IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Shiva Ram
20/04/2025

Outline

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

Executive Summary

Executive Summary:

- **Objective:** The objective of this project is to analyse SpaceX's Falcon 9 rocket launch data and develop a predictive model to estimate the success rate of Falcon 9's first stage rocket landings, enabling data-driven decision-making to improve reliability and cost optimisation for future missions.
- **Methodology Summary:** Collected data using SpaceX's API and web scraping techniques. Pre-processed the data through wrangling to handle inconsistencies and prepared for exploratory analysis and visualization. Generated training and test datasets from the analysis output to develop machine learning predictive models.
- **Results Summary:** Analysis of historical data reveals a significant improvement in SpaceX's first stage rocket landing success rate since 2013. Among the predictive models tested, the Decision Tree classifier achieved the highest accuracy at 89%, outperforming Regression, Support Vector Machine (SVM), and K-Nearest Neighbours (KNN), where their highest accuracy is around 83.3%.

Introduction

Project Background and Contexts

SpaceX aims to maximize the cost benefits of successful first stage landings as they are critical to overall mission costs and reliability. By analysing historical data and pre-launch parameters, predicting the likelihood of a successful first stage landing enables better mission planning, risk management, and operational efficiency.

Problem Statement:

This project is about assessing and predicting the Falcon 9 first stage landing success probability prior to launches. Through predictive model using historical mission data estimate the probability of a first stage landing success, enabling improved decision-making, risk assessment and mission planning through data modelling, data analysis and actionable insights.

Section 1

Methodology

Methodology - Data Collection (SpaceX API)

The data for this project was sourced using two methods: (1) API method for structured data (2) Web scraping method for getting HTML data.

API Method

- Collected data from an API using HTTP request method.
- From the response, collected features data - 'Boosterversion, Launchsites' , 'Payloads', 'Cores', and appended to the list.
- Filtered the dataset to include Falcon 9 rockets only, since the predictive modelling for success / failure rates will be built on Falcon 9 rockets.

  Data Normalization:
  - Normalized JSON data from the SpaceX API into a tabular structure using Python's json_normalize() to simplify access to nested fields.

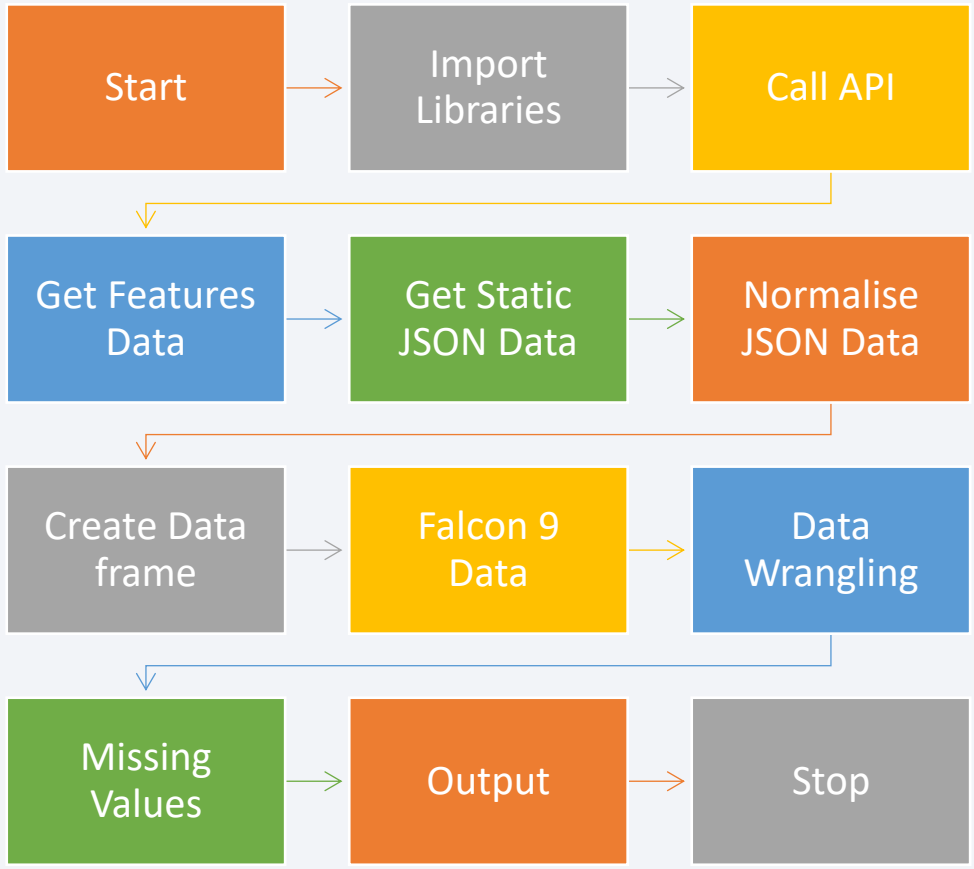
  Data Filtering:
  - Focused on relevant features, such as booster type, launch site, payload mass, and landing success outcomes, by removing unnecessary columns.
  - Filtered the dataset specifically for Falcon 9 missions to align with project objectives.

  Data Wrangling:
  - Handled missing values by imputing numerical fields (e.g., payload mass) with the mean and removing incomplete records that lacked critical data.
  - Standardized formats for date fields, ensuring consistency for time-series analysis.
- Save the cleansed data to a CSV file.

Flow Chart - Data Collection (SpaceX API)

Key Phrases for Each Step

1. **Start**: Initialize the data collection process.
2. **Import Libraries**: Load necessary libraries (requests, pandas, numpy, datetime, json).
3. **Call API**: Fetch data from SpaceX REST API.
4. **Get Features Data**: Extract boosterversion, launchsite, payload, and core from the API response.
5. **Get Static JSON Data**: Retrieve static JSON data if needed.
6. **Normalize JSON Data**: Convert nested JSON data into a flat table.
7. **Create Data frame**: Organize data into a pandas DataFrame.
8. **Falcon 9 Data**: Select records related to Falcon 9 launches.
9. **Data Wrangling**: Clean and transform data for analysis.
10. **Missing Values**: Handle any missing or null values in the dataset.
11. **Output**: Save the cleansed data to a CSV file.
12. **Stop**: End the data collection process.



GitHub Link - https://github.com/rams-star/shiva-test/blob/Space-X-Project/01%20jupyter-labs-spacex-data-collection-api.ipynb

Methodology - Data Collection (Web Scraping)

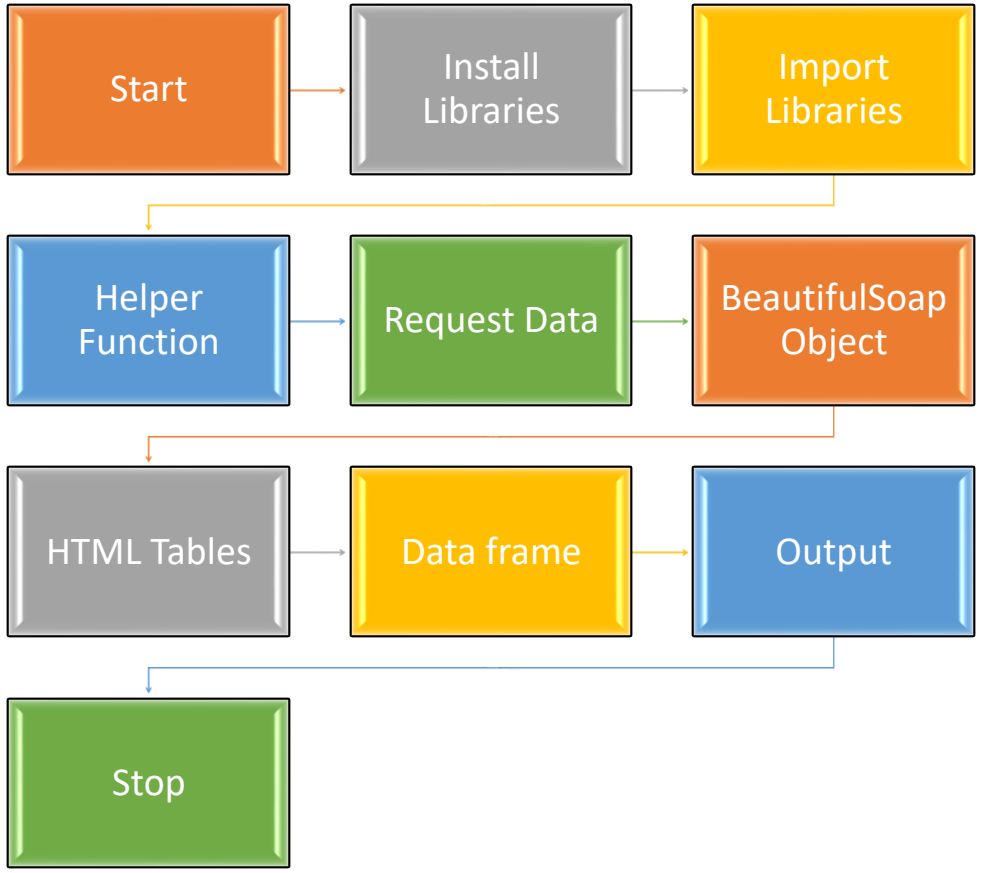
Web Scraping Method

- Collected Falcon 9 launch data from Wikipedia website.
- Extracted all column / variable names from HTML table header.
- Created a data frame by parsing the launch HTML tables.
- Stored the web scraped data frame onto a CSV file.

Flow chart - Data Collection (Web Scraping)

Key Phrases for Each Step

1. **Start**: Initialize the web scraping process.
2. **Install Libraries**: BeautifulSoap, Requests
3. **Import Libraries**: Load necessary libraries (requests, pandas, sys, BeautifulSoap, unicodedata).
4. **Helper Function**: To process web scrapped HTML table.
5. **Request Data**: Request Falcon 9 launch Wiki page from its URL.
6. **BeautifulSoap Object**: Create BeautifulSoap object from response text content.
7. **HTML Tables**: Extract all columns / variable names from HTML table header.
8. **Data Frame**: Create a data frame by parsing the launch HTML tables.
9. **Output**: Save the cleansed data frame data to a CSV file.
10. **Stop**: End the web scraping process.



GitHub Link - https://github.com/rams-star/shiva-test/blob/Space-X-Project/02%20jupyter-labs-webscraping.ipynb

Methodology - Data Wrangling

Source Data:

- Used CSV file created from API data collection method as source data.

Data Analysis:

- Identified and calculated percentage of the missing values in each attribute.
- Calculated number of launches on each site.
- Calculated the number and occurrence of each orbit.
- Calculated the number and occurrence of mission outcome of the orbits.
- Created a landing outcome label from Outcome column.

Output:

- Saved the wrangled data to a CSV file for exploratory data analysis (EDA) purpose.

Flow chart - Data Wrangling

Key Phrases for Each Step

1. **Start**: Initialize the data wrangling process.
2. **Install Libraries**: Pandas, Numpy
3. **Import Libraries**: Load necessary libraries (pandas, numpy).
4. **Data Collection**: Rretrieve data from an API and saved it as a CSV file.
5. **Data Cleaning:**
   - Identified missing values in each attribute and calculated their percentages.
   - Handled missing values (e.g., by imputation or removal).
6. **Data Transformation:**
   - Calculated the number of launches at each site.
   - Determined the number and occurrence of each orbit.
   - Analysed the mission outcomes for each orbit.
   - Created a new label for landing outcomes based on the Outcome column.
7. **Data Storage:**
   - Saved the cleaned and transformed data to a new CSV file for further analysis.
8. **Stop**: End the web scraping process.

GitHub Link - https://github.com/rams-star/shiva-test/blob/Space-X-Project/03%20labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Summary:

Plotted series of **scatter plots** to see what combinations will result in more likely to land successfully at the first stage.

1) Flight number vs Payload Mass – To see that as the flight number increases the first stage is more likely to land successfully. The payload mass also appears to be a factor; even with more massive payloads, the first stage often returns successfully.
2) Flight number vs Launch Site – To see the patterns of the successful / failed landings on each launch site.
3) Payload Mass vs Launch Site – To see if any rocket has been launched from each launch site based on payload mass.
4) Flight Number vs Orbit – To see the relationship between flight number and different orbits.
5) Payload Mass vs Orbit type – To see the successful landing on each orbit based on payload mass.

Bar Plot – To check if there are any relationship between success rate of each orbit type.

Line Plot – To see the success landing rate at first stage, year on year from 2010.

GitHub Link - https://github.com/rams-star/shiva-test/blob/Space-X-Project/05%20jupyter-labs-eda-dataviz.ipynb.ipynb

Methodology - Exploratory data analysis (EDA) using visualization and SQL

SQL for Data Exploration:

- Wrote SQL queries to segment and analysed data from the structured database.
- Identified mission outcome for Falcon 9 using GROUP BY and aggregate functions.

Visualization Tools:

- Leveraged Python libraries, matplotlib and Seaborn, to plot distributions, trends, and correlations.
- Created multiple scatter plots to visualise the relationships between Flight number, Launch Sites, Payload Mass and Orbit types.
- Created bar plots to show landing success rates for each orbit type.

Insights Derived: Discovered that Falcon 9 boosters showed a consistent success rate improvement after 2013.

Feature Engineering: Applied features on categorical columns and also created dummy variables for machine learning.

Data Storage: Stored the dataframe in CSV file for building models.

EDA with SQL

Summary - Used my_data1.db database for EDA purpose. Used sqlite to write SQL queries

1. Dropped if SPACEXTABLE existed and created SPACEXTABLE for query purpose.
2. Query to display unique launch sites.
3. Query to display 5 records for launch sites having starting letters as CCA.
4. Query to display total payload mass for customer NASA (CRS).
5. Query to display average payload mass for booster version F9 v1.1.
6. Query to display the data when the first successful landing outcome in ground pad was achieved.
7. Query to display names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
8. Query to display total number of successful and failure mission outcomes.
9. Using sub-query method display all the booster versions that have carried the maximum payload mass.
10. Query to show the records which will display the month names, failure landing outcomes in drone ship, booster versions, launch site for the months in the year 2015. Used sub-string function to extract year from date column.
11. Query to “rank” the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

GitHub Link - https://github.com/rams-star/shiva-test/blob/Space-X-Project/04%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

Methodology - Interactive visual analytics using Folium and Plotly Dash

Using Folium:

- Marked latitudes and longitudes for all launch sites.
- Marked the success / failed launches for each site in the map.
- Created a cluster and used marker with customised icon to identify the success / failed launch.
- From the data, picked CCAFS SLC-40 (Cape Canaveral Space Launch Complex 40) as the launch site for calculation purpose.
- Calculated the closest coastal point from CCAFS SLC-40.
- Calculated the closest railway station, highway, city from CCAFS SLC-40.

Using Plotly Dash:

- Created a drop down for user to select launch site.
- Created a slider to show the min and max Pay load values.
- Created call-back function to show pie chart as per values selected from the dropdown and the slider.
- Pie chart will be plotted showing the success / failed rate.

Build an Interactive Map with Folium

Summary: For building an interactive map with folium, used folium circle, markers and polyline objects.

1. Folium Circle - To highlight the SpaceX launch sites, with a defined radius on the map for visibility.
2. Folium markers – To focus on the geospatial coordinates of SpaceX launch sites.
3. Folium polyline - To draw lines between two geospatial locations and calculate the distance. For e.g.,
   - To calculate the distance between launch site CCFS SLC-40 and nearest coast.
   - To calculate the distance between launch site CCFS SLC-40 and nearest railway station.
   - To calculate the distance between launch site CCFS SLC-40 and nearest highway road.
   - To calculate the distance between launch site CCFS SLC-40 and nearest city.

GitHub Link - https://github.com/rams-star/shiva-test/blob/Space-X-Project/06%20lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Summary: Built an interactive dashboard to allow the users to select SpaceX launch sites and adjust the payload mass to find out Falcon 9's first stage landing success probability.

1. **Drop down option** - To allow the user to filter the data by selecting a specific launch site or viewing data for all sites. When a site is selected, the pie chart and scatter chart update to reflect data for that site.
2. **Pie-chart** for successful launches –
   - •All Sites: When 'All Sites' is selected, the pie chart shows the total number of successful launches for each launch site.
   - •Specific Site: When a specific site is selected, the pie chart shows the proportion of successful vs. failed launches for that site.
   - •Interaction: The pie chart updates dynamically based on the selected launch site from the dropdown.
3. **Payload Range Slider**:
   - •This slider allows users to filter the data based on the payload mass range. Users can adjust the range to see how different payload masses affect launch success.
   - •Interaction: The scatter chart updates dynamically based on the selected payload range.
4. **Scatter Chart** for Payload vs. Launch Success:
   - •All Sites: When 'All Sites' is selected, the scatter chart shows the correlation between payload mass and launch success for all sites.
   - •Specific Site: When a specific site is selected, the scatter chart shows the correlation for that site only.
   - •Interaction: The scatter chart updates dynamically based on the selected launch site and payload range, allowing users to explore how payload mass impacts launch success across different sites.

GitHub Link - https://github.com/rams-star/shiva-test/blob/Space-X-Project/07%20PlotlyDash_SpaceX-dash-app_JupyterVersion.py

Methodology - Predictive Analysis using Classification

Using Classification:

- Classified and standardised the data created from EDA.
- Split the data into training and test dataset.
- Created objects (Logistic Regression, Support Vector Machine (SVM), Decision Tree, K Nearest Neighbours (KNN)) to find the best parameters from the dictionary.
- Calculated the accuracy of the test data using score method.
- Created Confusion matrix for Logistic Regression, Support Vector Machine (SVM), Decision Tree, K Nearest Neighbours (KNN)) to evaluate the performance.
- Identified SVM as the better model, compared to other models in terms of testing accuracy.
- Applied hyperparameter from the existing parameter and calculated logistic regression, SVM, decision tree and KNN test results and Cross Valuation (CV) score.

Predictive Analysis (Classification)

Summary

1. Defined a function to plot confusion matrix to evaluate the performance of classification model.
2. Built data frames built from Exploratory Data Analysis (EDA) process.
3. Created NumPy array from data column “Class”, applied to_numpy() method and assigned to a variable (Y).
4. Using transform method, standardised the data and reassigned to variable (X).
5. Using the function train_test_split, split the X and Y data into training and test data, and set the parameter test_size to 0.2 and random_state to 2.
6. **Logistic Regression Object** -
   - Created logistic regression object then created GridSearchCV object and fitted the object to find the best parameters.
   - Displayed the best parameters using the data attribute best_params_ and accuracy on the validation data using the data attribute best_score_
   - Calculated the accuracy on the test data using method score.
   - Plotted confusion matrix to interpret of logistic regression model.

GitHub Link - https://github.com/rams-star/shiva-test/blob/Space-X-Project/08%20SpaceX_Machine%20Learning%20Prediction.ipynb

Predictive Analysis (Classification)

7. **Support Vector Machine (SVM) Object** -
   - Created SVM object then created GridSearchCV object and fitted the object to find the best parameters.
   - Displayed the best parameters using the data attribute best_params_ and accuracy on the validation data using the data attribute best_score_
   - Calculated the accuracy on the test data using method score.
   - Plotted confusion matrix to interpret support vector machine model.

8. **Decision tree classifier Object** -
   - Created decision tree object then created GridSearchCV object and fitted the object to find the best parameters.
   - Displayed the best parameters using the data attribute best_params_ and accuracy on the validation data using the data attribute best_score_
   - Calculated the accuracy on the test data using method score.
   - Plotted confusion matrix to interpret support vector machine model.

GitHub Link - https://github.com/rams-star/shiva-test/blob/Space-X-Project/08%20SpaceX_Machine%20Learning%20Prediction.ipynb

Predictive Analysis (Classification)

7. **K Nearest Neighbour (KNN) Object** -
   - Created KNN object then created GridSearchCV object and fitted the object to find the best parameters.
   - Displayed the best parameters using the data attribute best_params_ and accuracy on the validation data using the data attribute best_score_
   - Calculated the accuracy on the test data using method score.
   - Plotted confusion matrix to interpret support vector machine model.

8. Found best performing classification model (as per their accuracy rate) is Support Vector Machine (SVM).

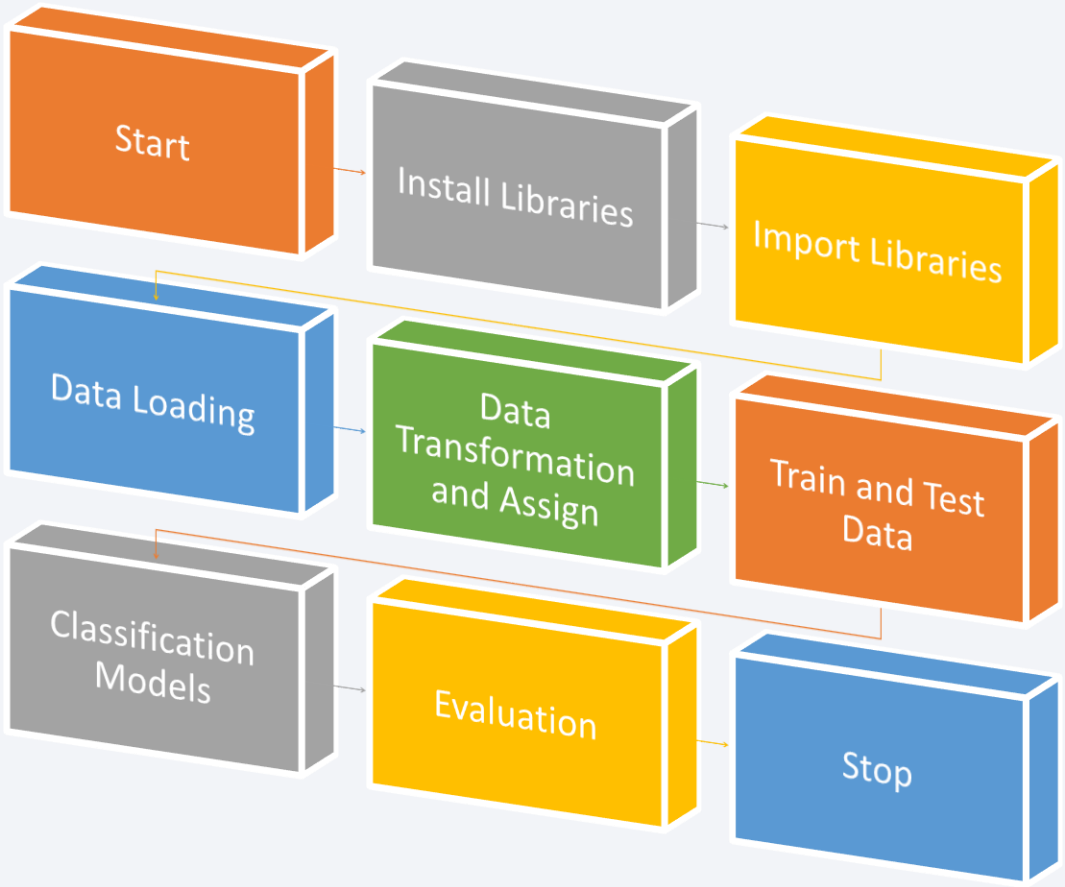
9. From hyperparameters, found decision tree model is better than other classification models.

GitHub Link - https://github.com/rams-star/shiva-test/blob/Space-X-Project/08%20SpaceX_Machine%20Learning%20Prediction.ipynb

Predictive Analysis (Classification) – Flow Chart

Key Phrases for Each Step

1. **Start**: Evaluate the performance of classification model.
2. **Install Libraries**: piplite
3. **Import Libraries**: Pandas, Numpy, Seaborn, Matplotlib, Sklearn.
4. **Data Loading**: Built data frames from EDA data.
5. **Data Transformation and Assign**:
   - Created NumPy array from data column “Class”, applied to_numpy() method and assigned to a variable (Y).
   - Using transform method, standardised and reassigned the data to variable X.
6. **Train and Test data:** Split data for training and testing purpose and set the parameters.
7. **Classification Models –** Build Regression, SVM, Decision Tree, KNN:
   - Calculate CV scores for all classification models.
   - Plot confusion matrix chart for model interpretation.
8. **Evaluation –** Identify the best performing model considering its internal parameters and pre-defined hyperparameters.
9. **Stop**: Evaluated the performance of classification model.



Data Science Project
Presentation prepared by: Shiva Ram

GitHub Link - https://github.com/rams-star/shiva-test/blob/Space-X-Project/08%20SpaceX_Machine%20Learning%20Prediction.ipynb

Results

Exploratory Data Analysis Results:

- Launch Sites - SpaceX uses 4 launching sites - CCAFS LC-40, CCAFS SLC-40, VAFB SLC-4E, SC LC-39A. Technically, they are 3 sites, because CCAFS LC-40, CCAFS SLC-40 are in the same geographical location.
- Total Payload Mass for all rocket launches used for NASA (customer) is 45,596 kgs.
- Average payload mass used for Falcon 9.1 series is 2,928.4 kgs.
- First successful ground landing rate was recorded as 22nd December 2015.
- 4 times Falcon 9 had successful drone landing with payload mass between 4,000 and 6,000 kgs.
- Mission was successful 100 times (includes 1 time success with payload mass stats unclear), and 1 time was failure.
- Heaviest payload mass recorded is 15,600kgs. 12 times, Falcon 9 carried this heavy payload mass.
- Booster F9 v1.1 failed twice to land drone ship successfully in 2015 from the same launch site CCAFS LC-40.
- Landing outcome - Failure (drone ship) has is number 1 in ranking order, followed Success (ground pad).

Results

Predictive Analysis Results:

Predictive analysis was done on different models to estimate the model accuracy. Result shows SVM has higher accuracy rate, compared to other classification models.

Model	Accuracy
Logistic Regression	79.2%
Support Vector Machine (SVM)	81.9%
Decision Tree	78.9%
K Nearest Neighbour (KNN)	75%

Applying hyperparameters model's accuracy rate changes. Result shows Decision tree has higher accuracy rate, compared to other classification models.

Model	Accuracy
Logistic Regression	83.4%
Support Vector Machine (SVM)	83.4%
Decision Tree	87.4%
K Nearest Neighbour (KNN)	86.1%

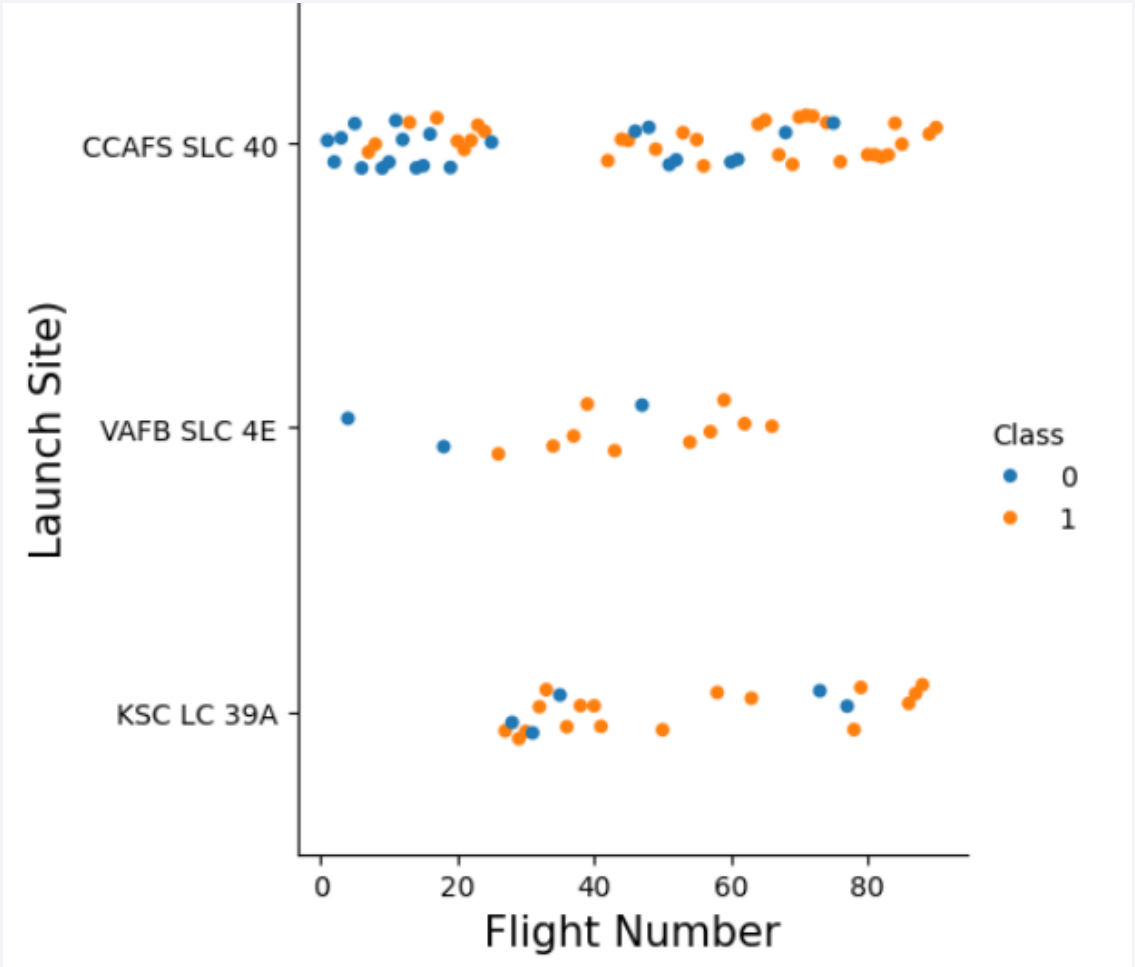
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

The scatter plot displays the following:

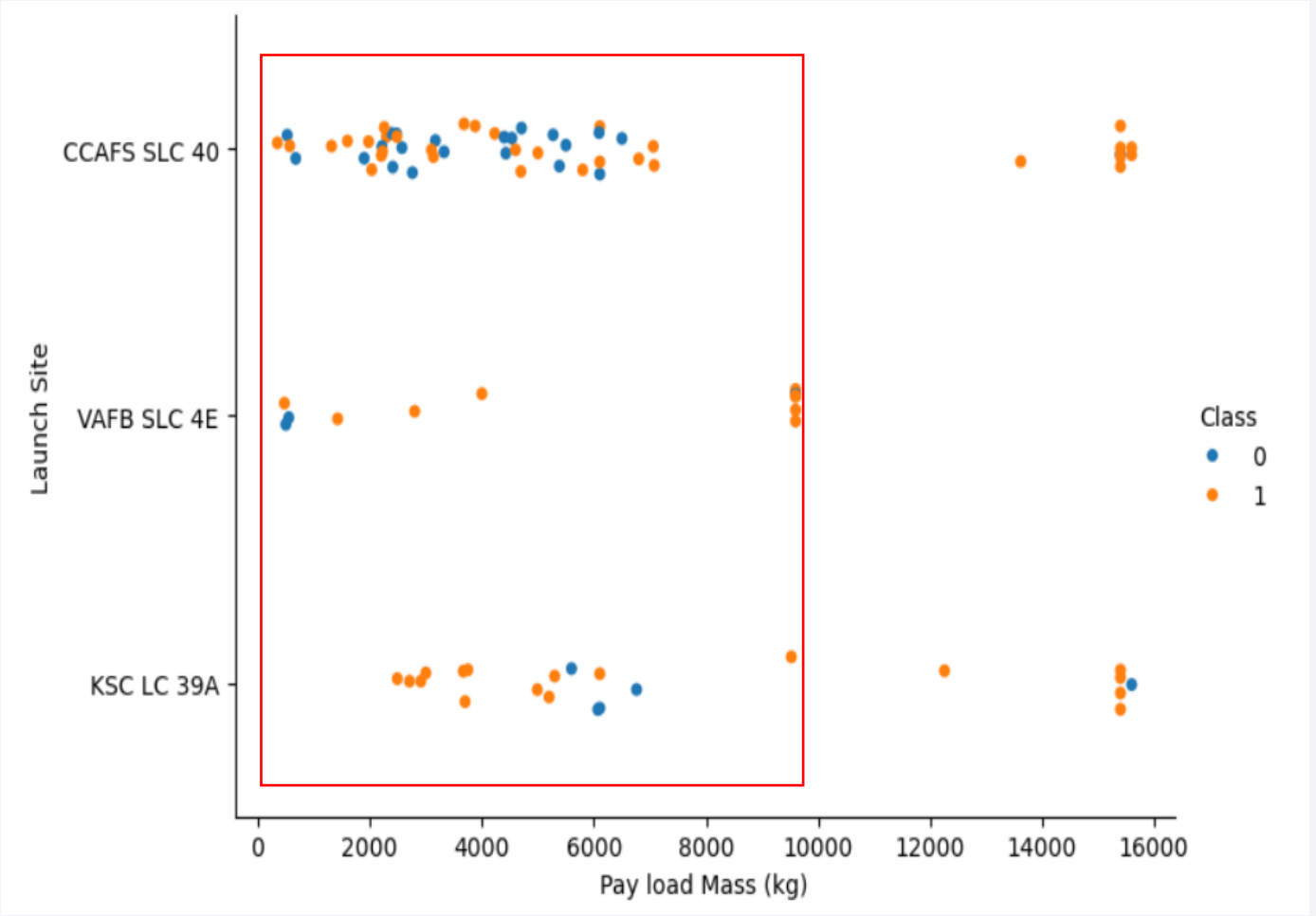
1. **X-axis (Flight Number):** Represents the sequential number of each flight.
2. **Y-axis (Launch Site):** Indicates the different launch sites where the flights took place.
3. **Class:**
   - 1 represents a successful launch.
   - 0 represents an unsuccessful launch.
4. **Flights Distribution:** Total 90 Falcon 9 flights launched between 2010 and 2020.
   - CCAFS SLC 40: Launches the most flights (55 flights).
   - KSC LC 39A: Launches the second most flights, followed by VAFB SLC 4E (22 flights).
   - KSC LC 39A: Started launching rockets after CCAFS SLC 40 and VAFB SLC 4E (13 flights).
5. **Class Comparison:**
   - KSC LC 39A: Has a higher success rate compared to other sites, based on the success/failure ratio.



Payload vs. Launch Site

The scatter plot displays the following:

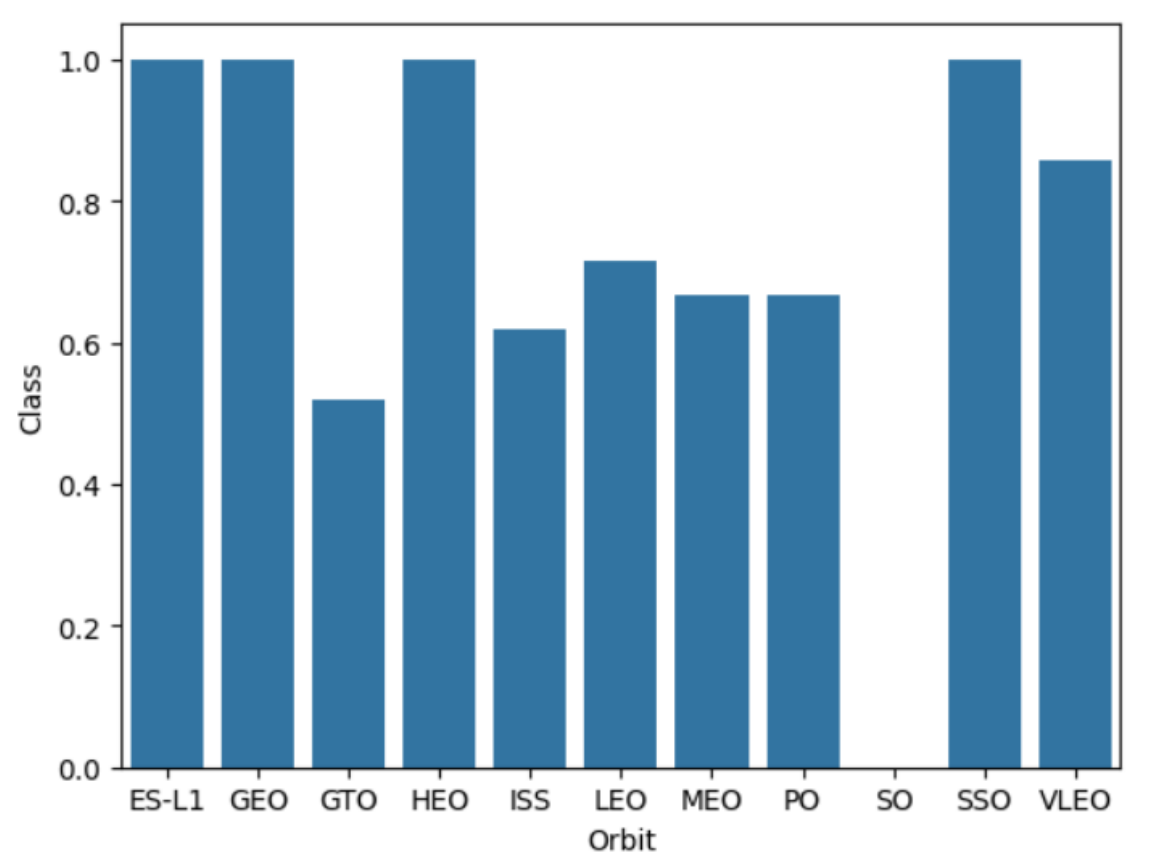
1. **X-axis (Pay load mass (kg)):** Range from 350kgs to 15,600kgs.
2. **Y-axis (Launch Site):** Indicates the different launch sites where the flights took place.
3. **Class:**
   - 1 represents a successful launch.
   - 0 represents an unsuccessful launch.
4. **Pay load mass (kg) :**
   - Majority of Falcon 9 rockets weighs below 10,000kgs.
   - VAFB SLC 4E site has not launched rockets above 10,000kgs.
5. **Launch Site Trends:**
   - CCAFS SLC 40 is used more frequently for rocket launch than other sites.



Success Rate vs. Orbit Type

The bar plot displays the following:

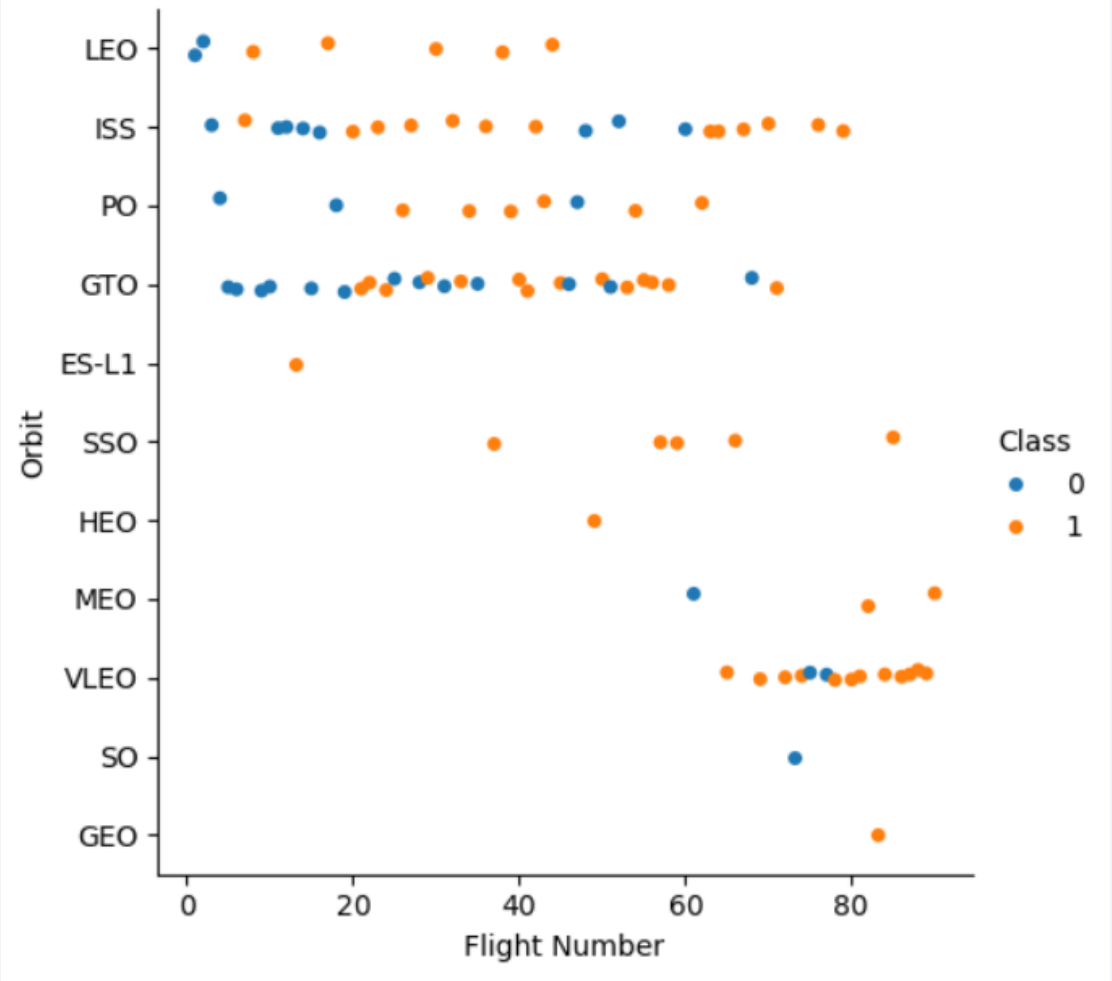
1. **X-axis (Orbits):** SpaceX Falcon 9 rockets use these orbital destinations.
2. Orbit Abbreviations:
   - ES-L1: Earth-Sun Lagrange 1 point
   - GEO - Geostationary Earth orbit
   - GTO - Geostationary transfer orbit
   - HEO - Highly elliptical orbit
   - ISS - International Space Station
   - LEO - Low Earth orbit
   - MEO - Medium Earth orbit
   - PO – Polar Orbit
   - SO / SSO - Sun-synchronous orbit
   - SSO – VLEO - Very low Earth orbit
3. **Y-axis (Class):** Indicates the launch success rate to the orbits.
4. **Success Rate:**
   - ES-L1, GEO, HEO and SSO has 100% success rate.
   - Least success rate is GTO (50%).



Flight Number vs. Orbit Type

The scatter plot displays the following:

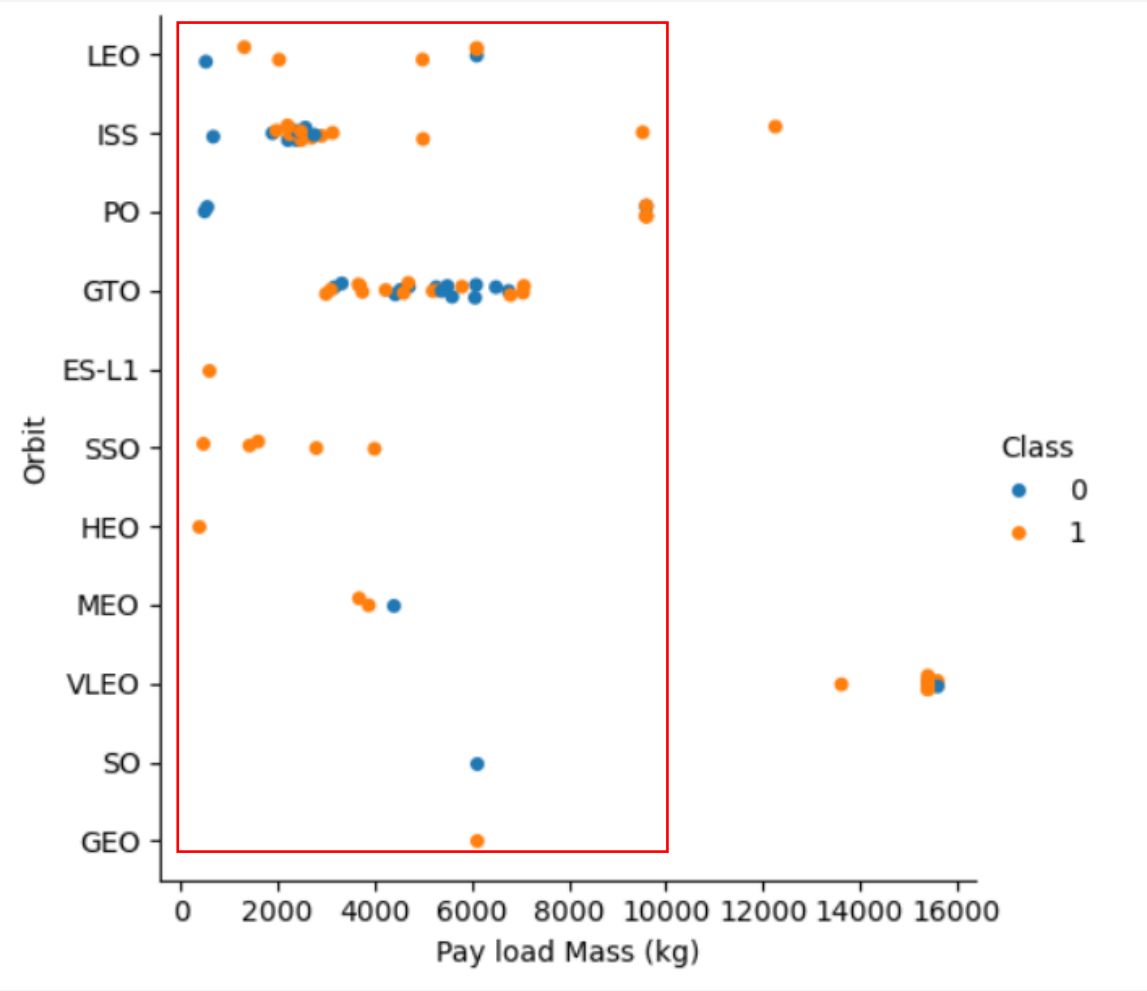
1. **X-axis (Flight Number):** Represents the sequential number of each flight.
2. **Y-axis (Orbit):** Indicates the different orbits where the flights orbited.
3. **Class:**
   - 1 represents a successful launch.
   - 0 represents an unsuccessful launch.
4. **Orbit Preference:** 90 times Falcon 9 has been orbited in the space between 2010 and 2020. In this period:
   - GTO is the most preferred orbit (27 launches), followed by ISS (21 launches).
   - ES-L1, GEO orbits are least preferred.



Payload vs. Orbit Type

The scatter plot displays the following:

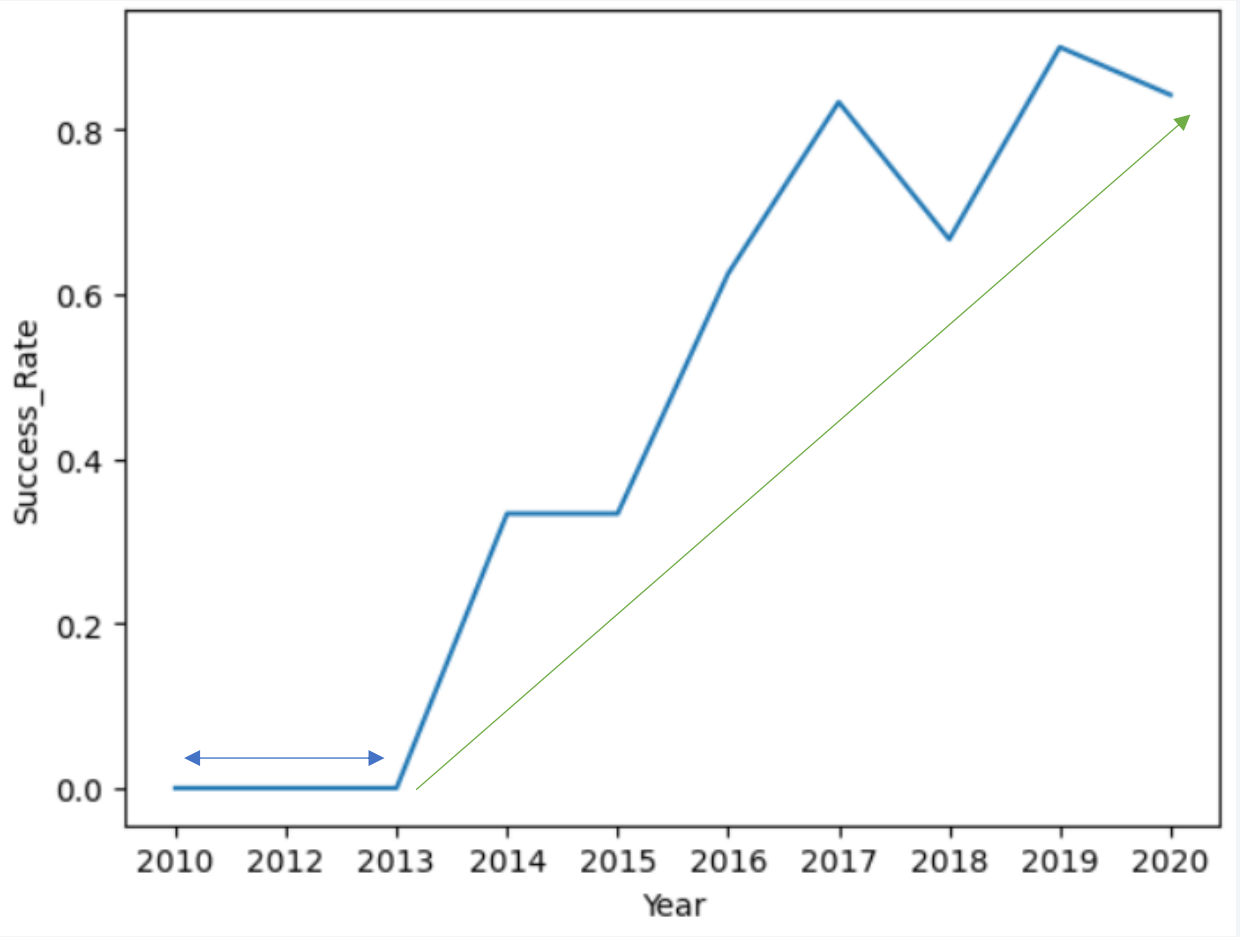
1. **X-axis (Pay load mass (kg)):** Range from 350kgs to 15,600kgs.
2. **Y-axis (Launch Site):** Indicates the different orbits where Falcon 9 rockets were deployed.
3. **Class:**
   - 1 represents a successful launch.
   - 0 represents an unsuccessful launch.
4. **Pay load mass (kg) :**
   - Majority of Falcon 9 rockets weighs below 10,000kgs.
   - Heavier payloads tend to be less frequent.
5. **Orbit Trends:**
   - GTO (Geostationary Transfer Orbit) is the most frequently used orbit.
   - ISS (International Space Station) is another common destination.
   - ES-L1 (Earth-Sun Lagrange Point 1) and GEO (Geostationary Orbit) are less frequently used.



Launch Success Yearly Trend

The line plot displays the following:

1. **X-axis (Year):** Indicates the time-period covered by the graph.
2. **Y-axis (Success Rate):** Indicates the success rate of Falcon 9 rocket launches.
3. **Yearly Trend:**
   - Until 2013: The success rate was nil.
   - After 2013: The success rate improved year on year.
   - As of 2020: The success rate is almost 80%.
4. **Visualization:**
   - The line plot provides a clear visual representation of the improvement in launch success rates over time.
   - The upward trend indicates significant advancements in Falcon 9 technology and operational reliability.



All Launch Site Names

Launch sites used by SpaceX are:

CCAFS LC-40, CCAFS SLC-40, VAFB SLC-4E, SC LC-39A

Note: Even though, query result from the data shows 4 launch sites, technically launch sites CCAFS LC-40 and CCAFS SLC-40 are same. CCAFS SLC-40 is more sophisticated to use newer rocket systems, whereas CCAFS LC-40 is the older name for the site. Hence, we can say SpaceX uses 3 launch locations (Cape Carnival Space Force Station, Vandenberg Space Force Base and Kennedy Space Center).

Query Result with explanation:

- Created database table SPACEXTBL in my_data1.db
- Used SELECT statement with DISTINCT keyword to query unique launch sites from table SPACETBL.
- The result shows a list of unique launch sites from the SPACEXTBL table. Each entry represents a distinct launch site name, ensuring no duplicates are included.

Display the names of the unique launch sites in the space mission

```
%%sql
select distinct Launch_Site from SPACEXTBL
```

```
 * sqlite:///my_data1.db
Done.
```

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

Launch Site Names Begin with 'CCA'

Query Result with explanation:

- Used SELECT statement from table SPACETBL and filtered to launch sites that has CCA characters and applied LIMIT 5 condition to display 5 records only.
- The result shows the first 5 launch sites from the SPACEXTBL table that start with 'CCA'. Each row represents a unique launch site, along with additional columns that provide more details about each site.

Display 5 records where launch sites begin with the string 'CCA'

```
%%sql
select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5
```

```
 * sqlite:///my_data1.db
Done.
```

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

Total Payload Mass

Query Result with explanation:

- This query uses the SELECT statement to calculate the total payload mass (in kilograms) for launches where the customer is 'NASA (CRS)'. The SUM function is used to add up the values in the PAYLOAD_MASS__KG_ column for these specific records.
- The result shows that the total payload mass launched by SpaceX for NASA (CRS) is 45,596 kg. This indicates the cumulative weight of all payloads carried by SpaceX rockets for NASA's Commercial Resupply Services (CRS) missions.

```
%%sql
select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = 'NASA (CRS)
```

```
 * sqlite:///my_data1.db
Done.
```

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

Query Result with explanation:

- This query uses the SELECT statement to calculate the average payload mass (in kilograms) for launches using Falcon 9 booster versions that are part of the 1.1 series. The AVG function is used to compute the mean value of the PAYLOAD_MASS__KG_ column for these specific records.
- The result shows that the average payload mass for Falcon 9 booster version 1.1 series is 2,982.4 kg. This indicates the typical weight of payloads carried by Falcon 9 rockets with this booster version.

```
%%sql
select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version like 'F9 v1.1'
```

```
 * sqlite:///my_data1.db
Done.
```

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

Query Result with explanation:

- This query uses the SELECT statement and MIN function to find out the starting date when the landing outcome from ground pad was successful.
- The result shows that the first successful landing on a ground pad occurred on 22nd December 2015. This indicates the date of the earliest successful ground pad landing recorded in the SPACEXTBL table.

```
select min(Date) as Success_Date_Launch from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'
```

```
 * sqlite:///my_data1.db
Done.
```

Success_Date_Launch
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Query Result with explanation:

- This query uses the SELECT statement to find out the booster version for successful landing outcome for drone ship where Payload mass is between 4000 and 6000 kgs.
- The result shows four booster versions meeting the criteria where payload mass is between 4,000kgs and 6,000kgs and, also drone ship landing was successful.

```
select Booster_Version from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000
```

```
 * sqlite:///my_data1.db
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

Query Result with explanation:

- This query uses the SELECT statement to find out the number of missions with different outcomes from the SPACEXTBL table. It groups the results by the Mission Outcome and counts how many times each outcome occurs. The results are as follows:
  - Failure (in flight): There was 1 mission that failed during flight.
  - Success: There were 99 missions that were successful.
  - Success (payload status unclear): There was 1 mission where the launch and flight were successful, but the status of the payload was unclear.

- The data shows that SpaceX has a very high mission success rate, with 99 out of 101 missions being successful. There was only one in-flight failure and one mission where the payload status was unclear, highlighting the overall reliability of their missions.

```
%%sql
select Mission_Outcome, count(Mission_Outcome) as Mission_Outcome_Count from SPACEXTBL group by Mission_Outcome
```

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	Mission_Outcome_Count
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

Query Result with explanation:

- This query uses the SELECT statement to retrieve the Booster Version and PAYLOAD_MASS__KG_ from the SPACEXTBL table. It specifically looks for the row(s) where the PAYLOAD_MASS__KG_ is equal to the maximum payload mass in the table. The subquery (select max(PAYLOAD_MASS__KG_) from SPACEXTBL) finds the maximum payload mass, and the main query retrieves the corresponding booster version and payload mass.
- This query identifies the booster version that carried the heaviest payload which is 15,600kgs. It finds the maximum payload mass from the table and then retrieves the booster version associated with that payload.

```
%%sql
select Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTBL
    where PAYLOAD_MASS__KG_ in (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

Query Result with explanation:

- This query uses the SELECT statement to retrieve specific columns from the SPACEXTBL table. It formats the Date column to show the month and year in the format MM YYYY and renames this column as Month_Year. It also selects the Booster_Version, Landing_Outcome, and Launch_Site columns. The query filters the results to include only those rows where the year in the Date column is 2015 and the Landing_Outcome is 'Failure (drone ship)'.
- This query extracts data for missions in 2015 that had a landing failure on a drone ship. It formats the date to show the month and year, and retrieves the booster version, landing outcome, and launch site for these specific missions.

```
%%sql
select substr(Date, 6,2) || ' ' || substr(Date,0,5) as Month_Year,  Booster_Version, Landing_Outcome, Launch_Site
    from SPACEXTBL where strftime('%Y', Date) = '2015' and Landing_Outcome = 'Failure (drone ship)'
```

```
 * sqlite:///my_data1.db
Done.
```

Month_Year	Booster_Version	Landing_Outcome	Launch_Site
01 2015	F9 v1.1 B1012	Failure (drone ship)	CCAFS LC-40
04 2015	F9 v1.1 B1015	Failure (drone ship)	CCAFS LC-40

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

Query Result with explanation:

- This query uses the SELECT statement to retrieve the Landing_Outcome and count the number of occurrences of each landing outcome from the SPACEXTBL table. It also assigns a rank to each landing outcome based on the count, in descending order. The query filters the results to include only those rows where the Date is between '2010-06-04' and '2017-03-20' and the Landing_Outcome is either 'Failure (drone ship)' or 'Success (ground pad)'. The results are grouped by Landing_Outcome.
- This query counts and ranks the landing outcomes of SpaceX missions between June 4, 2010, and March 20, 2017, specifically focusing on failures on drone ships and successes on ground pads. It provides a count of each outcome and ranks them based on their frequency.

```
%%sql
select Landing_Outcome, count(Landing_Outcome) as 'Count',
    row_number() over (order by count(Landing_Outcome) desc) as Rank
from SPACEXTBL
where Date Between '2010-06-04' and '2017-03-20' and Landing_Outcome in ('Failure (drone ship)', 'Success (ground pad)')
    group by Landing_Outcome
```

```
 * sqlite:///my_data1.db
Done.
```

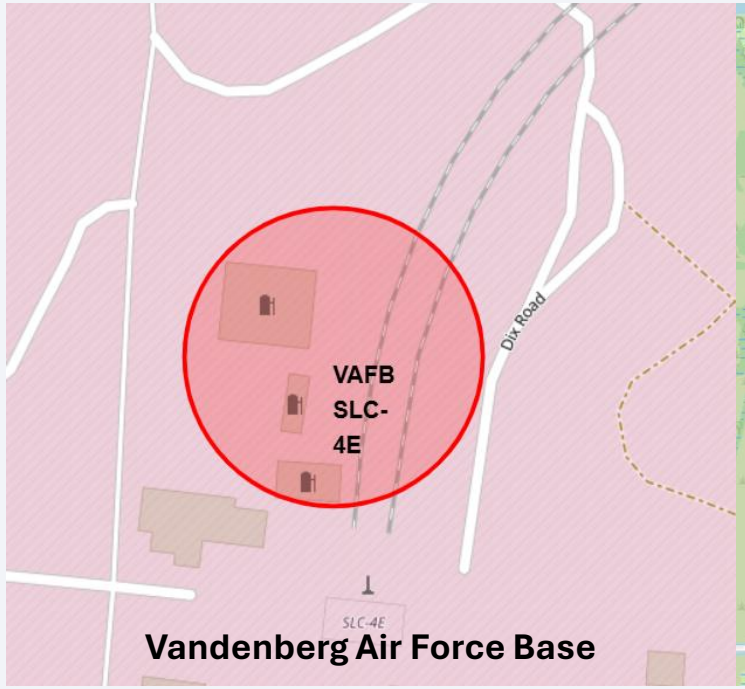
Landing_Outcome	Count	Rank
Failure (drone ship)	5	1
Success (ground pad)	3	2

Section 3

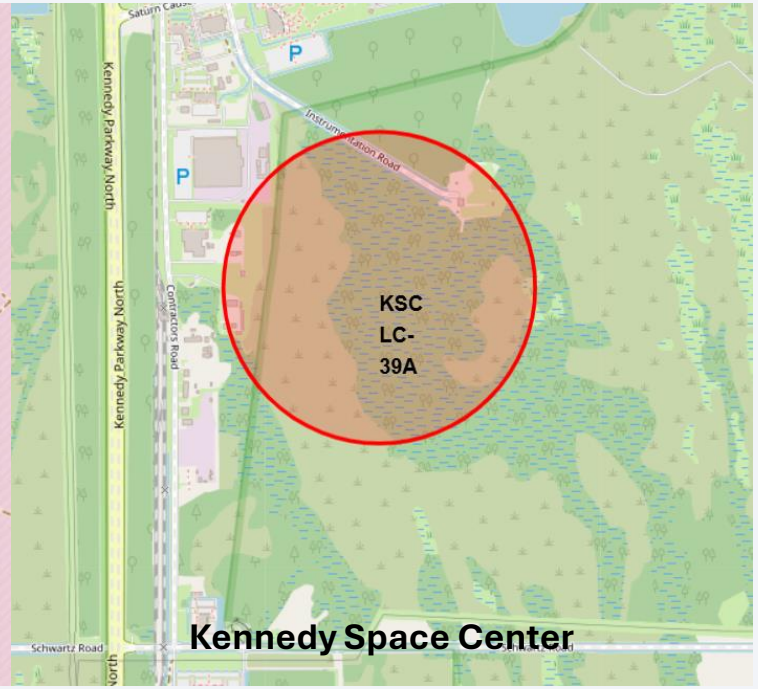
Launch Sites Proximities Analysis

Data Science Project
Presentation prepared by: Shiva Ram

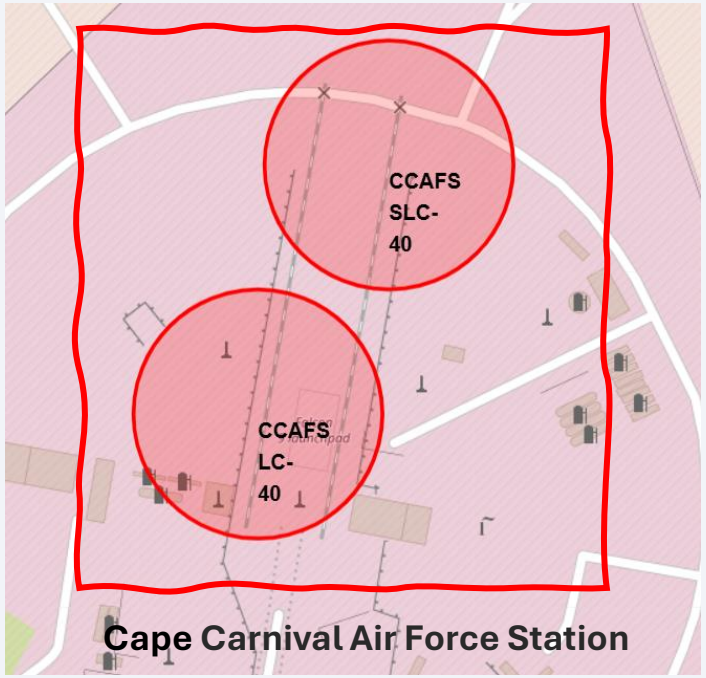
Launch sites



Vandenberg Air Force Base



Kennedy Space Center

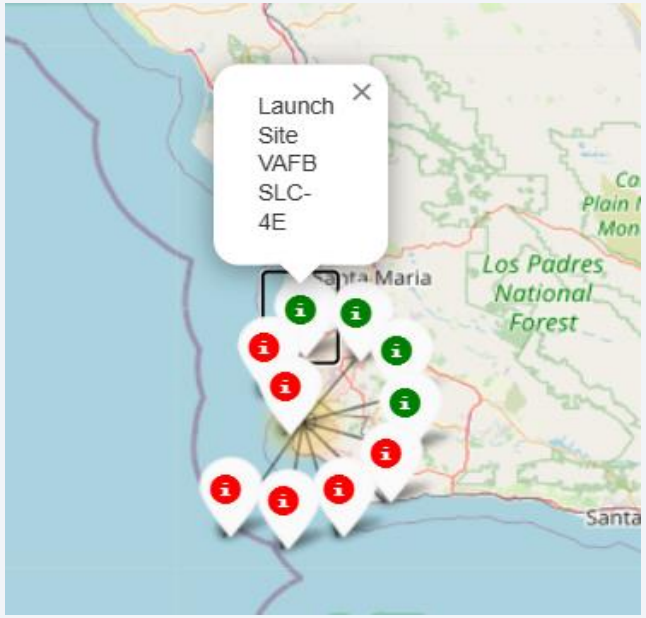


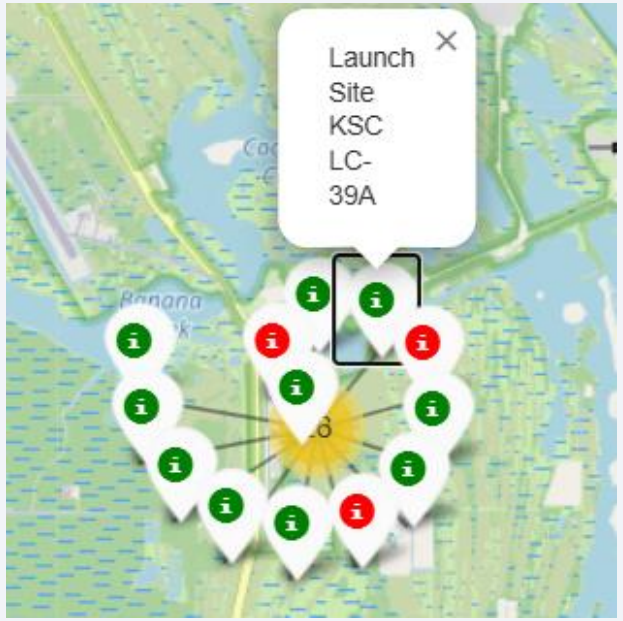
Cape Carnival Air Force Station

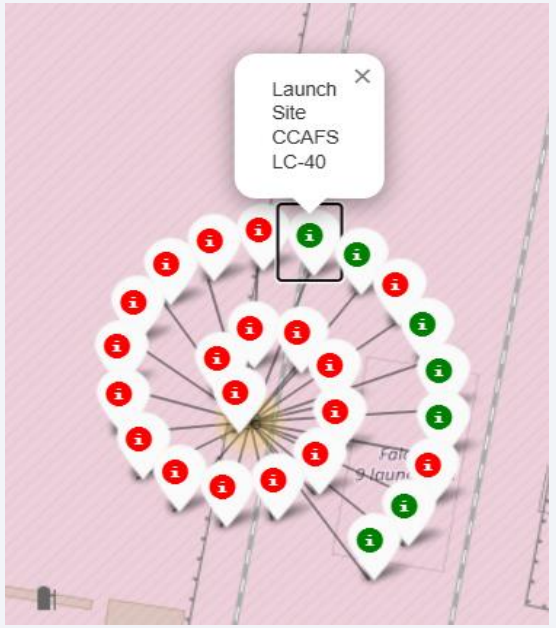
Key elements to note:

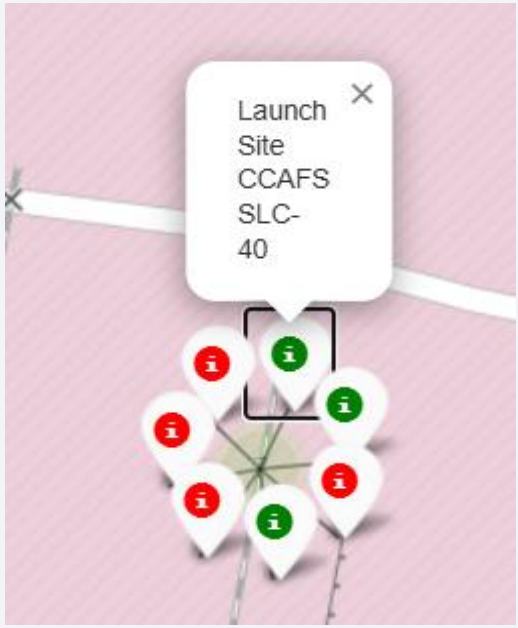
- Marked labels in circle represents as the launch sites used by SpaceX.
- CCAFS LC-40, CCAFS SLC-40 are in the same Cape Carnival Space Force Station.
- Vandenberg Air Force Base is located in west coast of United States of America in California.
- Kennedy Space Center and Cape Carnival Air Force Station are on the east coast of United States of America in Florida.

Launch sites - Launch outcome









Key elements to note:

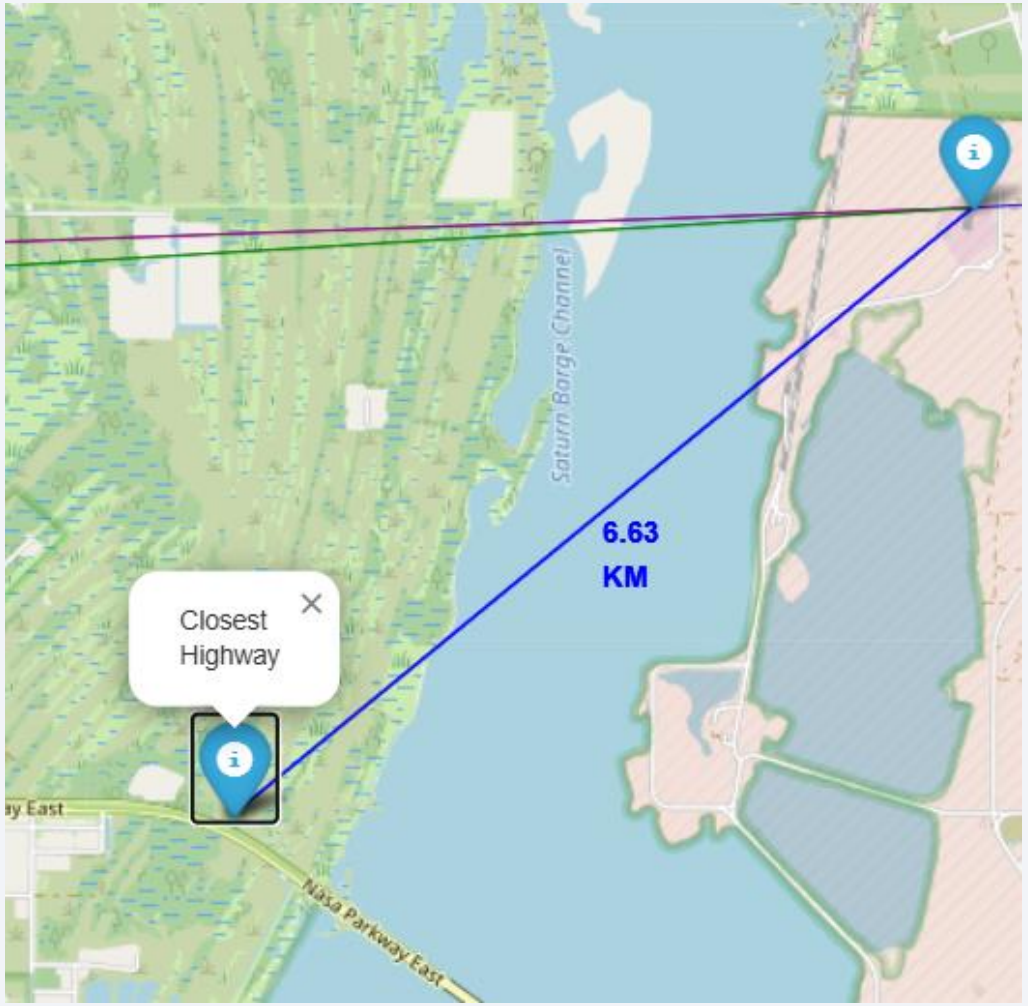
- From the map: “Green” icon represents successful launch and “Red” icon represents unsuccessful launch for SpaceX Falcon 9 rockets.
- Highest successful launch ratio is from KSC LC -39A site and the least is from CCAFS LC-40.

Closest proximity from launch site to Coastline

- This map shows the distance in kilometres between the launch complexes CCAFS LC-40 and CCAFS SLC-40 at Cape Canaveral Space Force Station and the coastline.
- Using the latitude and longitude coordinates for the launch sites (28.562302, -80.577356 for LC-40 and 28.563197, -80.576820 for SLC-40) and the coastline, the distance is calculated to be approximately **0.86** kilometres.
- This proximity is crucial for safety and trajectory planning during launches.

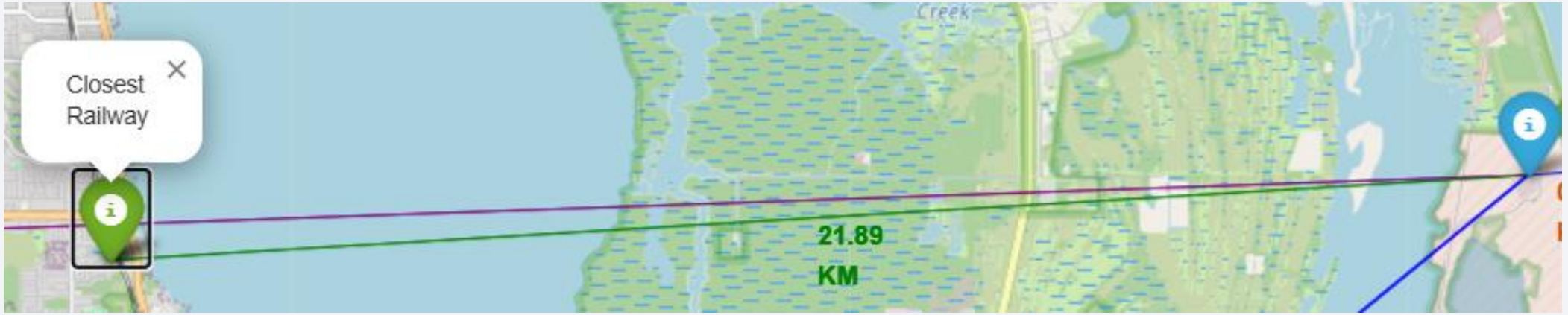
Closest proximity from launch site to highway

- This map shows the distance in kilometres between the launch complexes CCAFS LC-40 and CCAFS SLC-40 at Cape Canaveral Space Force Station and the nearest highway, NASA Parkway.
- Using the latitude and longitude coordinates for the launch sites (28.562302, -80.577356 for LC-40 and 28.563197, -80.576820 for SLC-40) and the nearest highway, the distance is calculated to be approximately **6.63** kilometres.
- This proximity is crucial for logistics, emergency access, and trajectory planning during launches.



Data Science Project
Presentation prepared by: Shiva Ram

Closest proximity from launch site to railway station



- This map shows the distance in kilometres between the launch complexes CCAFS LC-40 and CCAFS SLC-40 at Cape Canaveral Space Force Station and the closest railway station, Titusville.
- Using the latitude and longitude coordinates for the launch sites (28.562302, -80.577356 for LC-40 and 28.563197, -80.576820 for SLC-40) and the nearest railway station, the distance is calculated to be approximately **21.89** kilometres.
- This proximity is crucial for logistics, emergency access, and trajectory planning during launches.

Closest proximity from launch site to city

- This map shows the distance in kilometres between the launch complexes CCAFS LC-40 and CCAFS SLC-40 at Cape Canaveral Space Force Station and the closest city, Orlando.
- Using the latitude and longitude coordinates for the launch sites (28.562302, -80.577356 for LC-40 and 28.563197, -80.576820 for SLC-40) and the nearest railway station, the distance is calculated to be approximately **79.40** kilometres.
- This proximity is crucial for logistics, emergency access, and trajectory planning during launches.

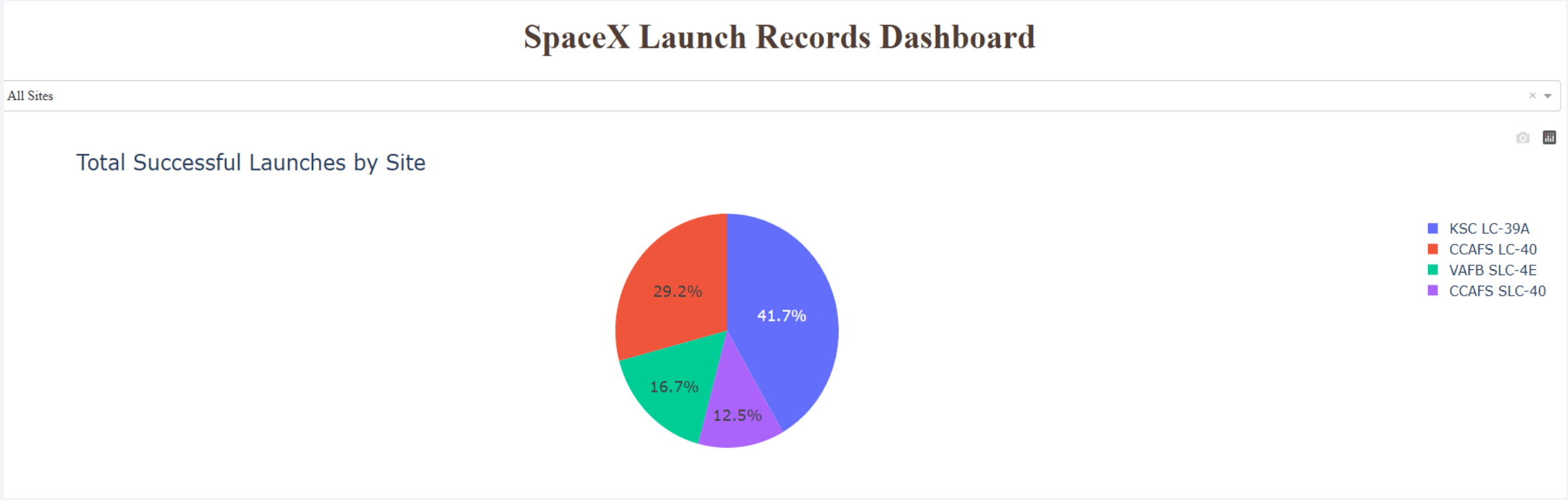
Section 4

Build a Dashboard with Plotly Dash

Data Science Project

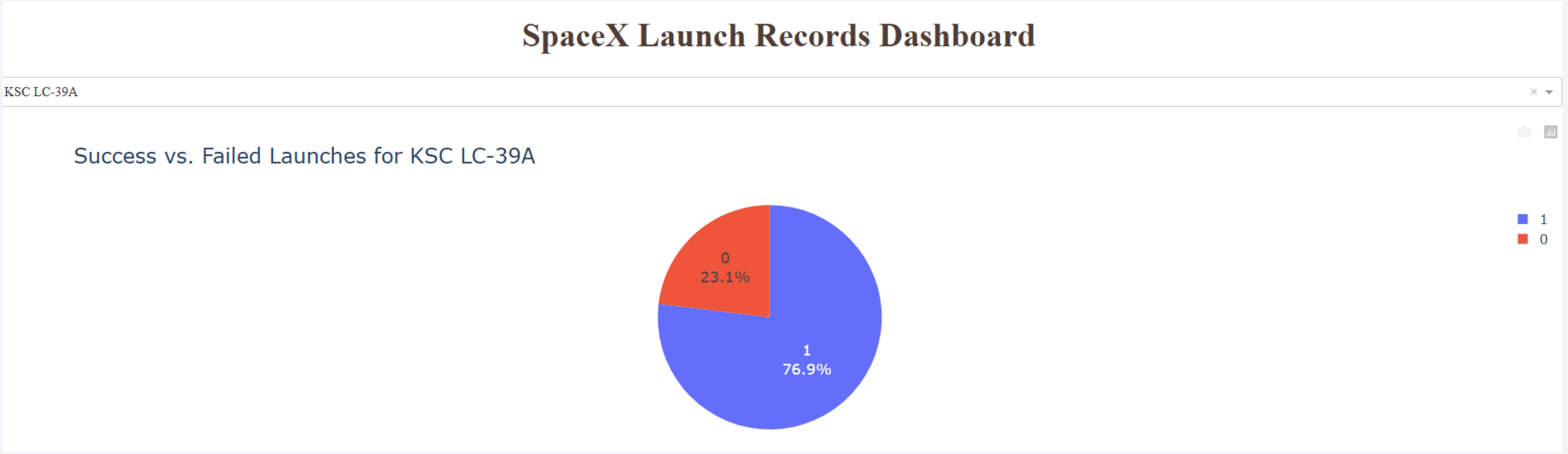
Presentation prepared by: Shiva Ram

SpaceX – All Sites launch success count



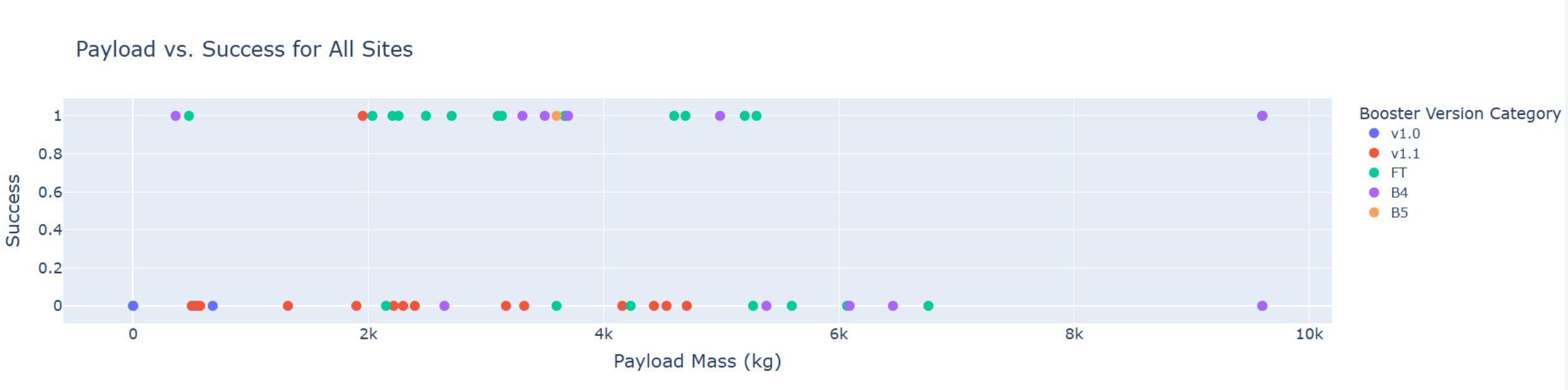
- SpaceX uses four launching sites as per the data used for this project.
- Pie chart shows all four sites and their launch successful rates.

Highest launch site success ratio



- Selecting KSC LC – 39A (Kenny Space Centre) from the drop down, pie chart shows 76.9% of the launches are successful.

Payload Mass vs Launch Outcome



- Scatter plot shows the success rate for all launch sites and Falcon 9 payload mass.
- Heaviest payload mass is 9,600kgs. Majority of the launches has payload mass between 2,000kgs and 5,000kgs.

Section 5

Predictive Analysis (Classification)

Data Science Project
Presentation prepared by: Shiva Ram

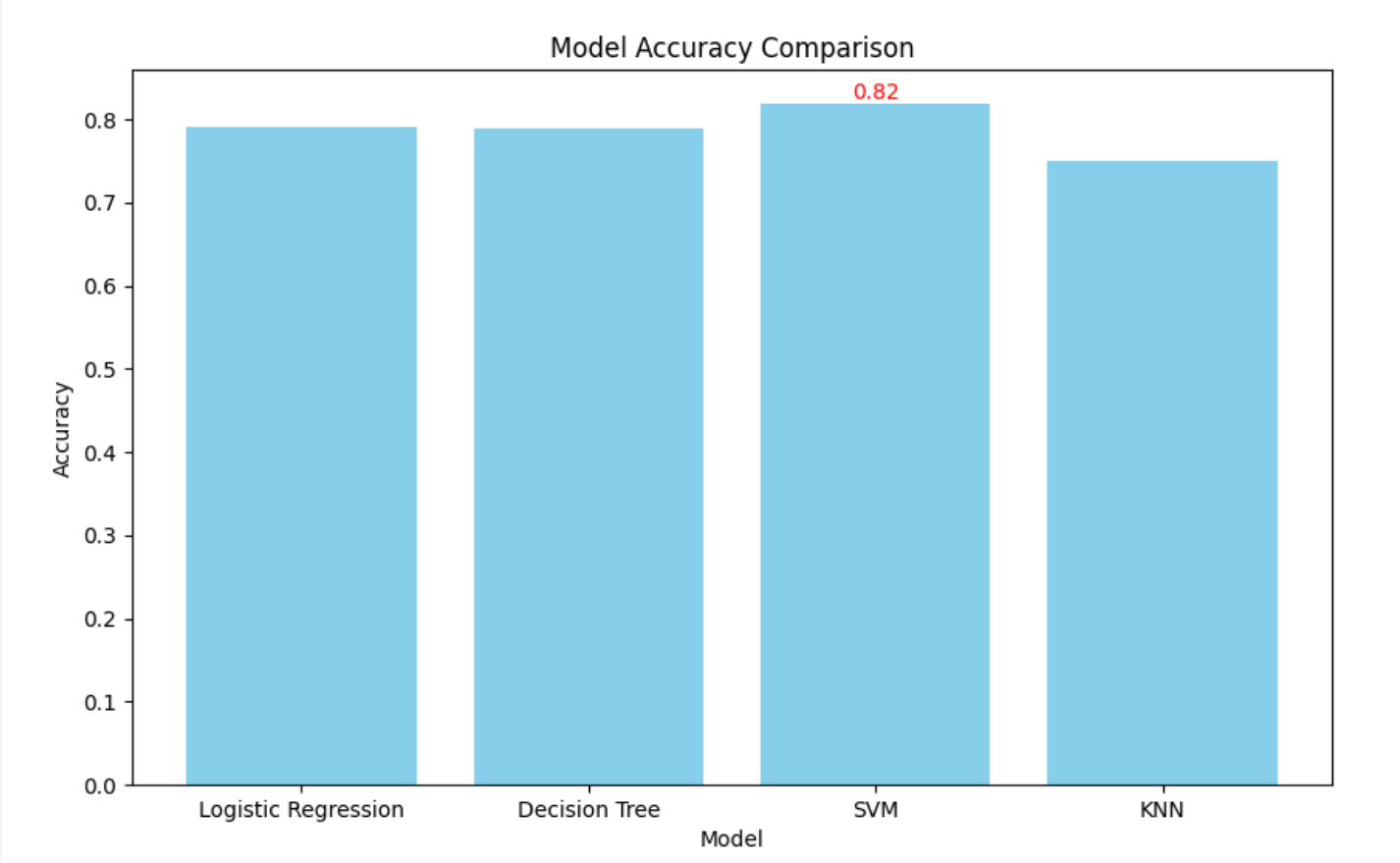
Classification Accuracy

From the SpaceX historical data, trained the following classification models:

- Logistic Regression
- Decision Tree
- Support Vector Machine (SVM)
- K-Nearest Neighbours (KNN)

After training the models, accuracy scores noted for plotting bar chart.

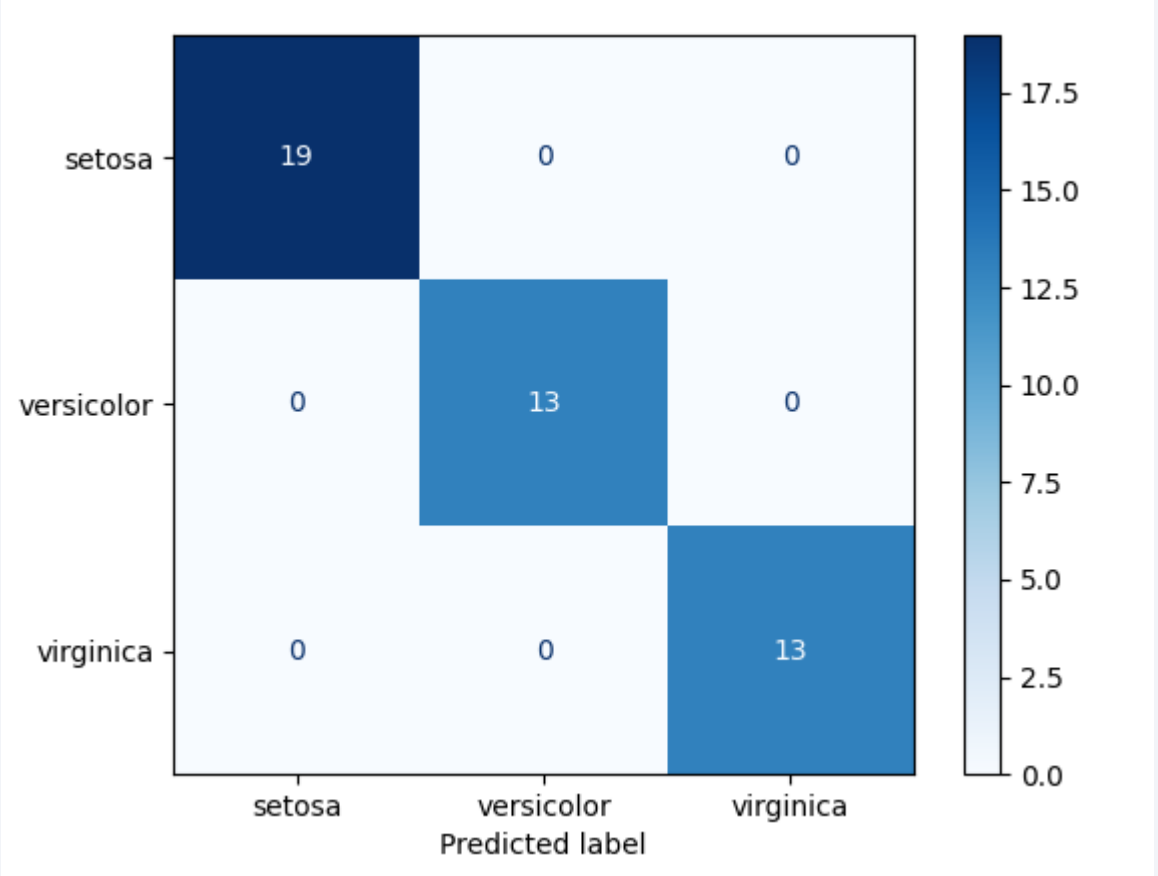
- Logistic Regression: 79.2%
- Decision Tree: 78.9%
- SVM: 82%
- KNN: 75%



Confusion Matrix

Explanation of the Confusion Matrix

- The confusion matrix is a table used to describe the performance of a classification model. It shows the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions.
- In this confusion matrix:
- True Positives (TP): The model correctly predicted the positive class.
- True Negatives (TN): The model correctly predicted the negative class.
- False Positives (FP): The model incorrectly predicted the positive class.
- False Negatives (FN): The model incorrectly predicted the negative class.
- The diagonal elements represent the number of correct predictions for each class, while the off-diagonal elements represent the number of incorrect predictions.



Conclusions

- The objective of this project was to determine the successful landing probability rates for Falcon 9 rockets across all SpaceX launch sites. Historical launch data was sourced from SpaceX via API calls, and additional data was obtained from Wikipedia.

- Data preparation involved querying structured data using SQLite and extracting web data using the BeautifulSoup Python library. Exploratory Data Analysis (EDA) was conducted using SQL, with various scatter plots and box charts revealing key factors contributing to Falcon 9's launch success. The analysis showed a significant improvement in launch success rates from 2013 onwards.

- Machine learning models, including regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbour, were built using the prepared data. SVM demonstrated the highest accuracy among the classification models, while hyperparameter tuning indicated that Decision Tree models performed best.

- Additionally, proximity calculations using Folium maps identified critical infrastructure near Cape Canaveral Space Force Station, essential for safety, trajectory planning, logistics, and emergency access.

- In conclusion, the Falcon 9 reusable rocket has proven to be a pivotal factor in SpaceX's growth and success, with significant improvements in launch success rates over the years.

Appendix

Python code for building Classification Accuracy

(Refer slide 54)

```
import pandas as pd
import matplotlib.pyplot as plt
# Step 1: Collect accuracy scores
data = {
'Model': ['Logistic Regression', 'Decision Tree', 'SVM', 'KNN'],
'Accuracy': [0.792, 0.789, 0.819,  0.75]}
# Step 2: Create a DataFrame
df = pd.DataFrame(data)
# Step 3: Plot the bar chart
plt.figure(figsize=(10, 6))
plt.bar(df['Model'], df['Accuracy'], color='skyblue')
plt.xlabel('Model')
plt.ylabel('Accuracy')
plt.title('Model Accuracy Comparison')
# Step 4: Highlight the highest accuracy
max_accuracy = df['Accuracy'].max()
max_model = df.loc[df['Accuracy'] == max_accuracy, 'Model'].values[0]
plt.text(df['Model'].tolist().index(max_model), max_accuracy, f'{max_accuracy:.2f}', ha='center', va='bottom', color='red')

plt.show()
```

Appendix

Python code for building Confusion Matrix

(Refer slide 55)

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
from sklearn.svm import SVC
# Load dataset
data = load_iris()
X = data.data
y = data.target
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# Train the best performing model (SVM in this case)
model = SVC()
model.fit(X_train, y_train)
# Predict the labels for the test set
y_pred = model.predict(X_test)
# Compute the confusion matrix
cm = confusion_matrix(y_test, y_pred)
# Display the confusion matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=data.target_names)
disp.plot(cmap=plt.cm.Blues)
plt.show()
```

Thank you!

Data Science Project
Presentation prepared by: Shiva Ram