IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rea Burla
9/6/2024

Outline

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

Executive Summary

1. Methodologies Summary:
2. Data Collection
3. Data Processing
4. Exploratory Data Analysis (EDA)
5. Interactive Visual Analytics
6. Predictive Analysis Using Classification Models

- Summary of all results

Introduction

- **Project Background and Context**

The commercial space age is here, with companies like Virgin Galactic, Rocket Lab, Blue Origin, and SpaceX leading the charge in making space travel more affordable and accessible. SpaceX, in particular, has achieved significant milestones, including sending spacecraft to the International Space Station and launching the Starlink satellite internet constellation. One key factor in SpaceX's success is the reusability of their rockets' first stages, which significantly reduces launch costs.

- **Problems to Solve**

The goal is to determine the cost of each SpaceX Falcon 9 launch and predict whether the first stage will be reused. This involves gathering and analyzing launch data from SpaceX using APIs and web scraping techniques. The project aims to create dashboards for visualizing this data and to train machine learning models to predict the likelihood of successful landings and reusability of the first stage based on various mission parameters.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API, Web Scraping
- Perform data wrangling
  - Handling Missing Values, One-Hot Encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Feature Selection, Hyperparameter Tuning, Train-Test Split, Metrics, Model Selection

Data Collection

The data collection process for this project involved the following steps:

1. API Data Collection:

- SpaceX REST API: Utilized the SpaceX REST API to gather detailed information about past launches.

- Endpoint Used: The specific endpoint used was api.spacexdata.com/v4/launches/past.

- Data Retrieved: Information included rocket details, payload specifications, launch and landing outcomes, etc.

- HTTP GET Requests: Employed the Python *requests* library to perform GET requests to the API and obtain the data in JSON format.

- JSON to DataFrame: Converted the JSON data to a Pandas DataFrame using *json_normalize* for easier manipulation and analysis.

2. Web Scraping:

- Wikipedia: Web scraped Wikipedia pages related to Falcon 9 and Falcon Heavy launches to complement the data collected from the API.

- HTML Parsing: Used the *BeautifulSoup* library to parse HTML tables and extract relevant data.

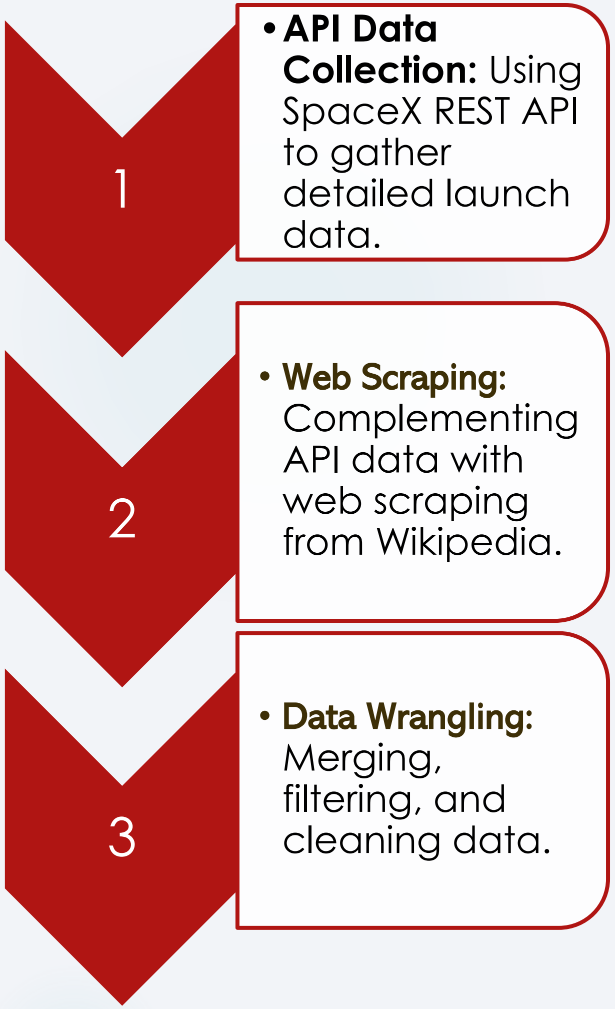
- Data Cleaning: Parsed and cleaned the HTML table data, converting it into a Pandas DataFrame.

3. Data Wrangling:

- Merging Data: Combined data from the API and web scraping into a unified dataset.

- Filtering: Filtered out irrelevant data such as Falcon 1 launches to focus only on Falcon 9.

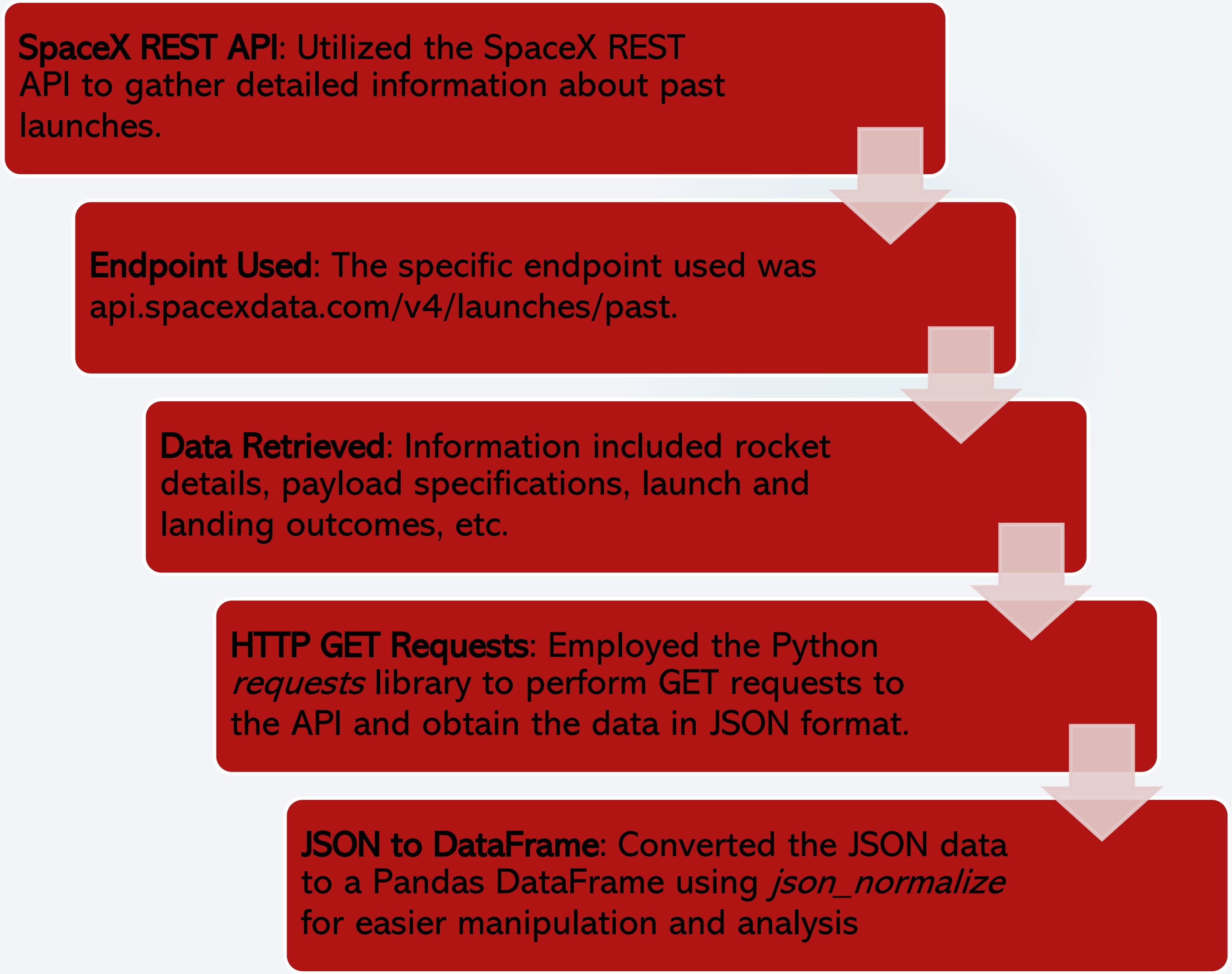
- Handling Null Values: Addressed missing values in key columns by calculating and filling with the mean value.



- **API Data Collection:** Using SpaceX REST API to gather detailed launch data.
- **Web Scraping:** Complementing API data with web scraping from Wikipedia.
- **Data Wrangling:** Merging, filtering, and cleaning data.

Data Collection – SpaceX API

- https://github.com/reab5555/IBM_Projects_Demo/blob/3a3c824d9e9c8be673852ed04208150954efe348/jupyter-labs-spacex-data-collection-api.ipynb



SpaceX REST API: Utilized the SpaceX REST API to gather detailed information about past launches.

Endpoint Used: The specific endpoint used was api.spacexdata.com/v4/launches/past.

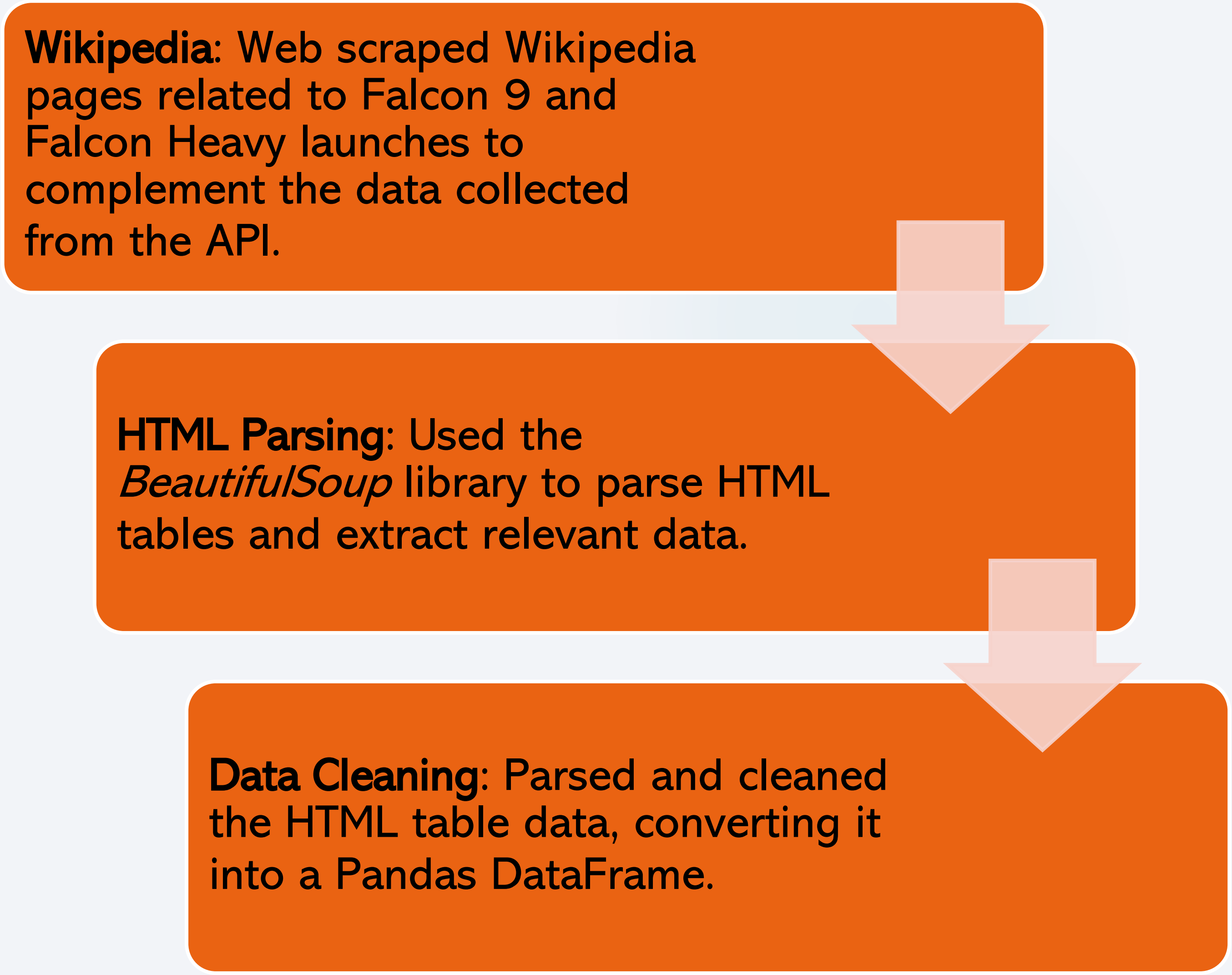
Data Retrieved: Information included rocket details, payload specifications, launch and landing outcomes, etc.

HTTP GET Requests: Employed the Python *requests* library to perform GET requests to the API and obtain the data in JSON format.

JSON to DataFrame: Converted the JSON data to a Pandas DataFrame using *json_normalize* for easier manipulation and analysis

Data Collection - Scraping

- https://github.com/reab5555/IBM_Projects_Demo/blob/cf6553ef1a02e66e48f91f4c52382aa58edd8e30/jupyter-labs-webscraping.ipynb



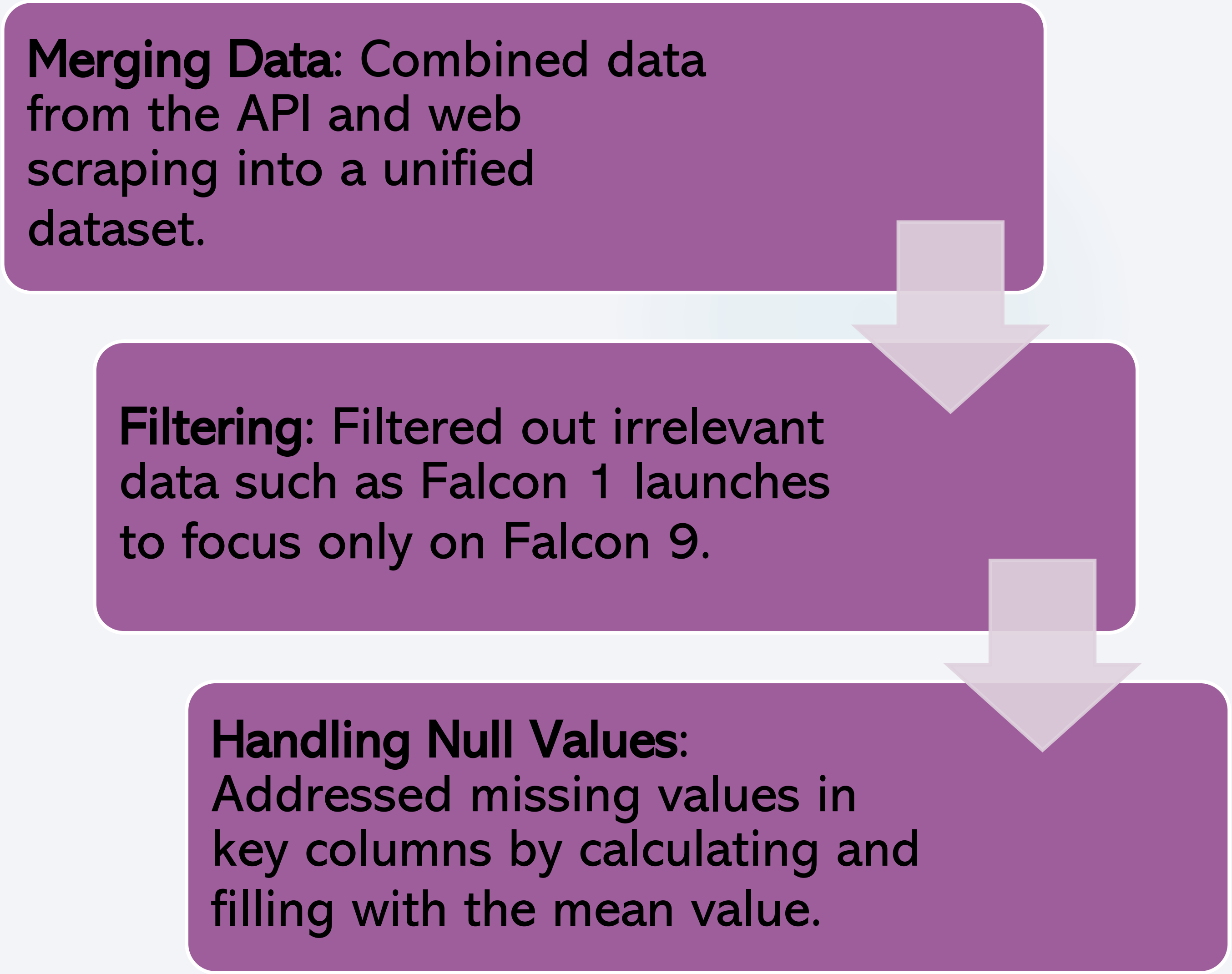
Wikipedia: Web scraped Wikipedia pages related to Falcon 9 and Falcon Heavy launches to complement the data collected from the API.

HTML Parsing: Used the *BeautifulSoup* library to parse HTML tables and extract relevant data.

Data Cleaning: Parsed and cleaned the HTML table data, converting it into a Pandas DataFrame.

Data Wrangling

- https://github.com/reab5555/IBM_Projects_Demo/blob/2b565f2894742d3b06957526023dc1d85f6af1ab/labs-jupyter-spacex-datawrangling.ipynb



Merging Data: Combined data from the API and web scraping into a unified dataset.

Filtering: Filtered out irrelevant data such as Falcon 1 launches to focus only on Falcon 9.

Handling Null Values: Addressed missing values in key columns by calculating and filling with the mean value.

EDA with Data Visualization

Scatter Plots:

Flight Number vs. Payload Mass

To visualize the relationship between the flight number (indicating continuous launch attempts) and the payload mass. This helps in understanding if these factors influence the launch outcome.

Flight Number vs. Launch Site

To explore the distribution of launches across different sites and their outcomes. This helps in identifying patterns related to launch site performance.

Payload Mass vs. Launch Site

To examine how payload mass varies with different launch sites and their outcomes. This helps in understanding if specific sites handle certain payload ranges better.

Flight Number vs. Orbit Type

To explore the relationship between the number of flights and the type of orbit. This helps in understanding if experience (flight number) influences success for specific orbits.

Payload Mass vs. Orbit Type

To examine the relationship between payload mass and orbit type. This helps in identifying if certain orbits are more likely to succeed with specific payload ranges.

Bar Chart: Success Rate by Orbit Type

To visualize the success rate of different orbit types. This helps in identifying which orbits are more likely to have successful landings.

Line Chart: Yearly Success Trend

To analyze the trend of launch success rates over the years. This helps in understanding if SpaceX has improved its success rate over time.

- https://github.com/reab5555/IBM_Projects_Demo/blob/101b2520e3f4b51eca99cb358c59f5b1cc63e675/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

1. SELECT DISTINCT Launch_Site FROM SPACEXTABLE; To identify all unique launch sites in the dataset.

2. SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%'; To filter and display records where the launch site names begin with 'CCA'.

3. SELECT SUM(Payload_Mass__kg_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)'; To calculate the total payload mass for launches conducted by NASA (CRS).

4. SELECT AVG(Payload_Mass__kg_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1'; To determine the average payload mass carried by the F9 v1.1 booster version.

5. SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)'; To find the date of the first successful landing on a ground pad.

6. SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND Payload_Mass__kg_ BETWEEN 4000 AND 6000; To list boosters that successfully landed on a drone ship with payload mass in the specified range.

7. SELECT Landing_Outcome, COUNT(*) FROM SPACEXTABLE GROUP BY Landing_Outcome; To count the total number of successful and failed landing outcomes.

8. SELECT Booster_Version FROM SPACEXTABLE WHERE Payload_Mass__kg_ = (SELECT MAX(Payload_Mass__kg_) FROM SPACEXTABLE); To identify boosters that carried the maximum payload mass.

9. SELECT SUBSTR(Date, 6, 2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE Landing_Outcome = 'Failure (drone ship)' AND SUBSTR(Date, 0, 5) = '2015'; To display records with specific details for failed drone ship landings in 2015.

10. SELECT Landing_Outcome, COUNT(*) AS OutcomeCount FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY OutcomeCount DESC; To rank the count of landing outcomes between the specified dates in descending order.

https://github.com/reab5555/IBM_Projects_Demo/blob/56d57162b93aefba7d20c1e93f72cb04c37cca9c/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Marker for the Launch Site
- Marker for the Closest Coastline Point
- Marker for the Closest City
- Marker for the Closest Railway
- Marker for the Closest Highway
- PolyLine Connecting Launch Site to Closest Coastline
- PolyLine Connecting Launch Site to Closest City
- **Geographical Context:** The markers and lines provide a visual representation of the launch site's proximity to key geographical features (coastline, city, railway, highway).
- **Distance Visualization:** Displaying distances directly on the map helps in understanding the accessibility and logistical considerations of the launch site.
- **Interactive Analysis:** The interactive map allows users to explore the spatial relationships and distances visually, making the data more intuitive and easier to analyze.

- https://github.com/reab5555/IBM_Projects_Demo/blob/830b419eba6811c8b5172fe25439a97a0623ce3e/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- **Launch Site Drop-down Input Component:** To provide users with the flexibility to filter data based on specific launch sites. This allows users to focus on data from a particular site or view an aggregate of all sites.
- **Success Pie Chart:** To give a quick visual summary of the success rates of launches for different sites.
- **Payload Range Slider:** To allow users to filter the data based on payload mass, which is an important factor in launch outcomes.
- **Success-Payload Scatter Chart:** To visualize the relationship between payload mass and launch success, with additional context provided by the booster version.

- https://github.com/reab5555/IBM_Projects_Demo/blob/f19fa5ff22b2cdecfbc4b2d48a57f78cb64a6ca3/ploty_dash_launch_sites.ipynb

Predictive Analysis (Classification)

Feature Selection:
- Identified features such as Flight Number, Payload Mass, Orbit, Launch Site, Grid Fins, Reused, Legs, and Landing Pad.

Model Selection:
- Chose models: Logistic Regression, Decision Trees, K Nearest Neighbors, and Support Vector Machine (SVM) for evaluation.

Train-Test Split:
- Split the data into training and testing sets to evaluate the model performance.

Model Training:
- Trained each model using the training set.

Model Evaluation:
- Evaluated each model on the test set using metrics: accuracy, precision, recall, and F1-score.

Model Improvement:
- Conducted hyperparameter tuning to optimize model performance.
- Used techniques such as Grid Search and Cross-Validation to find the best parameters.

Model Selection:
- Chose the model with the highest performance metrics (e.g., highest accuracy or F1-score).

- https://github.com/reab5555/IBM_Projects_Demo/blob/6059ccd94e1811b6961721982ebb3bb691c8cd24/SpaceX_Machine_Learning_Prediction.jupyterlite.ipynb

Results

Exploratory Data Analysis (EDA) Results

- Flight numbers of 36 to 45 there were only success of landings.
- For launch sites of CCAFS SLC 40 and KSC LC 39A, flight numbers from 78 to 90 there were only success of landings.
- For the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000)
- ES-L1, GEO, HEO, and SSO where the orbits with most success landings.
- In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.
- LEO, VLEO, and SSO seems to have the most success rates.
- The overall success rate is increasing since 2013 until 2017, and then increasing again after 2018.

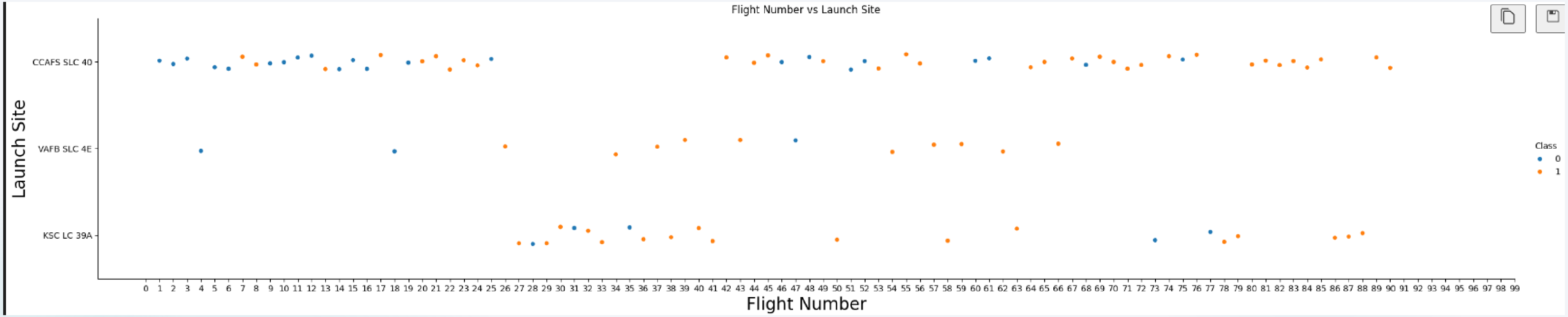
Predictive Analysis Results

- Flight attributes like for example payload mass, orbit, etc. can predict success outcome of landing.
- The best and most accurate model for success outcome prediction was decition tree model with an accuracy score of 0.94.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



Flight numbers vs their Launch Sites:

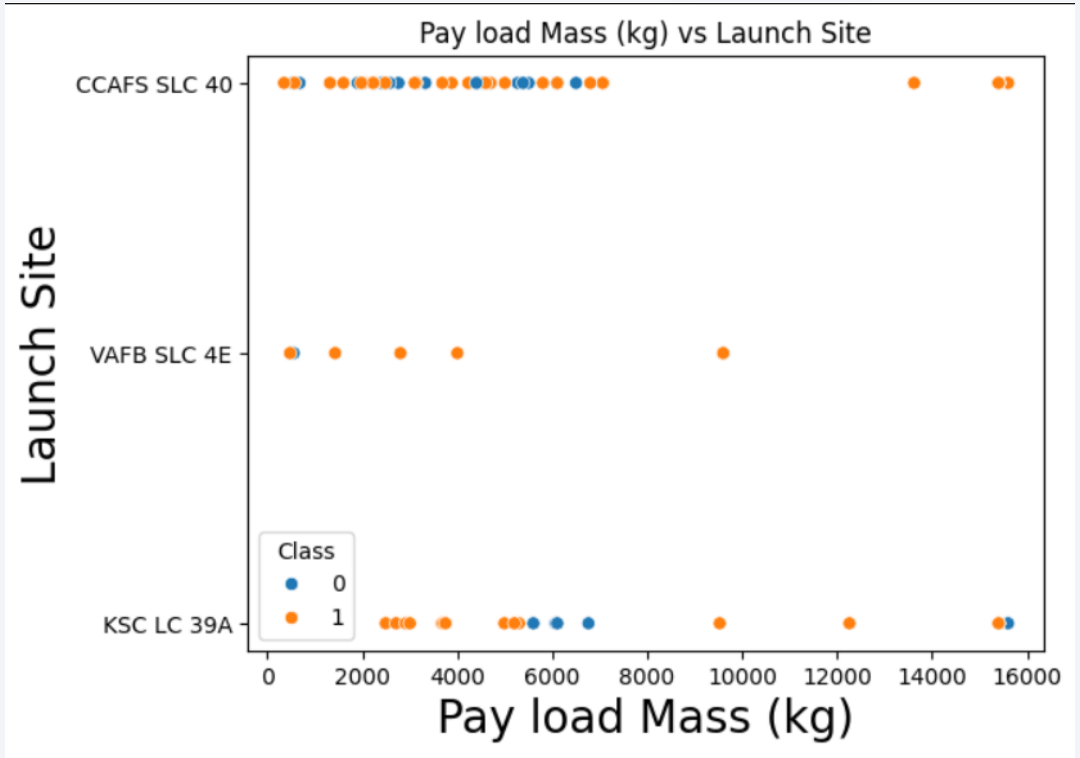
Flight numbers of 36 to 45 there were only success of landings.

Class 0 = Landing Failure

Class 1 = Landing Success

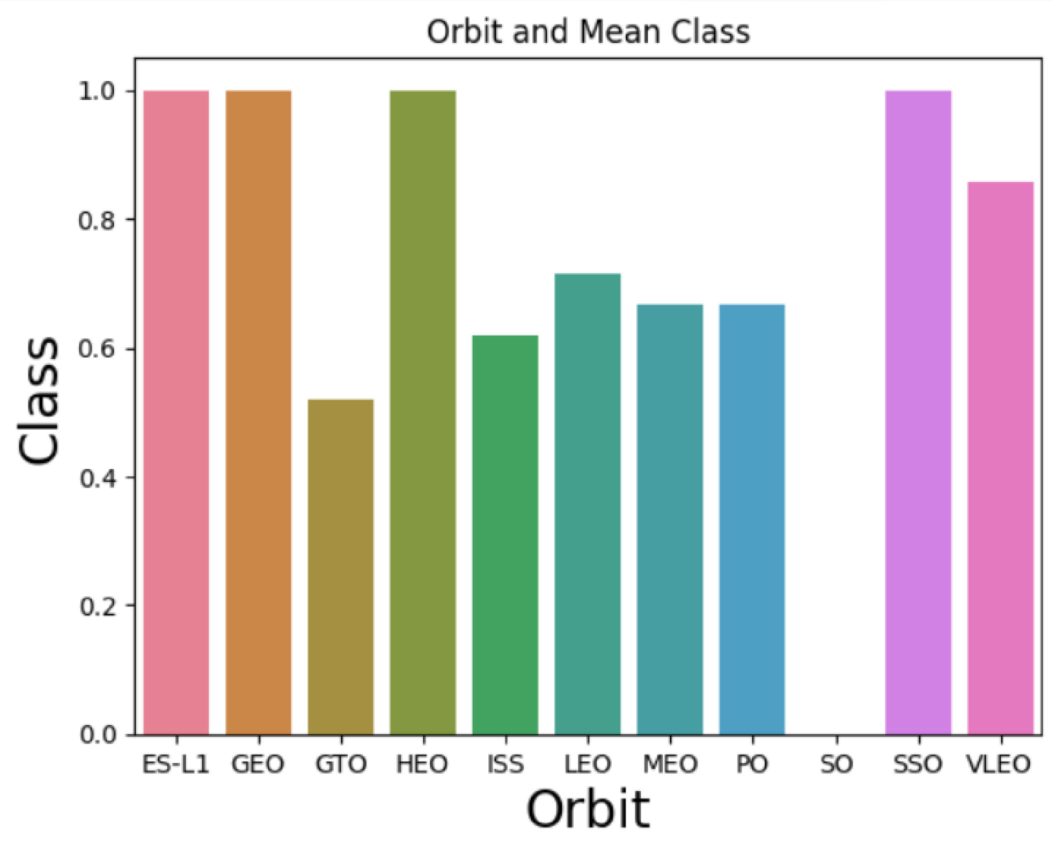
Payload vs. Launch Site

For the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000)



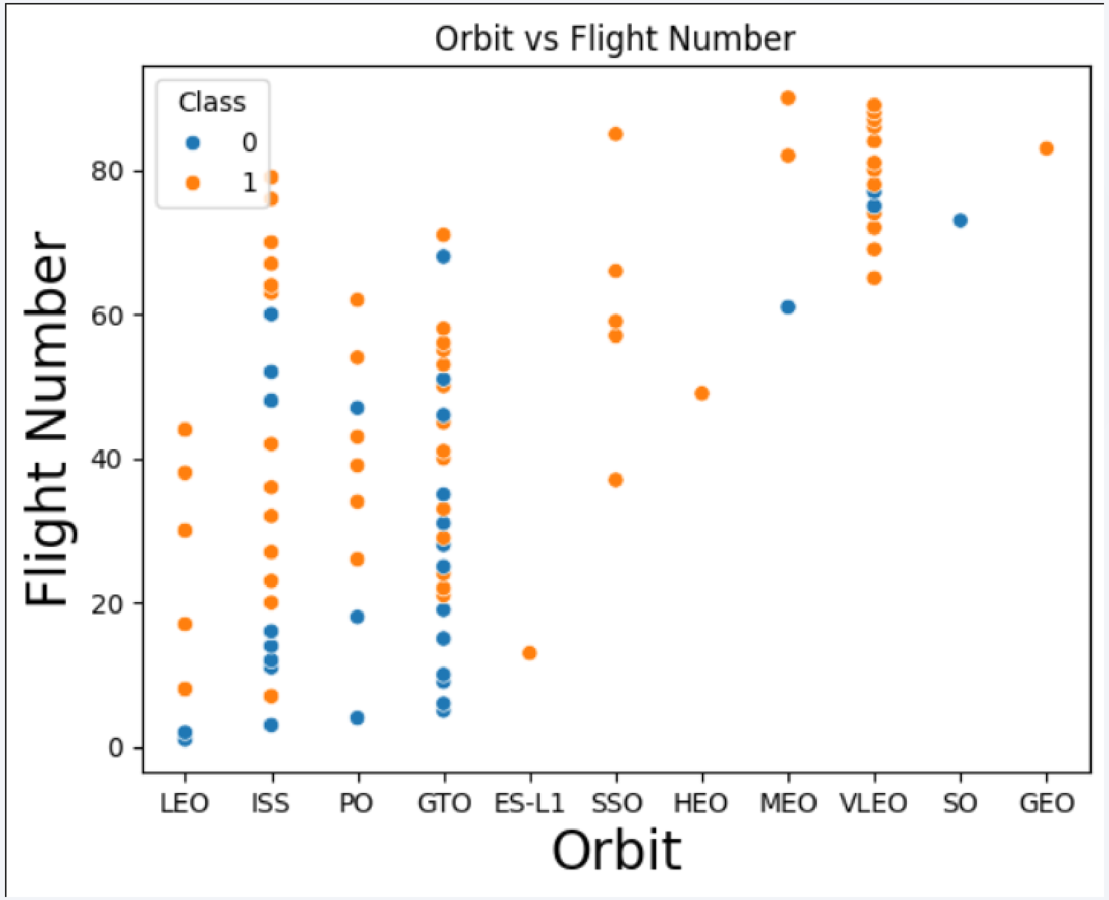
Success Rate vs. Orbit Type

ES-L1, GEO, HEO, and SSO where the orbits with most success landings.



Flight Number vs. Orbit Type

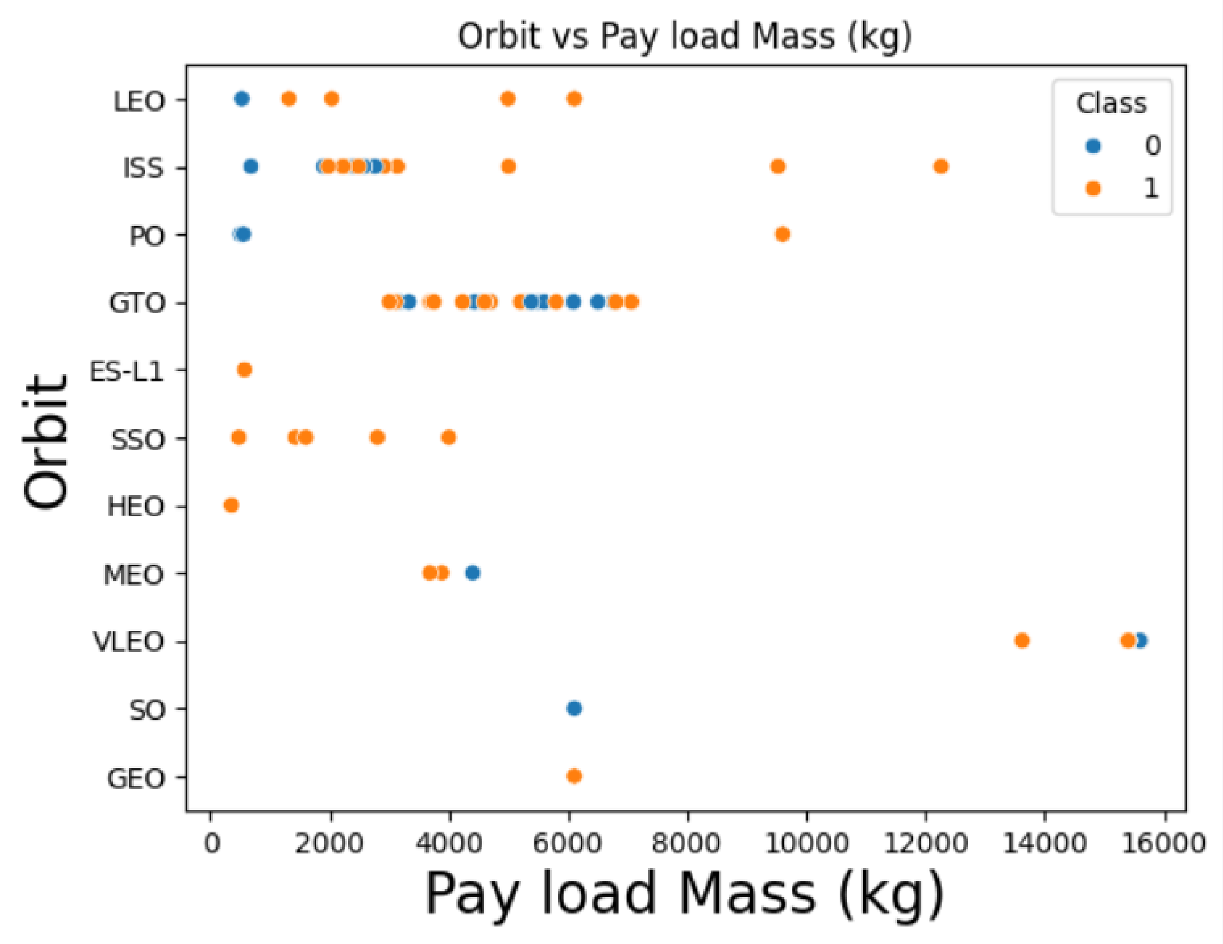
in the LEO orbit the Success appears related to the number of flights. on the other hand, there seems to be no relationship between flight number when in GTO orbit.



Payload vs. Orbit Type

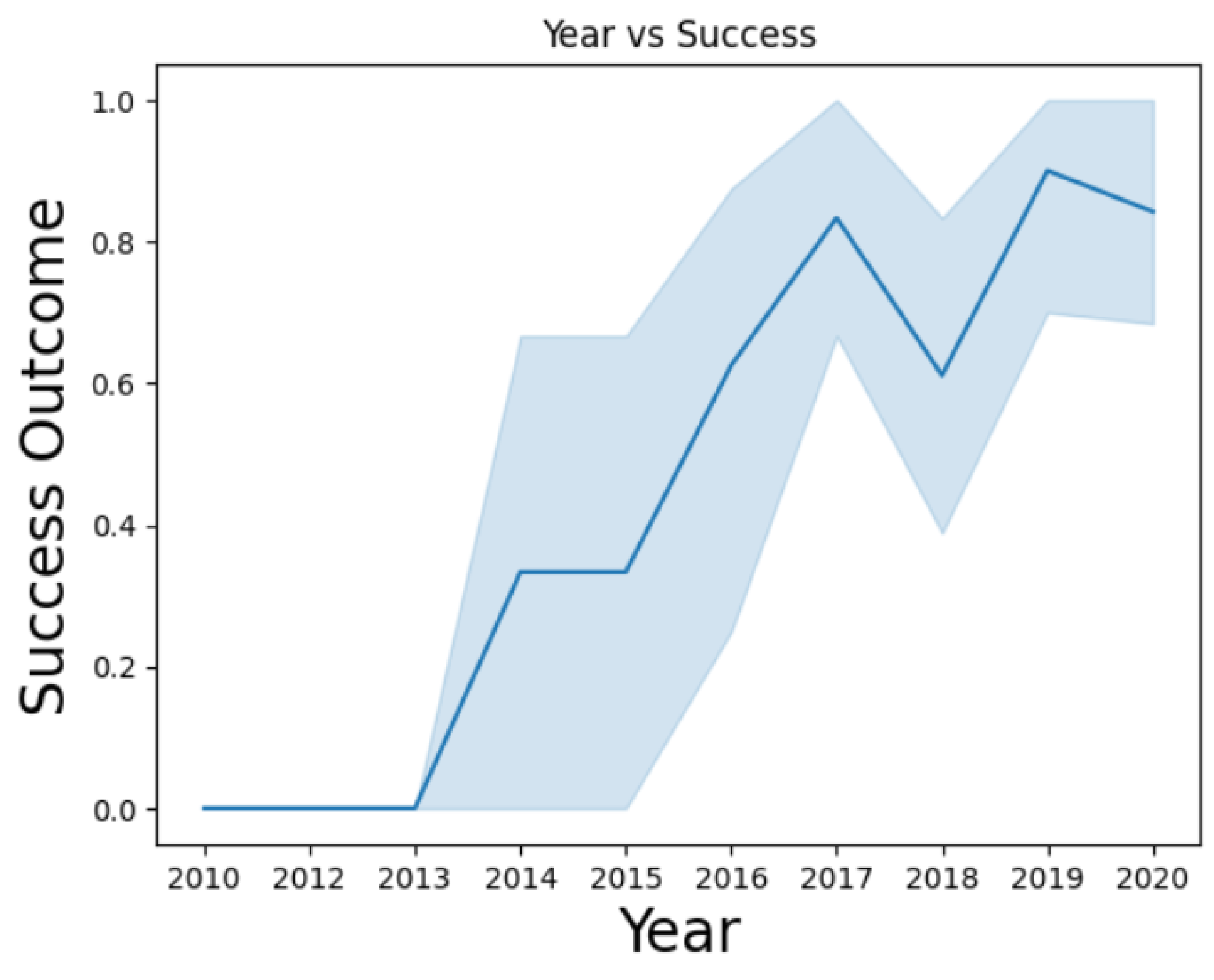
With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



Launch Success Yearly Trend

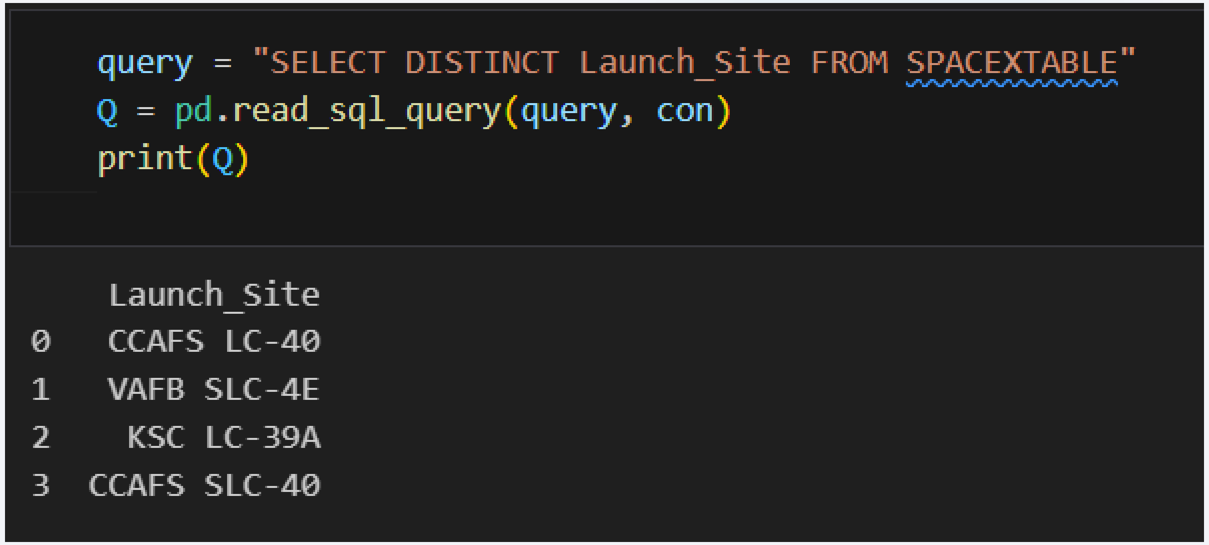
You can observe that the success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.



All Launch Site Names

Four Launch Sites in total:

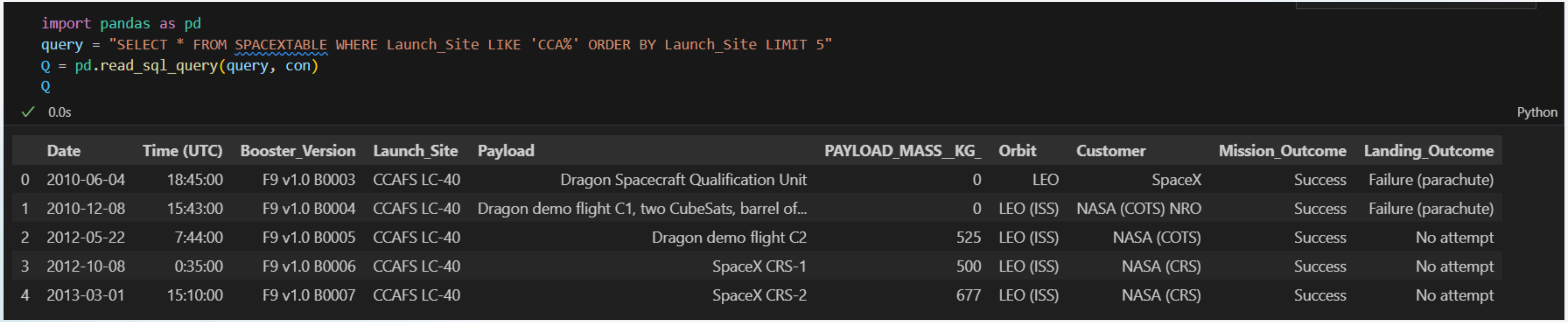
- CCAFS LC-40
- VAFB SLC-4E
- KSC LC-39A
- CCAFS SLC-40



```
query = "SELECT DISTINCT Launch_Site FROM SPACEXTABLE"
Q = pd.read_sql_query(query, con)
print(Q)
```

```
    Launch_Site
0   CCAFS LC-40
1   VAFB SLC-4E
2    KSC LC-39A
3  CCAFS SLC-40
```

Launch Site Names Begin with 'CCA'



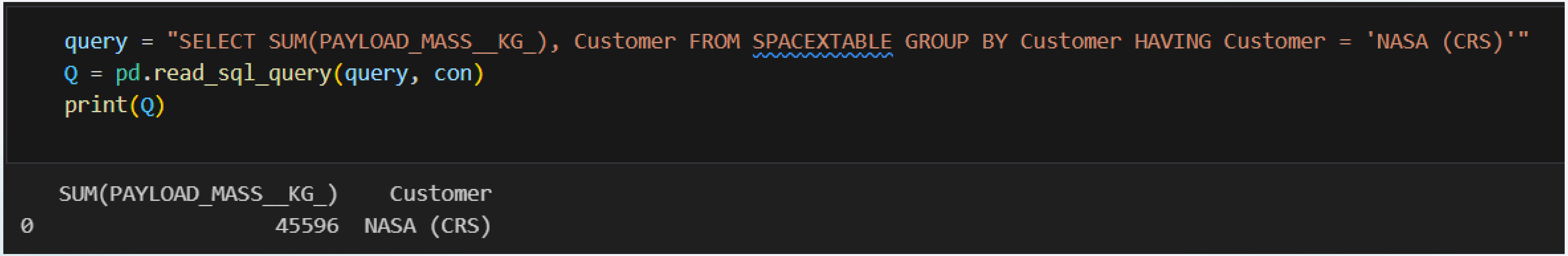
```
import pandas as pd
query = "SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' ORDER BY Launch_Site LIMIT 5"
Q = pd.read_sql_query(query, con)
Q
```

✓ 0.0s Python

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
0	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- 5 exemplary queries of launch sites that begins with ‘CCA’

Total Payload Mass

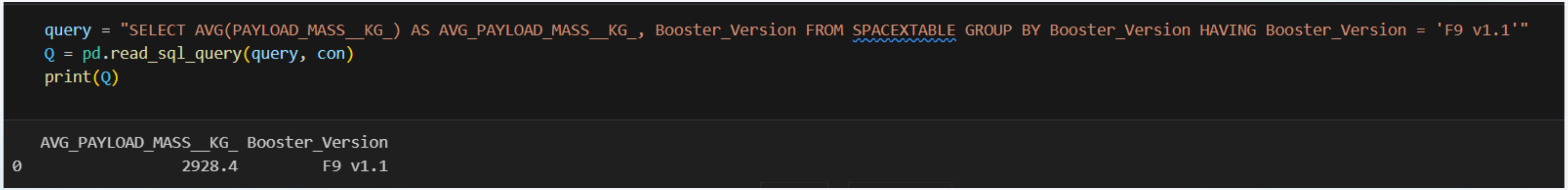


```
query = "SELECT SUM(PAYLOAD_MASS__KG_), Customer FROM SPACEXTABLE GROUP BY Customer HAVING Customer = 'NASA (CRS)'"
Q = pd.read_sql_query(query, con)
print(Q)
```

```
   SUM(PAYLOAD_MASS__KG_)    Customer
0                    45596  NASA (CRS)
```

The sum of total payload carried by boosters from NASA (45,596 kg).

Average Payload Mass by F9 v1.1

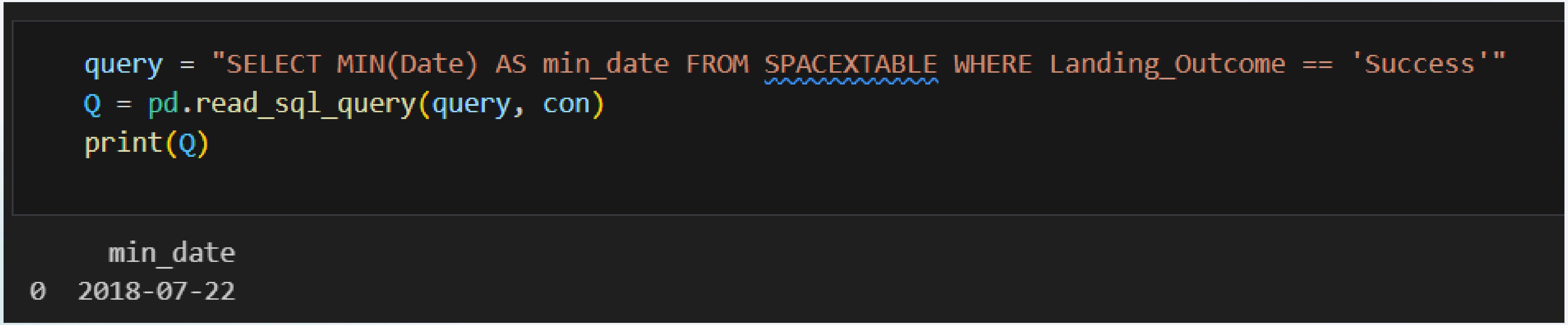


```
query = "SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS__KG_, Booster_Version FROM SPACEXTABLE GROUP BY Booster_Version HAVING Booster_Version = 'F9 v1.1'"
Q = pd.read_sql_query(query, con)
print(Q)
```

```
   AVG_PAYLOAD_MASS__KG_ Booster_Version
0                 2928.4         F9 v1.1
```

The average payload mass carried by booster version F9 v1.1 (2,928.4 kg)

First Successful Ground Landing Date

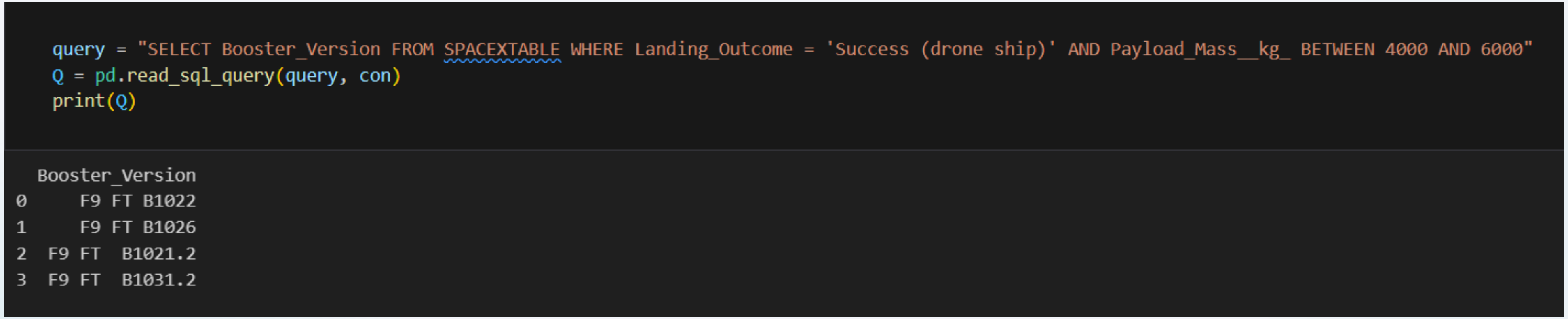


```
query = "SELECT MIN(Date) AS min_date FROM SPACEXTABLE WHERE Landing_Outcome == 'Success'"
Q = pd.read_sql_query(query, con)
print(Q)
```

```
     min_date
0  2018-07-22
```

22/7/2018 was the date with the first succesful ground landing.

Successful Drone Ship Landing with Payload between 4000 and 6000

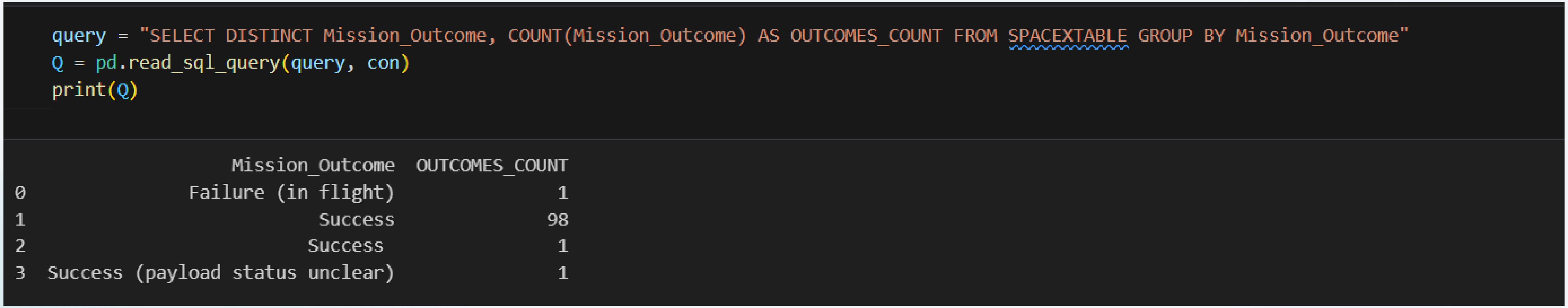


```
query = "SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND Payload_Mass__kg_ BETWEEN 4000 AND 6000"
Q = pd.read_sql_query(query, con)
print(Q)
```

```
  Booster_Version
0     F9 FT B1022
1     F9 FT B1026
2  F9 FT  B1021.2
3  F9 FT  B1031.2
```

The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes



```
query = "SELECT DISTINCT Mission_Outcome, COUNT(Mission_Outcome) AS OUTCOMES_COUNT FROM SPACEXTABLE GROUP BY Mission_Outcome"
Q = pd.read_sql_query(query, con)
print(Q)
```

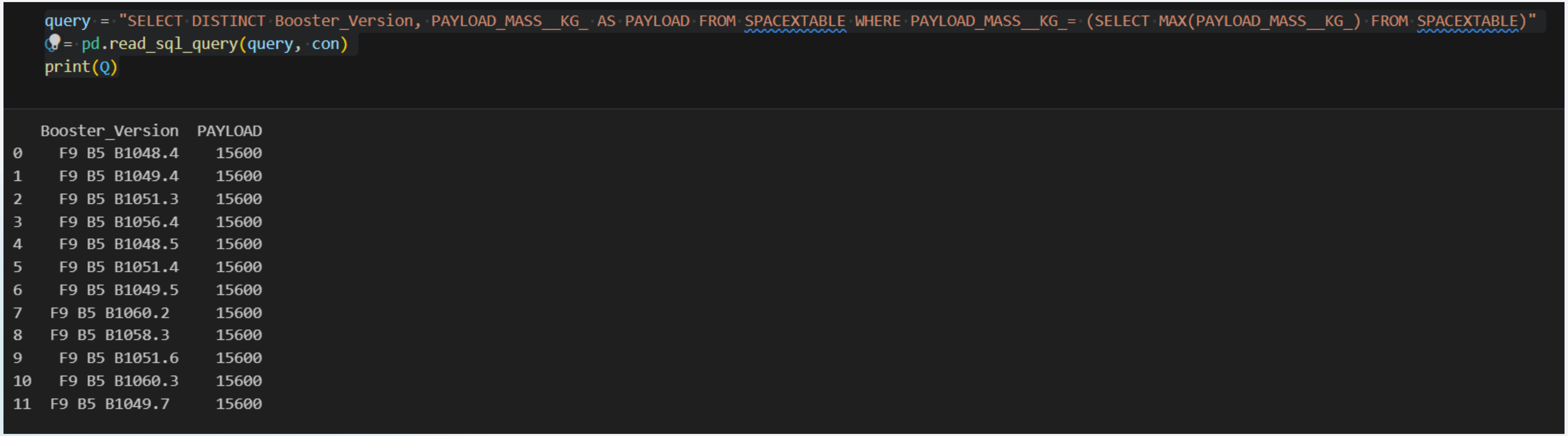
```
                    Mission_Outcome  OUTCOMES_COUNT
0               Failure (in flight)               1
1                           Success              98
2                          Success                1
3  Success (payload status unclear)               1
```

Total number of successful and failure mission outcomes

Success = 100

Failure = 1

Boosters Carried Maximum Payload



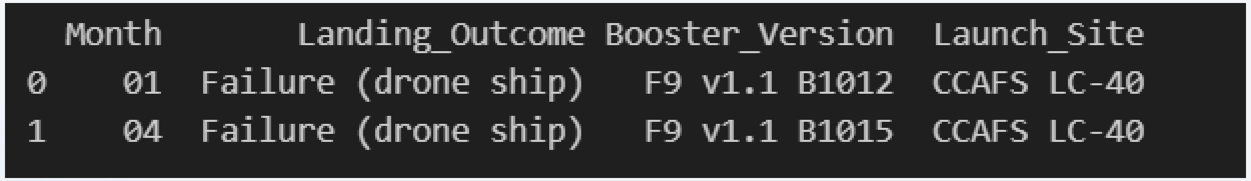
```
query = "SELECT DISTINCT Booster_Version, PAYLOAD_MASS__KG_ AS PAYLOAD FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_= (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)"
Q = pd.read_sql_query(query, con)
print(Q)
```

```
   Booster_Version  PAYLOAD
0    F9 B5 B1048.4    15600
1    F9 B5 B1049.4    15600
2    F9 B5 B1051.3    15600
3    F9 B5 B1056.4    15600
4    F9 B5 B1048.5    15600
5    F9 B5 B1051.4    15600
6    F9 B5 B1049.5    15600
7   F9 B5 B1060.2     15600
8   F9 B5 B1058.3     15600
9    F9 B5 B1051.6    15600
10   F9 B5 B1060.3    15600
11  F9 B5 B1049.7     15600
```

The names of the booster which have carried the maximum payload mass

Maximum payload = 15,600 KG

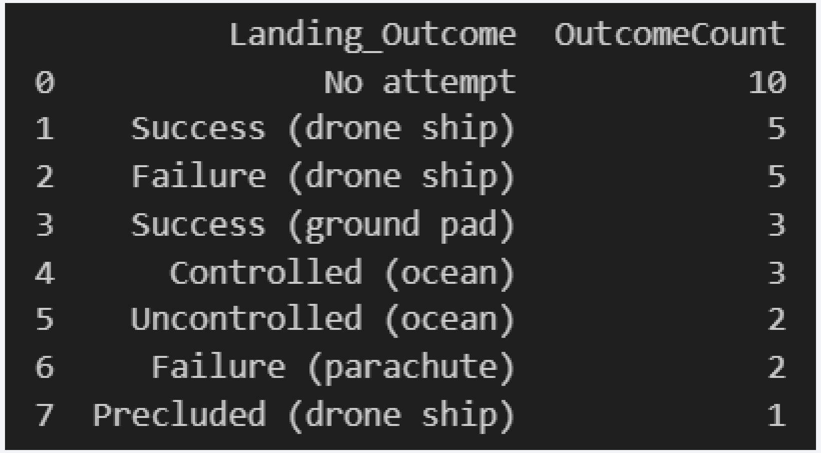
2015 Launch Records



```
  Month       Landing_Outcome Booster_Version  Launch_Site
0    01  Failure (drone ship)   F9 v1.1 B1012  CCAFS LC-40
1    04  Failure (drone ship)   F9 v1.1 B1015  CCAFS LC-40
```

List of the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

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



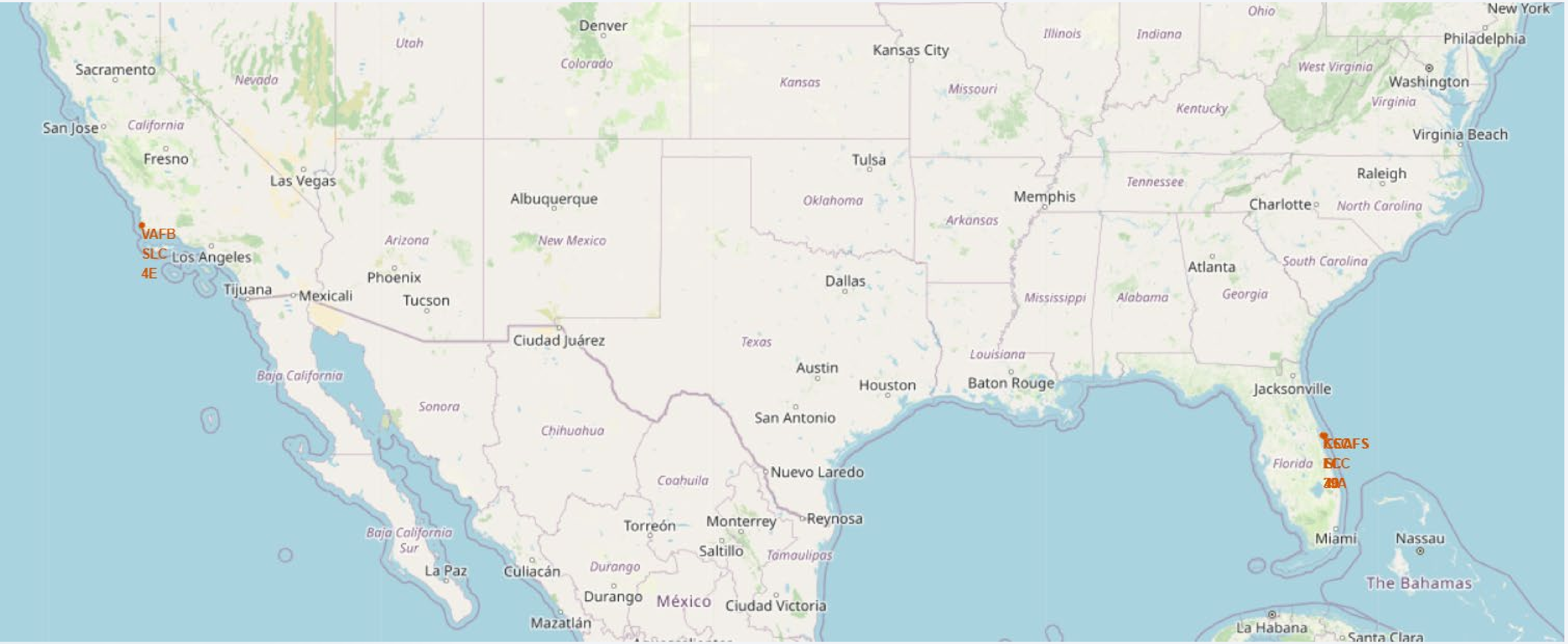
```
          Landing_Outcome  OutcomeCount
0              No attempt            10
1    Success (drone ship)             5
2    Failure (drone ship)             5
3   Success (ground pad)             3
4     Controlled (ocean)             3
5   Uncontrolled (ocean)             2
6    Failure (parachute)             2
7  Precluded (drone ship)             1
```

Ranks of 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 – most were No attempts and after that Drone ship success.

Section 3

Launch Sites Proximities Analysis

All launch sites locations



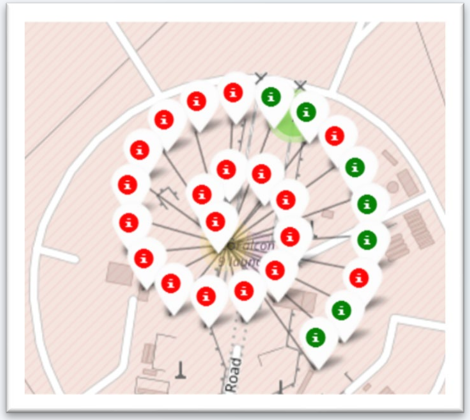
As you can see from the map, launch sites are located in Florida and California in the USA.

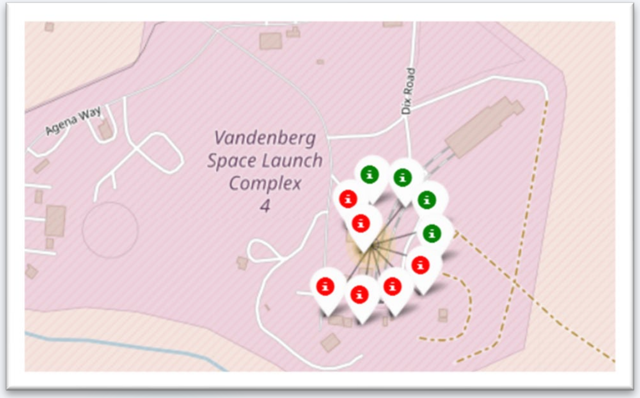
Launch Outcomes Maps

Labeled Launch Outcomes:

- Green = Success
- Red = Faliure

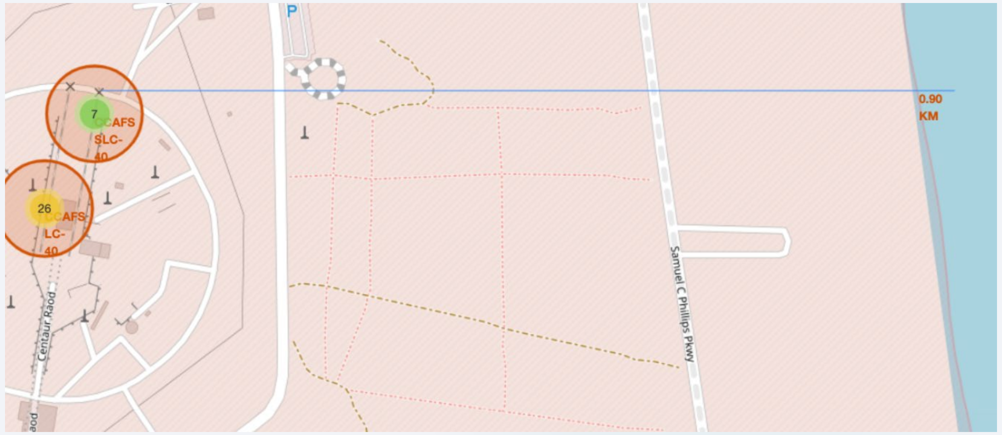
- Upper image: Falcon-9 Site
- Lower Image: Vandenber Site

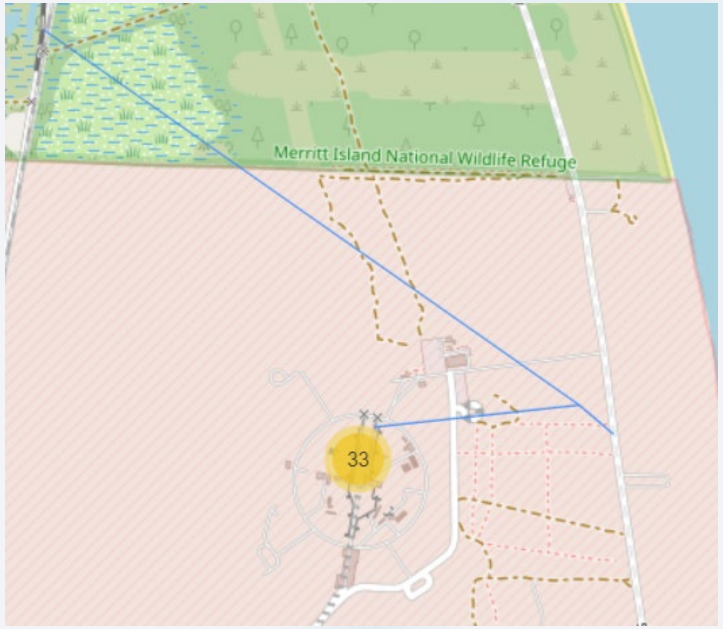


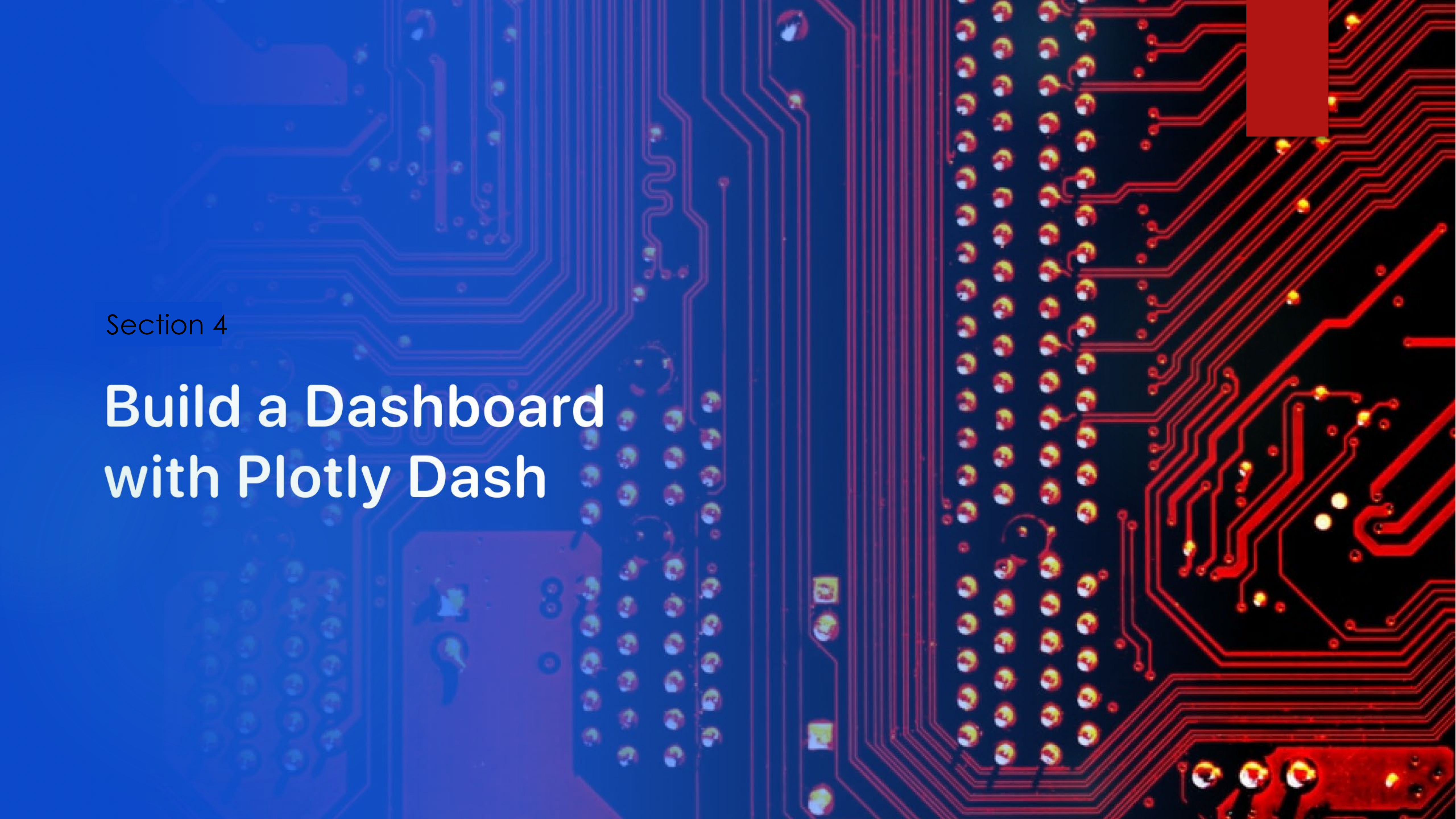


Near-Launch Site

- 1. To Coastline
- 2. To Highway/Railroad







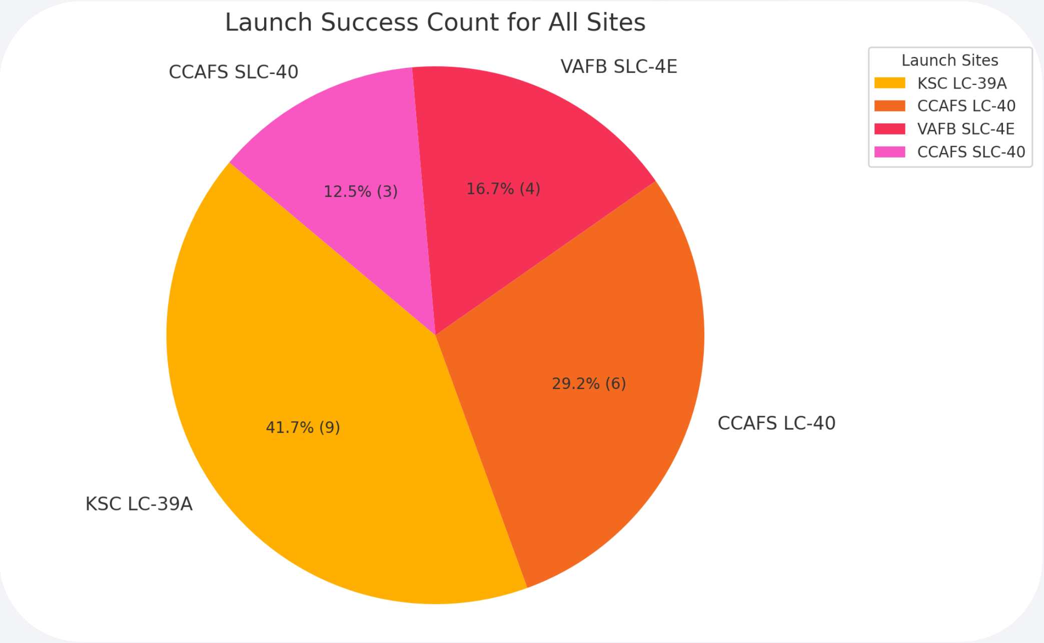
Section 4

Build a Dashboard with Plotly Dash

Launch success count for all sites

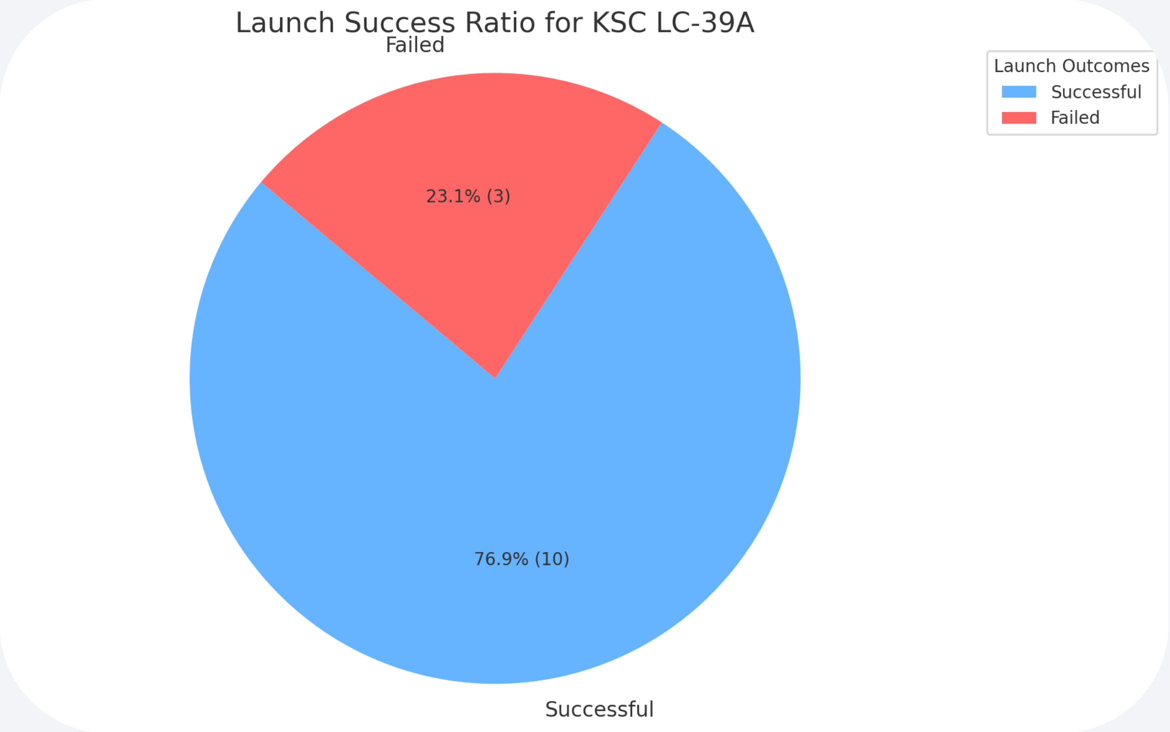
1. **KSC LC-39A**:
   This site has the highest proportion of successful launches, accounting for 41.7% of the total successful launches.
2. **CCAFS LC-40**:
   This site has the second-highest number of successful launches, with 29.2% of the total.
3. **VAFB SLC-4E**:
   This site accounts for 16.7% of the successful launches.
4. **CCAFS SLC-40**:
   This site has the lowest proportion, with 12.5% of the successful launches.

These findings indicate that the majority of successful launches occurred at the KSC LC-39A site, making it the most frequently used successful launch site for SpaceX missions.



Launch site with highest launch success ratio

These pie chart highlight KSC LC-39A as a highly reliable launch site with a significant proportion of successful launches.



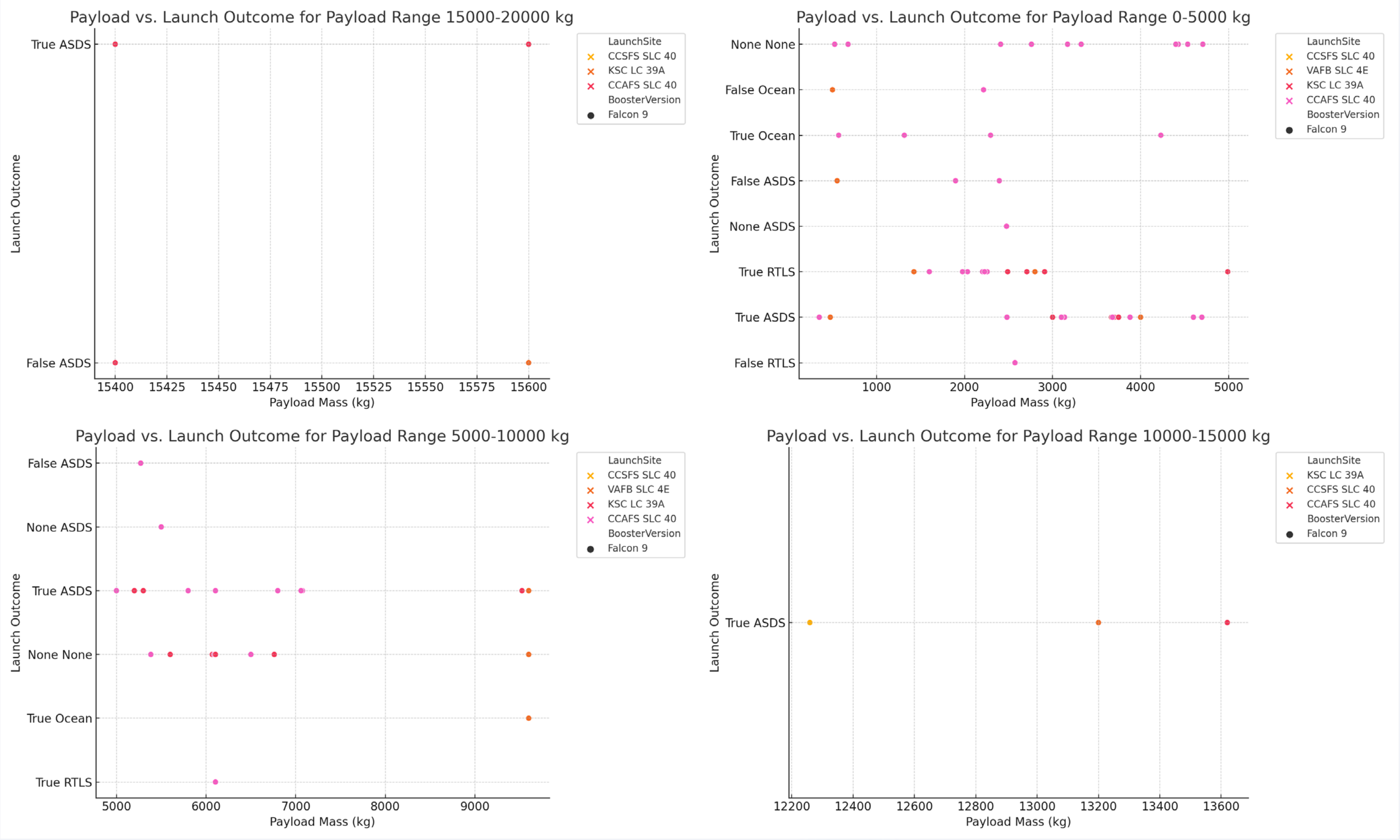
Success rates

1. There may be clusters indicating successful and unsuccessful launches, which can be analyzed based on launch sites and booster versions.

2. Notable trends could include which launch sites and booster versions have higher success rates for this payload range.

3. Identifying successful launch clusters and correlating them with specific launch sites and booster versions is essential.

4. Patterns indicating higher success rates for specific booster versions or launch sites might be observed.

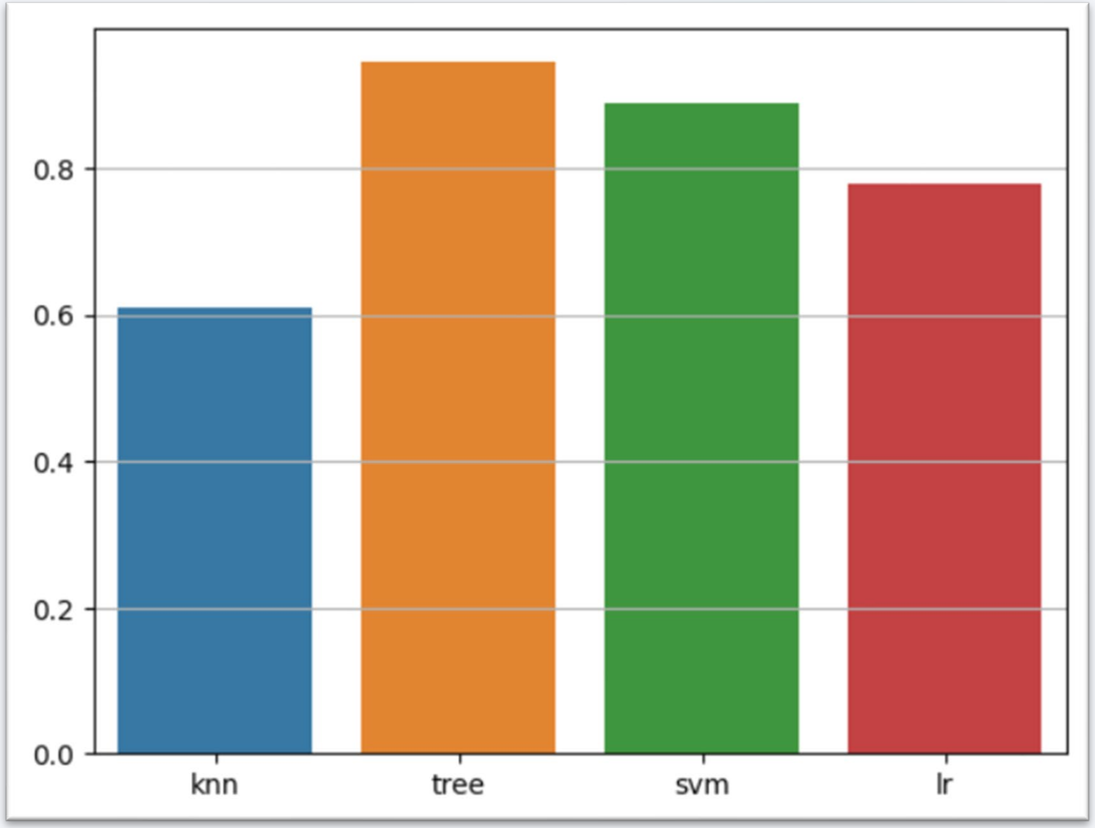


Section 5

Predictive Analysis (Classification)

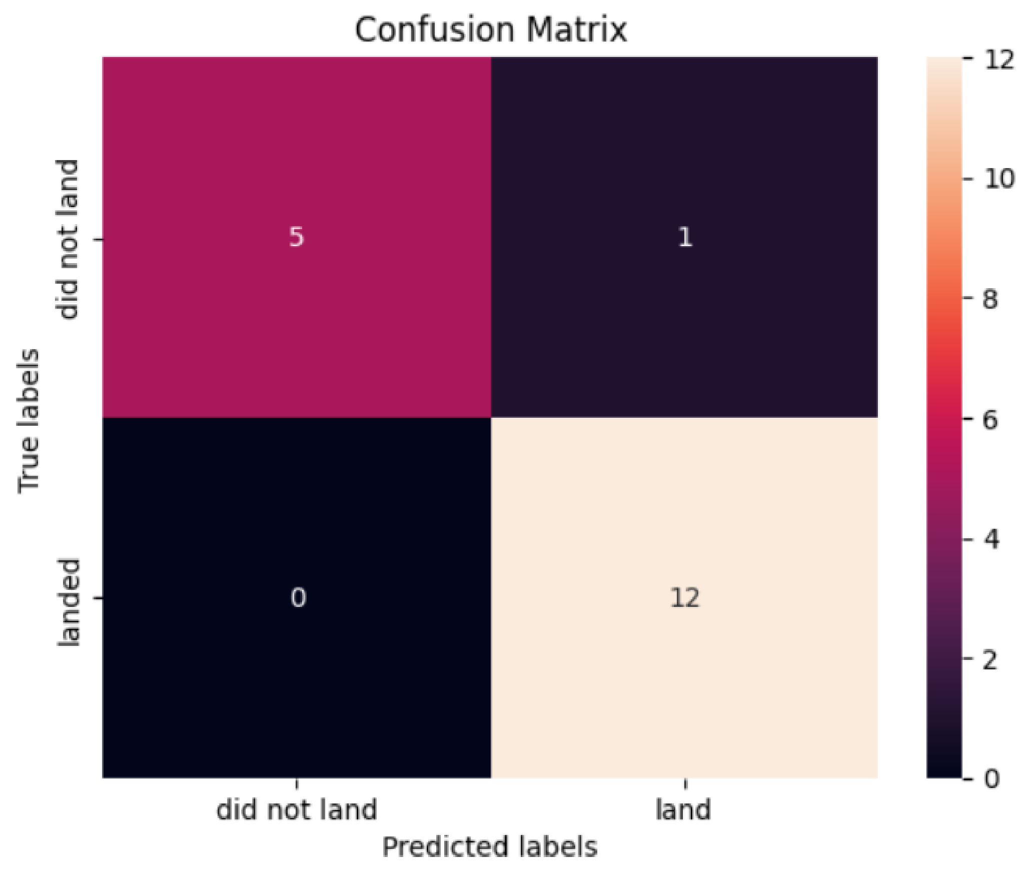
Classification Accuracy

- Decision Tree model received the highest accuracy among the classification models (0.944).



Confusion Matrix

- We can see in this confusion matrix that 12/13 were correctly labled as ‘land’, and 5/6 were correctly labled as ‘did not land’.
- This means that it was wrong in its predictions only on 2 predictions out of 17
- Accuracy = 0.944



Conclusions

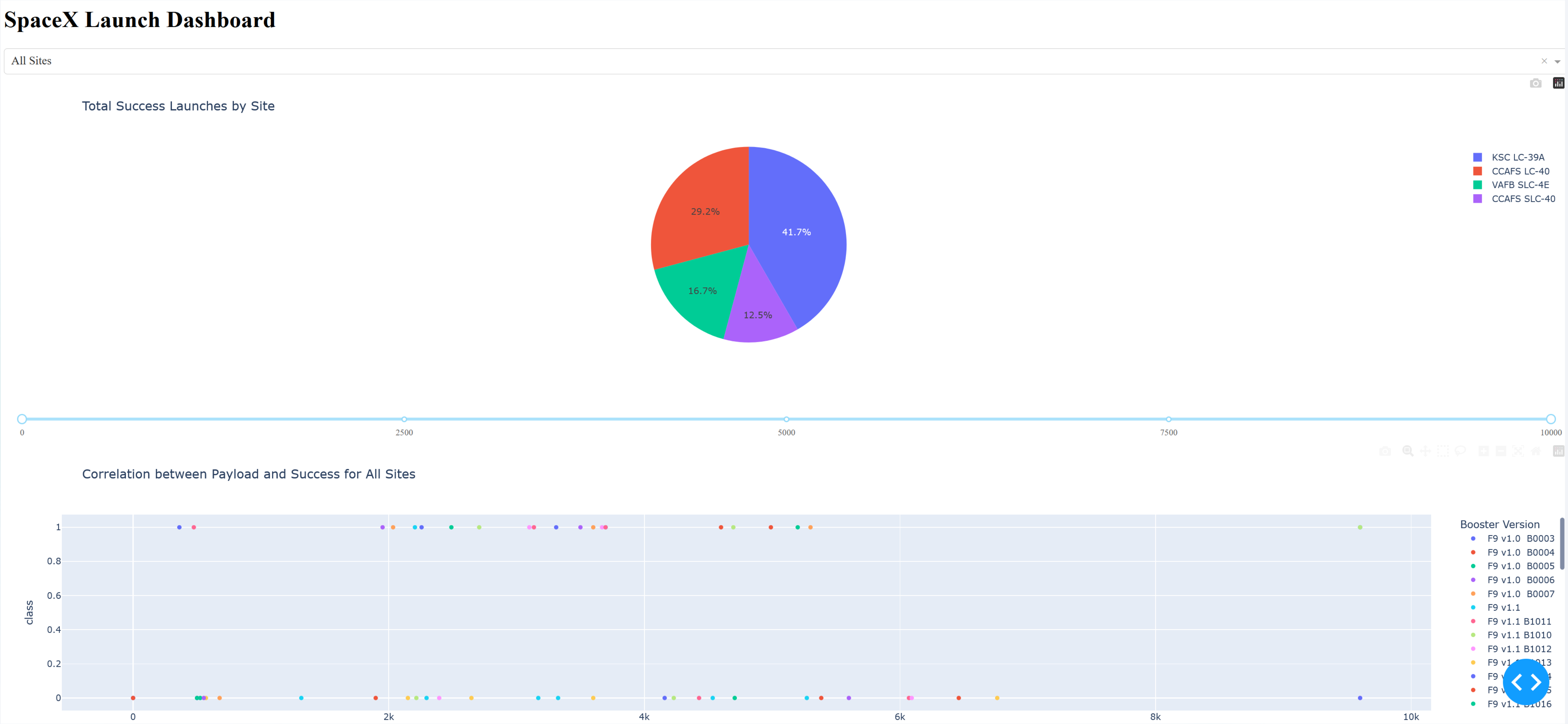
Visual Analytics and EDA:
- Interactive visual analytics using tools like Folium and Plotly Dash provided valuable insights into spatial relationships and trends, enhancing the understanding of launch outcomes.
- Exploratory Data Analysis (EDA) revealed key patterns, such as the relationship between flight numbers, launch sites, and payload mass, contributing to a deeper understanding of factors influencing launch success.

Influence of Payload and Orbit Types:
- Heavy payloads tend to have higher successful landing rates in specific orbits such as Polar, LEO, and ISS.
- Success rates for different orbits (ES-L1, GEO, HEO, SSO) are notably high.

Successful Launch Sites and Trends:
- The majority of successful launches occurred at the KSC LC-39A site.
- There is an increasing trend in overall launch success rates since 2013.

Predictive Model Performance:
- The decision tree model achieved the highest accuracy (0.944) among the classification models.

Appendix



SpaceX Launch Dashboard

All Sites

Thank you!