Winning Space Race with Data Science

Dror Avigdor
4/10/2025
GitHub: https://github.com/dror-avigdor

Outline

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

Executive Summary

- Space Y - for Competing Space X
- Goals:
  - First stage costs predictions.
  - First stage reuse success prediction.

Executive Summary

Methodologies Summary:

- **Data Collection:**
  - Retrieved **SpaceX launch data** via **REST API** and **Wikipedia web scraping**.
  - Extracted key features: **launch site, payload, orbit, landing outcome**.
- Data Wrangling & Exploration:
  - Cleaned missing values and standardized data.
  - Analyzed **launch frequency, orbit distribution, and landing success rates**.
- Predictive Model Development:
  - Used **Logistic Regression, SVM, Decision Trees, and KNN**.
  - **Hyperparameter tuning with GridSearchCV** to optimize performance.
  - Evaluated models using **accuracy metrics and confusion matrices**.

Executive Summary

Results Summary:

- **Exploratory Data Analysis Findings:**
  - Most launches occurred at CCAFS SLC 40.
  - **LEO, GTO, and ISS** were the most common orbits.
  - **Landing success rate: 66.67%**.
- Predictive Analysis Outcomes:
  - **SVM performed best** in classifying landing success
  - **Key features influencing success**: **Payload mass, launch site, booster type**
  - **Future improvements:** More training data and feature engineering.

Introduction

Project background:

- The commercial space industry is
- SpaceX leads the market with achievements like ISS missions, Starlink, and manned flights.
- Cost advantage: Falcon 9 launches at $62M, competitors at $165M+—thanks to first-stage reusability
- The first stage does most of the rocket's work but is sometimes lost due to mission needs or failures.

Introduction

Insights to find:

- Can a machine learning model help forecast landing outcomes?
- How can we accurately predict if SpaceX’s first stage will successfully land?
- What key factors influence whether SpaceX sacrifices the first stage for mission-specific reasons?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology: using Space-X API and Wikipedia.
- Perform data wrangling: using Python Pandas and Numpy libraries.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models:
  - Splitting the data set to training and testing sub-data sets.
  - Applying transformation to the data set.
  - Performing training and testing using several models - Logistic Regression, SVM, Decision Tree and K-Nearest Neighbor and reviewing best results.

Data Collection

- Data collection made of two sources:
  - Space-X API ()
    - *https://github.com/dror-avigdor/Space-Y/blob/main/jupyter-labs-spacex-data-collection-api.ipynb*
  - Wikipedia - by html parsing (using BeautifulSoup library)
    - *https://github.com/dror-avigdor/Space-Y/blob/main/jupyter-labs-webscraping.ipynb*

Data Collection – SpaceX API

Key Phrases & Summary

- We retrieved **historical launch data** from SpaceX using their **public REST API**.
- The API provides details on **rockets, launchpads, payloads, and core landings**.
- Data was **parsed and stored** in a Pandas DataFrame for further analysis.
- **Key endpoints used:**
  - `/launches/past` → Retrieves launch history.
  - `/rockets/{id}` → Fetches rocket specifications.
  - `/launchpads/{id}` → Returns launch site details.
  - `/payloads/{id}` → Provides payload mass & orbit data.
  - `/cores/{id}` → Gets landing outcomes & reuse details.

Data collection flow:

Start → Request launch data (`/launches/past`) → Parse JSON response

→ Extract Rocket, Payload, Launchpad, Core IDs → Make API calls for details

→ Store & Clean Data → Create Final DataFrame → Export as CSV → Use for Analysis

Data Collection – SpaceX API

Example API Request:

```
import requests
# Fetch past SpaceX launches

response = requests.get("https://api.spacexdata.com/v4/launches/past")
data = response.json()  # Convert response to JSON
```

GitHub URL:
https://github.com/dror-avigdor/Space-Y/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection – Scraping

Key Phrases & Summary

- We extracted **Falcon 9 launch records** from **Wikipedia** using `BeautifulSoup`.
- The process involved:
  - **Sending an HTTP GET request** to retrieve the HTML page.
  - **Parsing the launch table** to extract key data.
  - **Cleaning and formatting** information for analysis.
- Data collected includes:
  - **Launch site, payload details, orbit type, landing success, customer, booster version**.

Data collection flow:

Start → Request Web Page → Parse HTML with BeautifulSoup → Find Launch Table → Extract Data → Clean and Format Information → Store in Pandas DataFrame → Save as CSV → Use for Analysis

GitHub URL: https://github.com/dror-avigdor/Space-Y/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling & Preparation

Key Phrases & Summary

- **Data Cleaning & Exploration:**
  - Loaded SpaceX launch dataset.
  - Checked for missing values (**LandingPad had ~29% missing data**).
  - Ensured correct data types for numerical and categorical fields.
- **Launch Site & Orbit Distribution:**
  - Most launches occur from **CCAFS SLC 40** (55 launches).
  - Common orbits include **GTO (27), ISS (21), and VLEO (14).**
- **Landing Outcome Classification:**
  - Created **landing outcome labels**: `0` (unsuccessful) & `1` (successf

https://github.com/dror-avigdor/Space-Y/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

Exploratory Data Analysis (EDA) with SQL

Summary of SQL Queries Performed

- **Data Extraction & Filtering:**
  - Retrieved **distinct launch sites** from SpaceX missions.
  - Displayed **5 records where the launch site starts with 'CCA'**.
- **Payload Mass & Booster Analysis:**
  - Calculated **total payload mass for NASA CRS missions**.
  - Computed **average payload mass for booster version F9 v1.1**.
  - Identified **boosters that succeeded in landing on drone ships** and carried payloads **between 4000-6000 kg.**

https://github.com/dror-avigdor/Space-Y/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Exploratory Data Analysis (EDA) with SQL

Summary of SQL Queries Performed

- Mission Success & Failure Insights:
  - Found the **earliest success landing date**.
  - Counted **successful vs. failed missions** in a single query.
  - Retrieved **booster version with the highest payload mass**.
- Temporal & Outcome-Based Analysis:
  - Listed **failure outcomes on drone ships** for launches in **2015**.
  - Ranked **landing outcome counts** between **2010-06-04 and 2017-03-20** in descending order.
  -

https://github.com/dror-avigdor/Space-Y/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

Key Phrases & Summary

- **Model Building:**
  - Standardized the dataset for consistent scaling.
  - Created the 'Class' label to represent successful landings.
  - Split data into **training (80%) and test (20%)** sets.
- **Evaluation & Improvement:**
  - Tested multiple classifiers: **Logistic Regression, SVM, Decision Trees, and KNN**.
  - Used **GridSearchCV** to optimize hyperparameters.
  - **Confusion matrices** helped assess classification accuracy and false positives.

Predictive Analysis (Classification)

Key Phrases & Summary

- **Finding the Best Model*:**
  - Compared accuracy scores for each model.
  - Selected the best-performing model based on **test data evaluation**.

* Results might be change from other colleagues notebooks, since I've gave up some kernels for performance issues in my station.

Prediction Flow:

Data Collection → Standardization → Train-Test Split → Model Selection → Hyperparameter Optimization → Evaluate Performance → Choose Best Model

Results & Insights:

Exploratory Data Analysis (EDA) Results

- **Launch success trends:**
  - Most SpaceX launches occur from **Cape Canaveral & Kennedy Space Center**.
  - Falcon 9's **first stage reusability** strongly impacts launch costs.
- **Orbit analysis:**
  - Most missions target **LEO, ISS, and GTO**, influencing payload delivery.
- **Landing outcome distribution:**
  - Successful landings are **more common on drone ships (ASDS)** than land pads (RTLS).

Results & Insights:

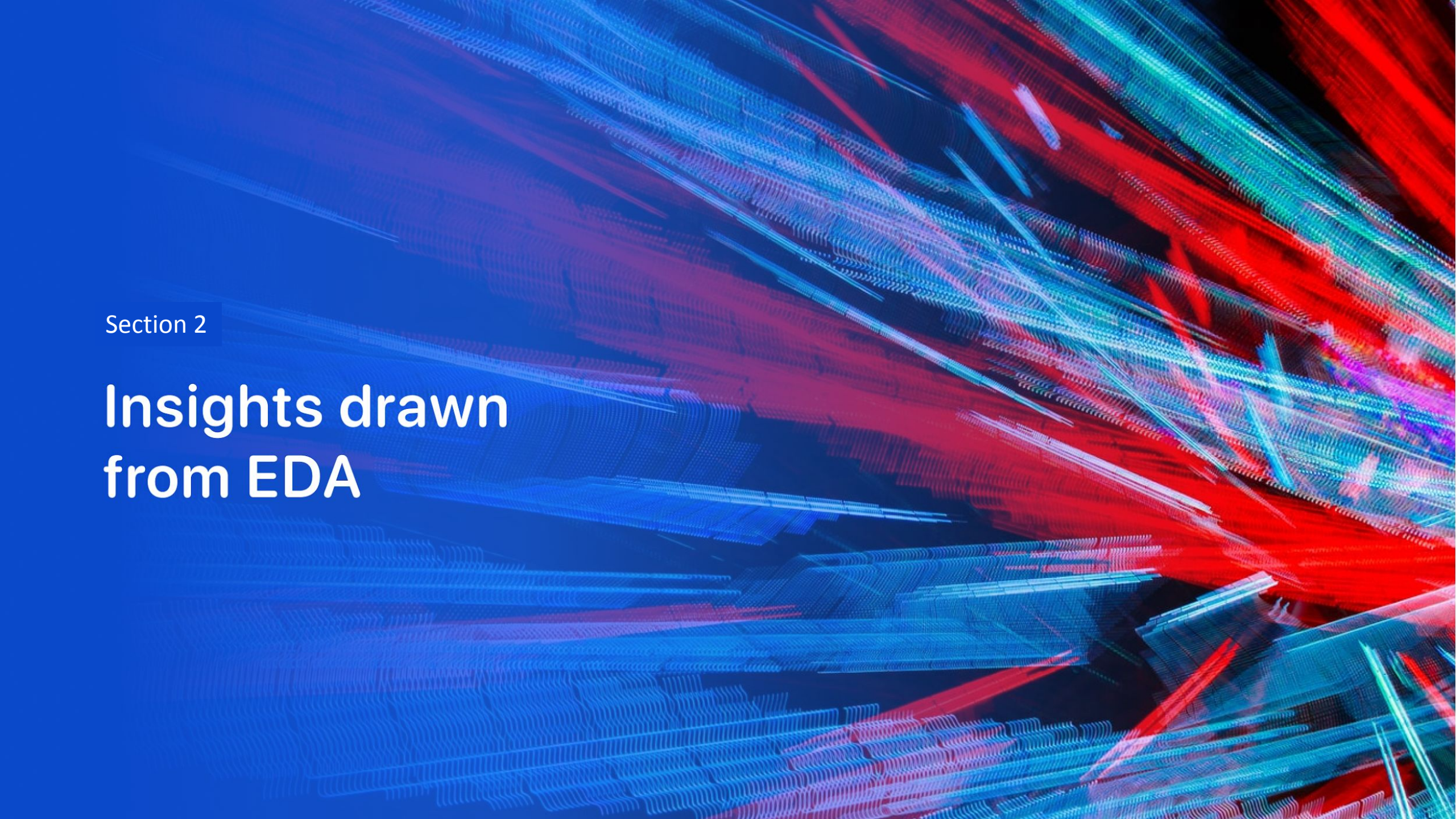
Predictive Analysis Results

- **Best performing model:**
  - **SVM had the highest test accuracy (94%)**, outperforming Logistic Regression (83.3%), Decision Trees (88%), and KNN (61%)*.
- **Key insights:**
  - **Payload mass, landing site, and booster type** strongly predict landing success.
  - **Hyperparameter tuning significantly improved model accuracy**.
- **Future improvements:**
  - More real-world mission data could refine predictions further.

*Results might be change from other colleagues notebooks, since I've gave up some kernels for performance issues in my station.

Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

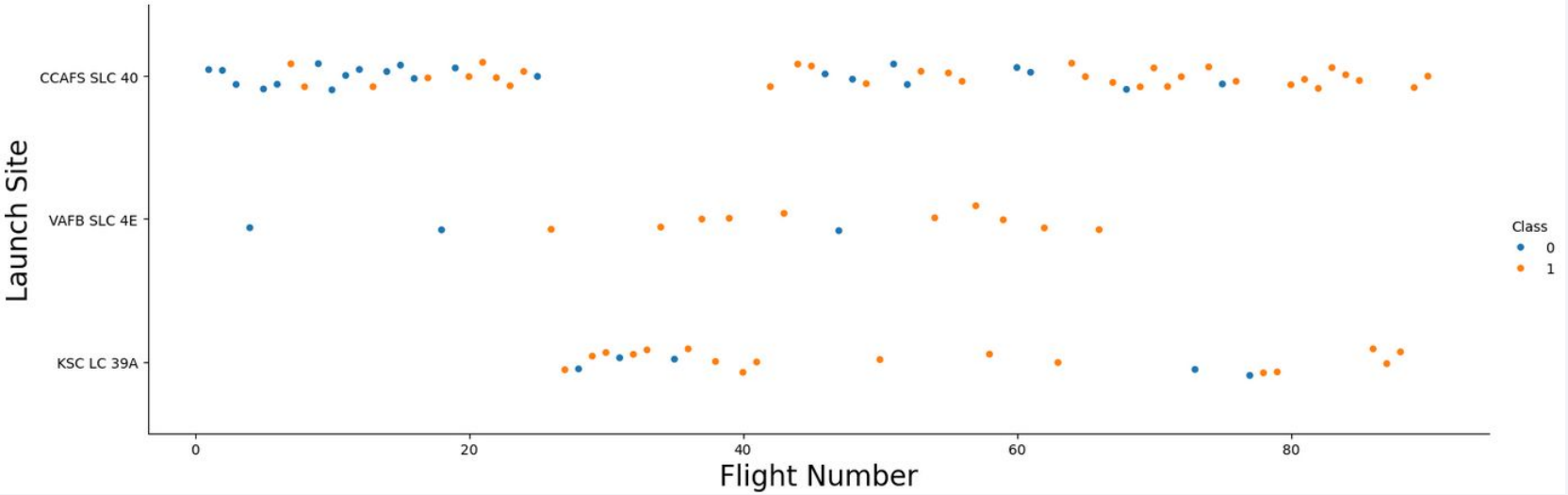


Figure: This scatter plot shows flight numbers and their launch sites, with colors indicating mission outcomes. It helps reveal patterns across different locations.

Payload vs. Launch Site

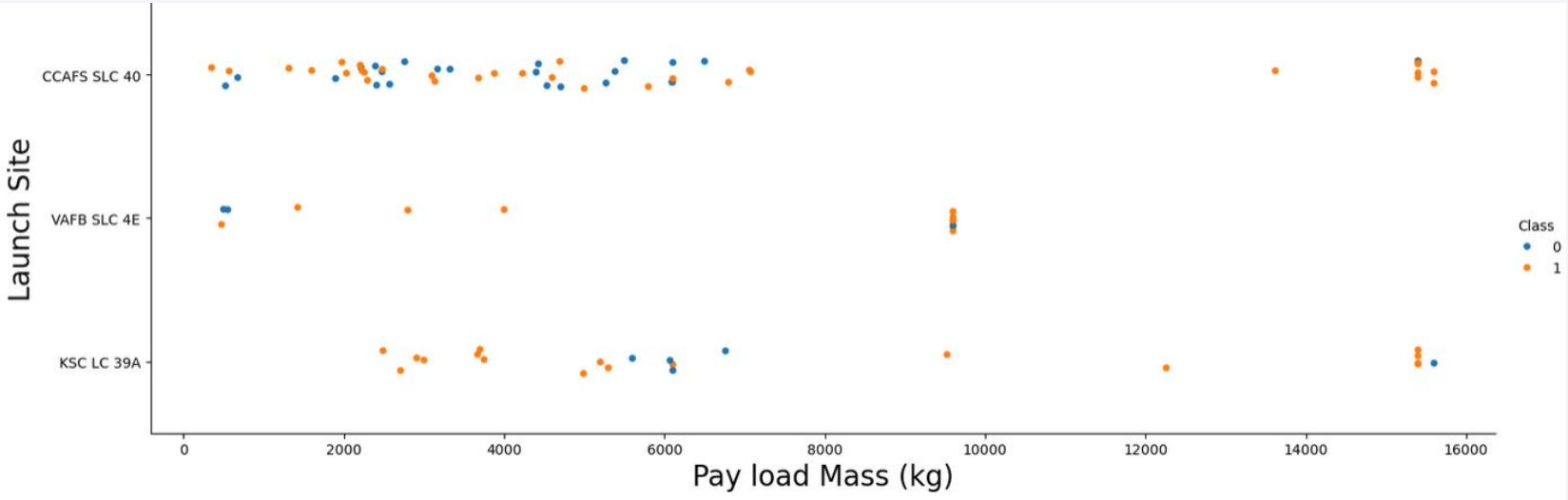


Figure: This scatter plot shows payload and their launch sites, with colors indicating mission outcomes. It helps reveal patterns across different locations.

Success Rate vs. Orbit Type

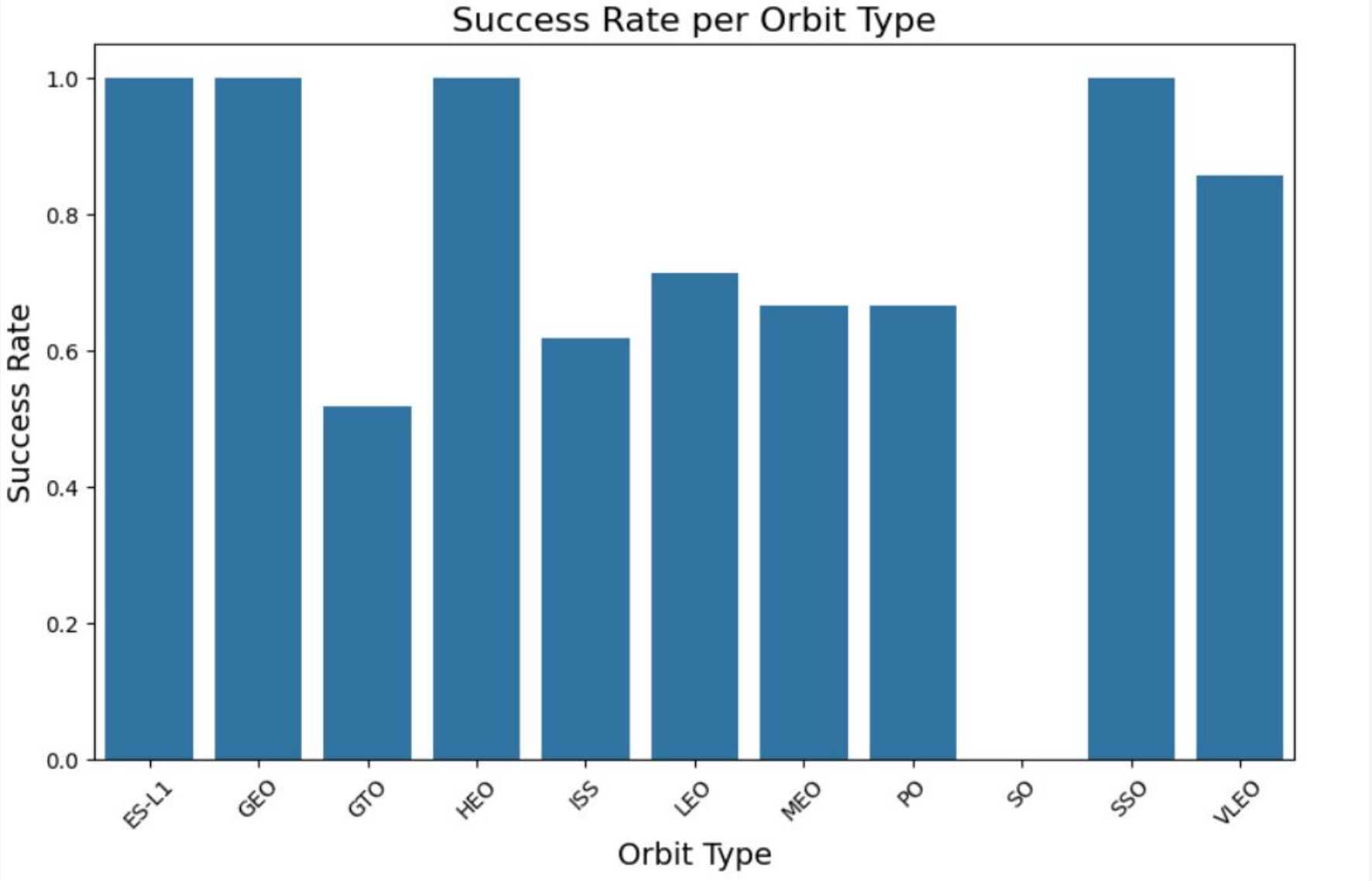


Figure: This scatter plot shows success rate per orbit type, indicating clear and absolute values of 100% success rate for several orbit types.

Flight Number vs. Orbit Type

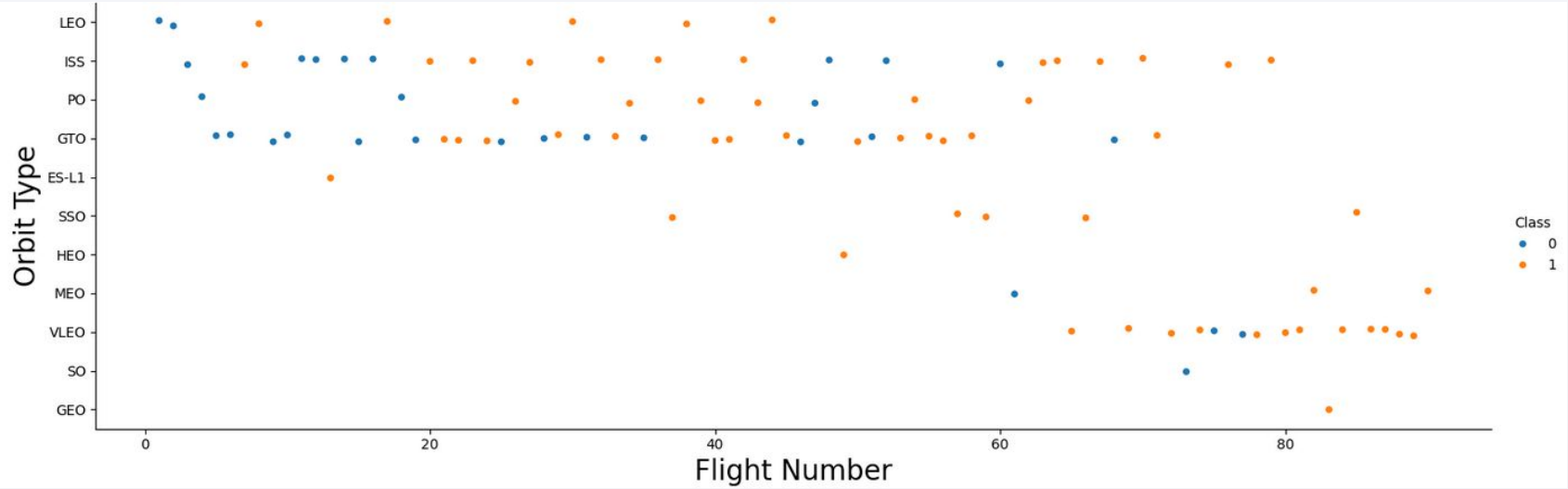


Figure: This scatter plot shows flight number and their orbit type, with colors indicating mission outcomes. It helps reveal patterns across sorts of orbit types.

Payload vs. Orbit Type

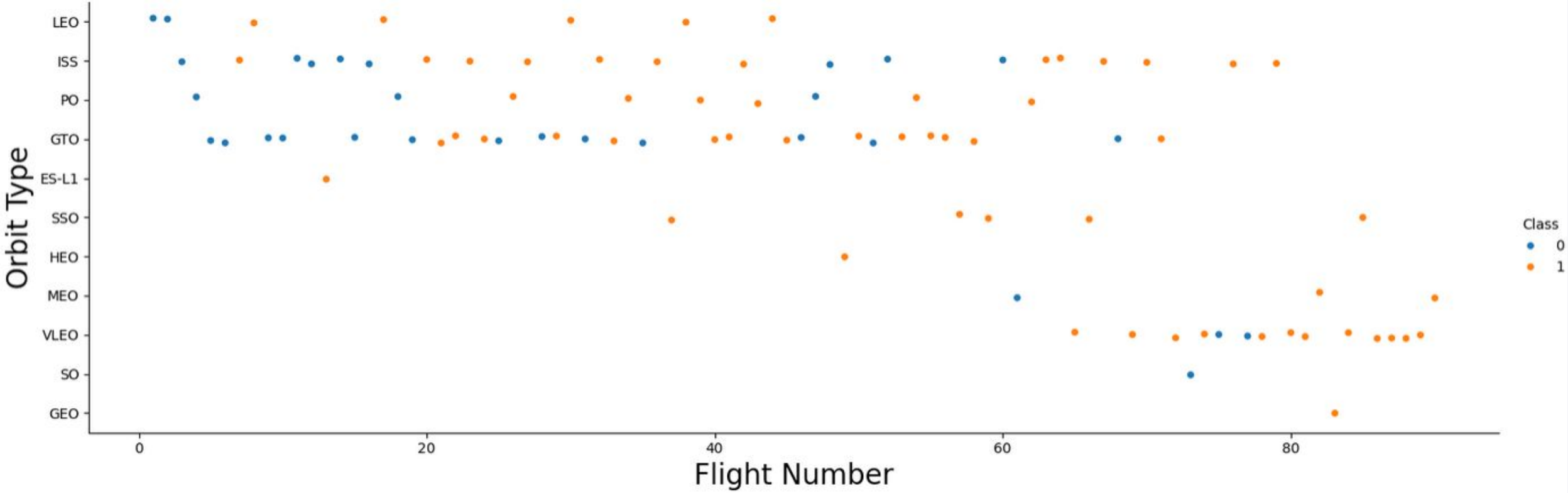


Figure: This scatter plot shows orbit type and vs. payload, with colors indicating mission outcomes. It helps reveal patterns across different locations.

Launch Success Yearly Trend

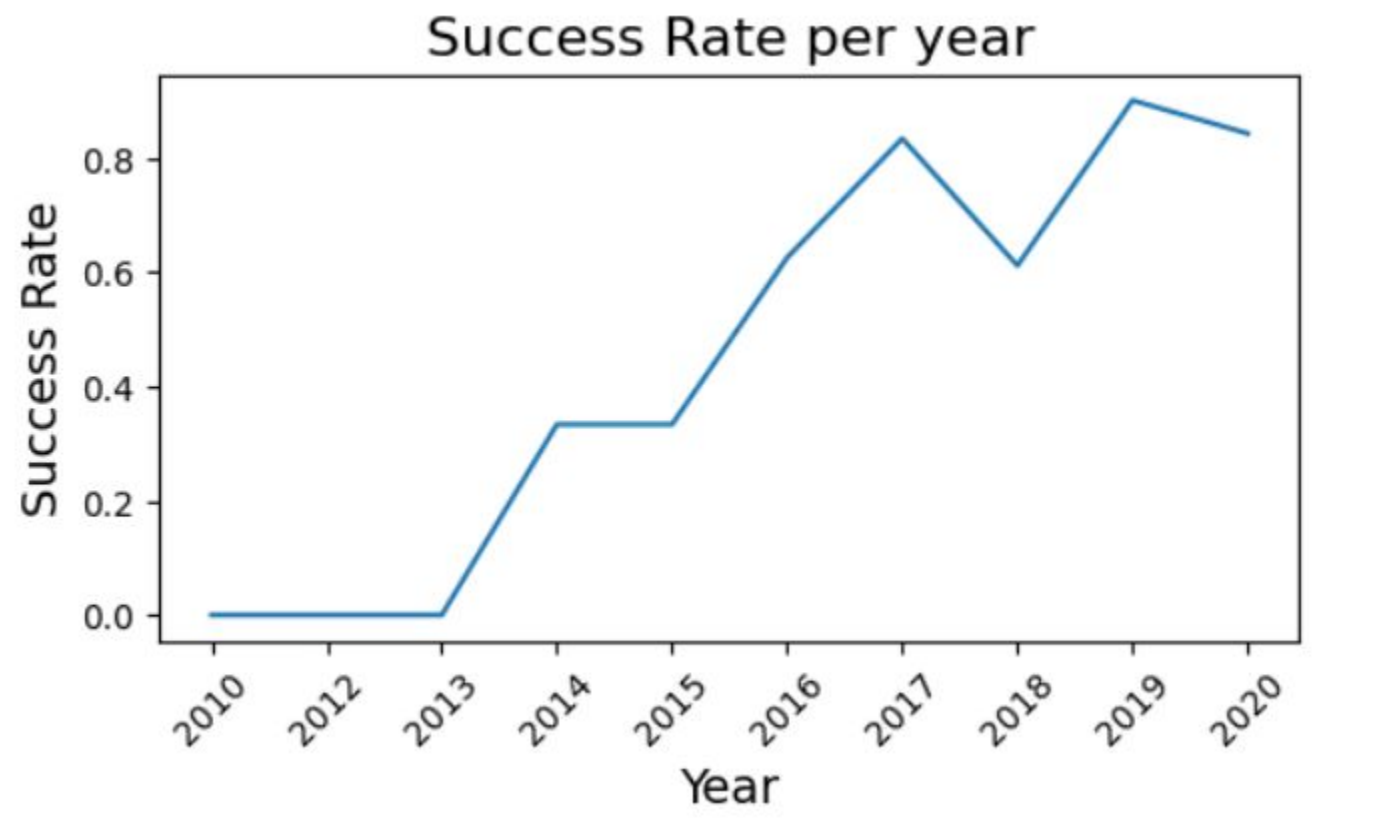


Figure: This line plot shows success rate, yearly, from 2010 to 2020, indicating an ongoing increase in success rate over the years.

All Launch Site Names

Launch_Sites

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

- Query:

  %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE

- Explanation: DISTINCT used to fetch unique values of a specified column.

Launch Site Names Begin with 'CCA'

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

- Query:

  *%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5*

- Explanation: Using ‘LIKE’ with ‘%’ for matching substring, LIMITing records to 5.

Total Payload Mass

SUM(PAYLOAD_MASS__KG_)
45596

- Query:

  %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer='NASA (CRS)'

- Explanation: Using ‘SUM’, specifying the column name.

Average Payload Mass by F9 v1.1

AVG(PAYLOAD_MASS__KG_)
2928.4

- Query:

  %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version='F9 v1.1'

- Explanation: Using AVG, specifying the column name.

First Successful Ground Landing Date

MIN(Date)
2015-12-22

- Query:

  %sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success%'

- Explanation: Using MIN for getting the minimal date, following the landing outcome condition.

Successful Drone Ship Landing with Payload between 4000 and 6000

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

- Query:

  %sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome='Success (drone ship)' AND PAYLOAD_MASS__KG_>4000 AND PAYLOAD_MASS__KG_<6000

- Explanation: DISTINCT - to get unique values of booster version

  Conditions: landing outcome explicitly verified as success with drone ship, AND payload limits - not inclusive.

Total Number of Successful and Failure Mission Outcomes

Two options provided:

First, using conditionals within the query, with COUNT(*):

```
1 %sql SELECT COUNT(CASE WHEN Mission_Outcome LIKE 'Success%' THEN 1 END) AS Successful_Missions, COUNT(CASE WHEN Mission_Outcome LIKE 'Failure%' THEN 1 END) AS Failed_Missions FROM SPACEXTABLE;
```

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

Successful_Missions	Failed_Missions
100	1

Second option, simply by listing all unique values of mission outcome (useful for our case where there are only 4 results):

```
1 %sql SELECT DISTINCT Mission_Outcome, COUNT(*) FROM SPACEXTABLE GROUP BY Mission_Outcome
```

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

Mission_Outcome	COUNT(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

Booster_Version	PAYLOAD_MASS__KG_
Collapse Output	
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

- Query:

  %sql SELECT DISTINCT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)

- Explanation: Using subquery for getting the maximum payload, and fetch the records in the outer query that matches that payload. Finally, using DISTINCT for not returning multiple records with same boosters.

2015 Launch Records

Month_name	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

- Query:
  %sql SELECT substr(Date, 6,2) as Month_name, Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Failure%' AND substr(Date, 0,5)='2015'
- Explanation: substr - for getting month and year from data, as a string..

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

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

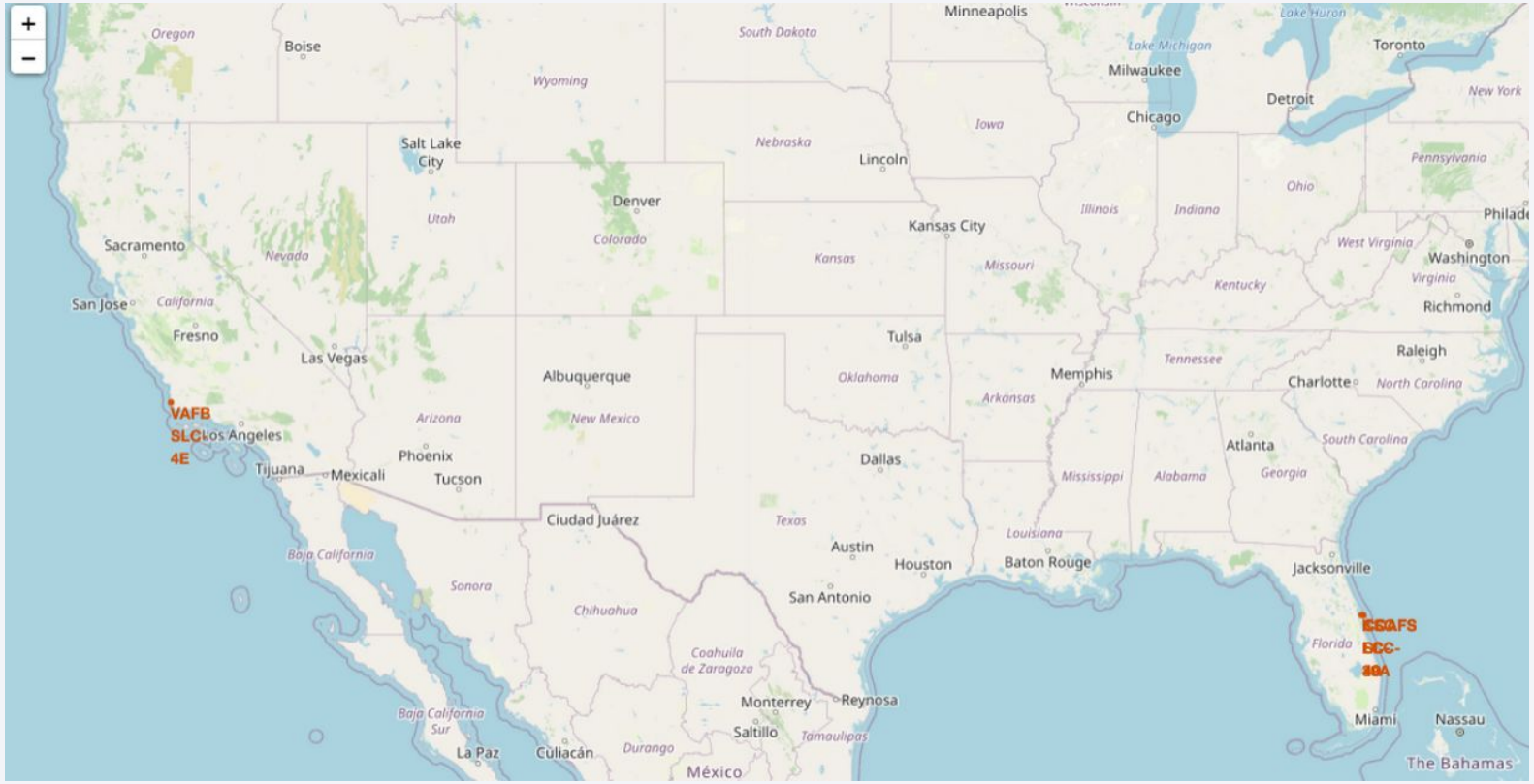
- Query:

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

- Explanation: Using COUNT(*), GROUPing the results by landing outcome, and ORDERing by the counting results.


Section 3
Launch Sites
Proximities Analysis

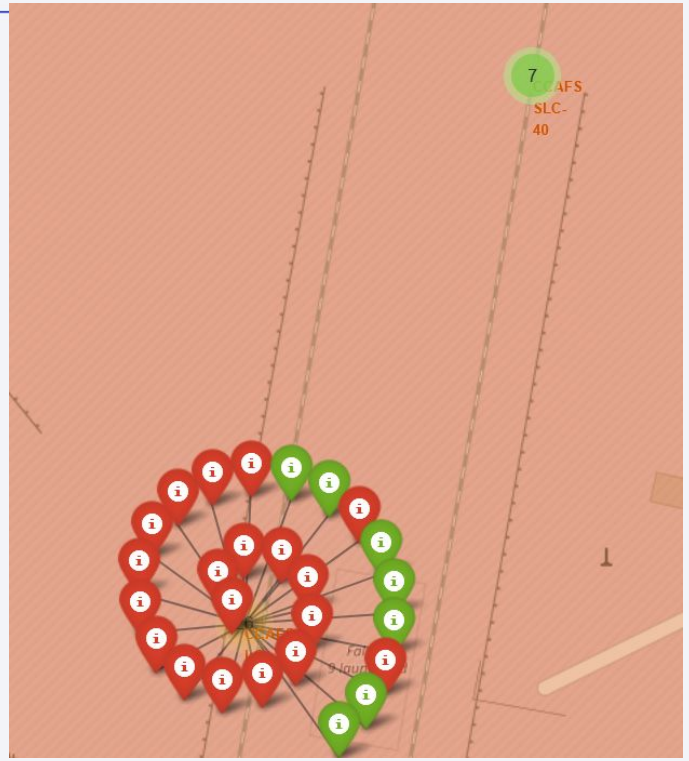
Lauch Sites Locations Map

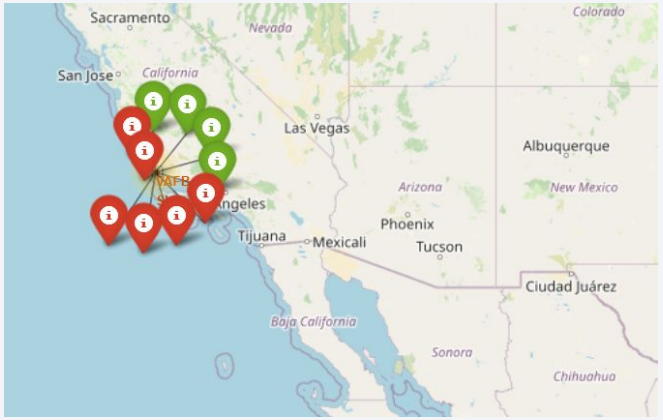


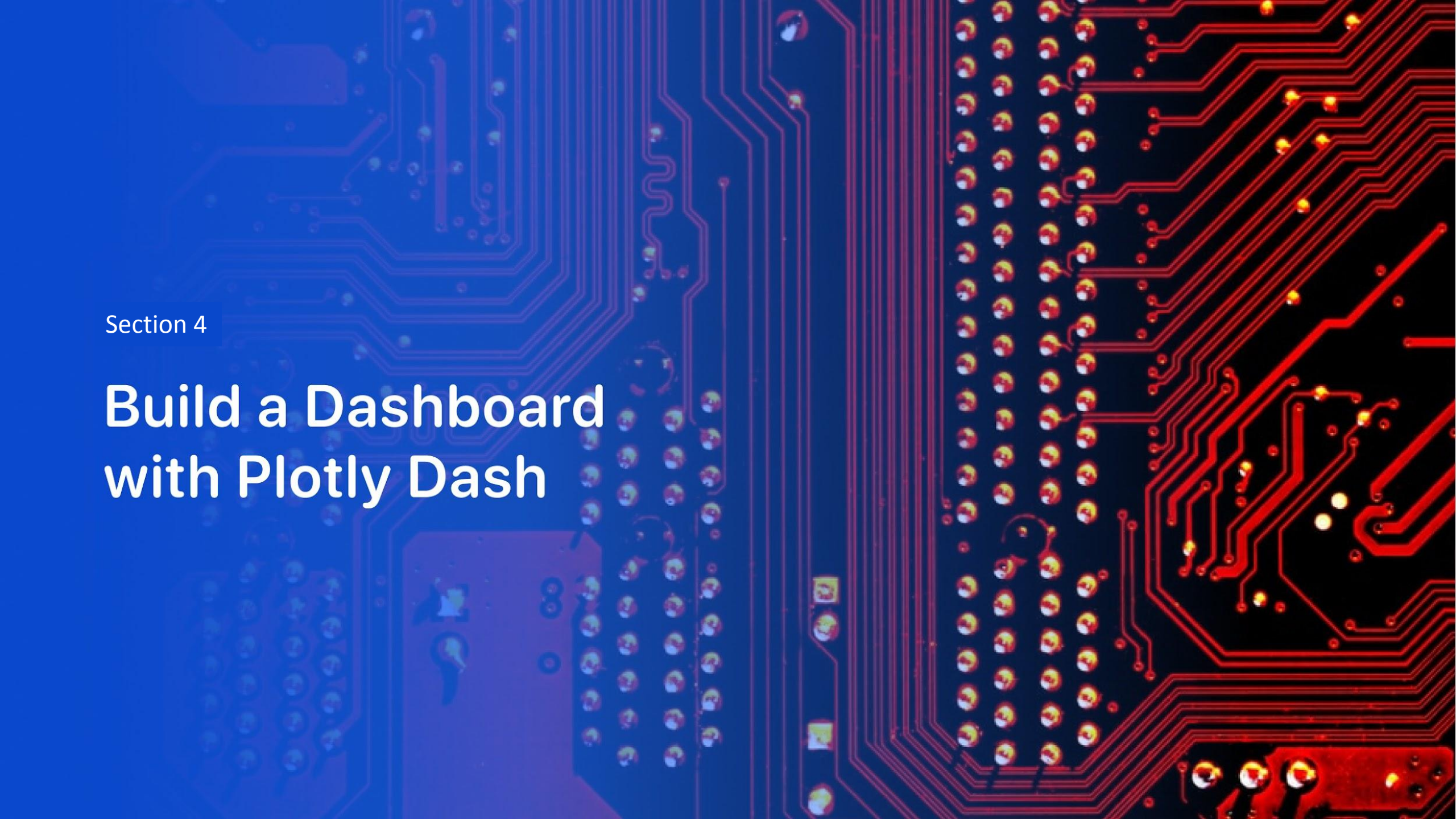
Insights:

- **Geographical Distribution** – The launch sites are strategically placed near the **east and west coasts of the United States**, ensuring efficient trajectories for different orbit types
- **Proximity to Water** – Most sites are located near oceans, allowing rockets to safely launch over water, reducing risks associated with debris and failed launches.
- **Latitude Influence** – The **closer to the equator**, the better for launching into **geostationary orbit** (like KSC LC-39A in Florida), as Earth's rotational speed helps conserve fuel.

Color Labeled Launch Outcomes






Section 4
Build a Dashboard with Plotly Dash

SuccessFul Launches per Launch Site

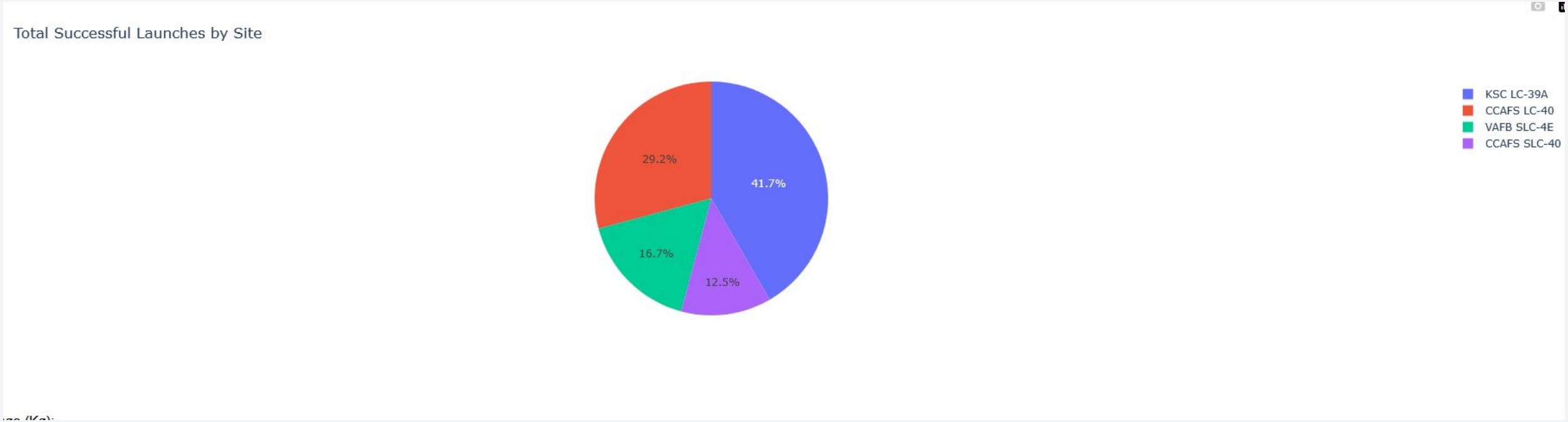


Figure: The pie chart reflects the success launches, per launch site. Clearly, it reflects that the most successful launch site is KSC LC-39 holding 41.7% of the success launches, and the lowest success rate is in CCAFS SLC-40 holding only 12.3% of the success launches.

Launches Success Rate in KSC LC-39A

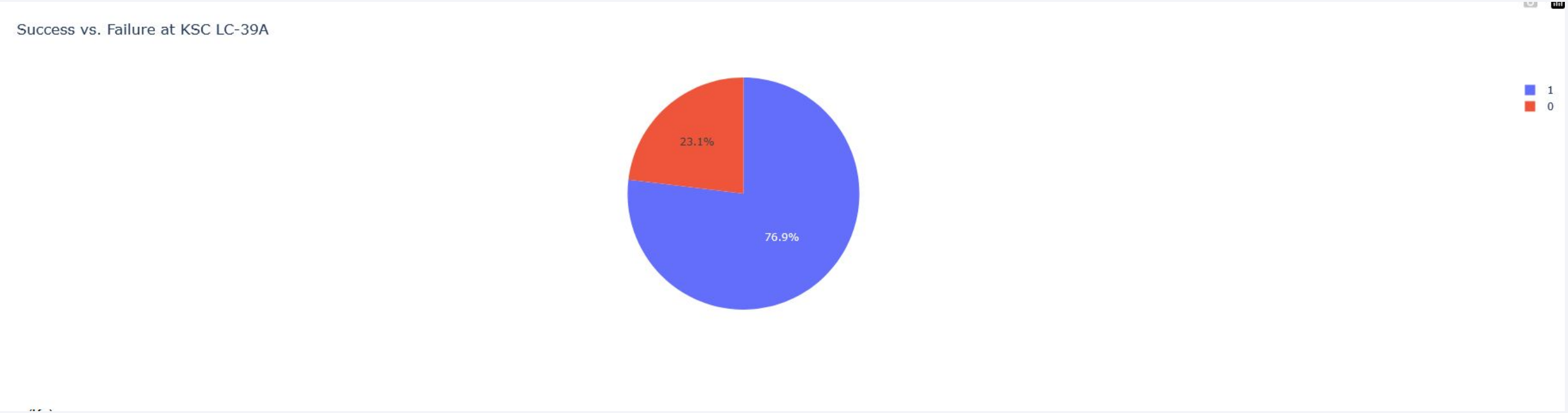
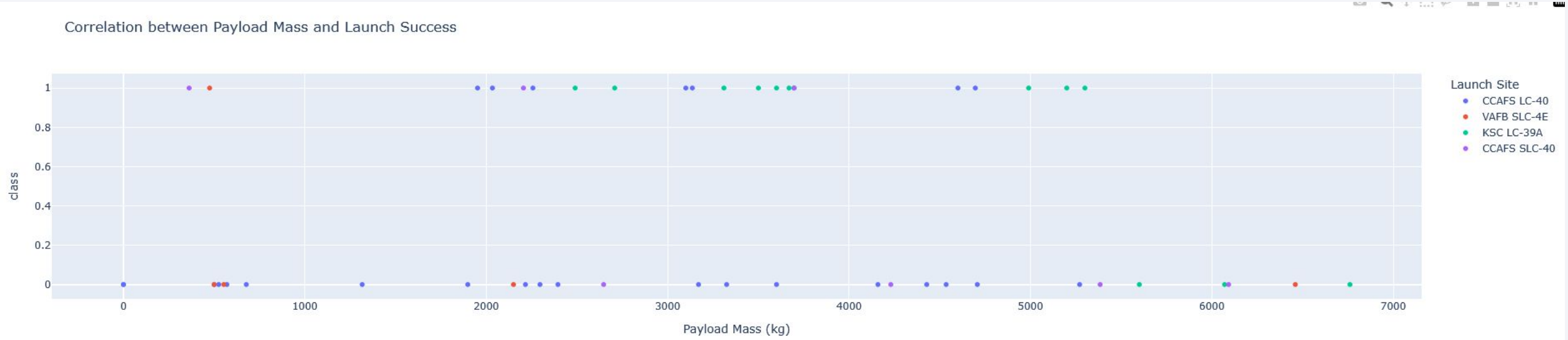


Figure: The most successful launch site, which is KSC LC-39A, has a success rate of 76.9%.

Payload vs. Launch Outcome

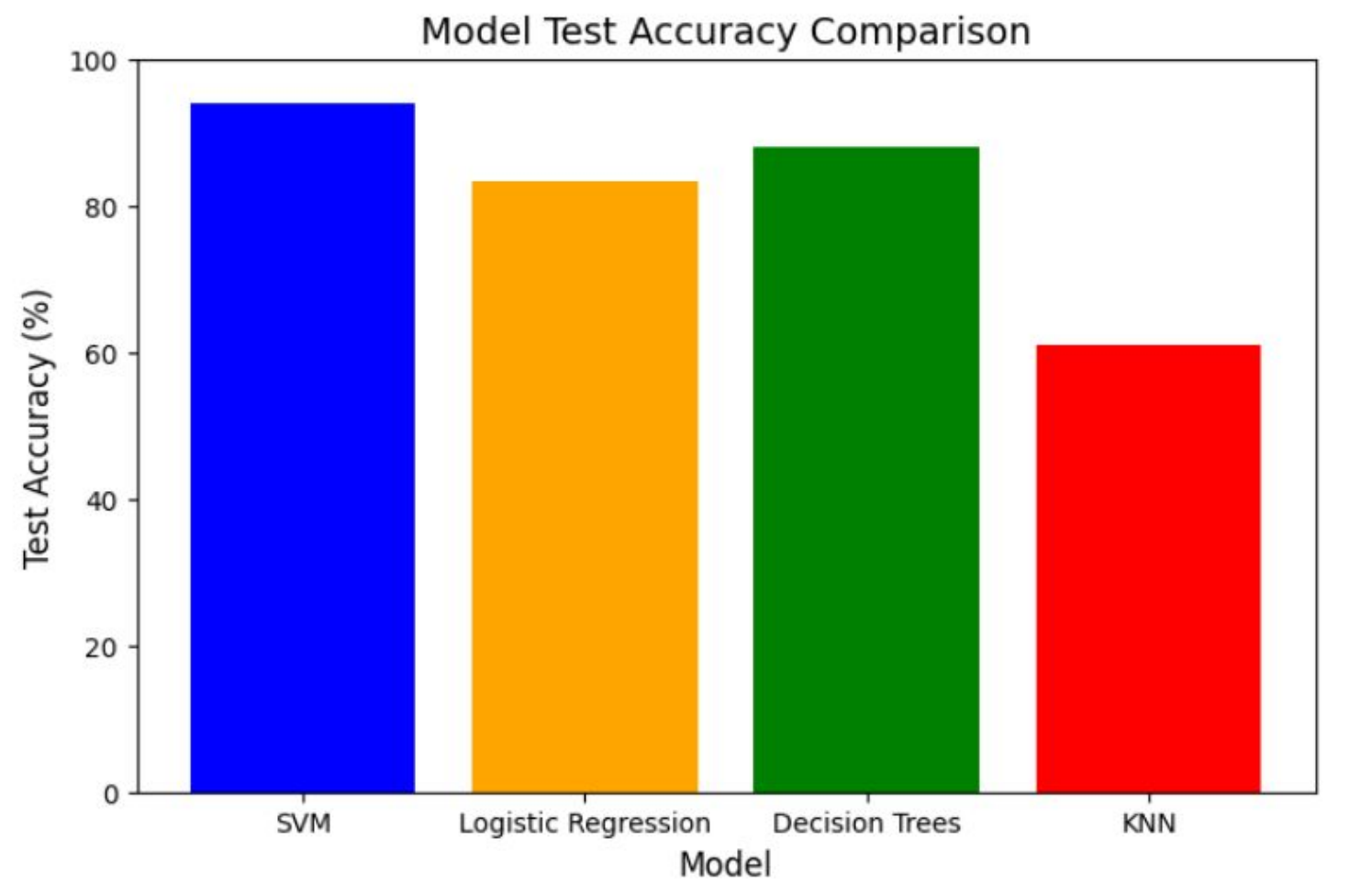


- **Figure:** Scatter plot shows Payload vs. Launch Outcome.
- **Insights from plot:**
  - All launches with payload above 60,000 kg have failed.
  - Most success launches made with a payload of 20,000-40,000 kg.
  - Launches with payloads out of the specified above boundaries, suffer from a low success rate.

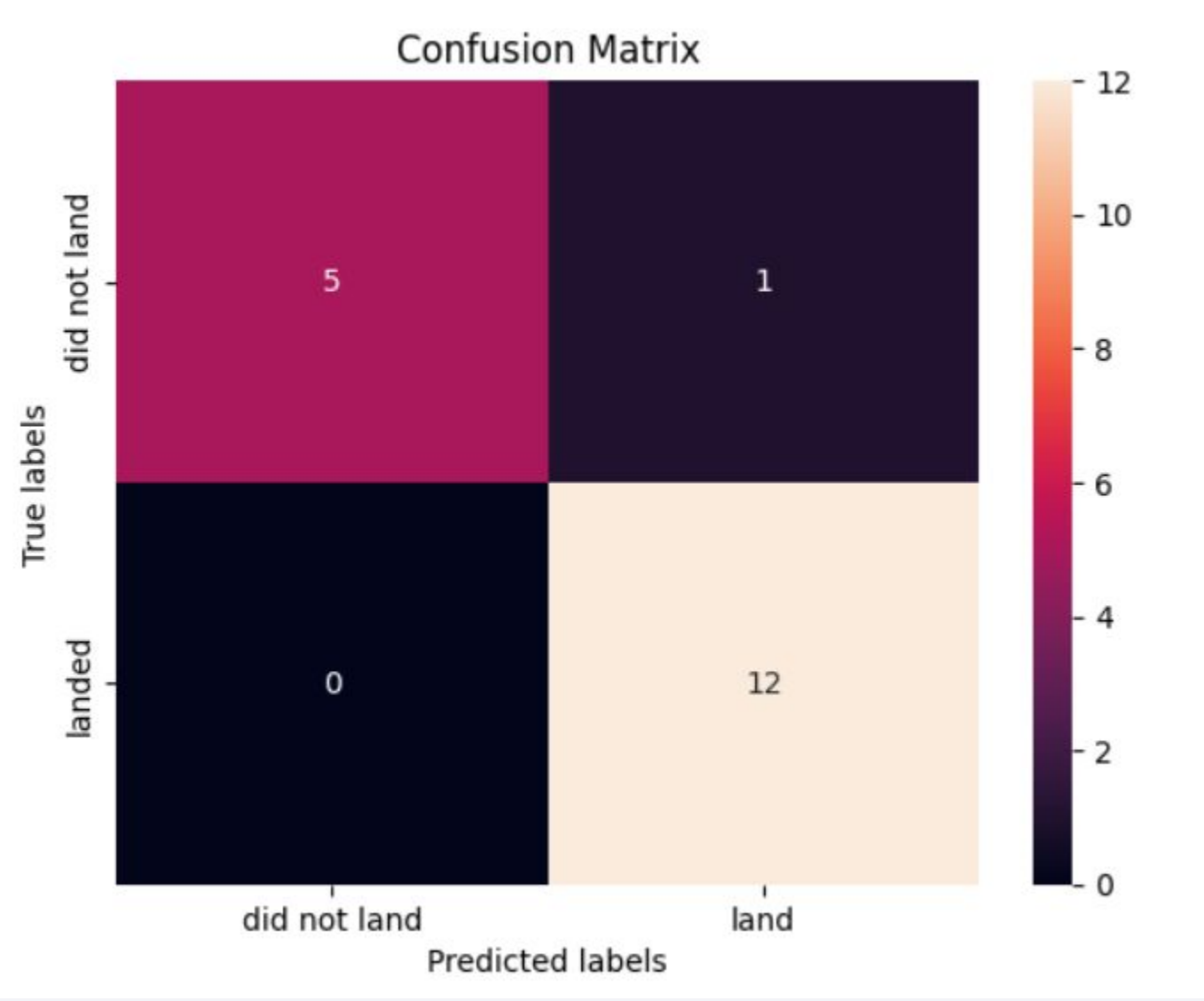
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



- Figure: Confusion matrix of SVM model prediction. The confusion matrix indicates good prediction rate, of 12 of 12 predicted as successfully landed, and 5 of 6 of failure landings.

Conclusions

Predictive Analysis Results

- **Best performing model:**
  - **SVM had the highest test accuracy (94%)**, outperforming Logistic Regression (83.3%), Decision Trees (88%), and KNN (61%)*.
- **Key insights:**
  - **Payload mass, landing site, and booster type** strongly predict landing success.
  - **Hyperparameter tuning significantly improved model accuracy**.

*Results might be change from other colleagues notebooks, since I've gave up some kernels for performance issues in my station.

Appendix

Project assets:

- Project Gitbub url: https://github.com/dror-avigdor/Space-Y
- Data Collection Space-X API notebook: https://github.com/dror-avigdor/Space-Y/blob/main/jupyter-labs-spacex-data-collection-api.ipynb
- Data Collection Web Scraping: https://github.com/dror-avigdor/Space-Y/blob/main/jupyter-labs-webscraping.ipynb
- EDA Visualization notebook: https://github.com/dror-avigdor/Space-Y/blob/main/edadataviz.ipynb
- EDA SQL notebook: https://github.com/dror-avigdor/Space-Y/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb
- Visual Analytics with Folium notebook: https://github.com/dror-avigdor/Space-Y/blob/main/lab_jupyter_launch_site_location.ipynb
- Data Wrangling notebook: https://github.com/dror-avigdor/Space-Y/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

Appendix

Project assets:

- Prediction Lab notebook: https://github.com/dror-avigdor/Space-Y/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Thank you!