IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Amine Hammoumi
09/05/2025

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context

1. SpaceX advertises low-cost Falcon 9 rocket launches (average of $62m vs. $165m of competitors
2. **using data science** to drive innovation and competitiveness in the commercial space industry.
3. **main objective** is to predict whether the first stage of the Falcon 9 will land successfully, using machine learning models and public data to inform launch pricing and strategy.

- Problems you want to find answers

1. **Predicting Successful Landings**: Determine the likelihood of the Falcon 9's first stage landing successfully. This involves analyzing historical data and applying machine learning models to make accurate predictions.
2. **Cost Reduction Strategies**: Explore how the reuse of the Falcon 9's first stage can lead to significant cost savings in space launches. Understanding the factors that contribute to successful landings can help optimize these strategies.

Section 1

Methodology

Methodology

Executive Summary

Data Collection Methodology

- **Sources:**
  - SpaceX API
  - Web scraping from the SpaceX Wikipedia page
- **Data Wrangling:**
  - Handled missing values by replacing them with mean values
- **Exploratory Data Analysis (EDA):**
  - Conducted visual and SQL-based analysis
    - Analyzed launch outcomes by orbit type
    - Explored relationships between payload mass, booster versions, and outcomes using SQL
    - Created visualizations by payload mass, launch time, orbit type, and launch site
- **Interactive Visual Analytics:**
  - Used **Folium** for geospatial mapping of launch sites
  - Built an interactive dashboard with **Plotly Dash**:
    - Analysis by launch site, payload mass, and booster version
- **Predictive Analysis:**
  - Applied classification models:
    - Logistic Regression
    - Support Vector Machine (SVM)
    - Decision Tree
    - K-Nearest Neighbors (KNN)
  - Performed hyperparameter tuning using **Grid Search**

Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts
- **SpaceX REST API:**
  - Accessed core launch data
  - Retrieved booster version details
  - Collected launch site and payload information
  - Utilized RESTful interface for structured data retrieval
- **Web Scraping (SpaceX Wikipedia Page):**
  - Sent HTML requests using HTTP GET
  - Used **Python** with the **BeautifulSoup** library for parsing HTML
  - Extracted structured data from HTML tables
  - Retrieved column names from table headers for data structuring

Data Collection – SpaceX API

Present your data collection with SpaceX REST calls using key phrases and flowcharts

1.Identify SpaceX API Endpoint

Example: https://api.spacexdata.com/v4/launches/latest

2.Send GET Request

Use Python's requests.get() .

3.Receive JSON Response

The API returns structured data in JSON format.

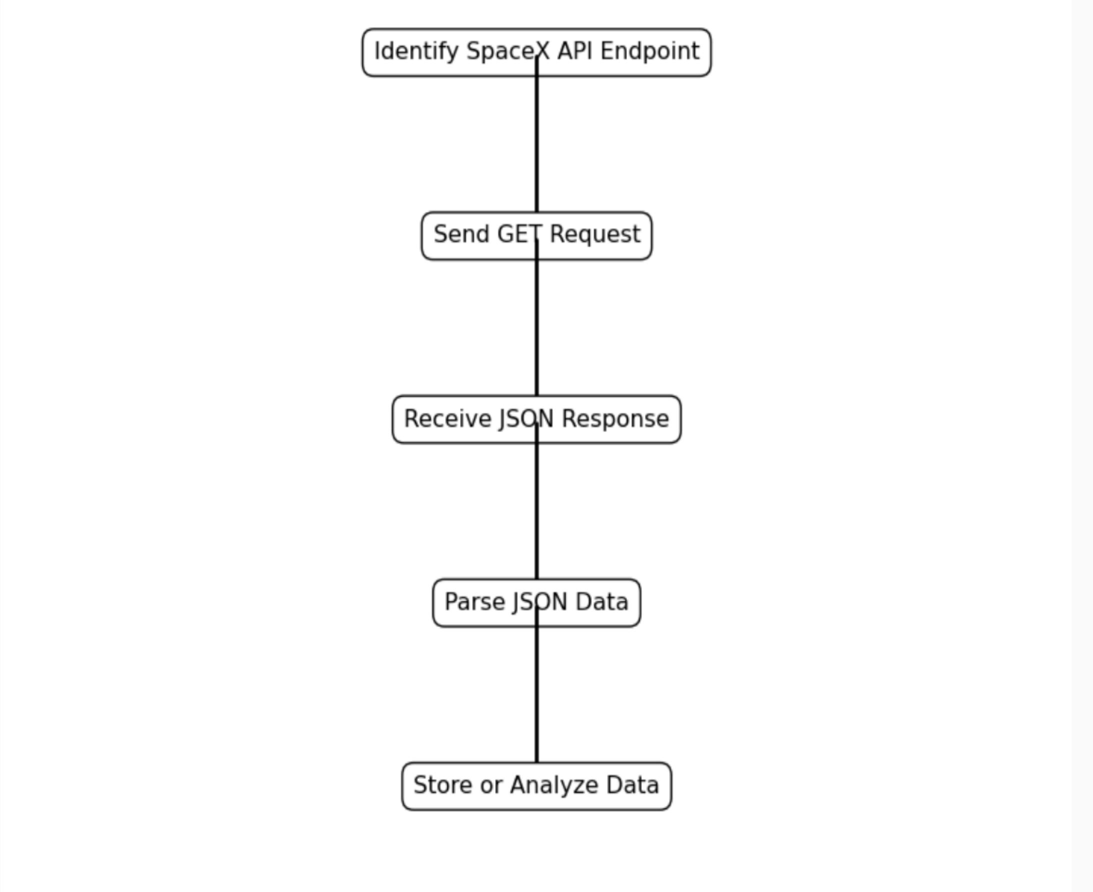
4.Parse JSON Data

Extract relevant fields using Python's

5.Store or Analyze Data

Save to a file, database, visualize/analyze directly.

[Github link : ds_IBM_certificate/jupyter-labs-spacex-data-collection-api-v2.ipynb at main · hammoumi123/ds_IBM_certificate]

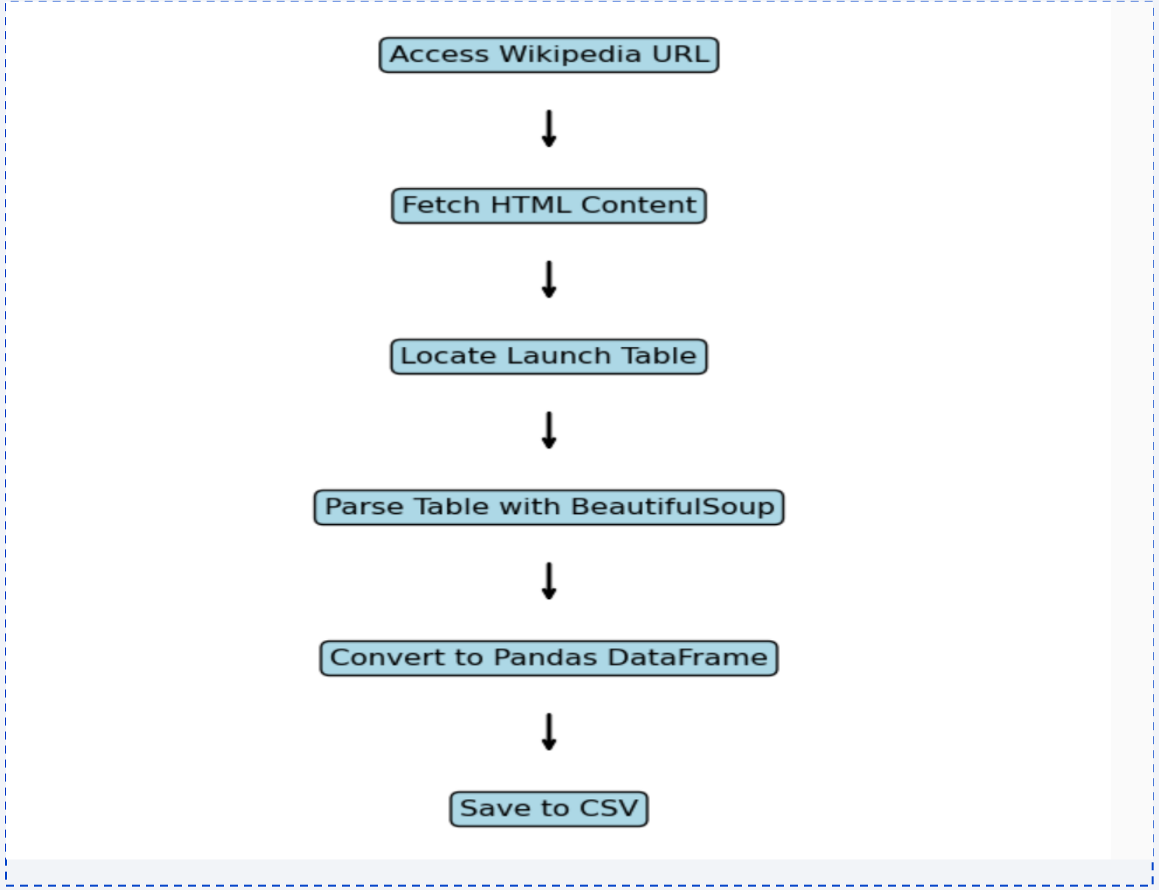


Data Collection - Scraping

1.Present your web scraping process using key phrases and

1.**Access Target URL**
(Wikipedia page with a launch table)
2.Fetch HTML Content
(Use requests.get() to retrieve the page)
3.Locate Data Table
(Identify the relevant <table> element)
4.Parse Table with BeautifulSoup
(Extract rows and columns from the HTML)
5.Convert to Pandas DataFrame
(Structure the data for analysis)
6.Export DataFrame to CSV
(Save the data locally for reuse as csv format)

- Add the GitHub URL:
  [ds_IBM_certificate/jupyter-labs-webscraping (1).ipynb at main · hammoumi123/ds_IBM_certificate]()



Data Wrangling

- Describe how data were processed

Dealing with missing values: The column for payload mass had handful of missing values. We replaced them with the mean value

- (standard in Analysis / ML)Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

- Github URL :

ds_IBM_certificate/labs-jupyter-spacex-Data wrangling-v2.ipynb at main · hammoumi123/ds_IBM_certificate

EDA with Data Visualization

Summarize what charts were plotted and why you used those charts

- Payload mass vs. Flight number vs. Success rate: This shows us the development of the payload mass and the success rate over time
- Launch site vs. Flight number vs. Success rate: This shows us the success rate of each launch site over time
- Launch site vs. Payload mass vs. Success rate: This shows us which payload is best to have success at a specific launch site
- Orbit type vs. Success rate: This can give us a hint which orbit types have the highest success rates
- Orbit type vs. Flight number vs. Success rate: This shows us the development of orbit types over time
- Orbit type vs. Payload mass vs. Success rate: Shows us the success rate for specific orbit type / payload mass clusters
- Success rate vs. Year: Shows the success development over time

GitHub URL of completed EDA with data visualization notebook, as an external reference and peer-review purpose :

ds_IBM_certificate/jupyter-labs-eda-dataviz-v2.ipynb at main · hammoumi123/ds_IBM_certificate

EDA with SQL

SQL queries

- Extract a list of all launch sites
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which carried the maximum payload mass
- List the failed landing_outcomes in drone ship, their booster versions, and launch site namesfor in year 2015
- 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

the GitHub URL of completed EDA with SQL notebook, as an external reference and peer-review purpose :

- ds_IBM_certificate/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · hammoumi123/ds_IBM_certificate

Build an Interactive Map with Folium

Map Objects

- Edged Circles (radius 1000m): Space launch sites
- Markers: for labelling all objects
- Marker Cluster: for creating a bunch of markers around space launch sites to indicate success (green) or failure (red) of the landing of the rocket's first stage
- Lines: Measure the distance between the launch site and the next coast or next city

[ds_IBM_certificate/lab-jupyter-launch-site-location-v2.ipynb at main · hammoumi123/ds_IBM_certificate]

Build a Dashboard with Plotly Dash

- **Summarize what plots/graphs and interactions you have added to a dashboard**
- Input Elements:
- Dropdown list for the launch site (with option to select all)
- Range Slider for selecting the payload mass
- Output Elements:
- Pie Chart: for showing the success rate of each launch site, or (if all sites are selected)
- showing the number of successful landing outcomes
- Scatterplot: Show success/failure by payload and booster version

ds_IBM_certificate/spacex_dash_app.py at main · hammoumi123/ds_IBM_certificate

Predictive Analysis (Classification)

- **Preprocessing**
  - One-Hot-Encoding for Categorical Features
  - Split data into dependent/independent variables and train/test data
  - Scale Data with StandardScaler
- **Model Building for each Method**
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree
  - K-Nearest Neighbor
- **Optimization**
  - Use Gridsearch for optimizing the models based on their hyperparameters
- **Evaluation**
  - Use Accuracy of Gridsearch for selecting the best parameter
  - Use Score to compare each classification method

[ds_IBM_certificate/SpaceX-Machine-Learning-Prediction-Part-5-v1.ipynb at main · hammoumi123/ds_IBM_certificate]

Results

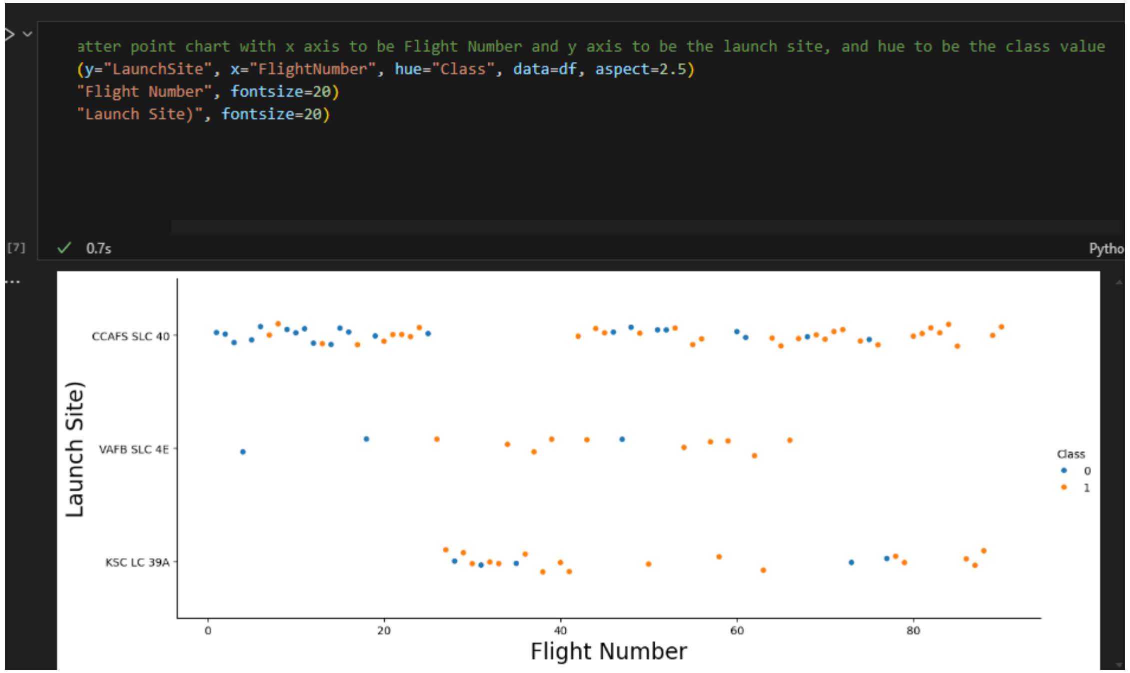
- **Exploratory data analysis results**
    - Launch success rate increases over time
    - Higher success rate for higher orbits
- **Interactive analytics demo in screenshots**
    - Higher success rate for higher payload mass
    - Low success rate for booster versions v1.0, v1.1, high success rate for FT, B4, B5
    - Higher success rate for Kennedy Space center and recent starts at Cape Canaveral
- **Predictive analysis results**
    - Best prediction results with Logistic Regression and Support Vector Machine

Section 2

Insights drawn from EDA

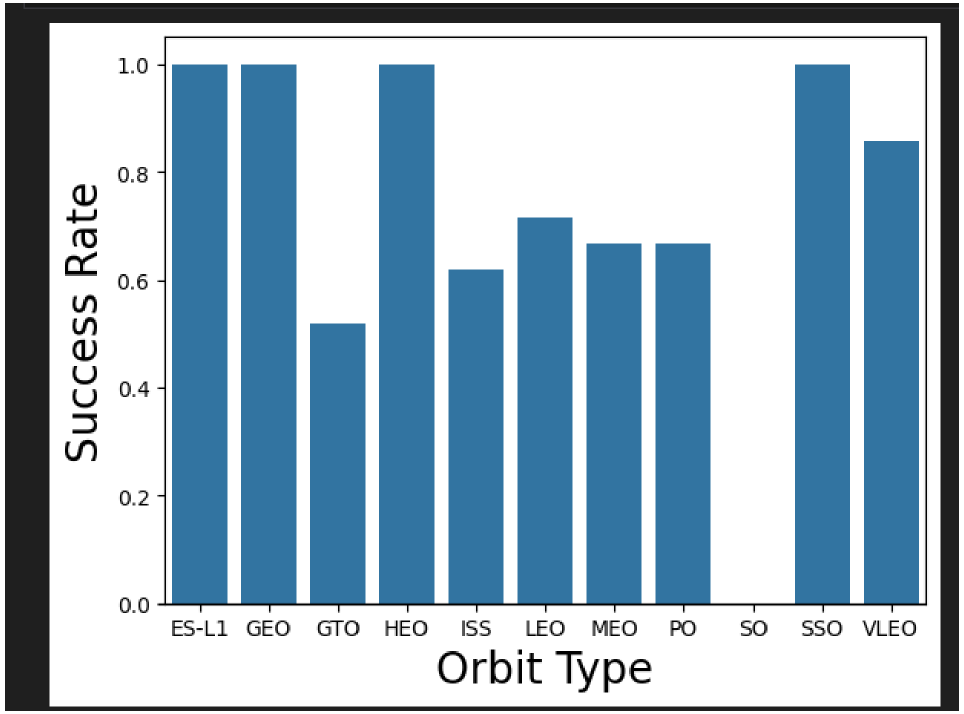
Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site
- **Strategic Insights:**
- The heavy use of CCAFS SLC 40 suggests it's the primary operational launch site
- VAFB SLC 4E usage might be reserved for specific orbital requirements (like polar/sun-synchronous orbits)
- KSC LC 39A's selective use could indicate it's reserved for special missions or higher-priority payloads



Payload vs. Launch Site

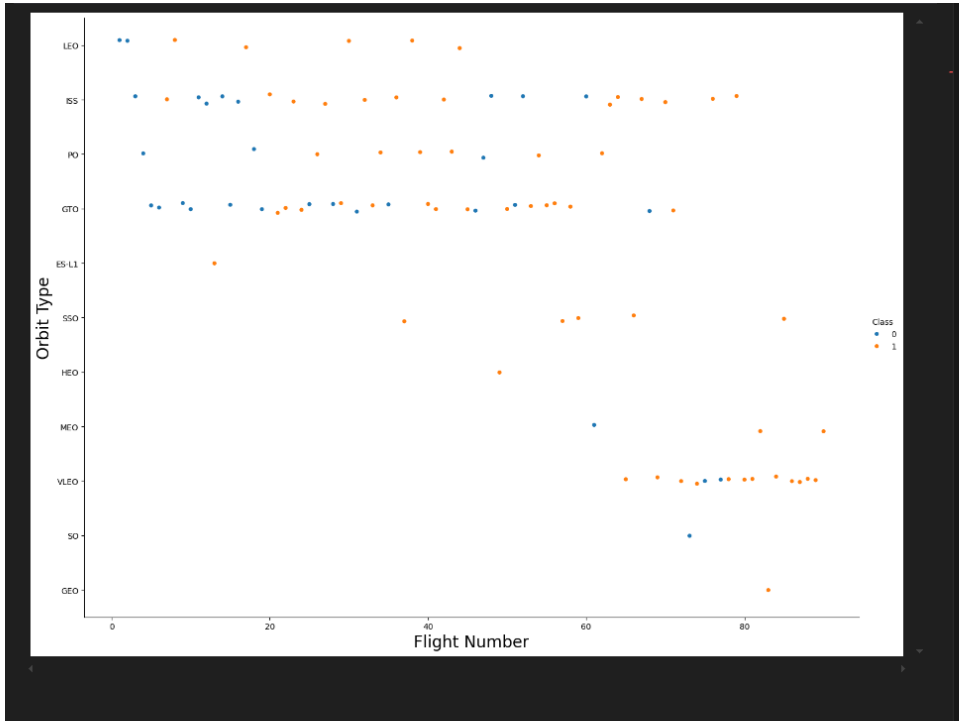
- Show a scatter plot of Payload vs. Launch Site
- **Distribution Insights:**
- Early flight numbers (1-20) show mostly lighter payloads from both classes
- As flight numbers increase, there's more diversity in payload weights
- Class 1 appears to handle the mission-critical heavy payload flights
- There's no clear linear trend - payload capacity seems to vary significantly regardless of flight sequence



Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type
- GTO missions show the lowest success rate (~52%) likely due to the challenging dual-burn sequence required to reach geostationary transfer orbit, while simpler direct-insertion orbits like GEO and SSO achieve perfect success rates.

Flight Number vs. Orbit Type



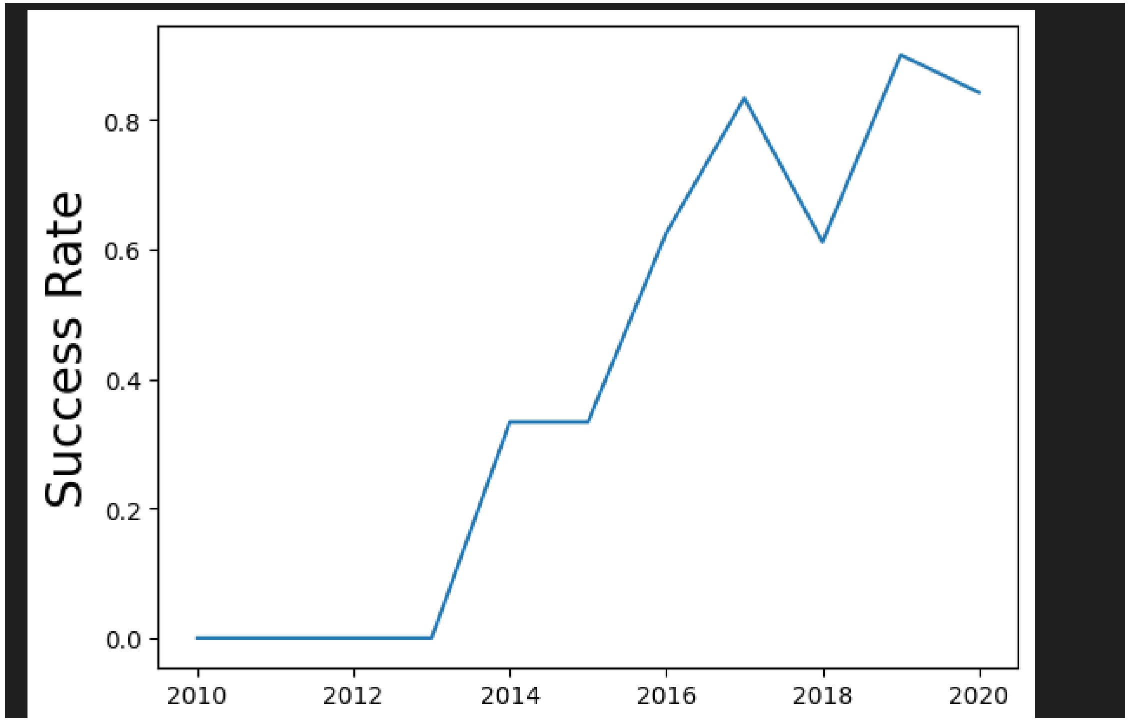
- Show a scatter point of Flight number vs. Orbit type
- GTO and LEO missions dominate the early flight numbers, while specialized orbits like VLEO and SSO appear predominantly in later flights, suggesting an evolution from basic orbital capabilities to more advanced mission profiles over time.

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type:
- Heavy payloads (above 10,000 kg) are exclusively sent to LEO and VLEO orbits, while higher-energy orbits like GTO and SSO are limited to lighter payloads under 6,000 kg due to the energy requirements of reaching those destinations.

Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- Success rates improved dramatically from 0% in early years to peak at ~90% by 2019, showing the typical learning curve of a maturing launch system with some volatility as mission complexity increased.



All Launch Site Names

Find the names of the unique launch sites

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

[12]
```
%sql select DISTINCT(Launch_Site) from SPACEXTBL
```
Python

```
* 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'

Find 5 records where launch sites begin with `CCA`

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

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

[31]

Python

* 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

Calculate the total payload carried by boosters from NASA :

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

[34] Python

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

Total_payload_mass
45596

Average Payload Mass by F9 v1.1

Calculate the average payload mass carried by booster version F9 v1.1

Task 4

Display average payload mass carried by booster version F9 v1.1

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

[35] Python

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

average_playload_mass
2928.4

First Successful Ground Landing Date

Find the dates of the firs : t successful landing outcome on ground pad

```
%sql SELECT MIN(date) AS first_successful_ground_pad_landing from  SPACEXTBL WHERE  Mission_Outcome = 'Success'  AND landing_
```

[53] Python

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

first_successful_ground_pad_landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql select Booster_Version as list_boosters from SPACEXTBL where landing_outcome like 'Success (drone ship)' and   PAYLOAD_M
```

[50] Python

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

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

Total Number of Successful and Failure Mission Outcomes

Calculate the total number of successful and failure mission outcome

```
[65]  %sql select count(*) as Total,Mission_Outcome   from SPACEXTBL GROUP BY Mission_Outcome
                                                                                        Python
```

* sqlite:///my_data1.db
Done.

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

```
[59]  %sql SELECT CASE WHEN Mission_Outcome LIKE 'Success%' THEN 'Success' WHEN Mission_Outcome LIKE 'Failure%' THEN 'Failure' ELSE
                                                                                        Python
```

* sqlite:///my_data1.db
Done.

Outcome_Category	total
Failure	1
Success	100

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

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

Python

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

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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql select substr(Date,6,2) as months ,Booster_Version,Launch_Site,Landing_Outcome from SPACEXTBL where Landing_Outcome = 'Failure (drone ship)'
```

[70] Python

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

months	Booster_Version	Launch_Site	Landing_Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)
01	F9 v1.1 B1017	VAFB SLC-4E	Failure (drone ship)
03	F9 FT B1020	CCAFS LC-40	Failure (drone ship)
06	F9 FT B1024	CCAFS LC-40	Failure (drone ship)

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

- 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

Task 10

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.

```
%sql select count(*) as count_of_landing from SPACEXTBL where (Landing_Outcome like 'Failure (drone ship)' or landing_outcome like 'Success (ground pad)') and Date between '2010-
```

Python

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

count_of_landing
5
3

Section 3

Launch Sites Proximities Analysis

Folium Map: Launch Sites

- Launch sites are at the East and West coast, near the southernmost U.S. mainland area, which is Florida and California

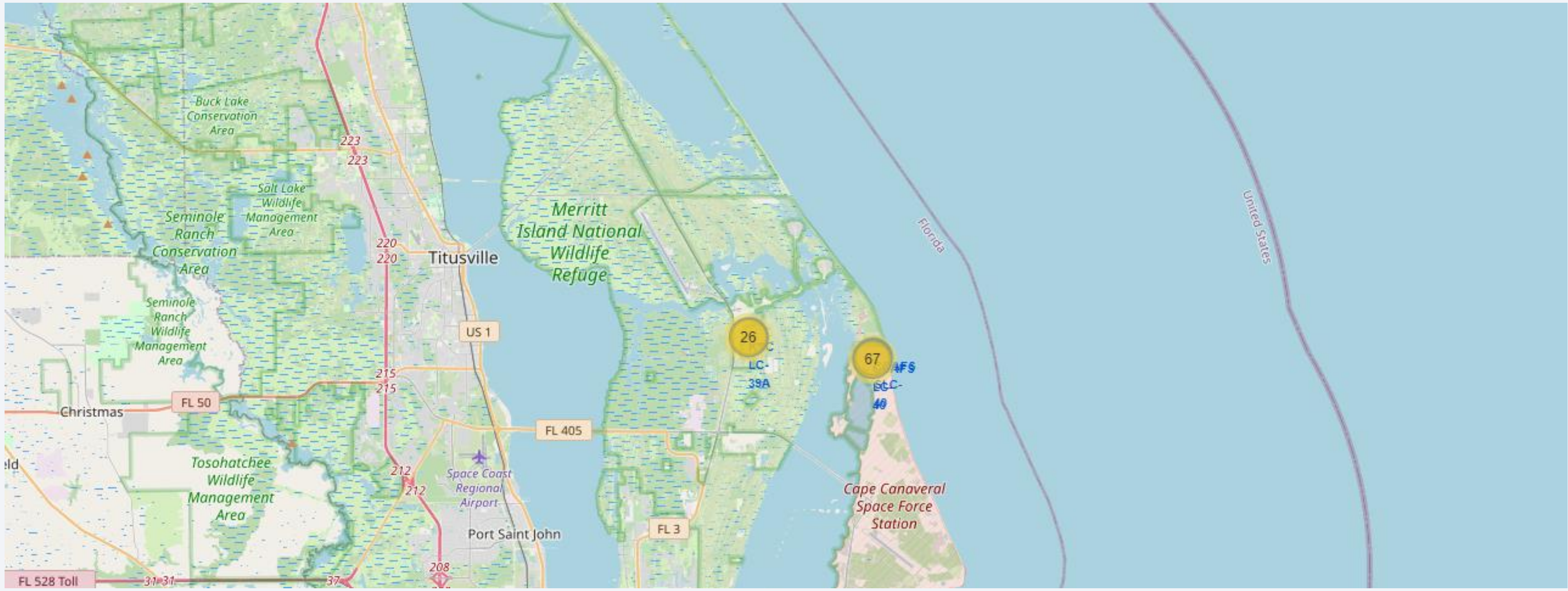
Folium Map: Proximity Vandenburg AFB

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Close to the Vandenburg AFB is the town of Lompoc. This might be an issue, if the stage-1 landing cannot be controlled, since rockets would usually start towards Eastern direction
- Explain the important elements and findings on the screenshot :

Folium Map: Proximity Kennedy SC / Cape Canaveral

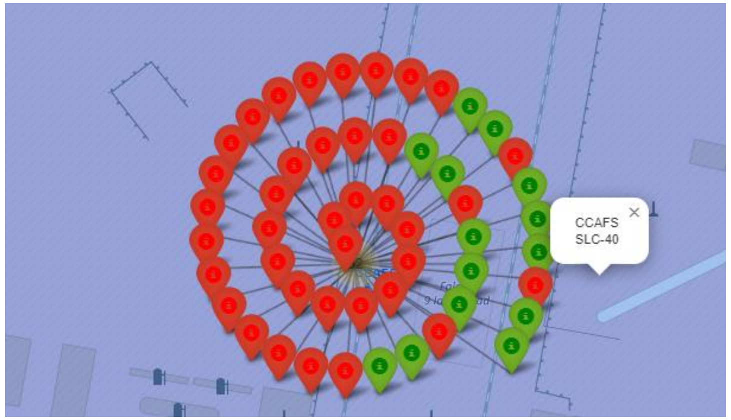
Explain the important elements and findings on the screenshot

- No city towards the Eastern Direction, ideal place for testing rocket launches

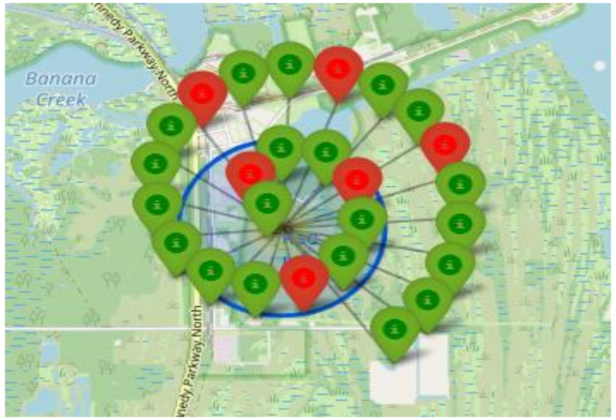


Folium Map:Stage-1 Landing Success by Launch Site

Vandenburg AFB



Cape Canaveral



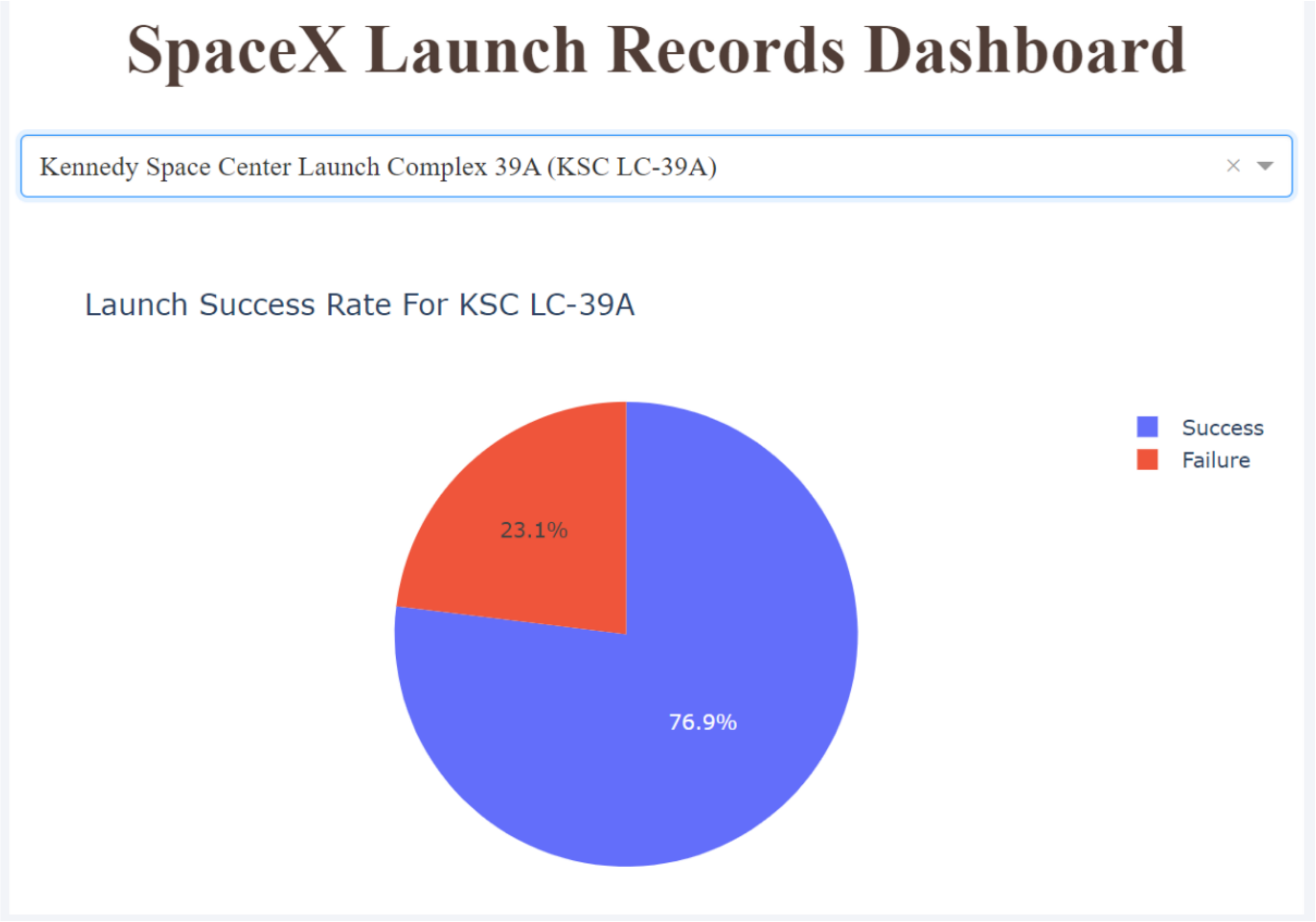
Kennedy Space Center

Section 4

Build a Dashboard with Plotly Dash

Dashboard: Success Rate Kennedy Space Center

- More than 3 of 4 landings have been successful at Kennedy Space Center

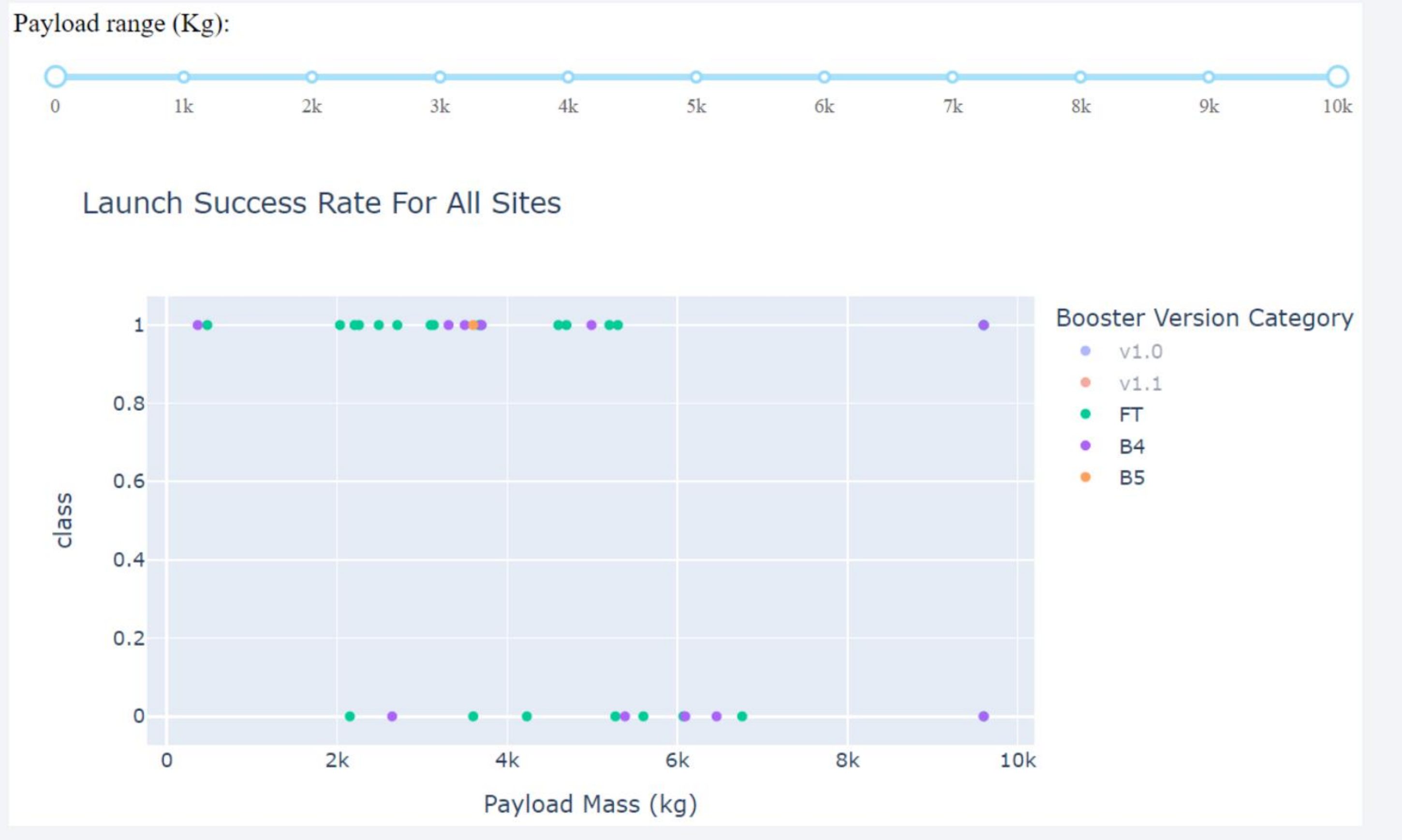


Dashboard: Booster Versions V1.0, V1.1

- Success rate for Booster versions v1.0 and v1.1 is quite small in the payload range to 10000kg

Dashboard: Booster Versions V1.0, V1.1

- Success rate for Booster versions FT, B4 and B5 is better in the payload range to 10000kg

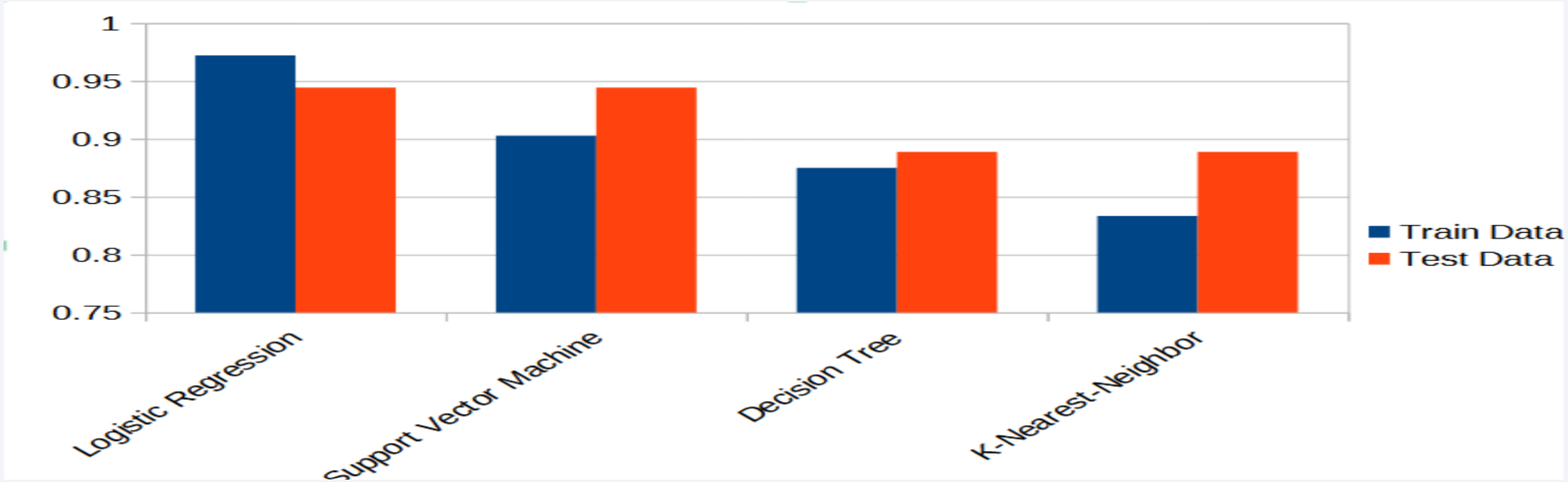


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Logistic Regression has the best result for train data
- Logistic Regression and Support Vector Machines have the best results on test data



Confusion Matrix

- True Positives: 12
- True Negatives: 5
- False Positives: 1
- False Negatives: 0

Conclusions

- Prediction with Logistic Regression is quite accurate
- Support Vector Machine also provide a good result for predicting the landing outcome
- None of the models had false negatives
- All models had at least one false positive

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

- Please visit this link below for all notebook used for this project :

- hammoumi123/ds_IBM_certificate

Thank you!