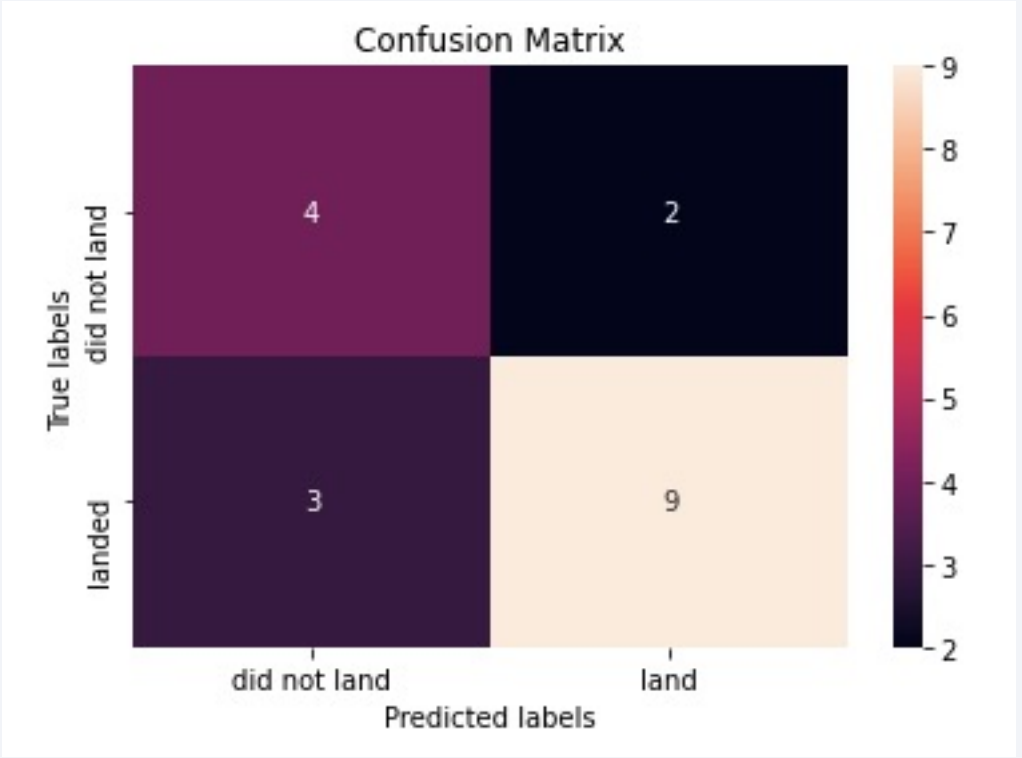
Classification Accuracy

- During the predictive analysis four Machine Learning based models have been used.
- The bar chart represents the performance metrics of the models to identify the best model.
- It can be observed that the Decision Tree classifier outperforms all other Machine Learning models and achieved 90.18% prediction accuracy.



Confusion Matrix

- The confusion matrix of the Decision Tree classifier (Best Model) has been presented here using heatmap by Seaborn Package.
- This helps to identify the classification and the misclassification of data points based on the true and predicted class labels.
- Here, True Positive (TP) = 9, True Negative (TN) = 4, False Positive (FP) = 3 and False Negative (FN) = 2.



Conclusions

- During the EDA it has been found that heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits. It has been observed that since 2013 the launch success rate kept increasing till 2020. Several SQL queries have been performed to extract certain information from the dataset.
- The Folium and Plotly have been used to generate several interactive visualizations to visualize certain information from the dataset.
- The different Machine Learning algorithms have been used for the predictive analysis to predict the future trend of landing outcomes from the dataset. During the analysis, the Decision Tree classifier performs better than other models by achieving 90.18% prediction accuracy.
- In future the EDA can be improved by using other interactive visualizations. The hyperparameter optimization for the machine learning models may enhance the prediction accuracy of these models.

Appendix

- Finding the best predictive model and visualize the model performance based on the prediction accuracy.

  Python Code:

```
Models = ['Logistic Regression', 'SVM', 'Decision Tree', 'K Nearest Neighbors’]

Accuracy = [round(logreg_cv.best_score_*100 , 2), round(svm_cv.best_score_*100 , 2),
round(tree_cv.best_score_*100 , 2), round(knn_cv.best_score_*100 , 2)]

model_df = pd.DataFrame()

model_df["Models"] = "”

model_df["Accuracy"] = ""

model_df['Models'] = Models

model_df['Accuracy'] = Accuracy

bar_plot = sns.barplot(x = 'Models', y = 'Accuracy', data = model_df)

bar_plot.set_xticklabels(bar_plot.get_xticklabels(), rotation=45)
```

Thank you!