SpaceX Falcon 9 Analysis and Prediction

Harshvardhan Jadhav

2 July, 2023

harshvardhanb.jadhav@gmail.com

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
  - Model Evaluation
- Discussion
  - Findings & Implications
- Conclusion

EXECUTIVE SUMMARY

- **Analysis and ML modeling**
  - Gathering data and importing it in data frame to perform analysis.
  - Perform EDA in Pandas data frame as well as by SQL queries, visualizing data to extract patterns.
  - Building an interactive dashboard using Plotly Dash and calculating distances using Folium by generating interactive maps.
  - Predicting the first stage land success by Machine learning algorithms with hyper-parameter tuning to find the best method.
- **Summary of results**
  - Collected data from API and web scraping and used it further for preprocessing.
  - Preprocessed and cleaned data which made EDA and SQL querying possible.
  - Fed the data into machine learning models by hyper-parameter tuning resulting in sufficiently usable models.

INTRODUCTION

- **Falcon 9** by SpaceX is a rocket launch especially known for its **reuse** of first stage after launch, which cuts down the costs of launches if first stage has landed successfully unlike other rocket launches which mostly do not reuse the first stage boosters.
  - Using this data from the SpaceX API, the cost of the launch can be determined from successfully predicting if first stage will **land** or **fail to land**.
  - The information produced can be used as per the nature of the objective undertaken by the organization.
- **Objective for analysis**:
  - Predicting successful landing of 1st stage boosters.
  - Insights that can help identify relationships between features
  - Productive use-cases which can help minimize/maximize for cost/output respectively.

METHODOLOGY

- **Data collection and preparation**: Python Pandas for data analysis and manipulation. Data collected using spaceX API and web scraping.
- **EDA and analysis**: Matplotlib / Seaborn visualization packages to find patterns and insights about data. Queried data by SQL queries to explore data by manipulating records.
- **Generating interactive dashboard**: Used Plotly Dash, Folium libraries to generate maps and mark data points by clusters over launch sites.
- **Model building**: From sci-kit learn package modeled data with SVM, Classification Trees (Decision Trees), Logistic Regression and optimized their hyper-parameters using grid-search cross validation to find best model.

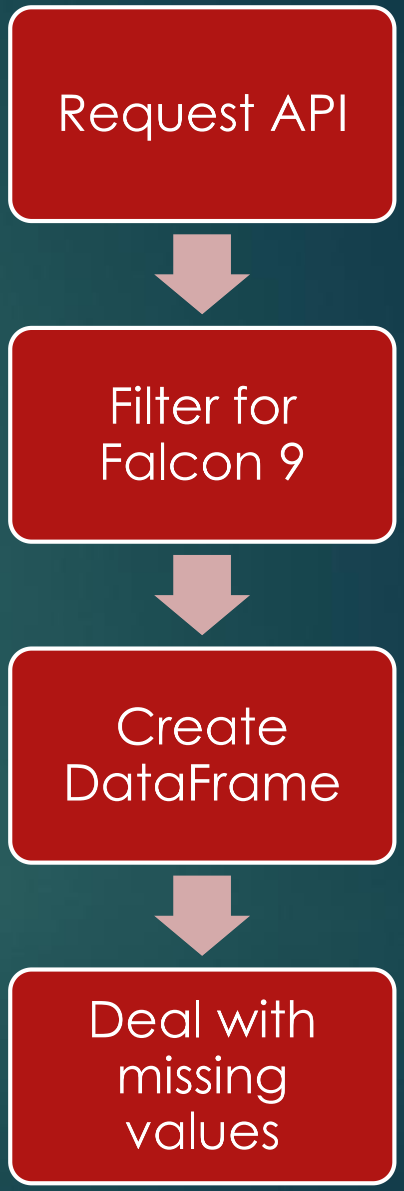
Data Collection API

- **Collecting Data:**
  - Requesting content from the API
  - Normalizing the data
  - Selecting a subset of data for required features
  - Create Lists and append data in them
  - Create a DataFrame for final data analysis

- **Filtering Data:**
  - Filtering records for only Falcon 9 launches

- **Deal with missing values:**
  - Impute the missing values with mean values

Github URL:
Data collection API source code



Data Collection Web Scraping

- **Collecting Data**:
  - Request Falcon 9 launch Wiki page from URL using BeautifulSoup
  - Extracting all columns from the HTML table header
  - Create a DataFrame by parsing the launch HTML tables

Github URL:
[Data Collection Web Scraping source code]

Data Wrangling Preprocessing

- **Preprocessing**
  - Examine the collected data.
  - Treat missing values if any.
  - Engineer features to create labels for feeding into ML models.

Github URL:
[Data Wrangling source code]

RESULTS

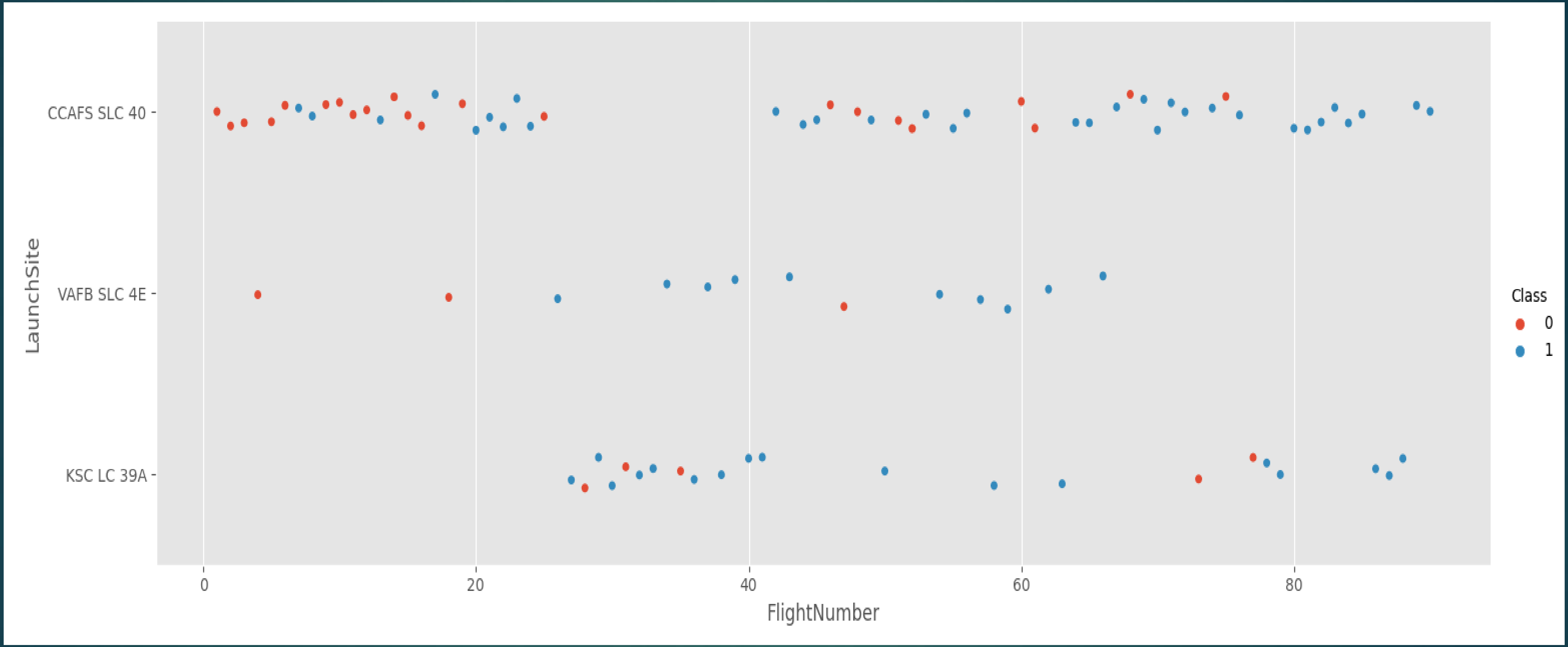
- **Summary of results**:
  - Performed EDA between features with visualization libraries like Seaborn and Matplotlib.
  - Performed EDA with SQL to answer queries.
  - Plotted interactive maps with Folium and added elements like markers, lines, labels and clusters to map data points.
  - Created a interactive Dashboard with Pandas Plotly and dash in a development environment, running the web app in a local server.

EDA with Matplotlib and Seaborn

- **Plots and Visualizations**:
  - Scatter plots are plotted to view the relationships between two features.
  - Bar plot is plotted to find the aggregate values of features at a glance.
  - Line plot is plotted to observe the trend over a time period.
  - Pie chart for understanding the composition of feature values.

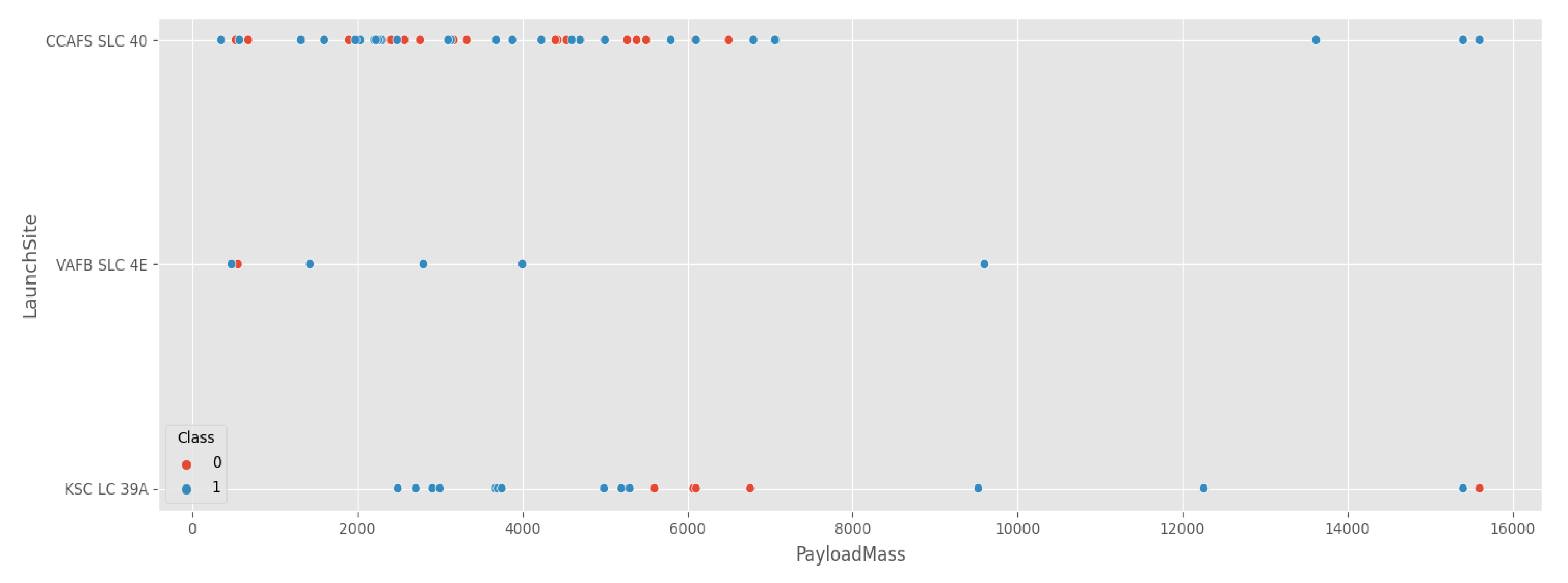
Github URL: [EDA with Data Visualization]

Flight Number by launch site



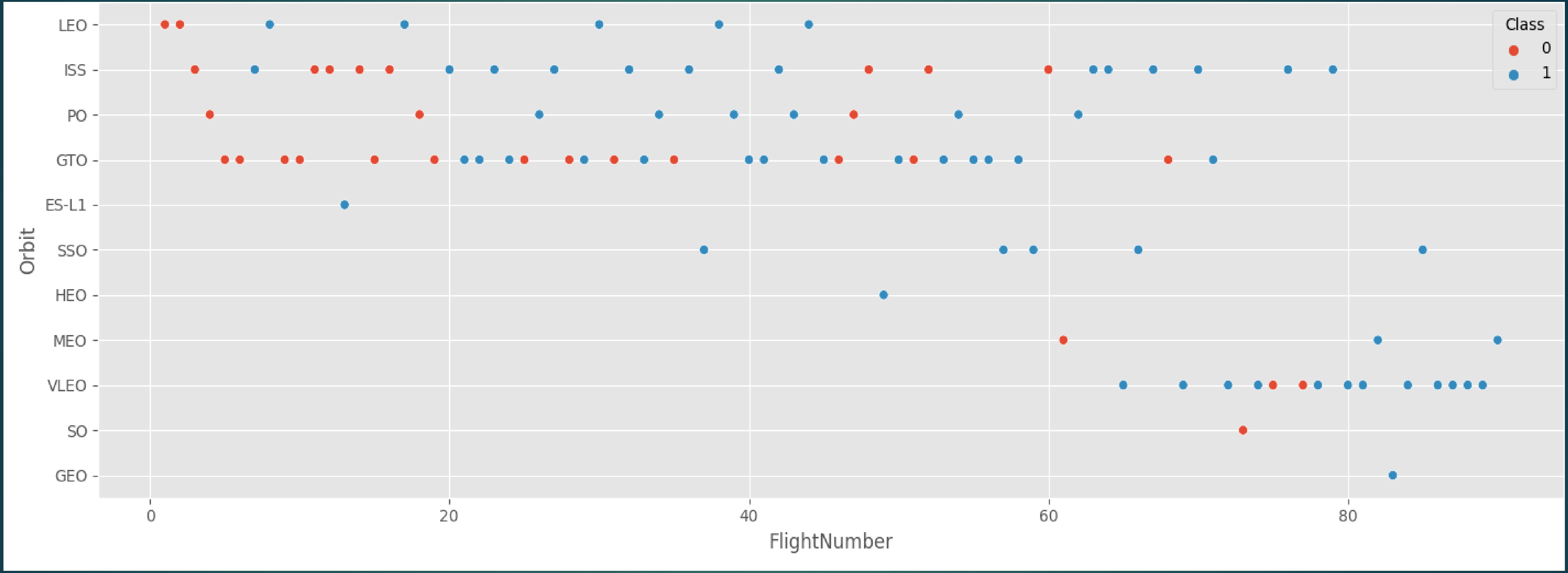
- **CCAFS SLC 40** and **KSC LC 39A** have launches after #70 except **VAFB SLC 4E** which has launches more around 20 to 60.

Payload by Launch Site



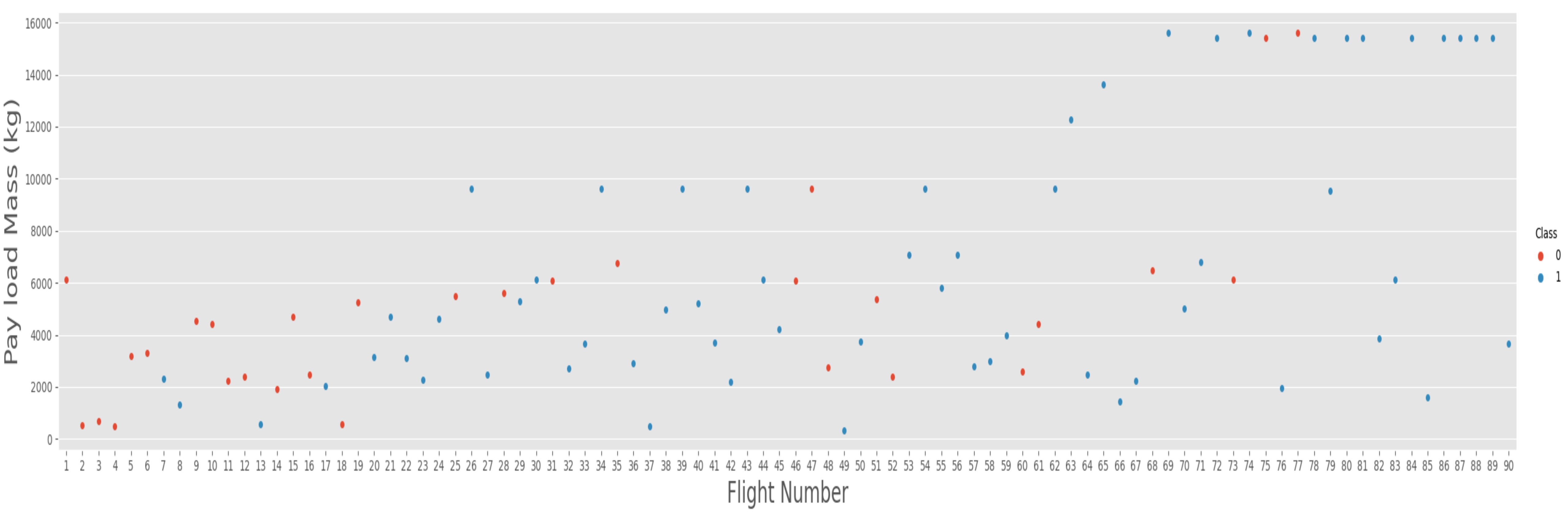
- **CCAFS SLC 40** and **KSC LC 39A** are the launch sites where the payload mass is greater than **10000 kg**.
- **VAFB SLC 4E** does not have launches where payload mass is greater than 10k and most are around **500 –**

Flight Number by Orbit



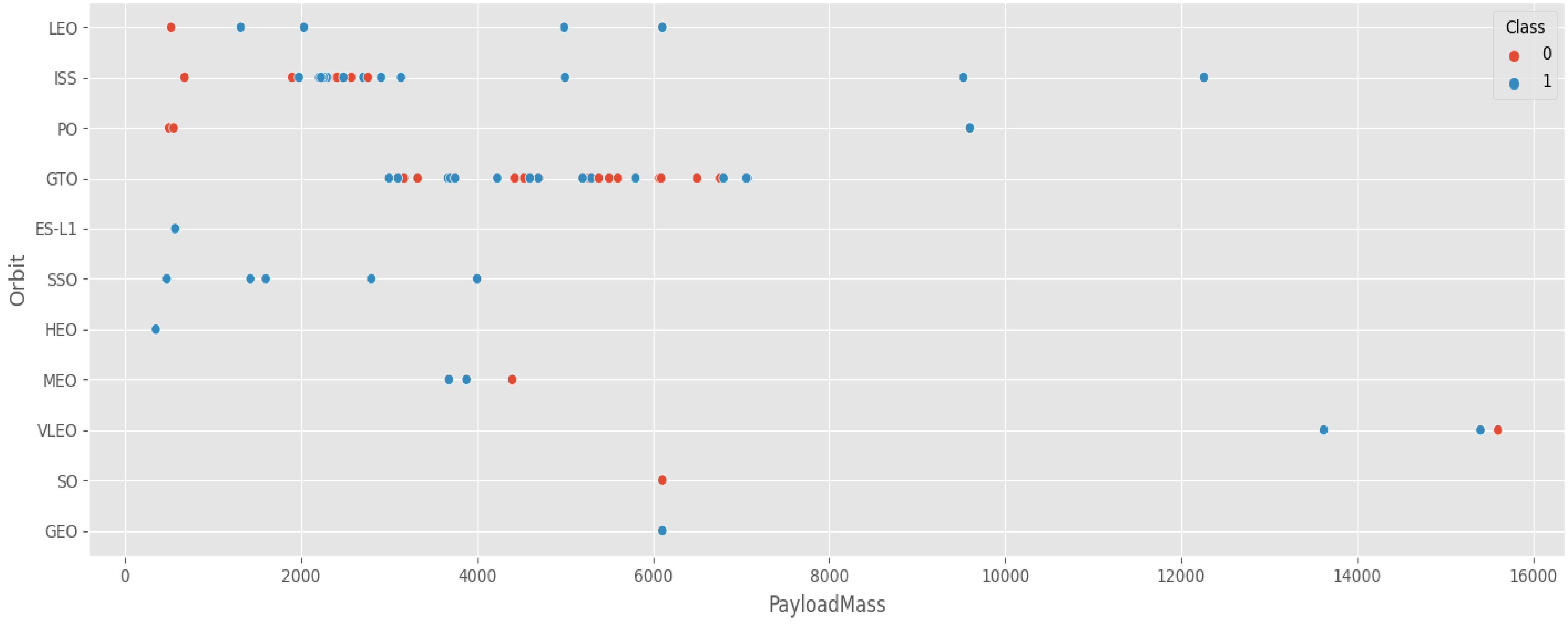
- Flight Number for **GEO, SO, VLEO, MEO, HEO, SSO** are mostly **after 35** and go **above 90.**
- **LEO, ISS, PO, GTO** mostly have flight numbers from the start to almost **80.**

Payload Mass (Kg) by Flight Number



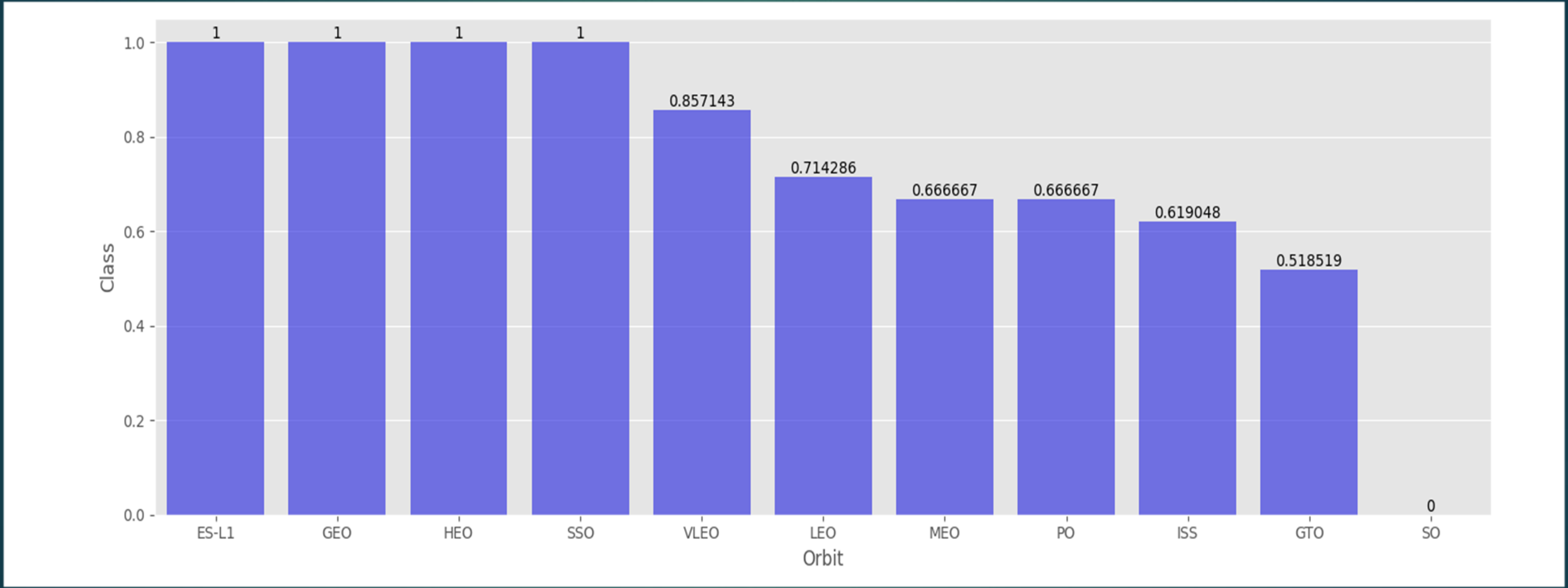
- **Lower Flight numbers** carry **lower payload** mass and mass increases with flights as observed.
- Flights after **#60** has much **heavier payload**.
- Launches with **heavier mass** are observed to be more **successful** rather than lower mass.

Payload Mass by Orbit



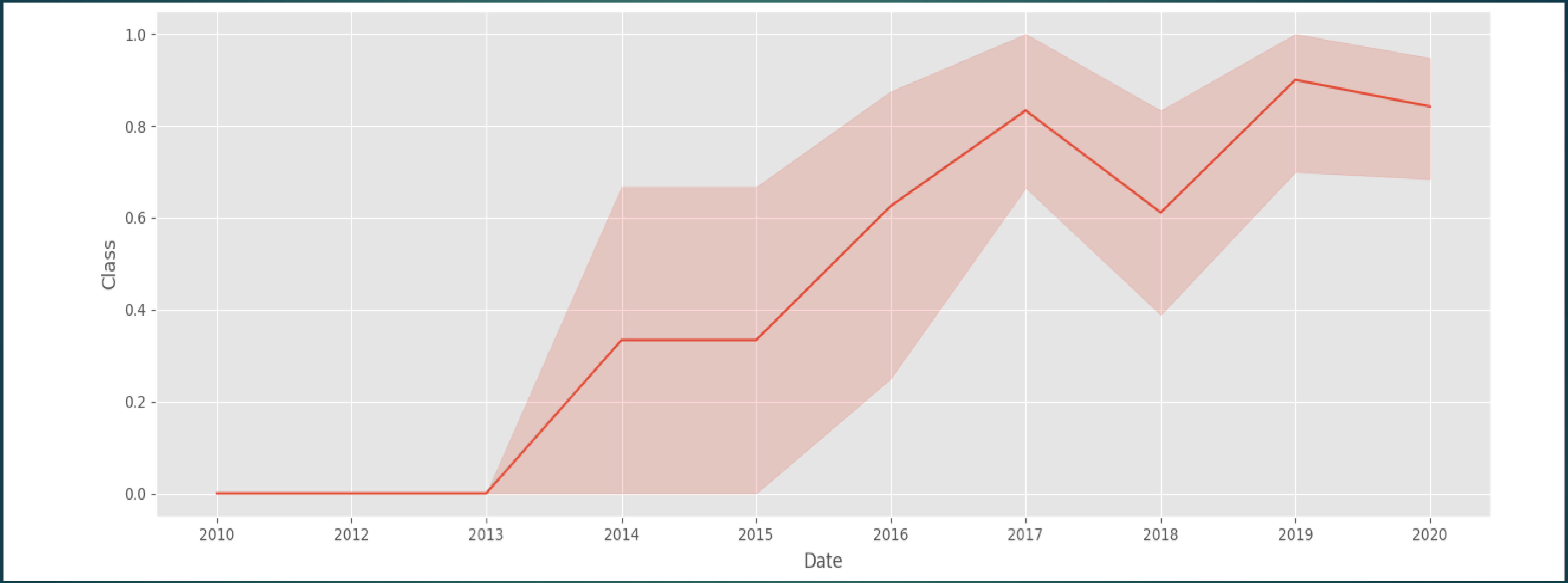
- **VLEO, ISS** are the orbits where payload mass is greater than **10000 kg**.
- **GTO** has most of the payload mass between **3000** to **7000 kg**.
- **LEO, ES-L1, SSO, HEO** has most payload mass between **100** to **4000 kg**.

Orbit success rate by Orbit Type



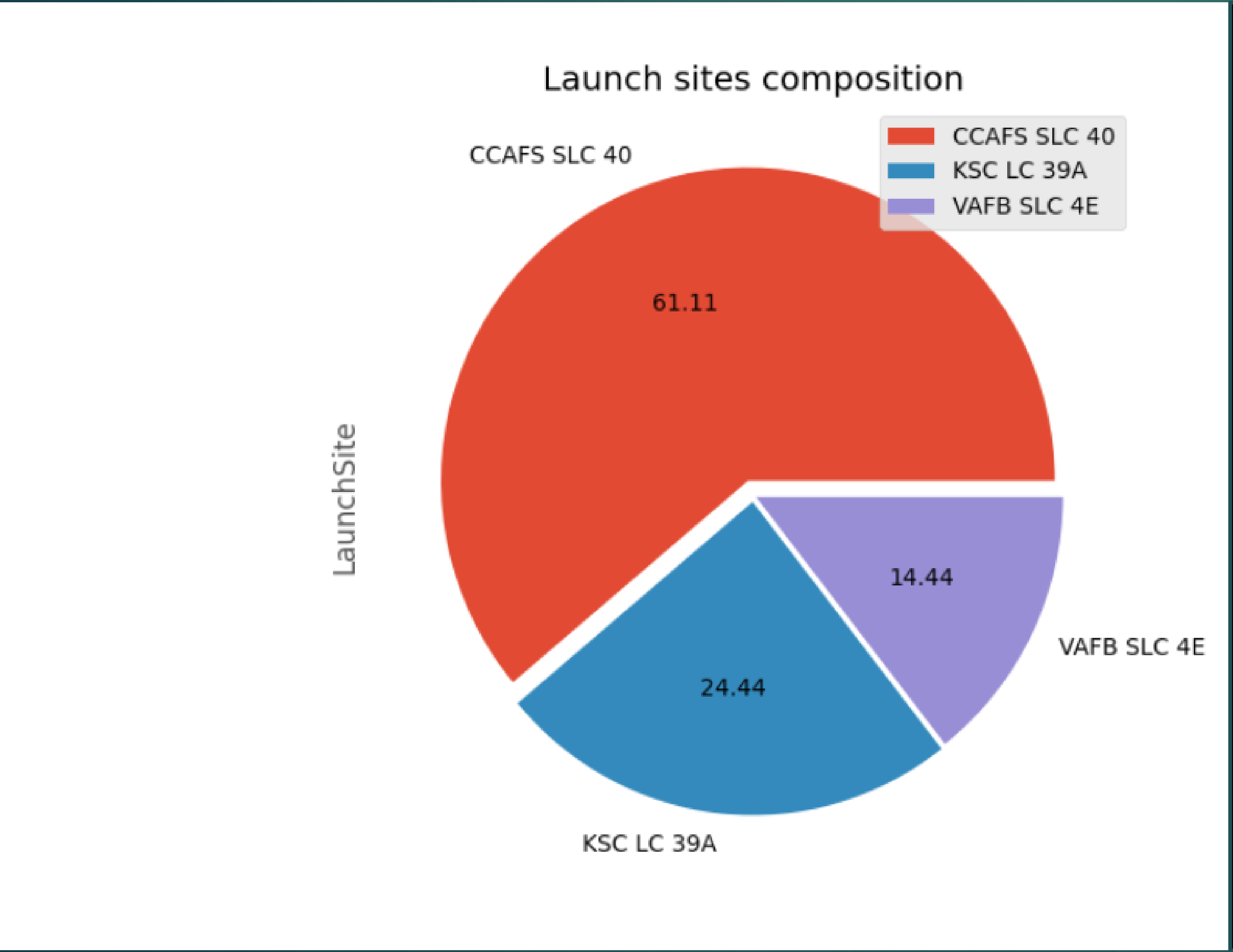
- **Orbit ES-L1, GEO, HEO, SSO** has the highest success rate.
- **SO, GTO** have less success rate.

Success trend (Yearly)



- The increase in success rate of launches started from **2013** and is continuously **improving**.

Launch Site composition



CCAFS SLC 40 has the highest launches.

VAFB SLC 4E has the least launches.

EDA with SQL

- Executed SQL queries:
  - %sql SELECT DISTINCT("Launch_Site") FROM SPACEXTBL;
  - %sql SELECT "Launch_Site" \ FROM SPACEXTBL \ WHERE "Launch_Site" LIKE "CCA%" LIMIT 5;
  - %sql SELECT SUM("PAYLOAD_MASS__KG_") \FROM SPACEXTBL \WHERE "Customer" = "NASA (CRS)“;
  - %sql SELECT AVG("PAYLOAD_MASS__KG_") \FROM SPACEXTBL \WHERE "Booster_Version" = "F9 v1.1“;
  - %sql SELECT DISTINCT("Landing_Outcome") FROM SPACEXTBL; %sql SELECT MIN("Date") FROM SPACEXTBL \ WHERE "Landing_Outcome" = "Success (ground pad)";

- Result / description of result:
  - Names of unique launch sites; **CCAFS LC-40, VAFB SLC-4E, KSC LC-39A, CCAFS SLC-40, None**
  - Displays 5 records where launch sites begin with the string ‘CCA’
  - Displays the Total Payload mass carried by boosters launched by NASA (CRS); **45596.0**
  - Displays average payload mass carried by booster version F9 v1.1; **2928.4**
  - Lists the date when the first successful landing outcome in ground pad was achieved; **01/08/2018**

Github URL:
[EDA with SQL source code]

EDA with SQL

- Executed SQL queries:
  - %sql SELECT "Booster_Version" FROM SPACEXTBL \ WHERE "Landing_Outcome" = "Success (drone ship)" \ AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
  - %sql SELECT DISTINCT("Mission_Outcome") FROM SPACEXTBL; %sql SELECT COUNT("Mission_Outcome") \ AS "Total number of outcomes" FROM SPACEXTBL;
  - %sql SELECT "Booster_Version","PAYLOAD_MASS__KG_" FROM SPACEXTBL \ WHERE "PAYLOAD_MASS__KG_" IN (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);

Github URL:
[EDA with SQL source code]

- Result / description of result:
  - Lists the name of boosters which have success in drone ship and have payload mass > 4000 but < 6000; **F9 FT B1022, F9 FT B1026, F9 FT B1021.2, F9 FT B1031.2**
  - Lists the total number of successful and failure mission outcomes; **101**
  - Lists the names of booster versions which carried the ma payload mass

F9 B5 B1048.4:15600.0; F9 B5 B1049.4:15600.0

F9 B5 B1051.3:15600.0; F9 B5 B1056.4:15600.0

F9 B5 B1048.5:15600.0; F9 B5 B1051.4:15600.0

F9 B5 B1049.5:15600.0; F9 B5 B1060.2:15600.0

F9 B5 B1058.3:15600.0; F9 B5 B1051.6:15600.0

F9 B5 B1060.3:15600.0; F9 B5 B1049.7:15600.0

EDA with SQL

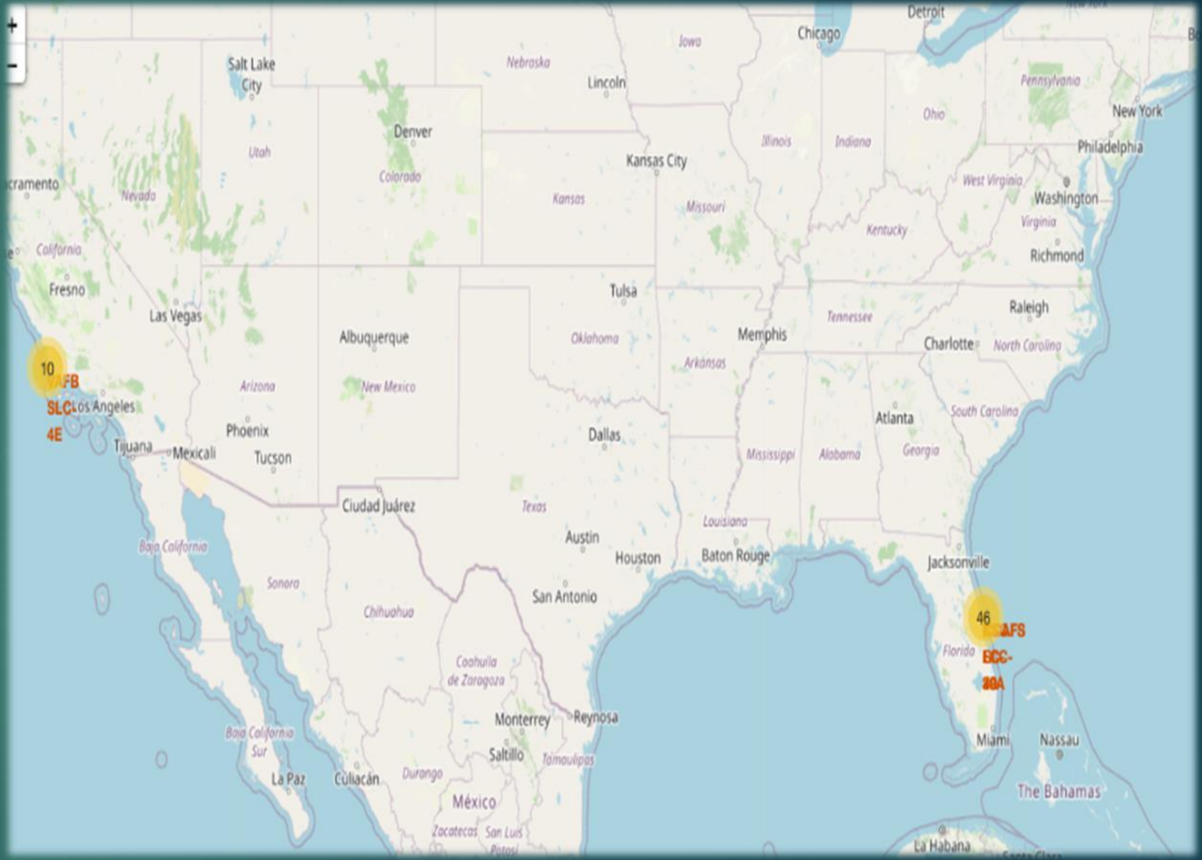
- Executed SQL queries:
  - %sql SELECT SUBSTR("Date",4,2) AS "Month","Date","Booster_Version","Launch_Site","Landing_Outcome" \FROM SPACEXTBL \WHERE "Landing_Outcome" = "Failure (drone ship)" AND SUBSTR("Date",7,4) = "2015";
  - %sql SELECT "Landing_Outcome","Date",COUNT(*) FROM SPACEXTBL \ WHERE "Date" BETWEEN "04/06/2010" AND "20/03/2017" \ GROUP BY "Landing_Outcome" HAVING "Landing_Outcome" IN ("Success" , "Success (ground pad)","Success (drone ship)") ORDER BY DESC
- Result / description of result:
  - Lists the records which will display the month names, failure landing outcomes in drone ship, booster versions, launch sites for the months in year 2015.
  - Ranks the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order;

Landing Outcome	**Date**	**COUNT(*)**
Success (ground pad)	**18/07/2016**	**7**
Success (drone ship)	**04/08/2016**	**8**
Success	**08/07/2018**	**20**

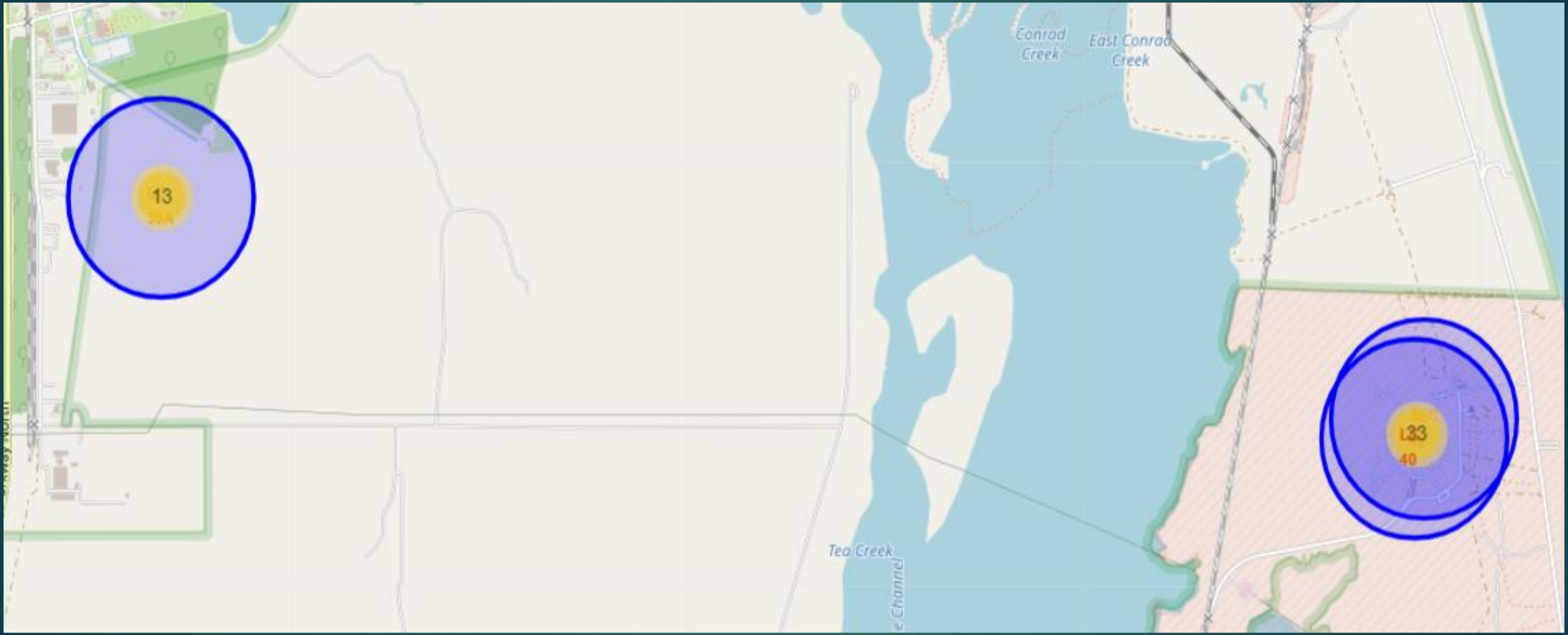
Github URL:
[EDA with SQL source code]

INTERACTIVE VISUAL MAPS

- Maps created with Folium to add elements and interactivity with the generated map plots
- Added elements like circle, markers, text for annotating over the map around the data points with the co-ordinates.
- Assigned labels for markers and color coded it, added a popup element to display data points when clicked over a cluster marker.
- Calculated the linear distance between launch sites and coast, nearest city.



FOLIUM MAPS



- Marked the area with a circle with the co-ordinates.
- Added and clustered the data points for the launch sites.

Github URL:
Folium maps source code

FOLIUM MAPS

- Launch sites with labels texts, cluster data points.
- Mapped Flight launches for the Launch sites.

Github URL:
[Folium maps source code]

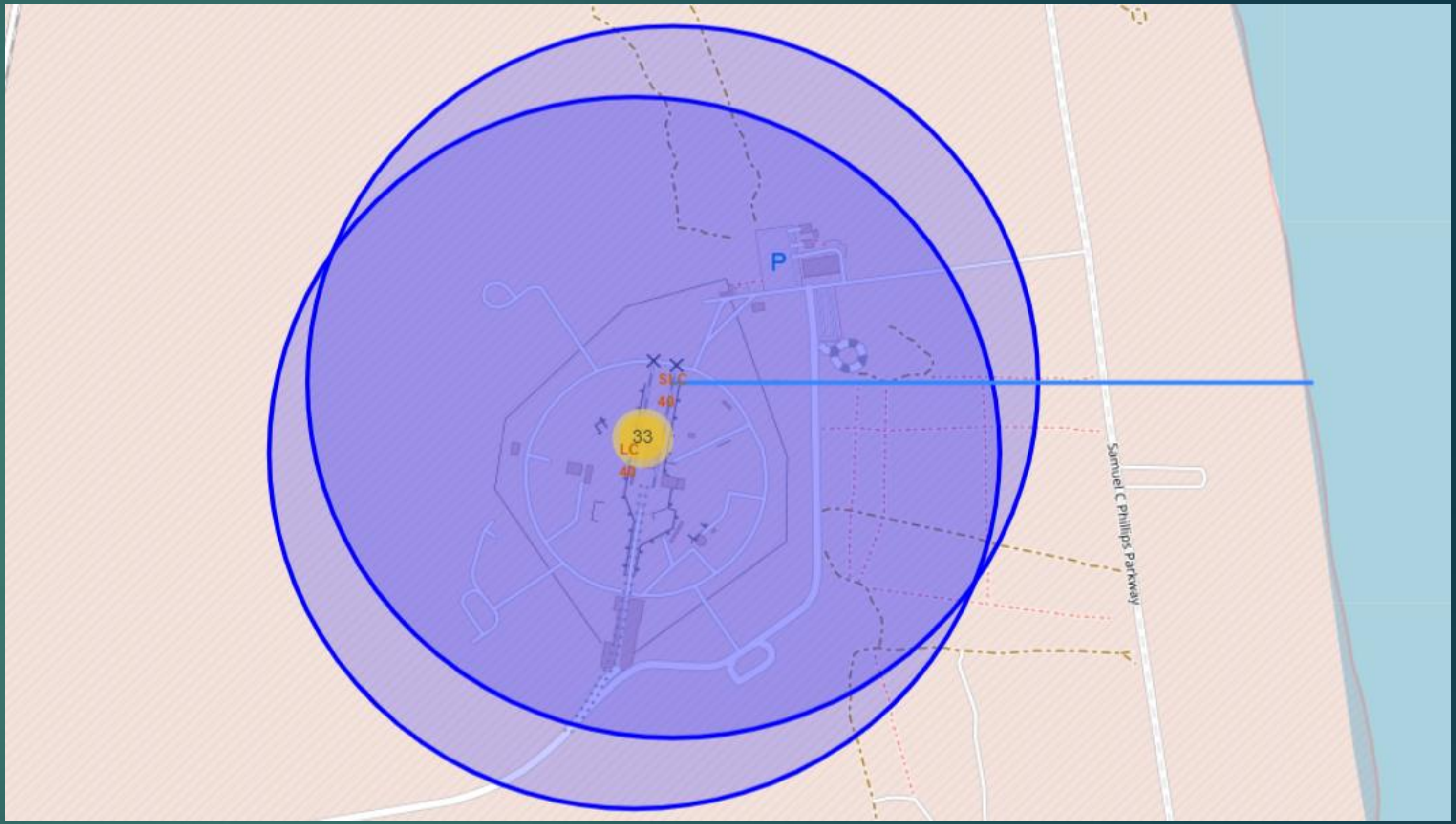
FOLIUM MAPS

- Data points clustered around center point of co-ordinates.
- When clicked the popup elements display the hidden data points.

Github URL: [Folium maps source code]

FOLIUM MAPS

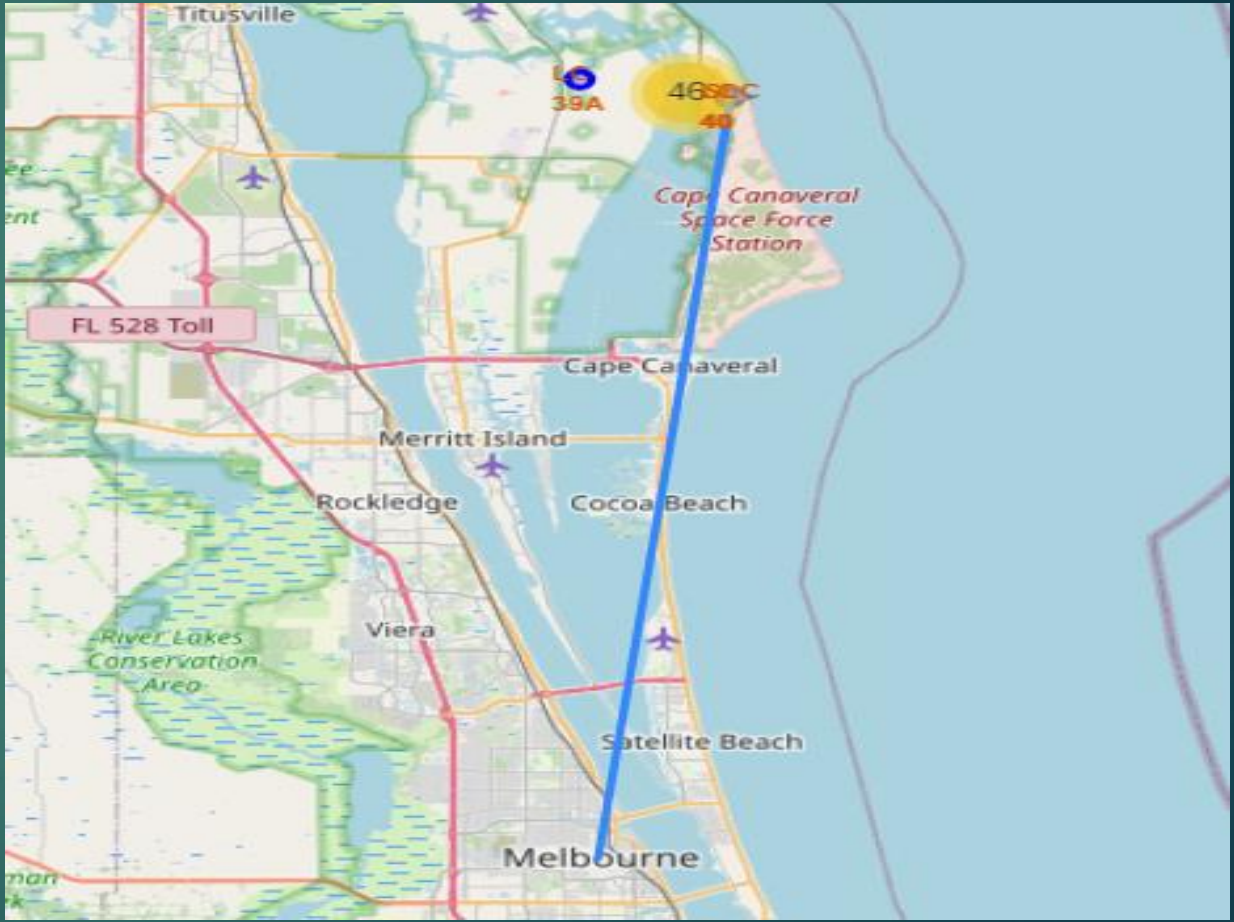
- Measuring the Linear distance from Launch site to coastline.
- Plotted the line with PolyLine element from Folium.



Github URL: [Folium maps source code]

FOLIUM MAPS

- Plotted the PolyLine from Launch Site to nearest city (Melbourne).



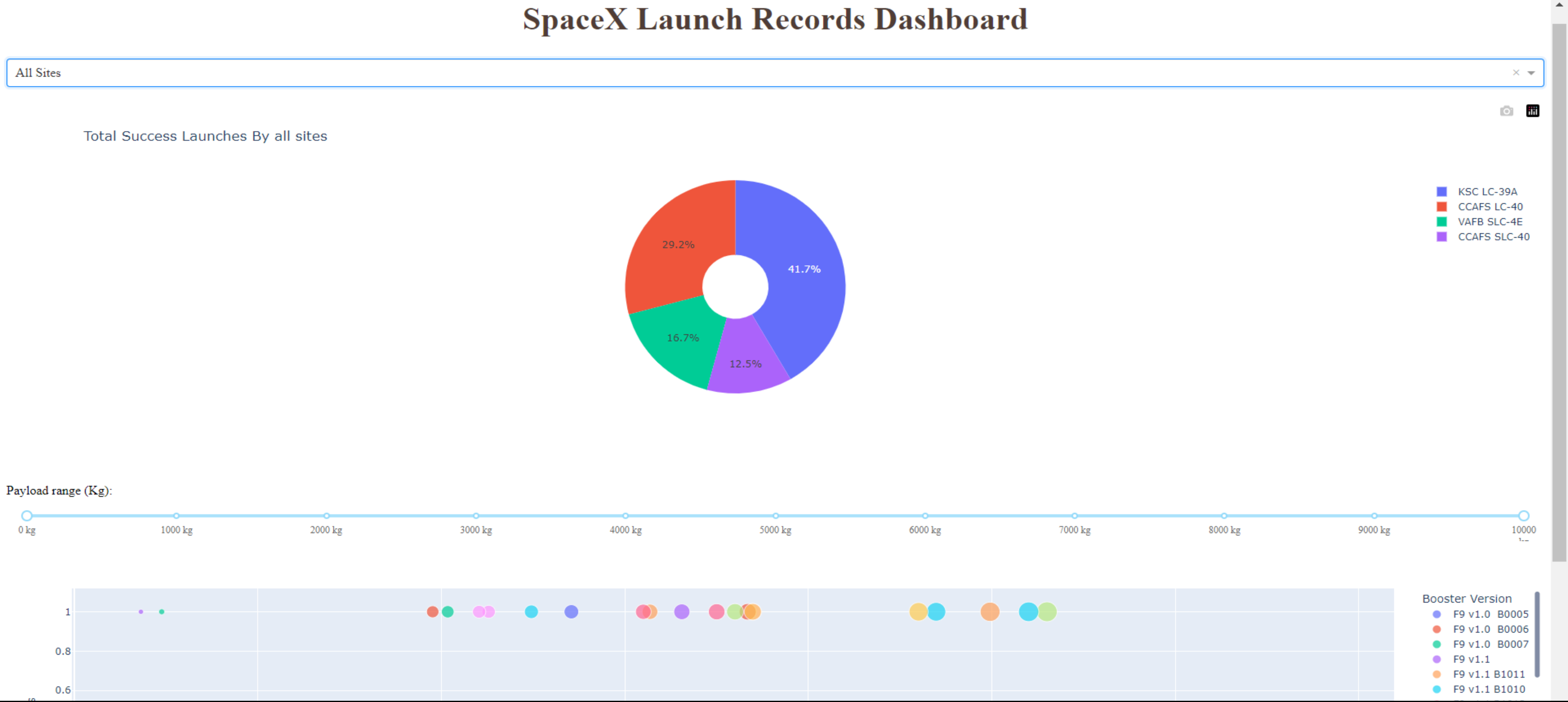
Github URL: [Folium maps source code]()

INTERACTIVE DASHBOARD

- Dashboard built using Pandas Plotly dash.
- Developed in a development environment, to run server locally on web browser.
- Added functionality of filtering records through Launch sites, Payload Mass slider to select between two values.
- Resulting in a Donut style chart representing the composition of launch sites, Scatter plot viewing the relationship between payload mass and class in different booster versions.

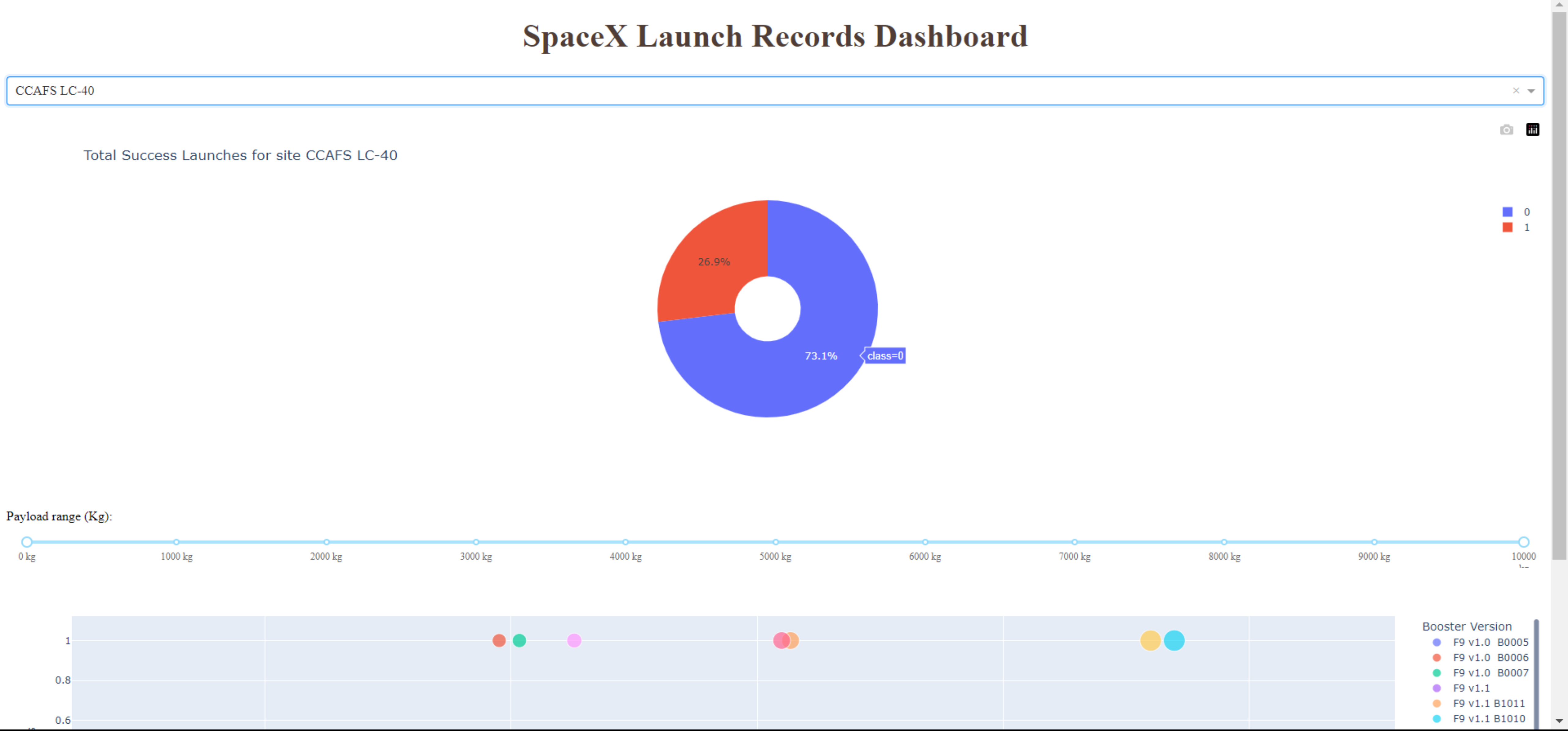
Github URL: [Dashboard source code]

DASHBOARD TAB 1



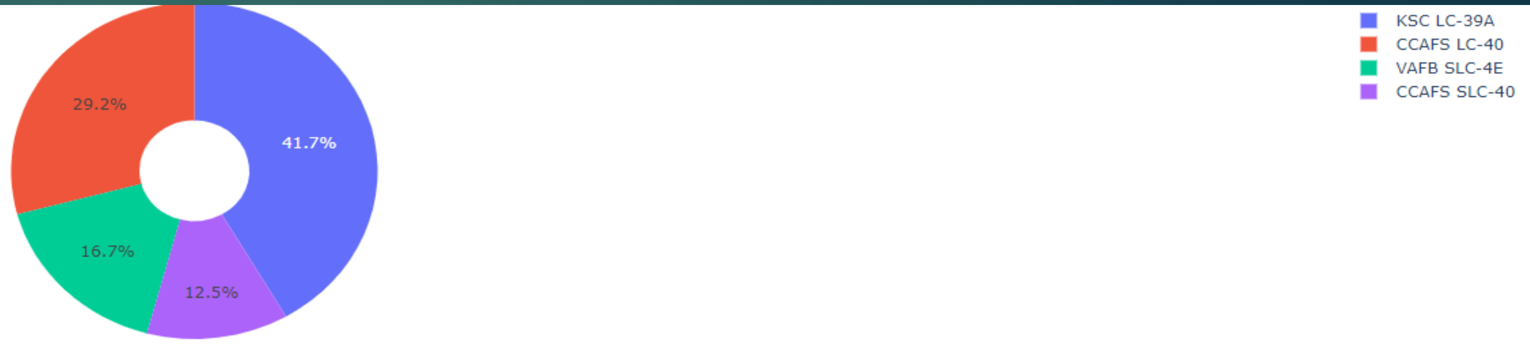
Github URL: [Dashboard source code]()

DASHBOARD TAB 2



Github URL: [Dashboard source code]

DASHBOARD TAB 3



Payload range (Kg):

Github URL: [Dashboard source code]

PREDICTIVE ANALYSIS

- Imported required libraries and functions and defined custom functions as well.
- Loading the cleaned preprocessed data, Identifying target and features.
- Preprocessing features and model selection methods for training and testing sets.
- Trained ML algorithms on training data and tuning hyper-parameters with GridSearchCV.
- Evaluate models with Testing set and calculating accuracy and f1 scores.

Import libraries and data → Preprocess and select features → Model Selection and Train/Test split → Training ML models → Model Evaluation

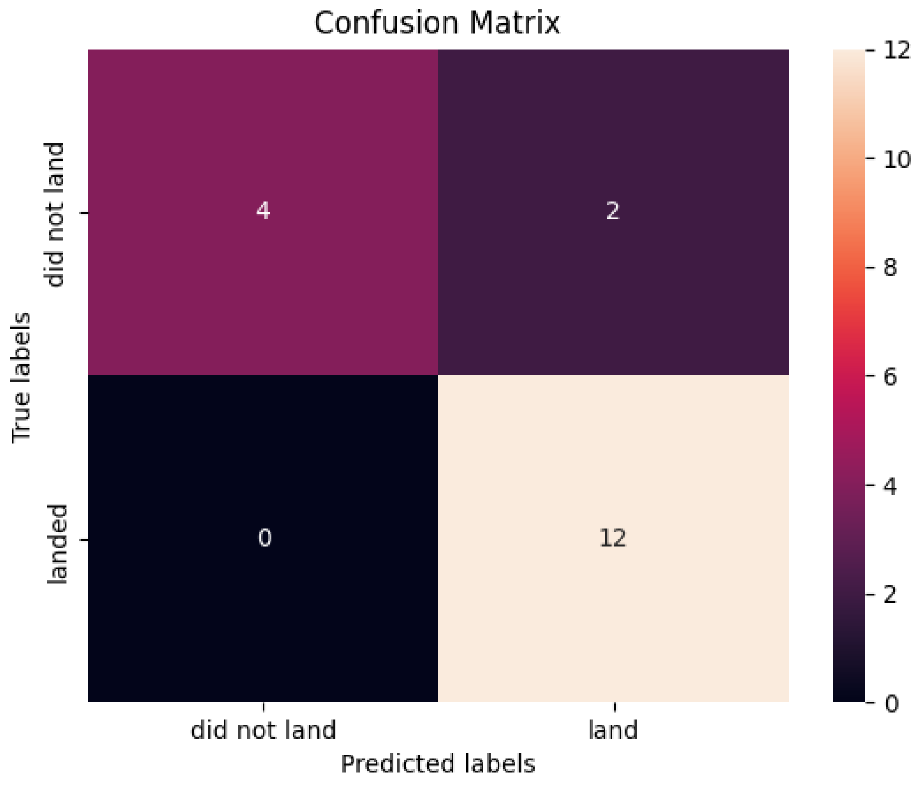
Github URL: Predictive analysis source code

FINDING BEST MODEL

- **Model Evaluation for finding best model:**
  - **Confusion matrix**: A matrix representing the Actual, Predicted values by True positives, True negatives, False negatives, False positives.
  - **Accuracy scores**: score ranging from 0 -1 signifying the correctly predicted values from total values.
  - **F1 score**: A harmonic mean between precision and recall.

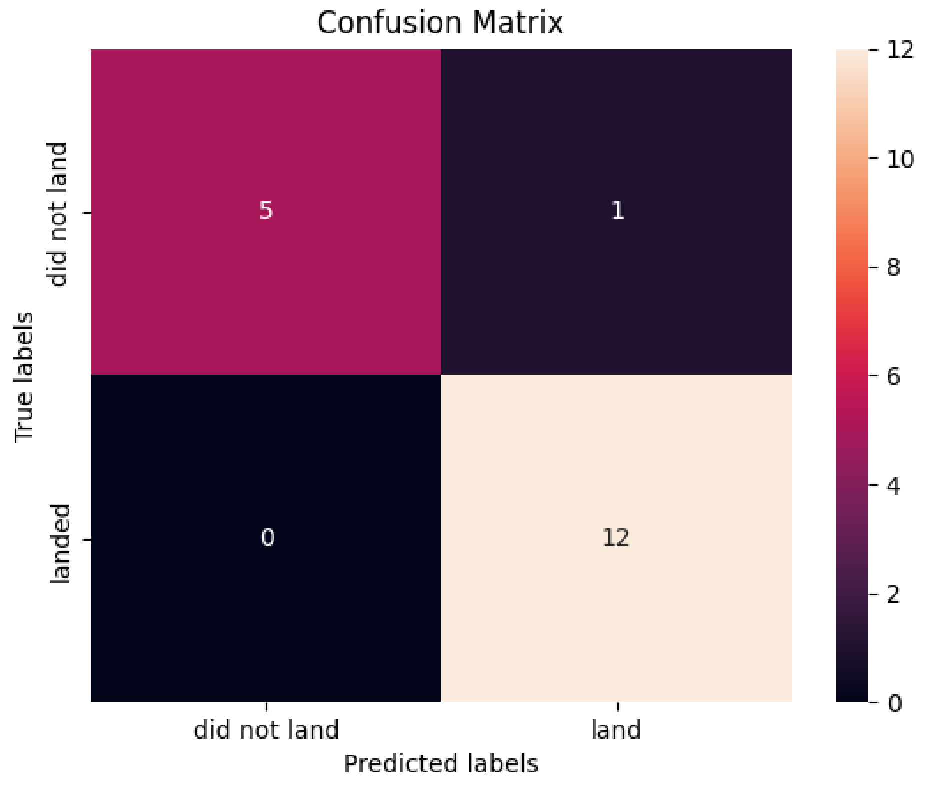
CONFUSION MATRIXES

Decision Tree Classifier



Parameters: {'criterion': 'entropy', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'}

KNN Classifier

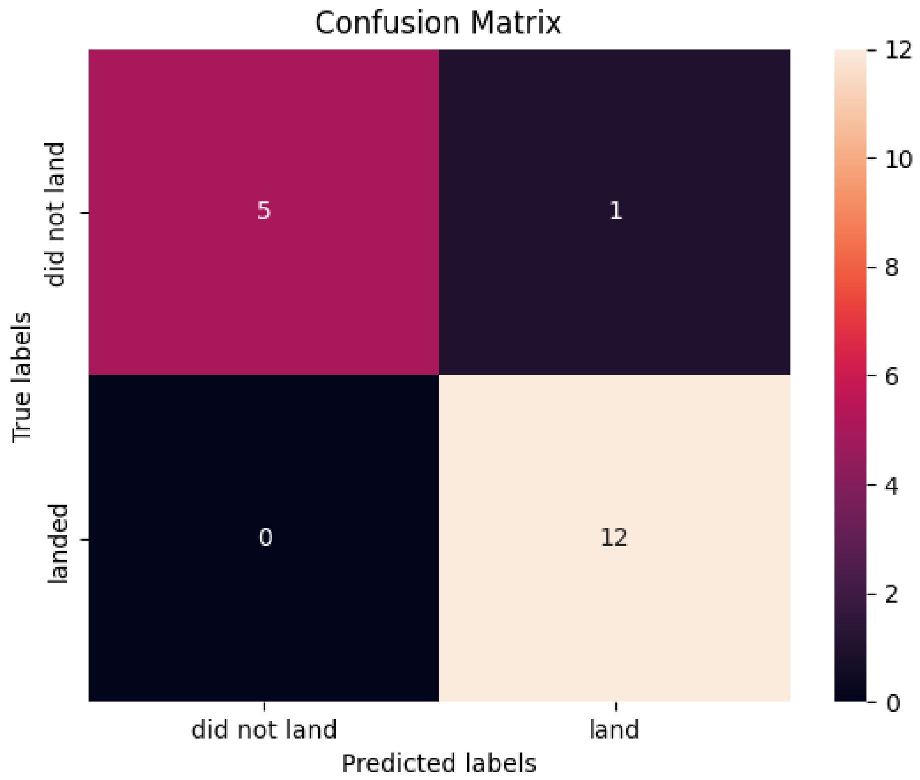


Parameters: {'algorithm': 'auto', 'n_neighbors': 5, 'p': 1}

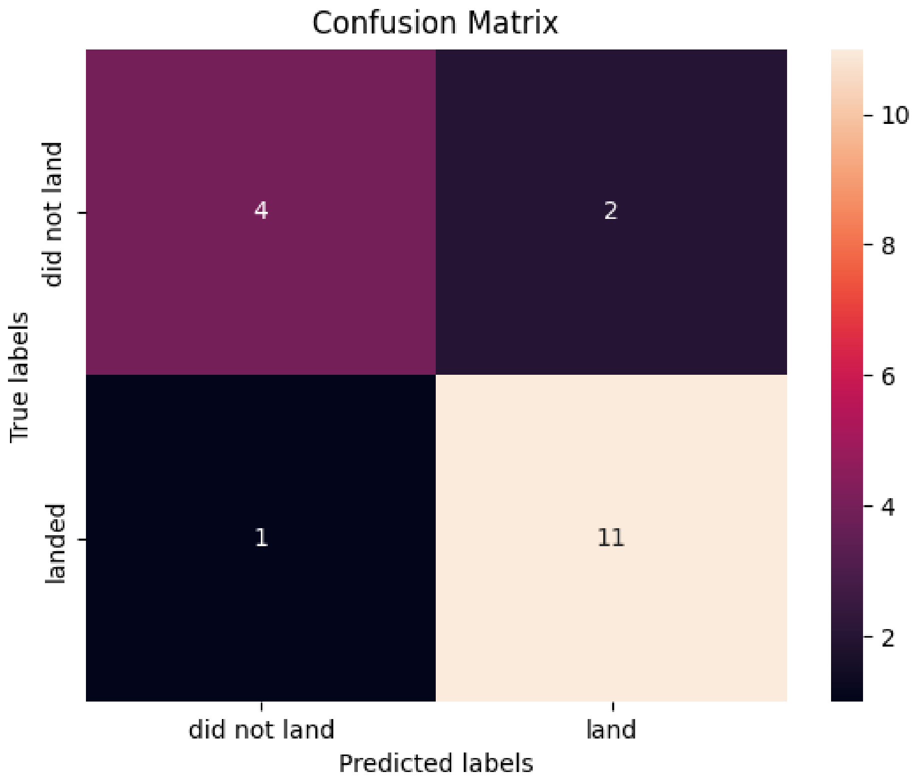
Github URL: [Predictive analysis source code]

CONFUSION MATRIXES

Logistic Regression



Support Vector Classifier



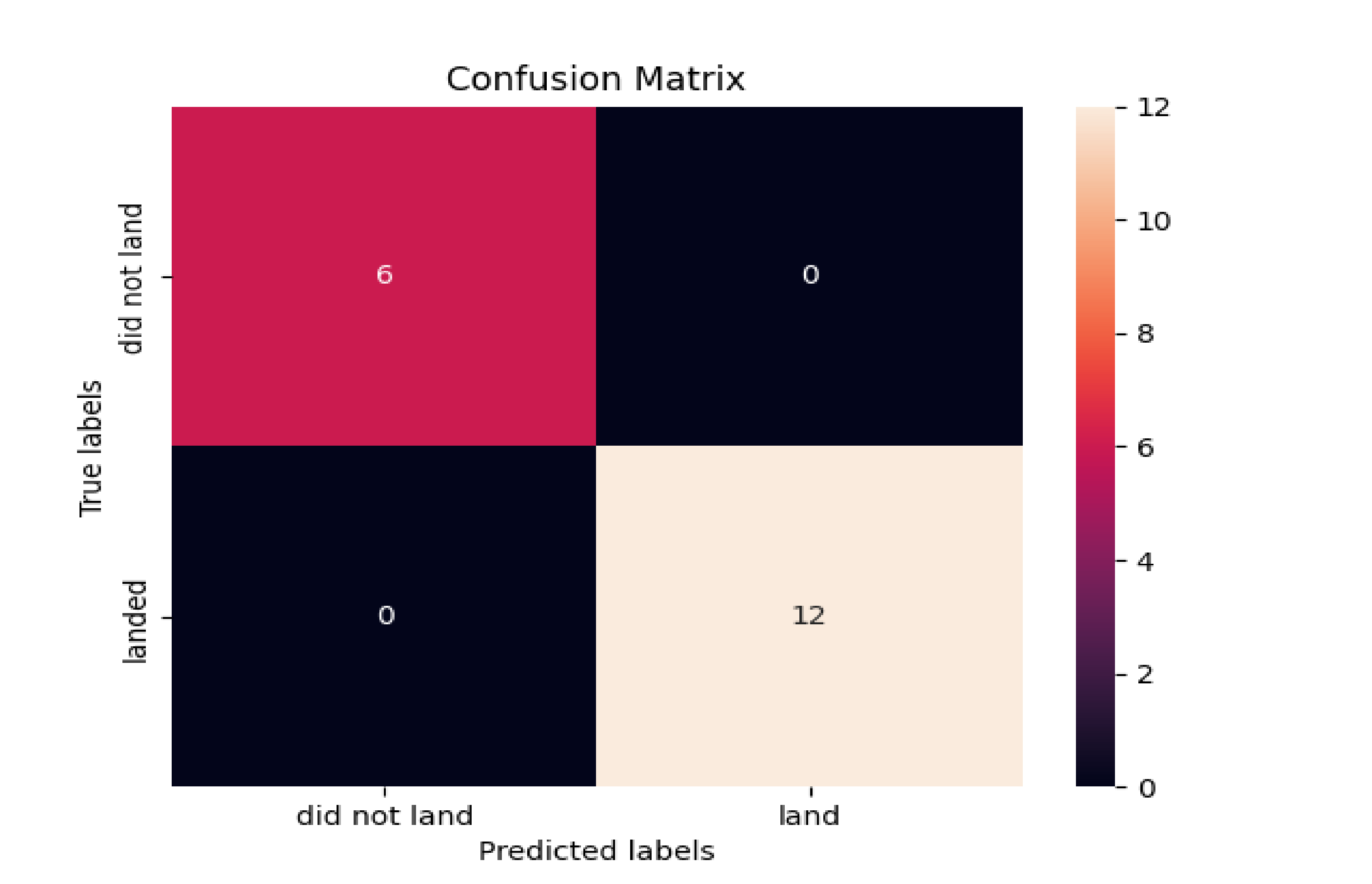
Parameters : {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}

Parameters : {'C': 1, 'gamma': 3, 'kernel': 'sigmoid'}

Github URL: [Predictive analysis source code]

CONFUSION MATRIX AND SCORE TABLE

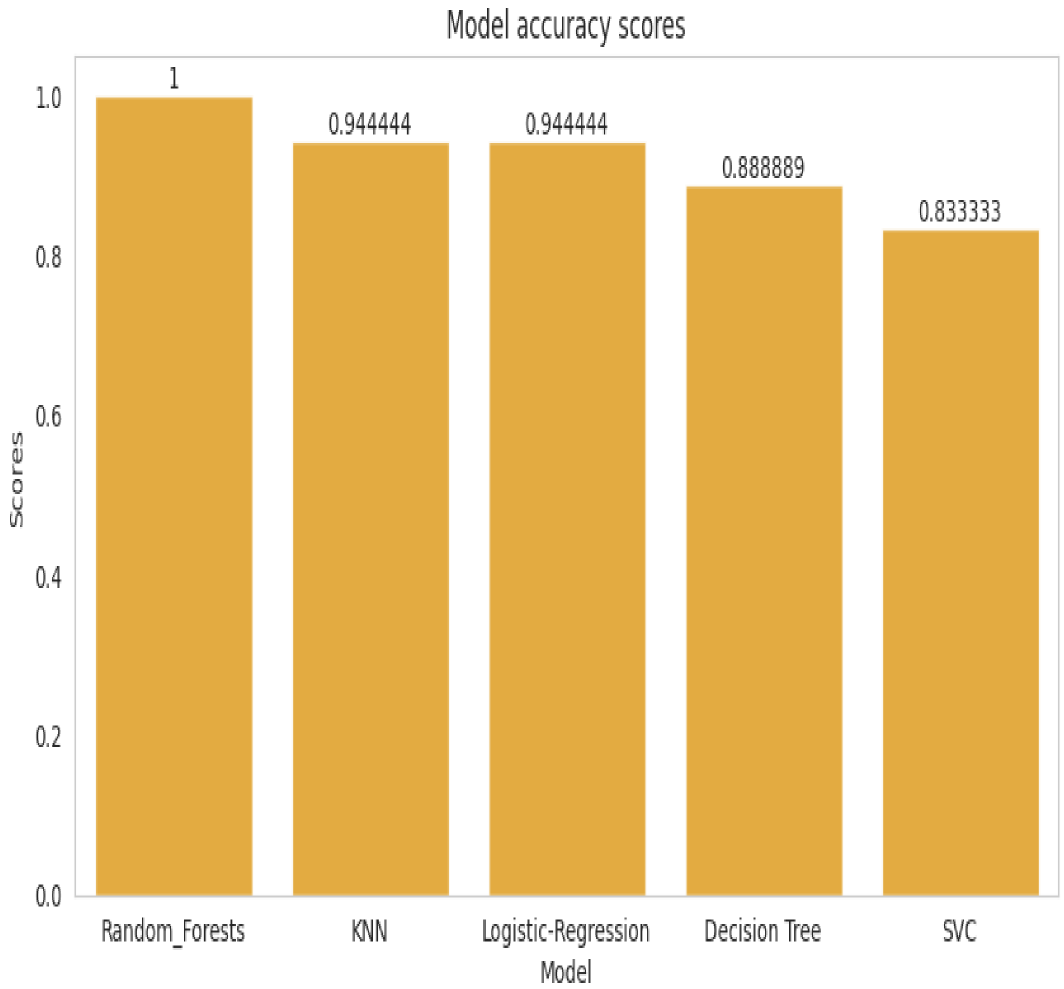
Random Forests Classifier

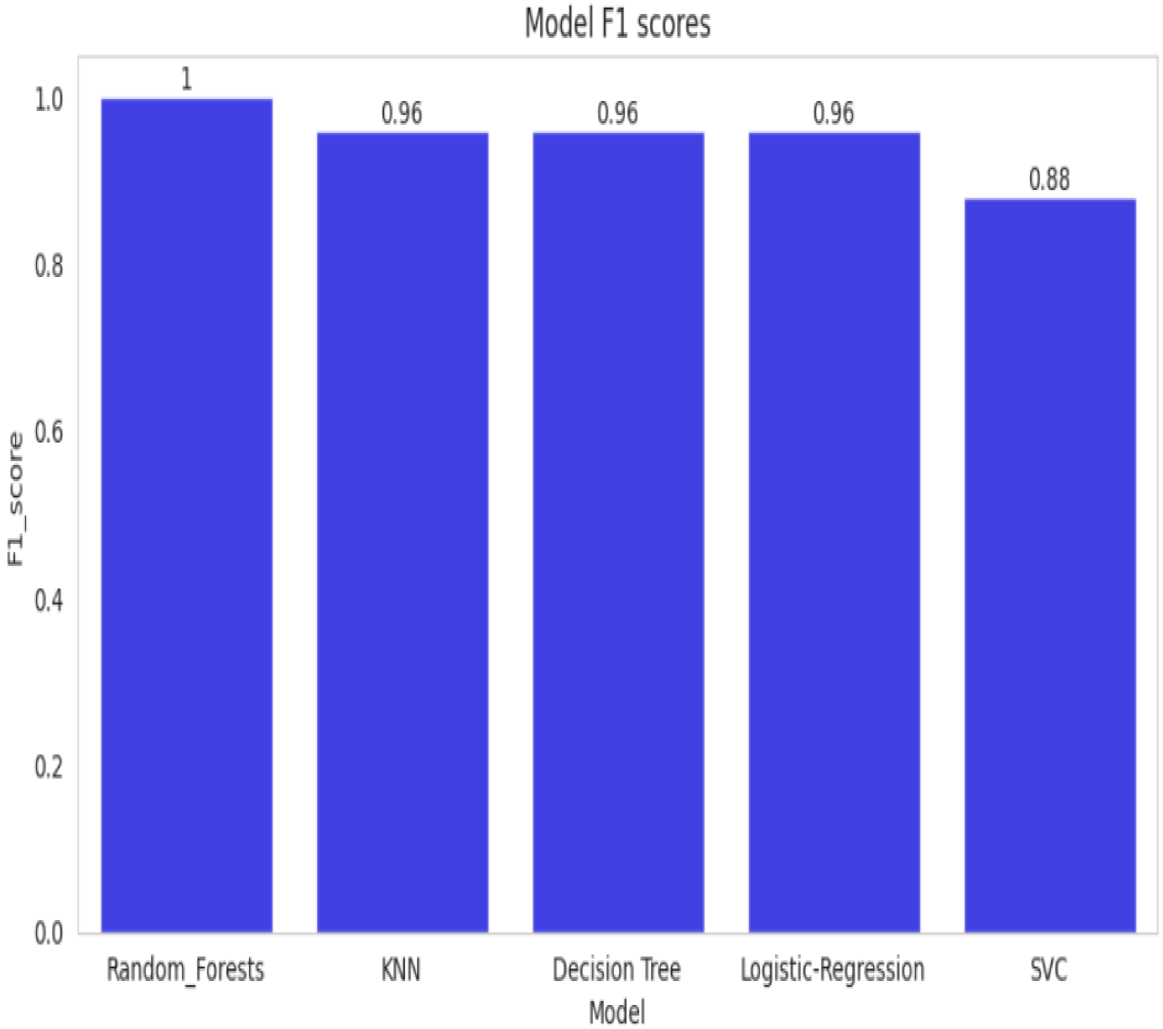


Model	Scores
Random Forests Classifier	1.000000
KNN	0.944444
Logistic-Regression	0.944444
Decision Tree	0.888889
SVC	0.833333

Github URL: [Predictive analysis source code]

Accuracy scores and F1 scores





Github URL: [Predictive analysis source code]

DISCUSSION

- Ensemble model like **Random Forest** did a better job, although more data could have been better to evaluate the models.
- The mean success rate is ***0.66 %***, and rate for successful launches are improving with time.
- It is also observed that launch sites are away from residential areas and closer to coast.

OVERALL FINDINGS

Findings:

- **CCAFS SLC 40** has most launches while **VAFB SLC 4E** has the least.
- **CCAFS SLC 40** and **KSC LC 39A** launch with payload mass more than **10000 kg**.
- Heavy payloads are not used on higher orbit rockets and success rate is not productive as well.
- Random Forests model evaluated as the best model with default parameters.

CONCLUSION

- Mean success rate of launches is approximate **66%**.
- Successful landings can be predicted with almost **80-90%** accuracy.
- **Random Forest** model did well in predicting and obtained decent *accuracy* and *f1 score*.
- Models can be used for predicting successful landing rates estimating costs for launches.

Thank you