IBM DATA SCIENCE CAPSTONE PROJECT

DAJI RAO AKASH SHINDE

GIT: https://github.com/dajiraoakash/Coursera/tree/main/Capstone

Table of Contents

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

Executive Summary

Data Collection and Feature Engineering:

- Data was gathered from the public SpaceX API and the SpaceX Wikipedia page.
- A 'class' column was created to label successful landings.
- Data was explored using SQL, visualizations, Folium maps, and dashboards.
- Relevant columns were selected as features for modeling.
- Categorical variables were converted to binary using one-hot encoding.
- Data was standardized, and GridSearchCV optimized machine learning model parameters.
- The accuracy scores of all models were visualized.

Machine Learning Model Performance:

- Four machine learning models were developed: Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K-Nearest Neighbors.
- All models achieved similar accuracy rates of approximately 83.33%.
- All models tended to over-predict successful landings.
- The analysis suggests that more data is necessary for improved model selection and accuracy.

INTRODUCTION

Background

- The commercial space industry is now a reality.
- SpaceX offers the most competitive pricing for launches (USD 62 million compared to USD 165 million).
- This cost advantage is primarily due to their capability to recover the first stage of their rockets.
- SpaceY aims to become a competitor to SpaceX in this market.

Problem Statement

- SpaceY has assigned us the task of developing a machine-learning model.

Methodology

- DATA COLLECTION
- WRANGLING
- VISUALIZATION
- DASHBOARD CREATION
- PREDICTIVE MODELLING

Data Collection Overview

Data Collection Methodology

- Data acquisition involved two primary methods: utilizing API calls to the public SpaceX API and web scraping information from a table on SpaceX's Wikipedia page.
- The subsequent slides will illustrate the data collection workflow for both the API and web scraping processes using flowcharts.

SpaceX API Data

FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

Wikipedia Web-Scraped Data

Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

Data Collection via SpaceX API

1. Request (SpaceX APIs): Initiate a data request from the SpaceX APIs.
2. JSON file + Lists (Launch Site, Booster Version, Payload Data): The API response is received as a JSON file containing lists of data, including Launch Site, Booster Version, and Payload Data.
3. Json_normalize to DataFrame data from JSON: The JSON data is processed and normalized into a DataFrame structure.
4. Filter data to only include Falcon 9 launches: The DataFrame is filtered to retain only data related to Falcon 9 launches.
5. Cast dictionary to a DataFrame: (This step appears to be in parallel with the filtering and might refer to a specific part of the JSON structure being converted.)
6. Dictionary relevant data: (This also appears parallel and likely indicates selecting specific data fields from the dictionary structure.)
7. Impute missing PayloadMass values with mean: Any missing values in the 'PayloadMass' column of the DataFrame are filled in using the mean value of the existing 'PayloadMass' data.

GIT: https://github.com/dajiraoakash/Coursera/blob/main/Capstone/Lab1-data-collection-api.ipynb

Data Collection via Web Scraping

1. **Request Wikipedia html:** Send a request to obtain the HTML content of the relevant Wikipedia page.
2. **BeautifulSoup html5lib Parser:** The received HTML is parsed using the BeautifulSoup library with the 'html5lib' parser to create a navigable tree structure.
3. **Find launch info HTML table:** Within the parsed HTML, locate the specific HTML table containing the launch information.
4. **Cast dictionary to DataFrame:** (This step appears to be in parallel with finding the table and likely refers to a dictionary created later being converted.)
5. **Iterate through table cells to extract data to a dictionary:** Loop through the cells of the identified launch information table to extract the data and store it in a dictionary format.
6. **Create dictionary:** (This appears parallel and likely represents the initialization or population of the dictionary with the extracted data.)
7. **GIT**: https://github.com/dajiraoakash/Coursera/blob/main/Capstone/LAb2-jupyter-labs-webscraping.ipynb

Data Wrangling

- **Define the Training Label:** Create a new column to serve as the training label for landing outcomes. Assign a value of 1 to represent a successful landing and 0 to represent a failure.
- **Source of Information:** The 'Outcome' column contains two pieces of information: 'Mission Outcome' and 'Landing Location'.
- **Create the 'class' Column:** Generate a new column named 'class'. This column will hold the training label.
- **Labeling Logic:**
  - If the 'Mission Outcome' is 'True', assign a value of 1 to the 'class' column.
  - Otherwise (if 'Mission Outcome' is 'False' or any other value), assign a value of 0 to the 'class' column.
- **Specific Value Mapping for 'class':**
  - Combinations of 'True ASDS', 'True RTLS', and 'True Ocean' should be mapped to a value of 1 in the 'class' column.
  - Combinations of 'None None', 'False ASDS', 'None ASDS', 'False Ocean', and 'False RTLS' should be mapped to a value of 0 in the 'class' column.

GIT: https://github.com/dajiraoakash/Coursera/blob/main/Capstone/Lab3-Data%20wrangling.ipynb

EDA with Data Visualization

- **Variables Analyzed:** Exploratory Data Analysis was conducted on the following variables: Flight Number, Payload Mass, Launch Site, Orbit, Class (the created label), and Year.
- **Visualizations Employed:** The following plots were utilized for the EDA:
  - Flight Number Compared to Payload Mass
  - Flight Number Compared to Launch Site
  - Payload Mass compared to Launch Site
  - Orbit Compared to Success Rate
  - Flight Number Compared to Orbit
  - Payload compared to Orbit
  - Success Rate Trend over Years
- **Plot Types:** Scatter plots, line charts, and bar plots were used as visualization techniques.
- **Objective of EDA:** The purpose of these comparisons was to identify potential relationships between the variables. This analysis aimed to determine if correlations or patterns existed that would make these variables useful for training the subsequent machine learning model.

GIT: https://github.com/dajiraoakash/Coursera/blob/main/Capstone/edadataviz.ipynb

EDA WITH SQL

- **Data Storage:** The collected dataset was loaded into an IBM DB2 database.
- **Querying Method:** SQL queries were executed using Python integration to interact with the database.
- **Purpose of Queries:** These queries were performed to gain a deeper understanding of the dataset's characteristics.
- **Specific Information Retrieved:** The queries focused on retrieving information about:
  - Launch site names
  - Mission outcomes
  - Various payload sizes for different customers
  - Booster versions
  - Landing outcomes

GIT: https://github.com/dajiraoakash/Coursera/blob/main/Capstone/Lab4-%20EDA%20SQL.ipynb

Build an interactive map with Folium

Folium maps visually represent launch sites, differentiating between successful and unsuccessful landings. They also illustrate the proximity of a launch site to important infrastructure like railways, highways, the coast, and nearby cities. This helps explain the strategic placement of launch sites and provides a visual understanding of landing success in these locations.

Git:
url:https://github.com/dajiraoakash/Coursera/blob/main/Capstone/lab_jupyter_launch_site_location.ipynb

BUILDING DASHBOARD WITH DASH

- The dashboard features a pie chart and a scatter plot for data visualization. The pie chart can display either the distribution of successful landings across all launch sites or the success rate of a specific launch site.
- The scatter plot allows the user to select either all launch sites or an individual site and then view data based on a payload mass range adjustable with a slider (from 0 to 10,000 kg). The pie chart's primary function is to visualize the success rate of launch sites. The scatter plot is designed to help analyze how landing success is influenced by the chosen launch site(s), the payload mass, and the booster version category (indicated by color and point size).

GIT: https://github.com/dajiraoakash/Coursera/blob/main/Capstone/spacex_dashapp.py

Build a Dashboard with Plotly Dash

Dashboard includes a pie chart and a scatter plot.

Pie chart can be selected to show distribution of successful landings across all launch sites and can be selected to show individual launch site success rates.

Scatter plot takes two inputs: All sites or individual site and payload mass on a slider between 0 and 10000 kg.

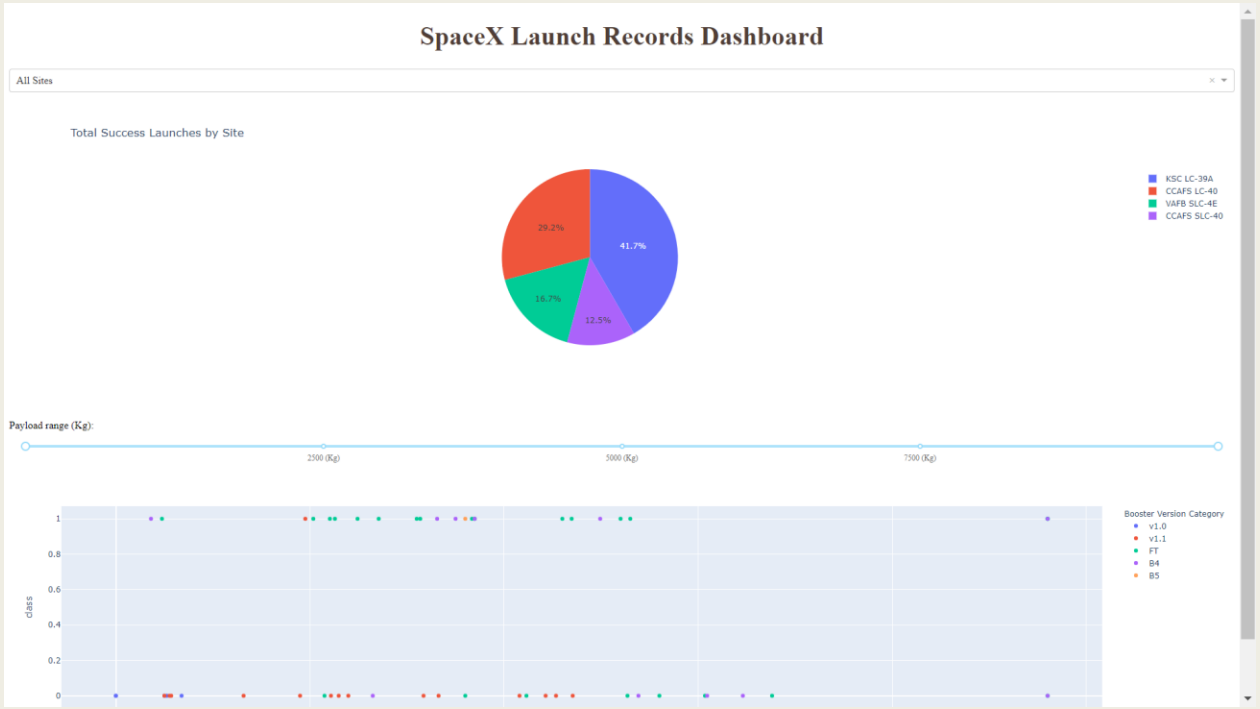
The pie chart is used to visualize launch site success rate.

The scatter plot can help us see how success varies across launch sites, payload mass, and booster version category.

GitHub url:

https://github.com/navassherif98/IBM_Data_Science_Professional_Certification/blob/master/10.Applied_Data_Science_Capstone/Week%203%20Interactive%20Visual%20Analytics%20and%20Dashboard/spacex_dash_app.py

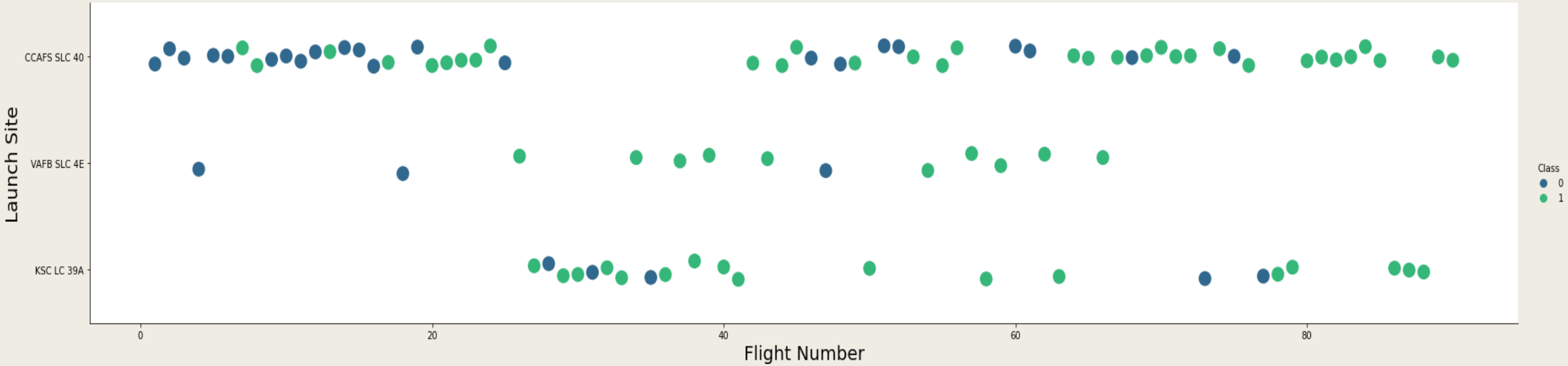
RESULTS



This slide provides a glimpse of the Plotly dashboard. Subsequent slides will present the findings from Exploratory Data Analysis (EDA) using visualizations, EDA using SQL queries, an interactive map created with Folium, and ultimately, the results of our machine learning model, which achieved approximately 83% accuracy.

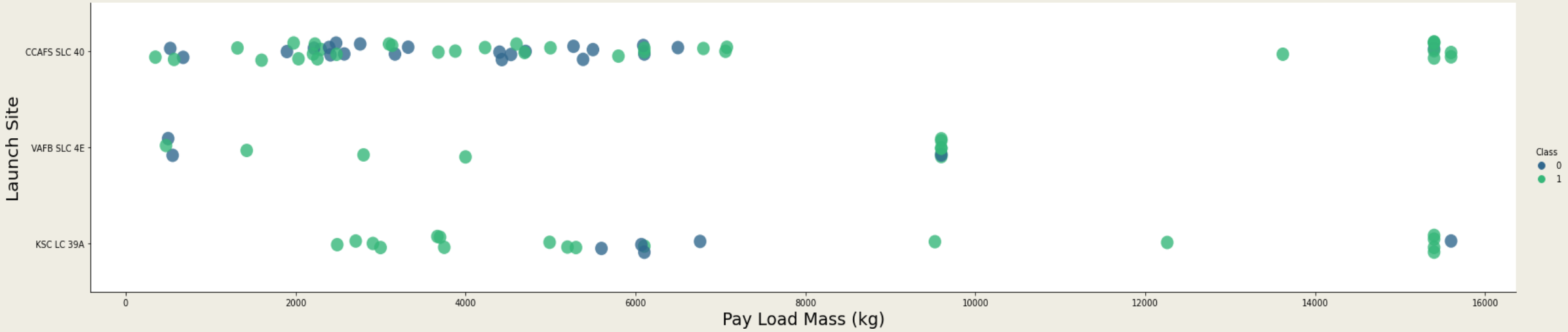
Exploratory Data Analysis (EDA) using visualizations

FLIGHT NUMBER VS LAUNCH SITE



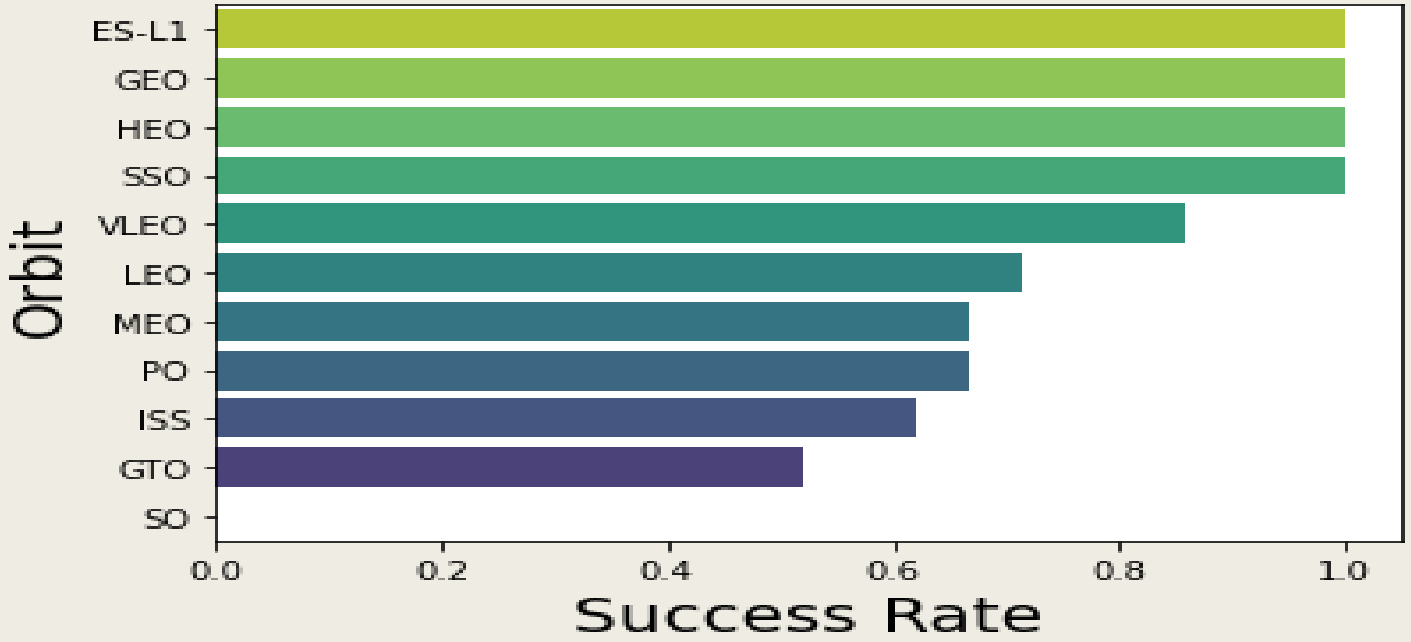
The graph indicates that the success rate of landings has generally improved over time, as suggested by the increasing Flight Number. A significant turning point or breakthrough likely occurred around flight 20, leading to a notable increase in the success rate. Furthermore, CCAFS (Cape Canaveral Space Force Station) appears to be the primary launch site, as it accounts for the highest number of launches in the dataset.

PAYLOAD VS LAUNCH SITE



The data suggests that the majority of payloads launched have a mass between 0 and 6000 kilograms. Additionally, there seems to be a correlation between the launch site used and the typical payload mass carried. Different launch sites appear to handle different ranges of payload weights.

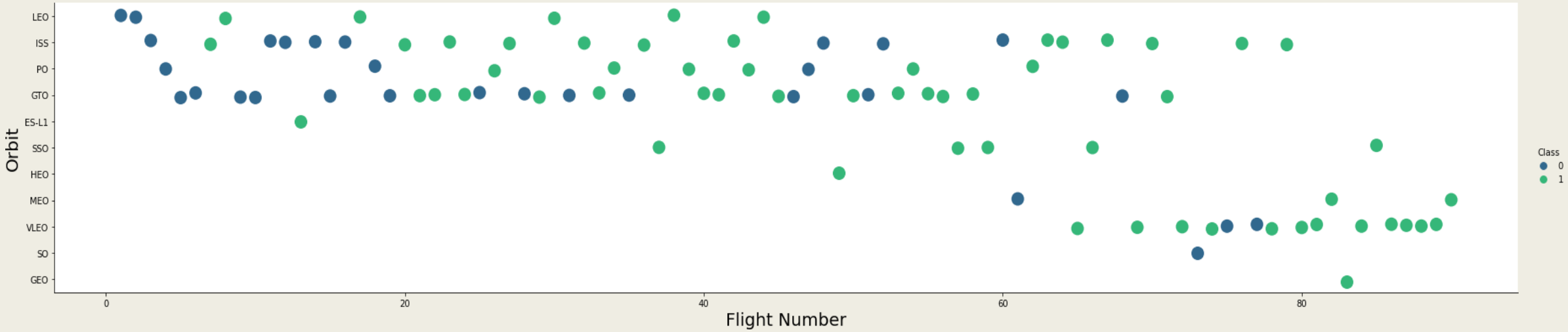
SUCCESS RATE VS ORBIT TYPE



The following orbital categories show their successful landing rates and the number of attempts (in parentheses):

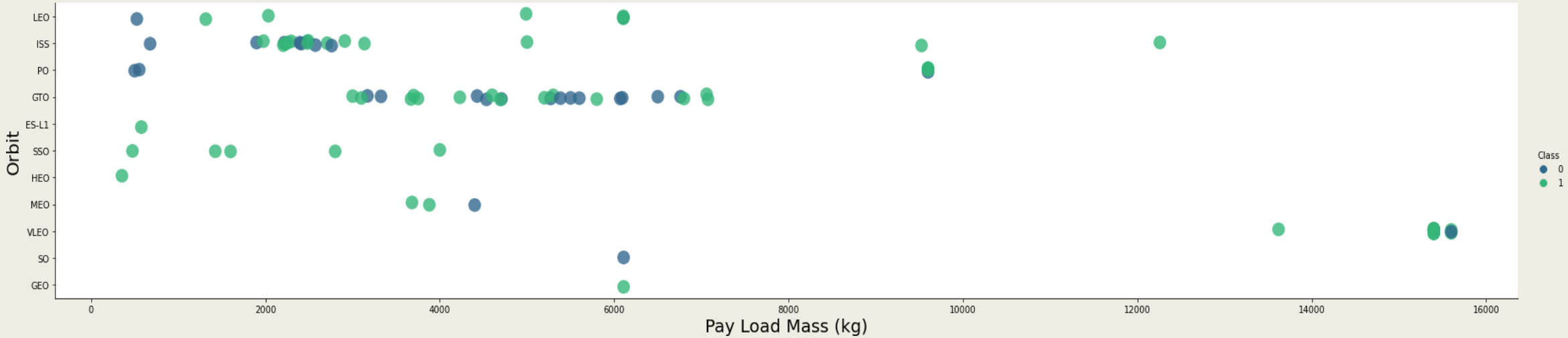
- **ES-L1 (1):** 100% success rate
- **GEO (1):** 100% success rateA
- **HEO (1):** 100% success rate
- **SSO (5):** 100% success rate
- **VLEO (14):** Decent success rate with a moderate number of attempts.
- **SO (1):** 0% success rate
- **GTO (27):** Approximately 50% success rate and represents the largest sample size among the orbital categories listed.

FLIGHT NUMBER VS ORBIT TYPE



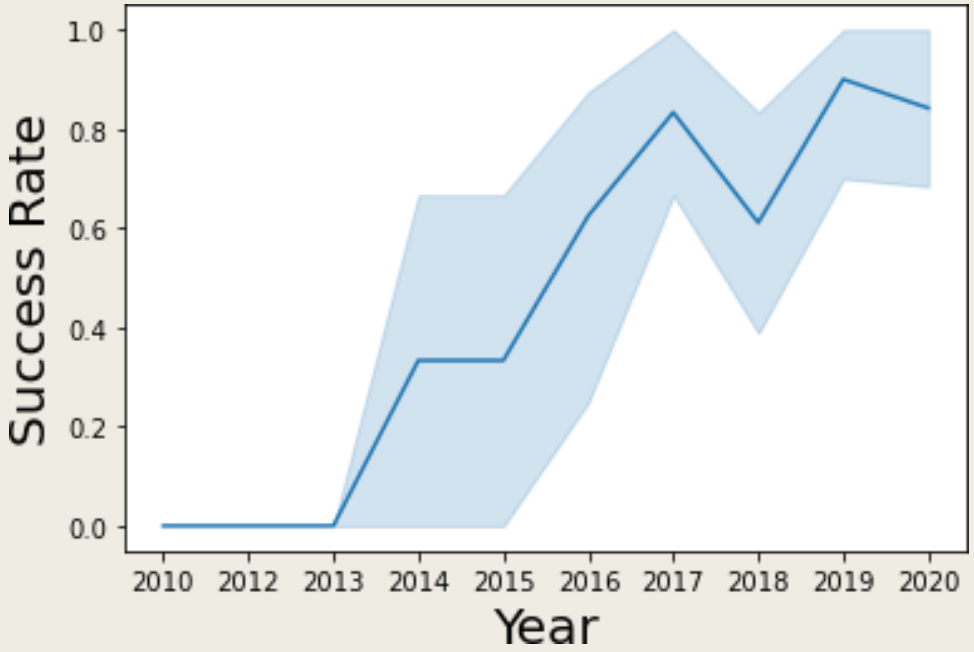
The preferred launch orbits have evolved throughout the flight numbers. This shift in orbit preference appears to be related to the launch outcome. SpaceX initially utilized Low Earth Orbit (LEO) with a moderate degree of landing success. More recently, there has been a return to Very Low Earth Orbit (VLEO) for launches. Overall, SpaceX seems to achieve better landing success rates when launching to lower orbits or Sun-Synchronous Orbit (SSO).

PAYLOAD VS ORBIT TYPE



There appears to be a relationship between payload mass and the intended orbit. Specifically, Low Earth Orbit (LEO) and Sun-Synchronous Orbit (SSO) tend to be associated with relatively lower payload masses. Conversely, Very Low Earth Orbit (VLEO), another orbit with a high success rate, seems to be used for payloads with masses in the higher end of the observed range.

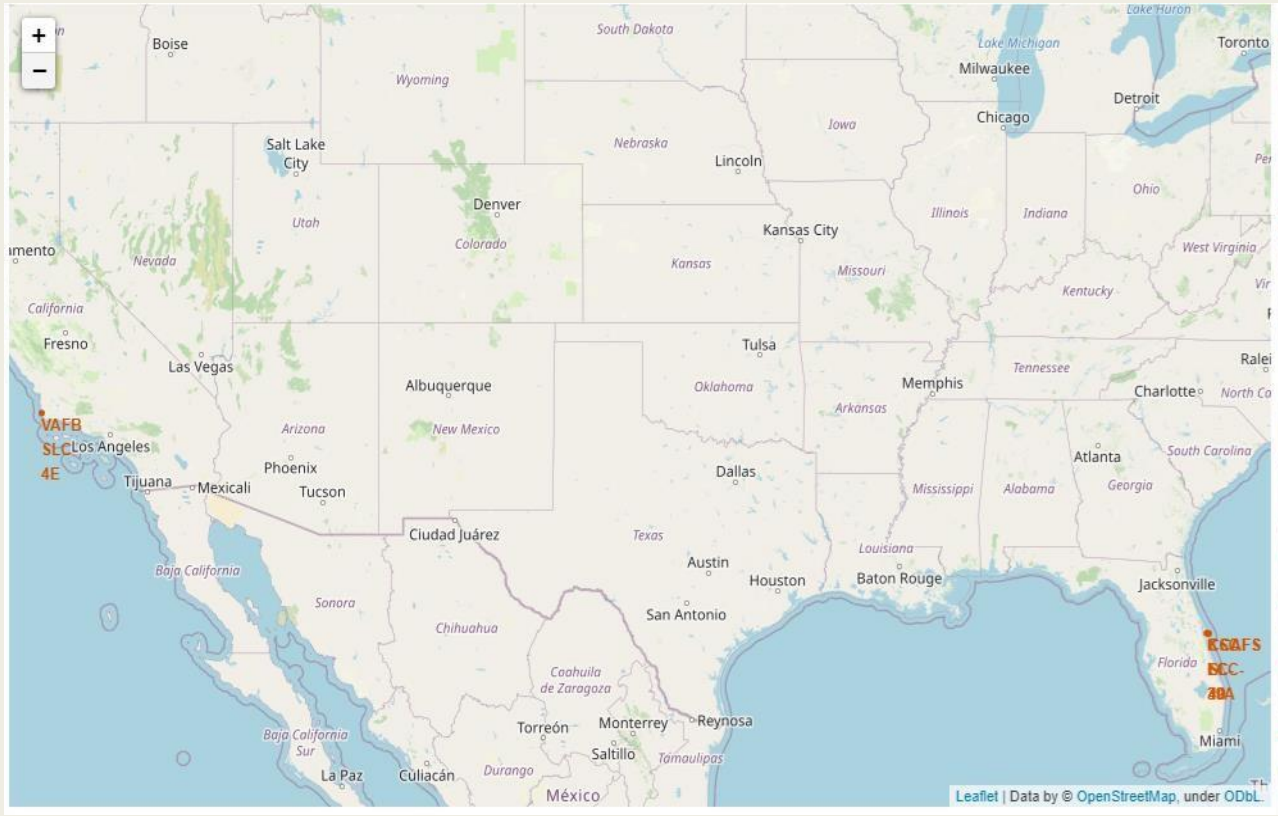
LAUNCH SUCCESS YEARLY TREND

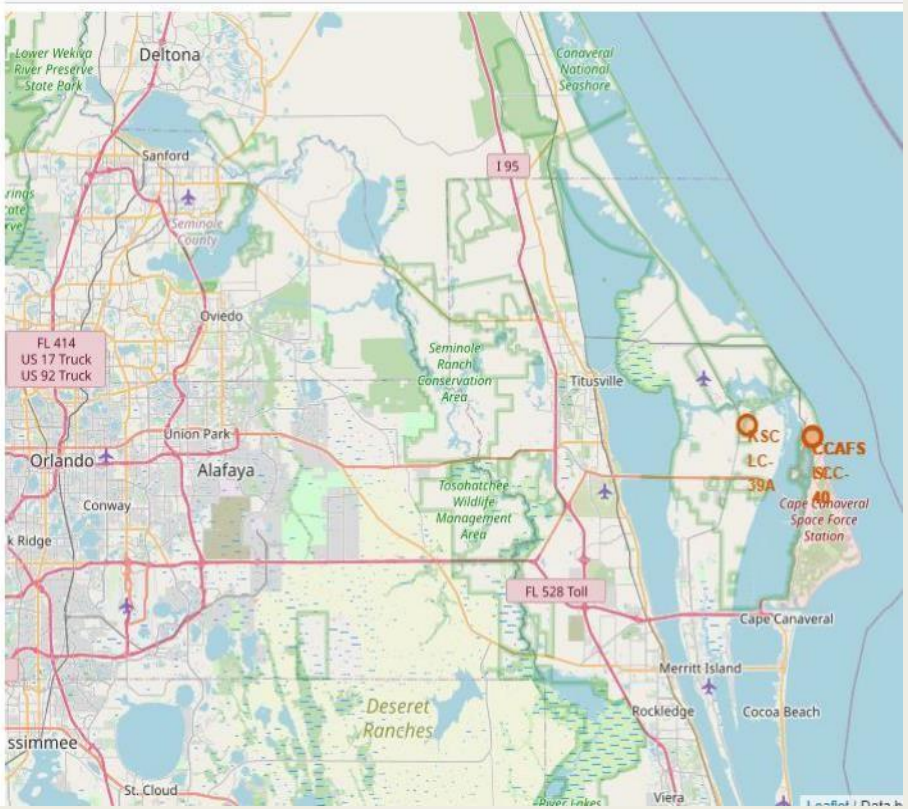


Since 2013, the success rate has generally trended upward, with a minor decrease observed in 2018. In recent years, the success rate has stabilized at approximately 80%.

Interactive Map with Folium

LAUNCH SITE LOCATIONS

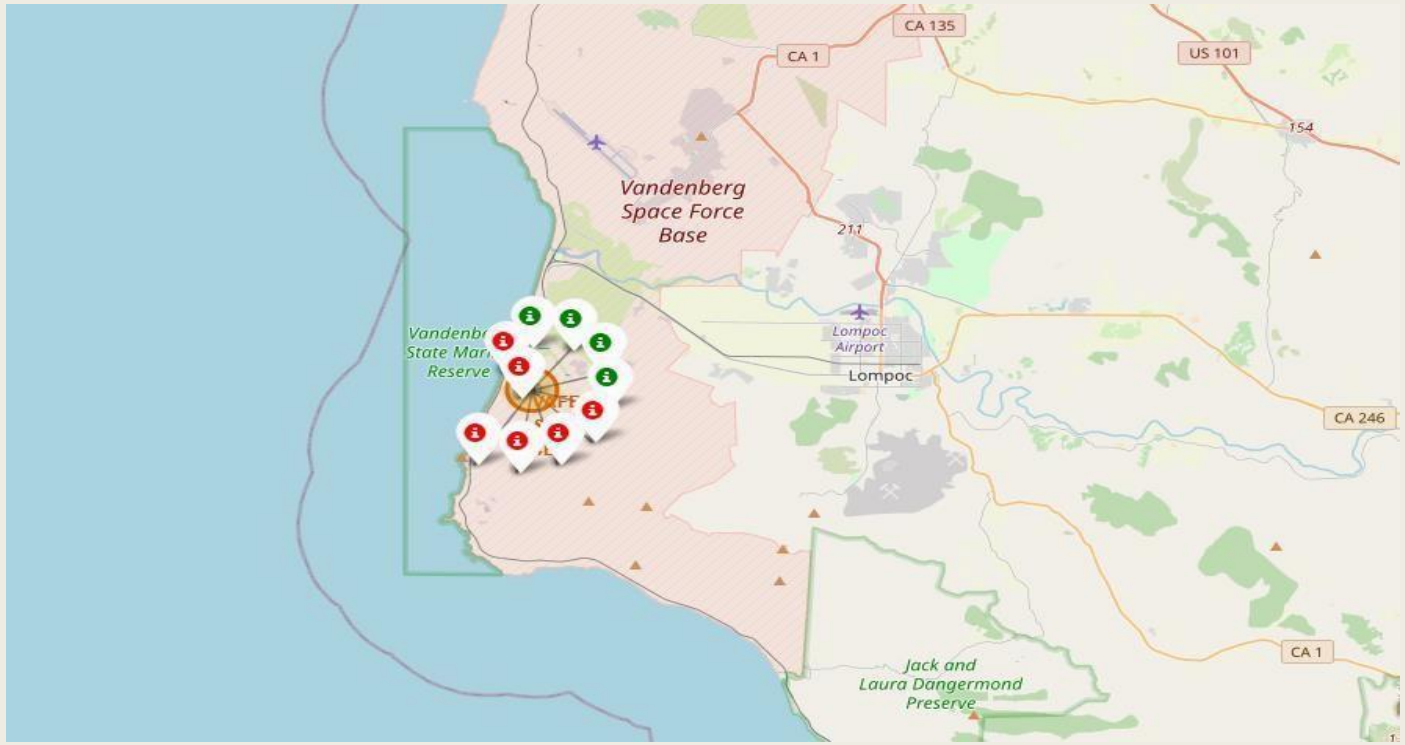




•The map on the left displays the locations of all launch sites with the overall map of the United States.

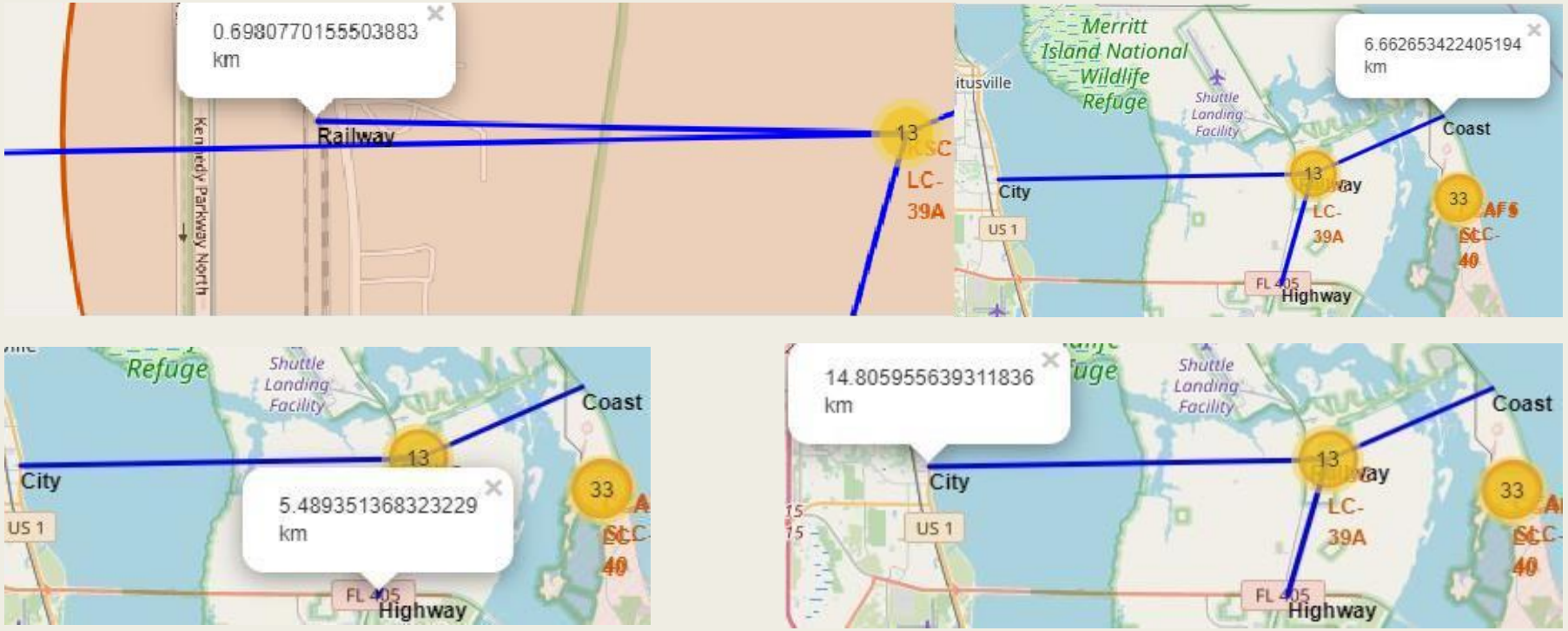
•The map on the right specifically focuses on the two launch sites situated in Florida, as they are geographically very close to one another.

Color-Coded Launch Markers



- On the Folium map, clusters of launch events are interactive and can be clicked.
- Clicking a cluster reveals individual landing outcomes, represented by different colored icons: green icons indicate successful landings and red icons indicate failed landings.

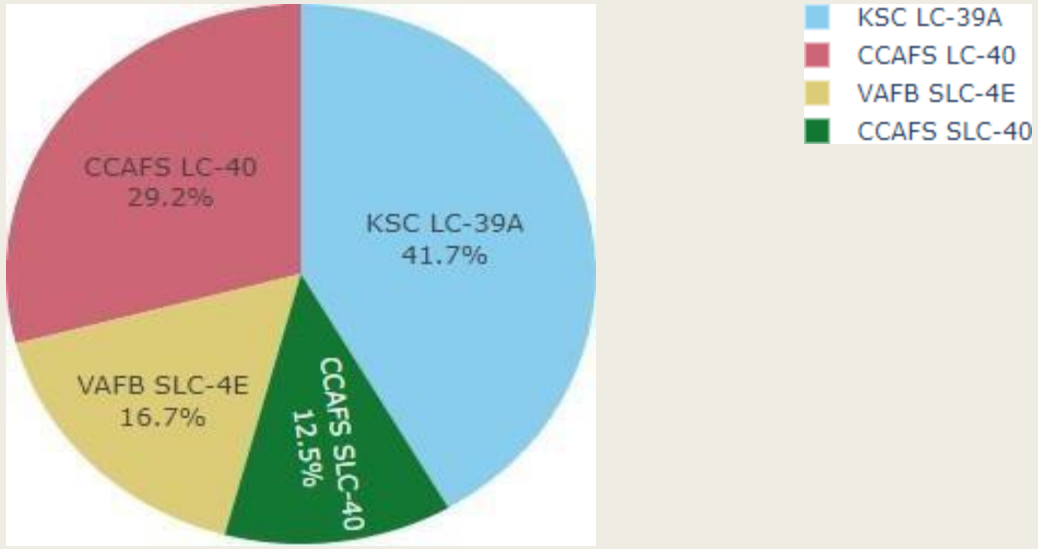
Key Location Proximities



Launch sites like KSC LC-39A are near railways and highways for transport, and close to coasts but far from cities for safety in case of launch failures over populated areas.

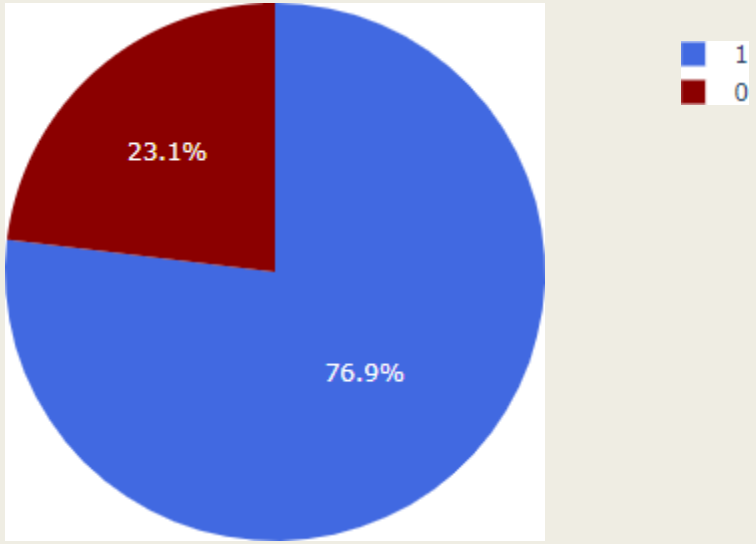
Build a Dashboard with Plotly Dash

Successful Launches Across Launch Sites



This shows where successful landings happened. CCAFS and KSC have the most, though CCAFS's successes mostly occurred under its former name (CCAFS LC-40). VAFB has the fewest, possibly due to fewer launches or more challenging conditions on the West Coast.

Highest Success Rate Launch Site



KSC LC-39A has the best success rate with 10 successful landings compared to 3 failures.

Payload Mass vs Success vs Booster Version Category

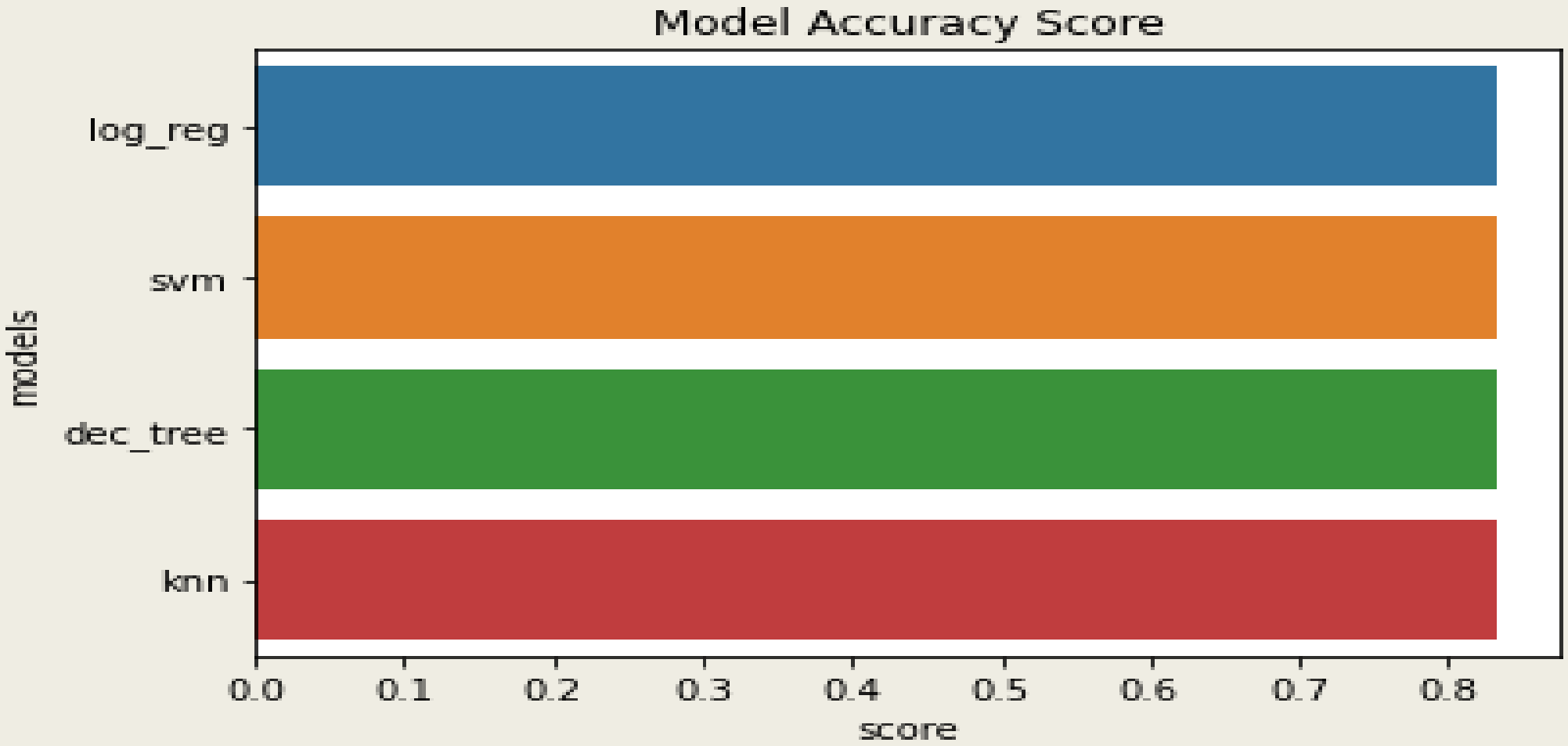
The Plotly dashboard includes a payload range selector, currently set from 0 to 10,000 kg, although the maximum payload is 15,600 kg. The scatter plot uses color to distinguish booster versions and point size to represent the number of launches. Successful landings are marked as '1' and failures as '0' (in the 'Class' variable). Notably, within the 0-6,000 kg payload range, there are two instances of failed landings with zero payload.

PREDICTIVE ANALYSIS

- GRIDSEARCHCV (with 10-fold cross-validation) was applied to :

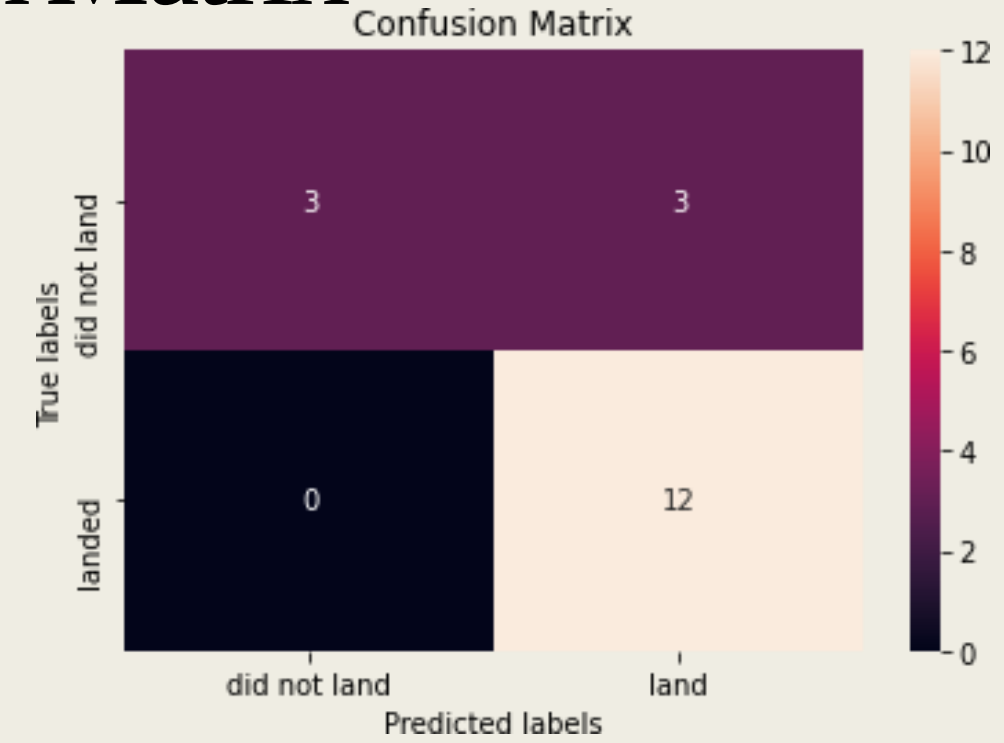
1. Logistic Regression
2. SVM (Support Vector Machine)
3. Decision Tree
4. KNN (K-Nearest Neighbors) models

CLASSIFICATION



All models achieved approximately the same accuracy (83.33%) on the test set. However, the test set was small. (only 18 samples), which can lead to significant variability in accuracy, as seen in repeated runs of the Decision Tree Classifier. More data is likely needed to confidently identify the best-performing model.

Confusion Matrix



Because all models showed equal performance on the test data, their confusion matrices are identical. The models correctly identified 12 actual successful landings. They also correctly predicted 3 actual unsuccessful landings. However, the models incorrectly predicted 3 unsuccessful landings as successful landings (these are false positives). This indicates a tendency for the models to over-predict successful landings.

CONCLUSION

- **Objective:** The project aimed to build a machine learning model for SpaceY to predict successful Stage 1 rocket landings, a crucial factor in cost savings (approximately USD 100 million per successful recovery) for competing with SpaceX.
- **Data Sources:** The data used for this project was collected from the public SpaceX API and by web scraping the SpaceX Wikipedia page.
- **Data Handling:** The collected data was labeled for training and stored in a DB2 SQL database for efficient management and querying.
- **Visualization:** A dashboard was created to visualize the data and potentially the model's performance.
- **Model Performance:** A machine learning model was developed and achieved an accuracy of 83% in predicting successful Stage 1 landings on the test data.
- **Potential Application:** Allon Mask of SpaceY can utilize this model before a launch to predict the likelihood of a successful Stage 1 landing. This prediction can inform the decision of whether to proceed with the launch, considering the potential cost savings from a successful recovery.
- **Recommendations:** It is recommended to gather more data to help in selecting the most effective machine learning model and to further enhance the prediction accuracy.

This model provides SpaceY with a valuable tool to assess the economic viability of their launches by predicting the success of Stage 1 recovery. The current accuracy of 83% offers a relatively high level of confidence for decision-making, but acquiring more data could lead to even more reliable predictions.

APPENDIX

- **GitHub URL:** https://github.com/dajiraoakash/Coursera/tree/main/Capstone

- **Instructors:** Rav Ahuja, Alex Aklson, Aije Egwaikhide, Svetlana Levitan, Romeo Kienzler, Polong Lin, Joseph Santarcangelo, Azim Hirjani, Hima Vasudevan, Saishruthi Swaminathan, Saeed Aghabozorgi, Yan Luo