Winning Space Race with Data Science

Supriya Jagdale

21/07/2024

Outline

- Executive summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

Summary of methodologies

- Data was collected from the SpaceX public API and publicly available data on Wikipedia.
- Data wrangling included extracting launch outcome information to serve as the dependent variable in the Machine Learning models.
- SQL queries and data visualizations (static plots and interactive maps) were created for exploratory analysis.
- Predictive analysis was performed using Classification algorithms like Logistic Regression, SVM (Support Vector Machine), Decision Tree, and KNN (k-Nearest Neighbors) Machine Learning models.

Summary of all results

- Launch data includes info about flight number, date of launch, payload mass, orbit type, launch site, mission outcome and other variables.
- Logistic Regression, SVM (Support Vector Machine), Decision Tree and KNN (k-Nearest Neighbors) all performed equally well for Machine Learning models on this dataset.

Introduction

- SpaceX offers Falcon 9 launches for 62 million dollars, significantly cheaper than competitors' 165 million dollars, due to first-stage reusability. Predicting successful landings can help estimate launch costs, providing valuable insights for companies competing with SpaceX. This project aims to develop a machine learning pipeline for this prediction.

- **Key questions include factors influencing landing success, cost differences**, interactive data visualization, and **the accuracy of predictive models.** Through detailed data collection, exploratory data analysis, and predictive modeling, this project aims to provide valuable insights and a robust predictive framework for SpaceX's launch operations.

Section 1

Methodology

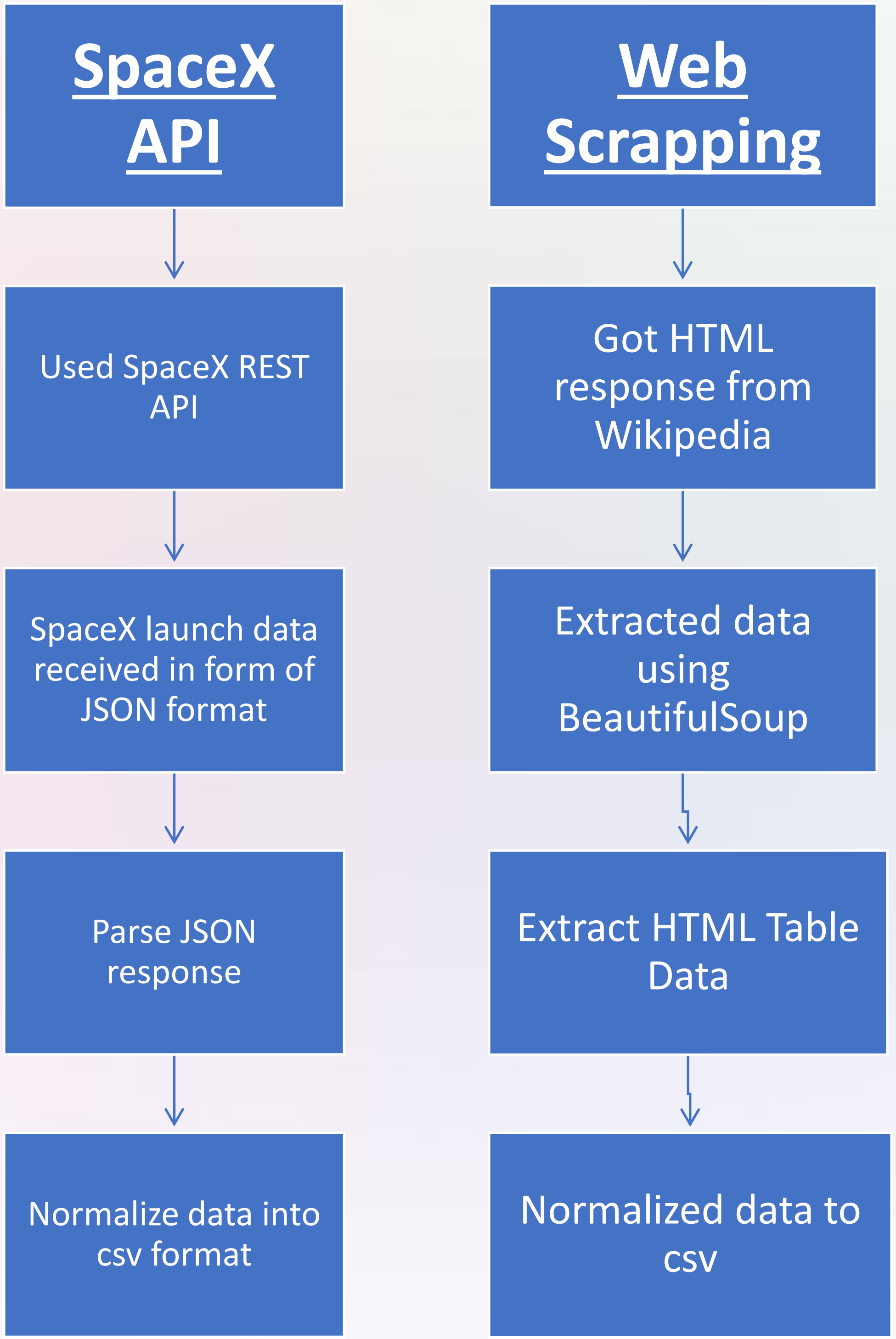
Methodology

- SpaceX REST API and Wikipedia launch table data was collected.
- Data was cleaned in preparation by handling null values and filtering data.
- Exploratory data analysis (EDA) was done using visualization by Pandas and Matplotlib and SQL.
- Interactive visual analytics were created using Folium and Plotly Dash.
- Predictive analysis using classification models was done.

Data Collection

- Data collection was done using get request to the SpaceX API.
- Next, we got the response content as a Json using .json() function call and turn it into and turn it into a pandas dataframe using .json_normalize().
- We then cleaned the data, checked for missing values and fill in missing values values where necessary.

- In addition, we performed web scraping from Wikipedia for Falcon 9 launch records launch records with BeautifulSoup.
- The objective was to extract the launch records as HTML table, parse the table and table and
- convert it to a pandas dataframe for future analysis.



Data collection-SpaceX API

1. Requested SpaceX Launch Data

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

2. Parsed JSON Response

```
# Use json_normalize meethod to convert the json result into a dataframe
response_json=response.json()
data=pd.json_normalize(response_json)
```

3. Data Preprocessing

```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a single
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

4. Requested SpaceX Launch Data

```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []

getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

5. Created Final DataFrame

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}

df_launch = pd.DataFrame(launch_dict)
print(df_launch)
```

6. Filtered for Falcon 9 Launches

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df_launch[df_launch['BoosterVersion'] != 'Falcon 1']
```

GitHub link:
https://github.com/suppijags/IBM-assignment/blob/main/1.%20jupyter-labs-spacex-data-collection-api.ipynb

Data collection-Web Scraping

1. Requested Falcon 9 Launch Wiki Page

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
page = requests.get(static_url)
```

2. Created BeautifulSoup Object

```
soup = BeautifulSoup(page.content, "html.parser")
```

3. Extracted HTML Table Headers from the <th></th> tags

```
html_tables=soup.find_all('table')
html_tables
```

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
for th in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

4. Created Data Dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

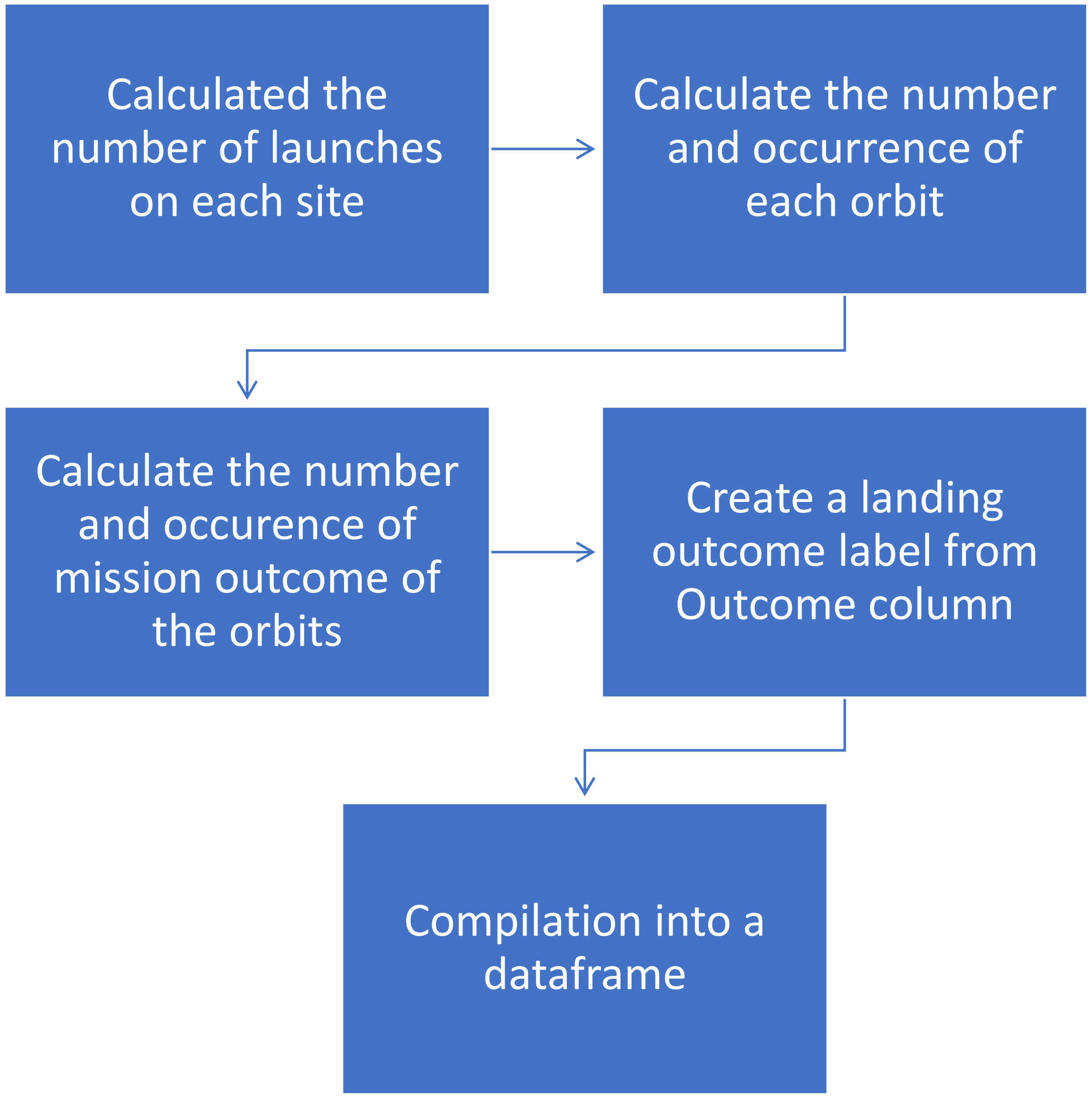
5. Converted Dictionary to DataFrame:

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

GitHub link: https://github.com/suppijags/IBM-assignment/blob/main/2.%20jupyter-labs-webscraping.ipynb

Data wrangling

- Handled null values by replacing Null values
- We created features from API Data by extracting detailed data for Booster, Launchpad, payload, and core to enhance the dataset.
- The .csv file from the first section contained data that required cleaning.
- The launch sites, orbit types, and mission outcomes were cleaned and standardized.
- Various mission outcome types were simplified into a binary classification: 1 indicating a successful Falcon 9 first stage landing, and 0 indicating a failure.
- GitHub link: https://github.com/suppijags/IBM-assignment/blob/main/3.%20labs-jupyter-spacex-Data%20wrangling.ipynb



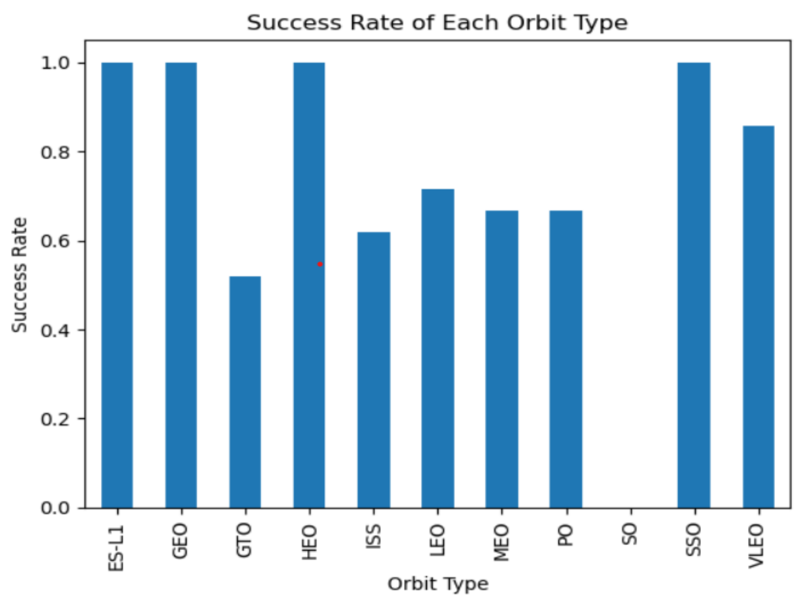
EDA with Data Visualization

- We performed Exploratory Data Analysis (EDA) with visualization and Feature Engineering using **Pandas and Matplotlib** to analyze space mission data.

- We visualized the relationships between various parameters, such as launch sites, payload masses, Orbit type, to compare sets of data between different groups at glance by using Scatterplots and Bar plots.

- We plotted a line plot for to see yearly trend of launch successes to observe improvements over time, highlighting specific years with higher success rates.

- GitHub link: https://github.com/suppijags/IBM-assignment/blob/main/4.%20jupyter-labs-eda-dataviz.ipynb



EDA with SQL

We loaded the SpaceX dataset into a PostgreSQL database in Jupyter notebook.

We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:

- Display the names of the unique launch sites in the space mission
- 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 have carried the maximum payload mass. Use a subquery
- 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.
- 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/suppijags/IBM-assignment/blob/main/5.%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We identified and marked all launch sites on the folium map, adding various map objects such as markers, circles, and lines to indicate the success or failure of launches at each site.

- We categorized the launch outcomes into two classes: 0 for failure and 1 for success.

- By using color-labeled marker clusters, we determined which launch sites had relatively high success rates.

- We measured the distances from each launch site to nearby features, such as railways, highways, and coastlines, and addressed questions like:
  - Are launch sites located near railways, highways, and coastlines?
  - Do launch sites maintain a certain distance from cities?
- GitHub link: https://github.com/suppijags/IBM-assignment/blob/main/6.lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

We developed an interactive dashboard using Plotly Dash:

•Constructed pie chart to display the total launches at specific sites and determine launch site with highest Launch Success Ratio

•We generated scatter plots to illustrate the relationship between Outcome and Payload Mass (Kg) for different booster versions.

•GitHub link: https://github.com/suppijags/IBM-assignment/blob/main/Spacex_dash.py

Predictive Analysis (Classification)

1 Preprocessing

We standardized the data to ensure consistent scaling followed by Splitting the data into training and testing sets to evaluate model performance.

2 Model Training

We trained various classification models, such as Logistic Regression, Logistic Regression, Logistic Regression, Support Vector Machines, Decision Tree Tree Classifier, and K-nearest Neighbors. to predict launch success.

3 Model Evaluation

We evaluated the model performance using metrics like accuracy and confusion metrics to ensure robust predictions We evaluated the best hyperparameter values obtained through Grid Search .

Predictive Analysis (Classification)

Flowchart of Machine learning

Dataframe was created with cleaned data → Data was split into testing and training datasets → Training dataset was used for training different data models → Each model was evaluated on testing data set → Accuracy scores were calculated to compare each model’s performance

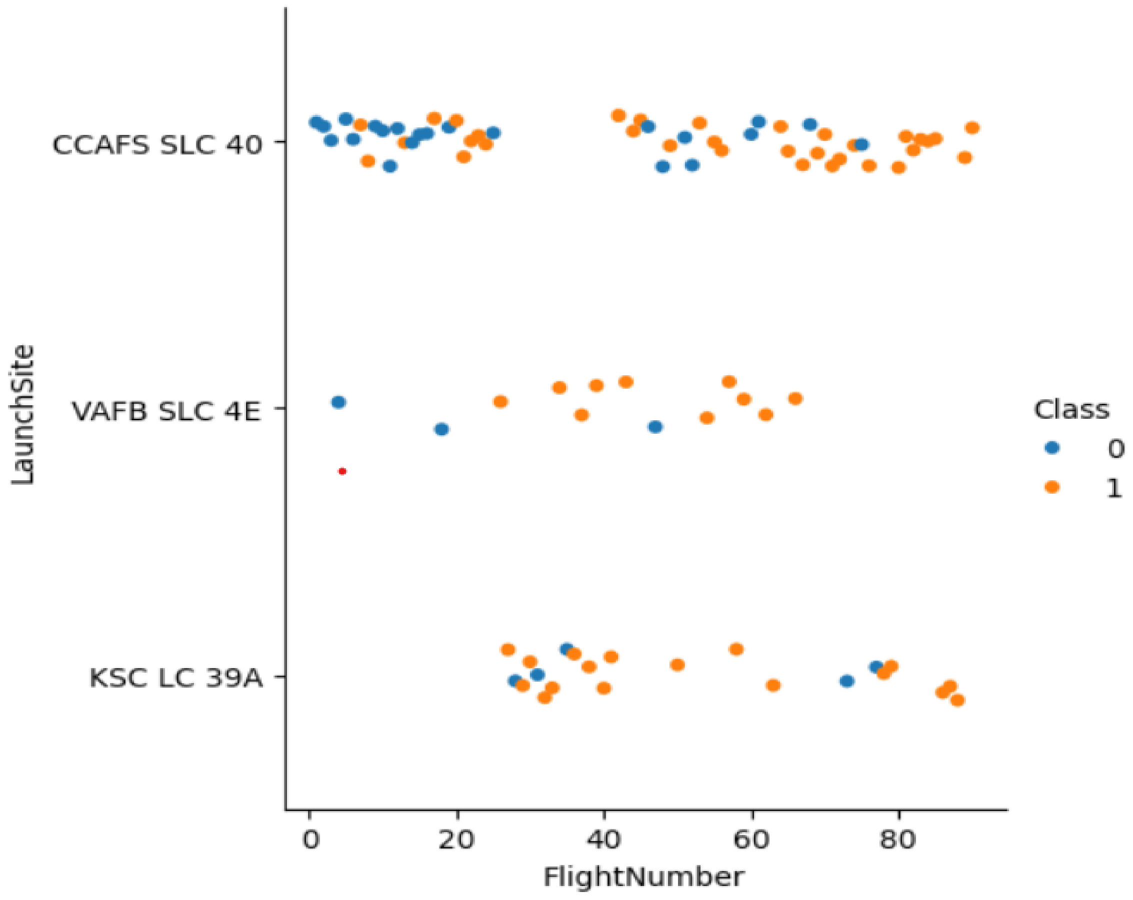
GitHub link: https://github.com/suppijags/IBM-assignment/blob/main/7.SpaceX_Machine%20Learning%20Prediction.ipynb

Section 2

Insights drawn from EDA

EDA with Visualization

1 Flight Number vs. Launch Site

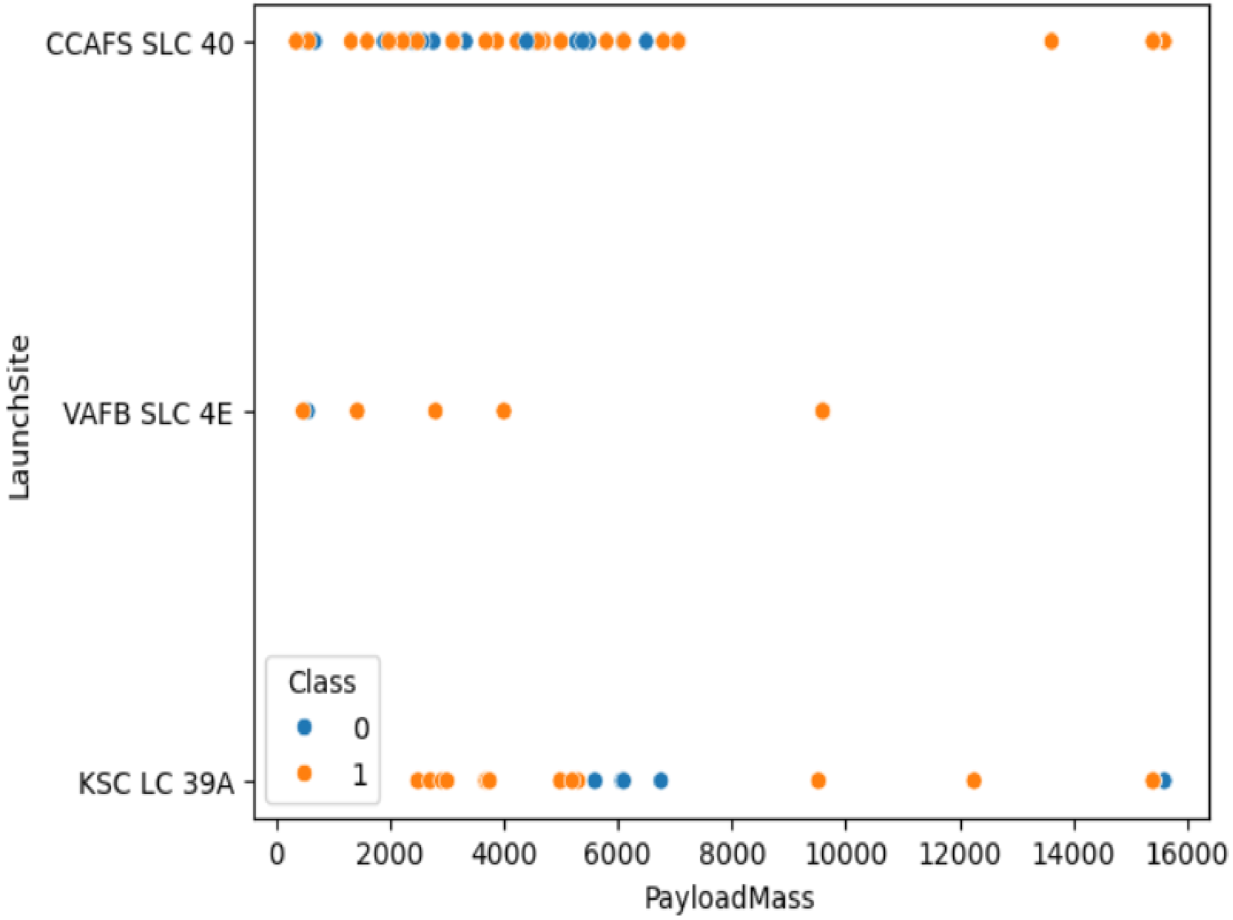


- Across all launch sites, there is a general trend of higher flight numbers being associated with a higher rate of successful landings.
- This suggests that as more flights are conducted, the likelihood of a successful landing increases, likely due to improvements in technology, experience, and operational procedures over time.
- The scatter plot also indicates that certain launch sites may have varying success rates, possibly due to different conditions or operational challenges specific to each site.

EDA with Visualization

2

Payload vs. Launch Site

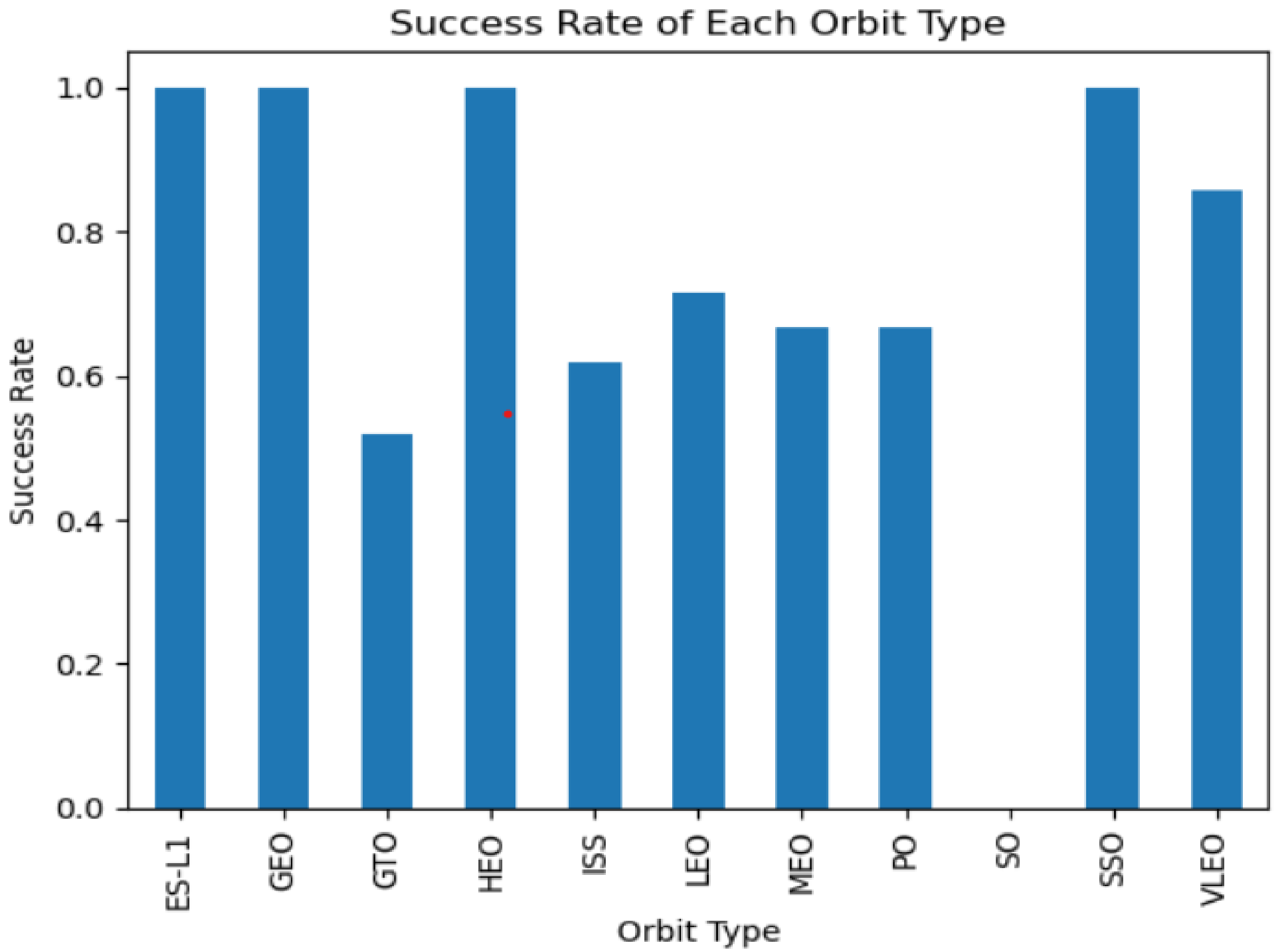


- VAFB SLC 4E appears to handle only lower payload masses compared to CCAFS SLC 40 and KSC LC 39A, which manage a wide range including very heavy payloads.
- Both CCAFS SLC 40 and KSC LC 39A show successful landings with heavy payloads (>10,000 kg), suggesting better capability or more experience with heavy payload missions.
- CCAFS SLC 40 and KSC LC

EDA with Visualization

3

Success Rates of Different Orbit Type



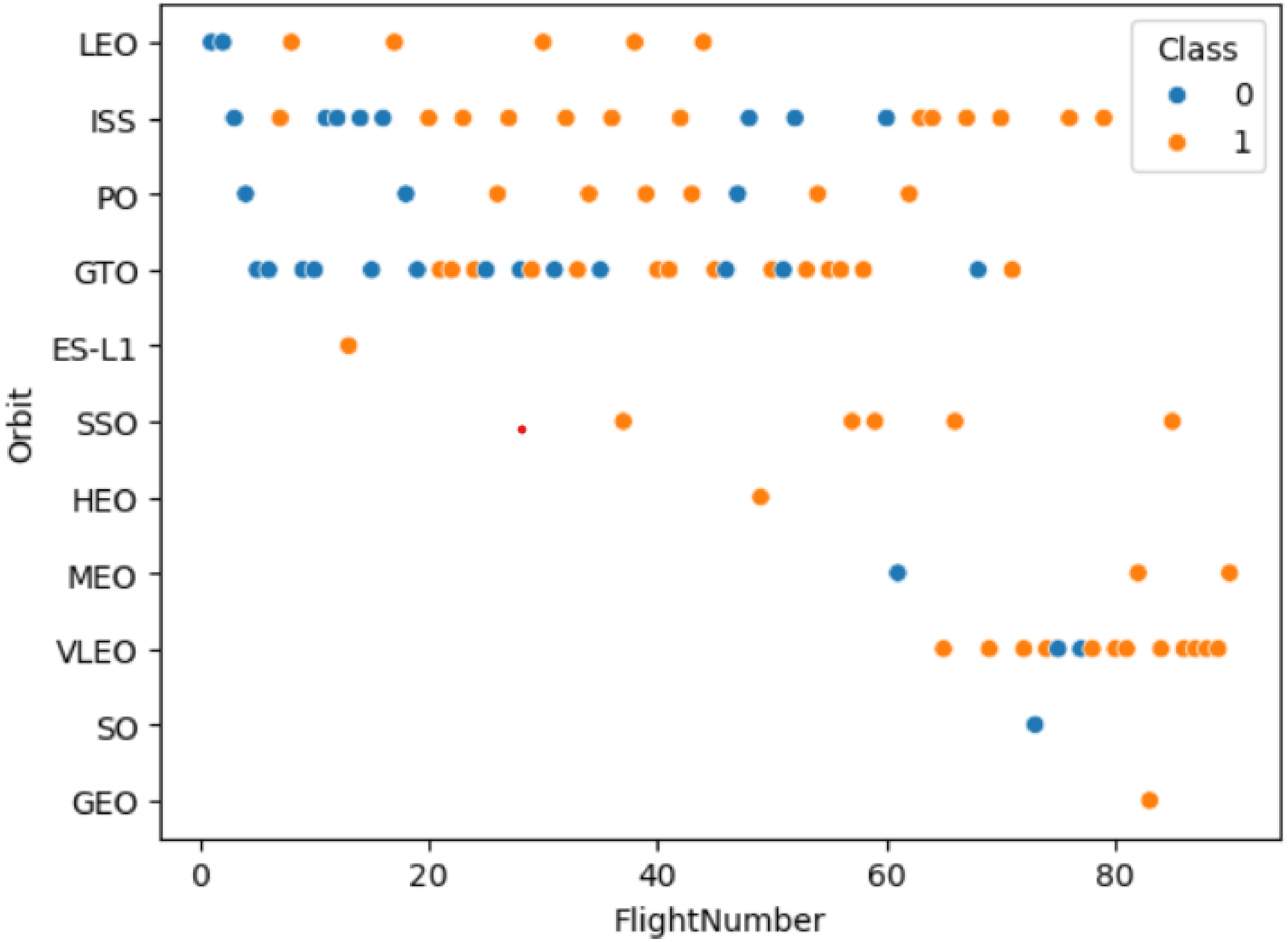
•Orbits such as ES-L1, GEO, SSO, and VLEO have notably high success rates, making them more reliable for missions.

•Moderate success rates are observed in orbits like LEO and ISS, indicating potential areas for improvement.

•GTO and HEO have lower success rates, suggesting these orbits might present more challenges or require additional focus for successful missions.

EDA with Visualization

4 Landing Success Across Different Orbits



•LEO and, to some extent, ISS orbits demonstrate that increased flight numbers correlate with higher success rates, reflecting the positive impact of experience and iterative improvements.

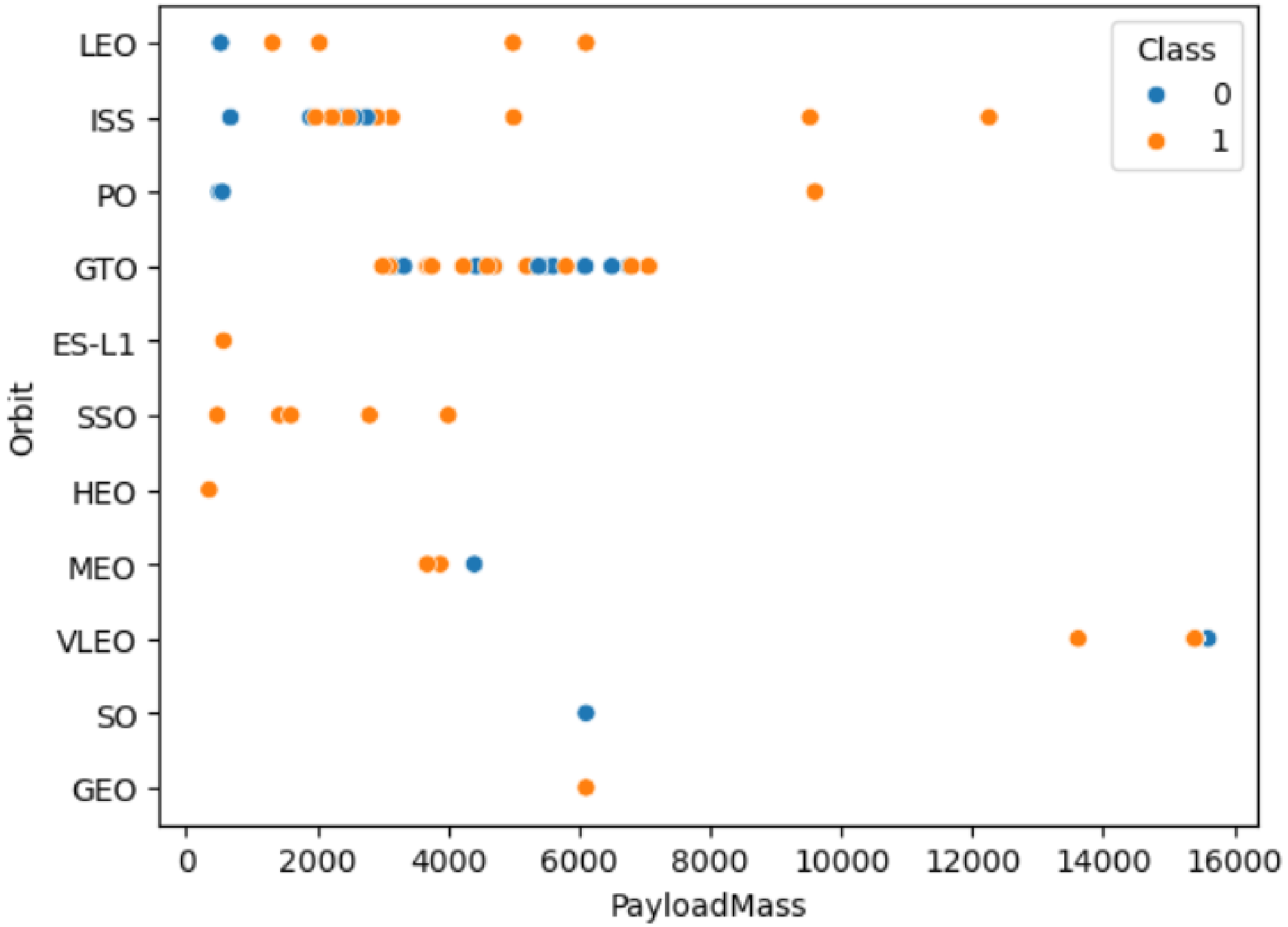
•The lack of a clear pattern in GTO suggests that the inherent difficulty of achieving successful landings in this orbit might overshadow the benefits gained from increased flight experience.

•Some orbits inherently show high success rates regardless of flight number, possibly due to their lower complexity or more mature technology used for these missions.

EDA with Visualization

5

Impact of Payload Mass on Landing Success across different orbit type

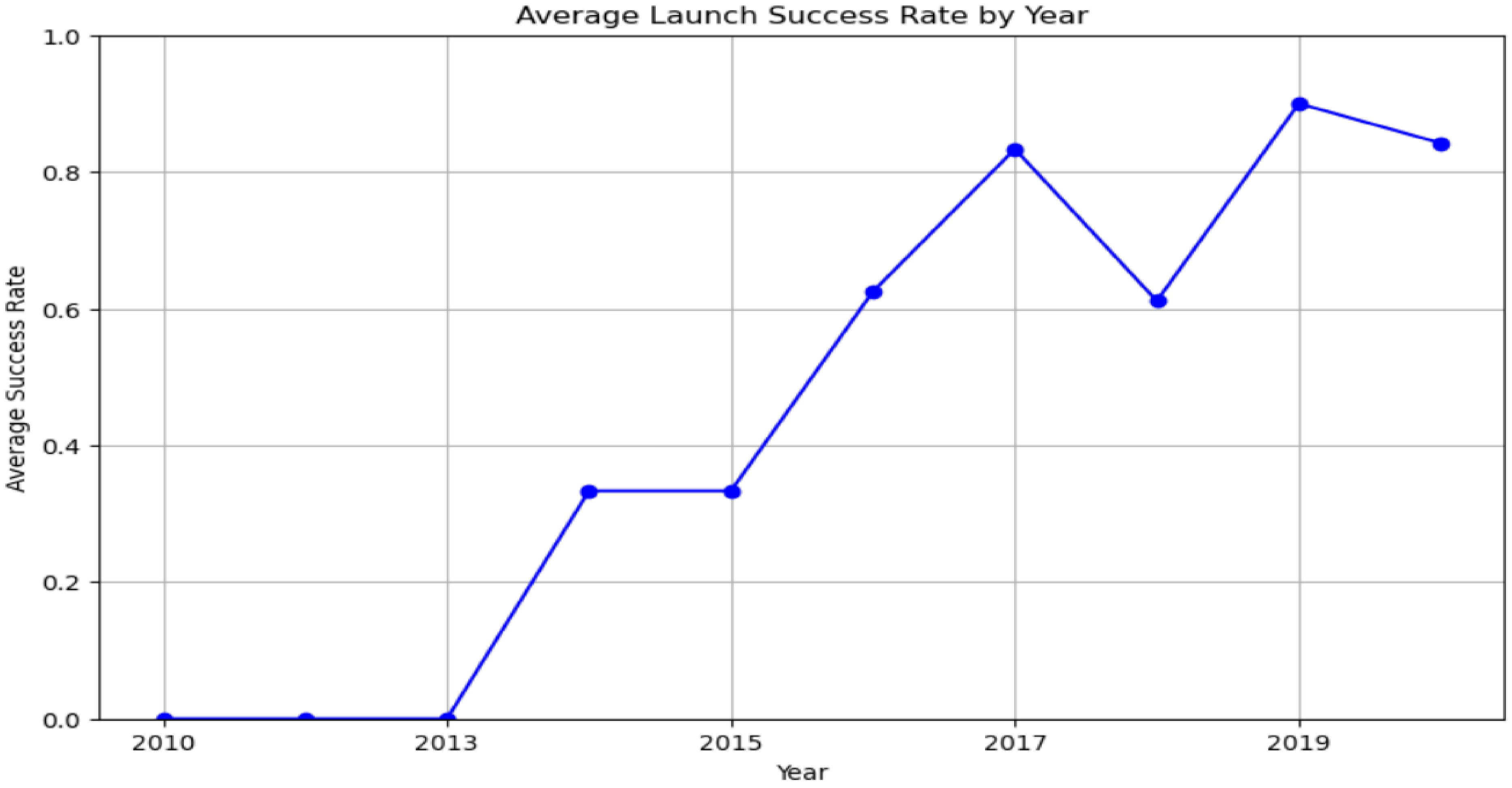


- **Polar (PO), LEO, and ISS Orbits**: These orbits demonstrate a more favorable trend for successful landings with heavier payloads, which could be attributed to better experience, technology, or conditions suited for these missions.
- **GTO Orbit**: The inconsistency in landing outcomes, irrespective of payload mass, indicates that factors other than payload weight, such as mission complexity or technical challenges, play a significant role in determining success rates for GTO missions.

EDA with Visualization

6

Falcon 9 Launch Success Rates (2010-2020)



The overall trend indicates a significant improvement in the success rate of Falcon 9 launches over the years. Starting from a low base, the success rate has steadily increased, peaking at around 80% in recent years.

EDA with SQL

Task 1: Display the names of the unique launch sites in the space mission

Query

```
%%sql
SELECT DISTINCT "Launch_Site"
FROM SPACEXTBL;
```

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

Result

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

Explanation: Using the word DISTINCT in the query means that it will only show Unique values in the Launch_Site column from SPACEXTABL. There are 4 unique launch sites.

EDA with SQL

Task 2: Display 5 records where launch sites begin with the string 'CCA'

Query

```
%%sql
SELECT *
FROM SPACEXTBL
WHERE "Launch_Site" LIKE 'CCA%'
LIMIT 5;
```

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

Result

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

Explanation: Using the 'LIMIT 5' in the query means that it will only show 5 records from SPACEXTABL and LIKE keyword has a wild card with the words 'CCA%' the percentage in the end suggests that the Launch_Site name must start with CCA.

EDA with SQL

Task 3: Display the total payload mass carried by boosters launched by NASA (CRS)

Query

```
%%sql
SELECT SUM("PAYLOAD_MASS__KG_") AS "Total_Payload_Mass"
FROM SPACEXTBL
WHERE "Customer" LIKE '%NASA (CRS)';
```

Result

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

Total_Payload_Mass
45596

Explanation: Using the function SUM summates the total in the column PAYLOAD_MASS_KG_ The WHERE clause filters the dataset to only perform calculations on Customer NASA (CRS).

The sum is 45596 Kg.

EDA with SQL

Task 4: Display average payload mass carried by booster version F9 v1.1

Query

```
%%sql
SELECT AVG("PAYLOAD_MASS__KG_") AS "Average_Payload_Mass"
FROM SPACEXTBL
WHERE "Booster_Version" = 'F9 v1.1';
```

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

Result

Average_Payload_Mass
2928.4

Explanation: Using the function AVG works out the average in the column PAYLOAD_MASS_KG_ The WHERE clause filters the dataset to only perform calculations on Booster_version F9 v1.1

EDA with SQL

Task 5: List the date when the first successful landing outcome in ground pad was achieved.

Query

```
%%sql
SELECT MIN("Date") AS "First_Successful_Landing_Date"
FROM SPACEXTBL
WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

Result

First_Successful_Landing_Date
2015-12-22

Explanation: Using the function MIN works out the minimum date in the column Date. The WHERE clause filters the dataset to only perform calculations on Landing_Outcome Success (drone ship)

EDA with SQL

Task 6: List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

Query

```
%%sql
SELECT "Booster_Version"
FROM SPACEXTBL
WHERE "Landing_Outcome" = 'Success (drone ship)'
AND "PAYLOAD_MASS__KG_" > 4000
AND "PAYLOAD_MASS__KG_" < 6000;
```

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

Result

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

Explanation: Selecting only Booster_Version. The WHERE clause filters the dataset to Landing_Outcome =Success (drone ship). The AND clause specifies additional filter conditions Payload_MASS_KG_ > 4000 AND Payload_MASS_KG_ < 6000.

EDA with SQL

Task 7: List the total number of successful and failure mission outcomes

Query

```
%%sql
SELECT "Mission_Outcome", COUNT(*) AS "Total_Count"
FROM SPACEXTBL
GROUP BY "Mission_Outcome";
```

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

Result

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

Explanation: The SQL query retrieves the count of each unique Mission_Outcome from the SPACEXTBL table. It groups the data by the Mission_Outcome field and counts how many times each outcome appears, labeling the counts as Total_Count. The result shows the number of number of successful and failure mission outcomes in the dataset.

EDA with SQL

Task 8: List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

Query

```
%%sql
SELECT "Booster_Version"
FROM SPACEXTBL
WHERE "PAYLOAD_MASS__KG_" = (
    SELECT MAX("PAYLOAD_MASS__KG_")
    FROM SPACEXTBL
    );
```

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

Result

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

Explanation: This SQL query retrieves the Booster_Version from the SPACEXTBL table where the PAYLOAD_MASS__KG_ is equal to the maximum payload mass in the table. The subquery SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL identifies the maximum payload mass, and the main query fetches all booster versions associated with this maximum payload mass. The result lists the booster versions corresponding to the heaviest payloads launched.

EDA with SQL

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.

Query

```
%%sql
SELECT
    substr("Date", 6, 2) AS "Month",
    "Landing_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM SPACEXTBL
WHERE "Landing_Outcome" = 'Failure (drone ship)'
AND substr("Date", 1, 4) = '2015';
```

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

Result

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

Explanation: The SQL query retrieves the average PAYLOAD_MASS__KG_ for each Rocket_Type from the SPACEXTBL table. It calculates the average payload mass for each type of rocket and returns the results with the Rocket_Type and its corresponding average payload mass. This helps in understanding the typical payload capacity associated with different rocket types.

EDA with SQL

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.

Query

```
%%sql
SELECT
    "Landing_Outcome",
    COUNT(*) AS "Outcome_Count"
FROM SPACEXTBL
WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY "Outcome_Count" DESC;
```

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

Result

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

Explanation: This SQL query counts the number of each Landing_Outcome in the SPACEXTBL table for the date range between '2010-06-04' and '2017-03-20’.

It groups the outcomes, orders them by the count in descending order, and displays the results, showing that "No attempt" had the highest count with 10 occurrences, followed by "Success (drone ship)" and "Failure (drone ship)" with 5 occurrences each.

Section 3

Launch Site Proximities Analysis

Interactive Map with Folium

Launch Sites

The Folium map displayed the locations of SpaceX's primary launch sites, along with key details about each site.

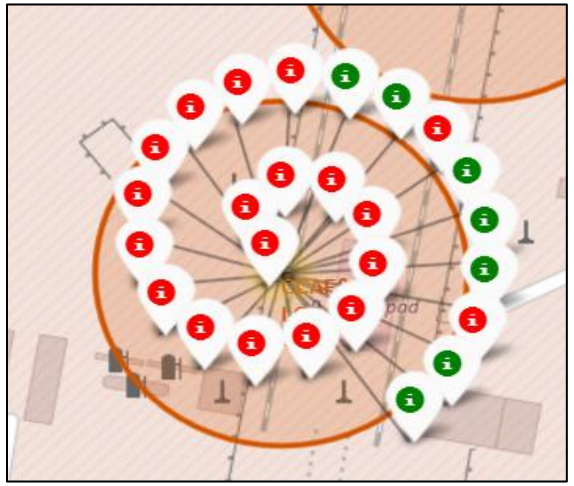
1. VAFB SLC-4E (California, USA)
Vandenberg Air Force Base Space Launch Complex 4E

2. KSC LC-39A (Florida, USA)
Kennedy Space Center Launch Complex 39A

3. CCAFS LC-40 (Florida, USA)
Cape Canaveral Air Force Station Launch Complex 40

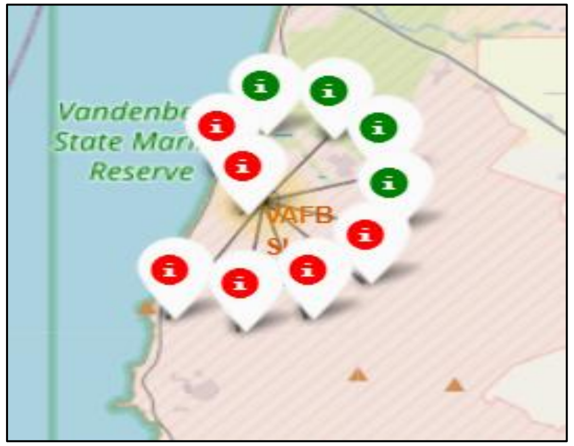
4. CCAFS SLC-40 (Florida, USA)
Cape Canaveral Air Force Station Space Launch Complex 40

Map Markers of Landings of Falcon 9

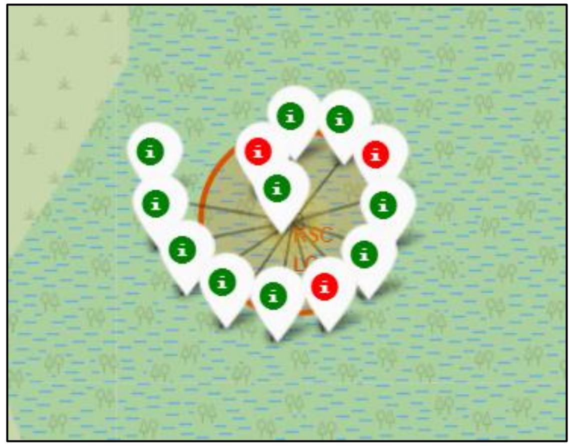


CCAFS LC-40
(Florida, USA)

CCAFS SLC-40
(Florida, USA)



VAFB SLC-4E
(California, USA)



KSC LC-39A
(Florida, USA)

Markers are clustered on the map according to the launch site co-ordinates.

The success rate of Falcon 9 first stage landings at a particular launch site can be inferred from the ratio of green success markers to red failure markers.

Launch Site Proximities Analysis

- **Are launch sites in close proximity to railways?**

No, as the distance between CCAFS LC-40 launch site and railways is 23.46km

- Are launch sites in close proximity to highways?

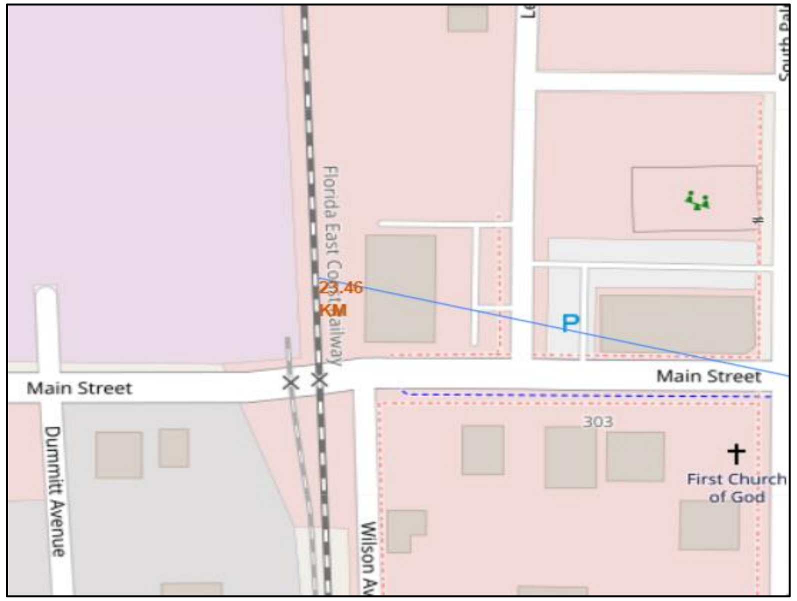
No, as the distance between CCAFS LC-40 launch site and highways is 26.95km

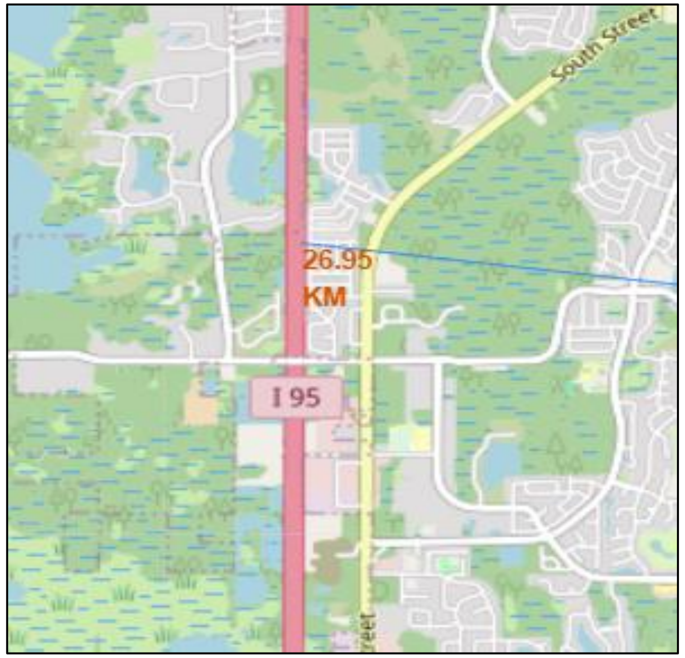
- Are launch sites in close proximity to coastline?

Yes, as the distance between CCAFS LC-40 launch site and coastline is only 0.94km

- Do launch sites keep certain distance away from cities?

Yes, CCAFS LC-40 launch site is 78.53km away from Orlando as per the map.

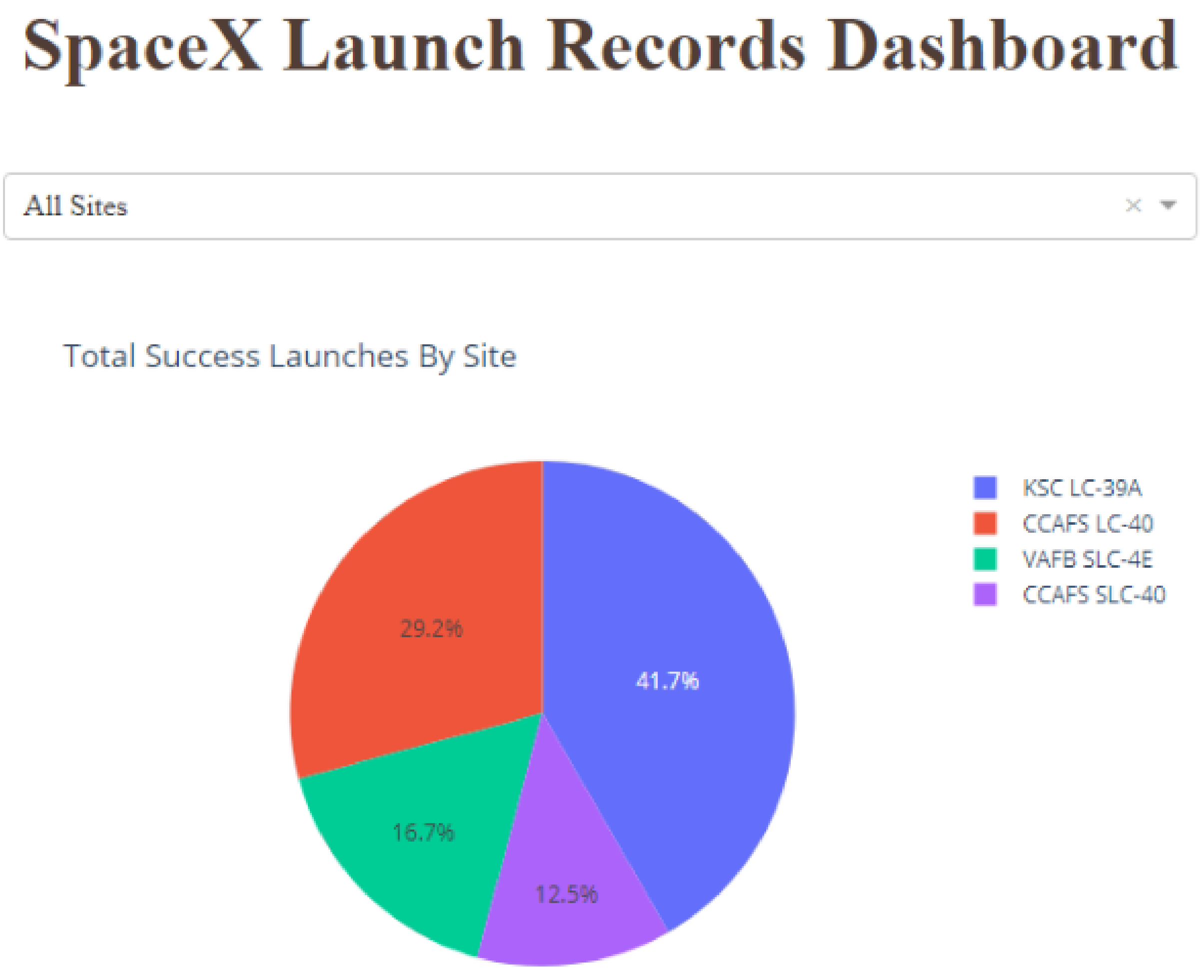




Section 4

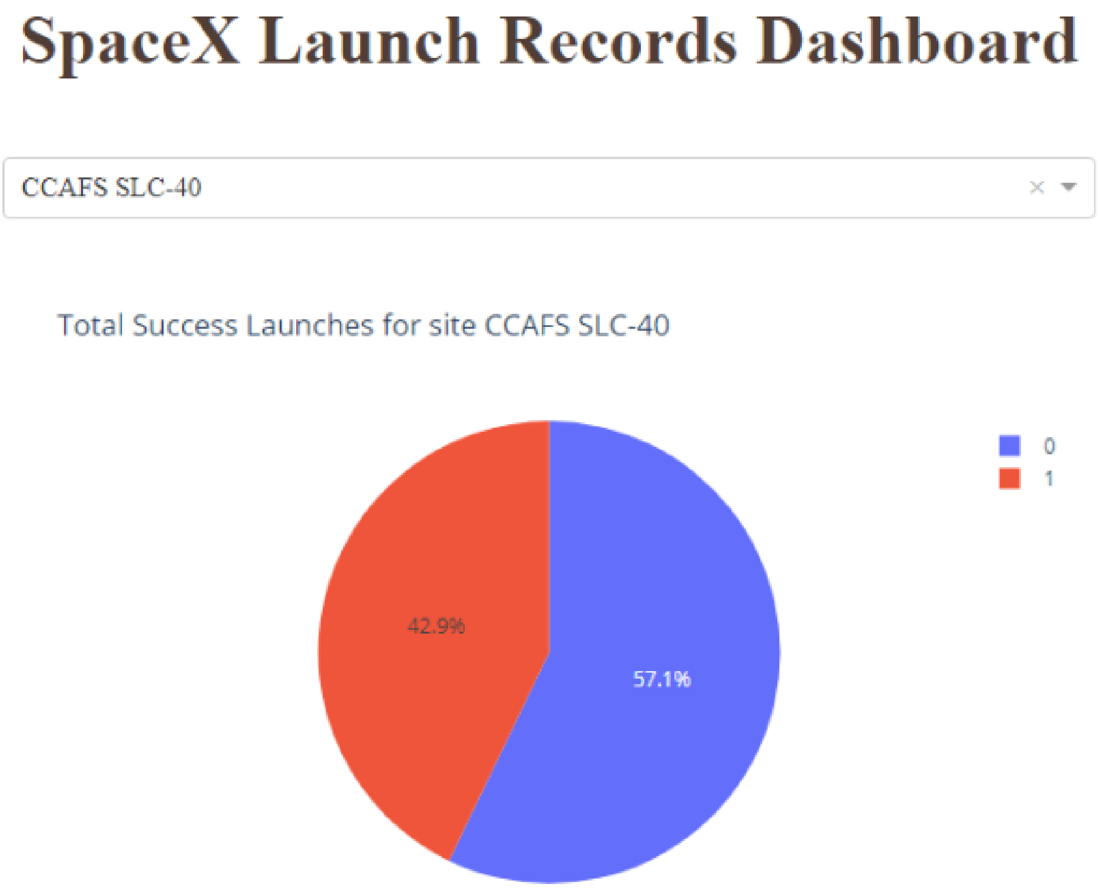
Build a dashboard with Plotly Dash

Plotly Dash Dashboard : Launch success count for all sites sites



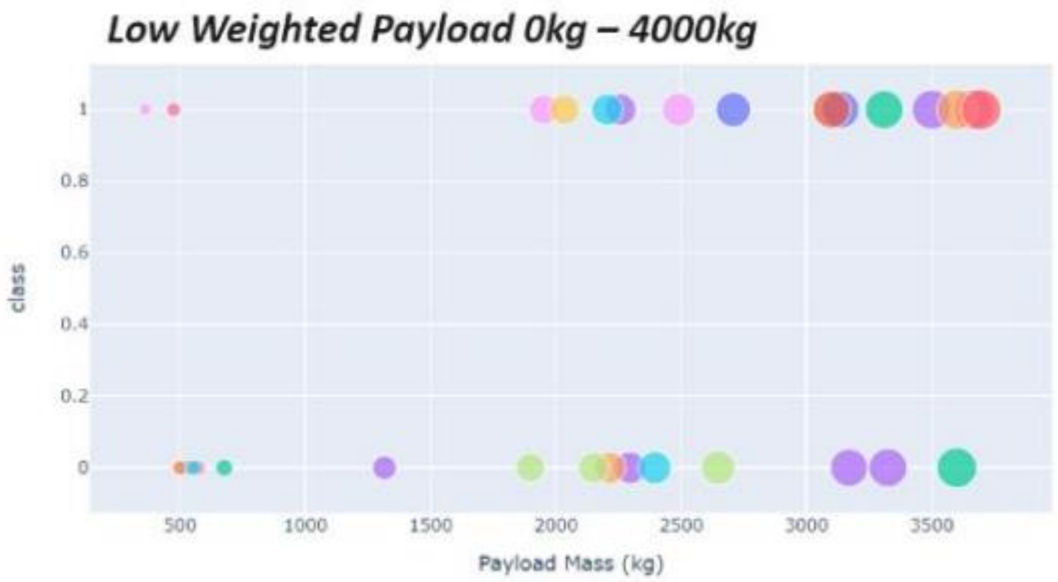
- The pie chart visually represents the success rates of launches from different sites, highlighting KSC LC-39A as the most successful launch site.
- This visual representation aids in quickly understanding the distribution of successful launches across the different sites.

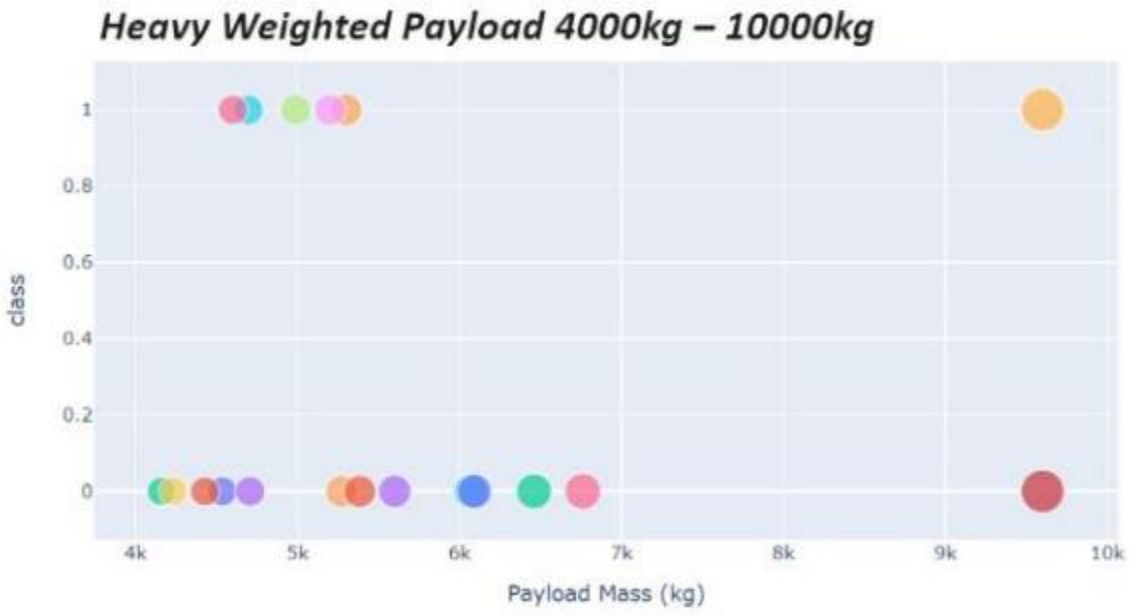
Plotly Dash Dashboard : Launch Site with Highest Launch Success Ratio



- The dashboard provides a quick and easy way to understand the performance of SpaceX launches at different sites.
- For CCAFS SLC-40, the success rate is 57.1%, as indicated by the pie chart.

Plotly Dash Dashboard : Payload vs Launch Outcome for selected range





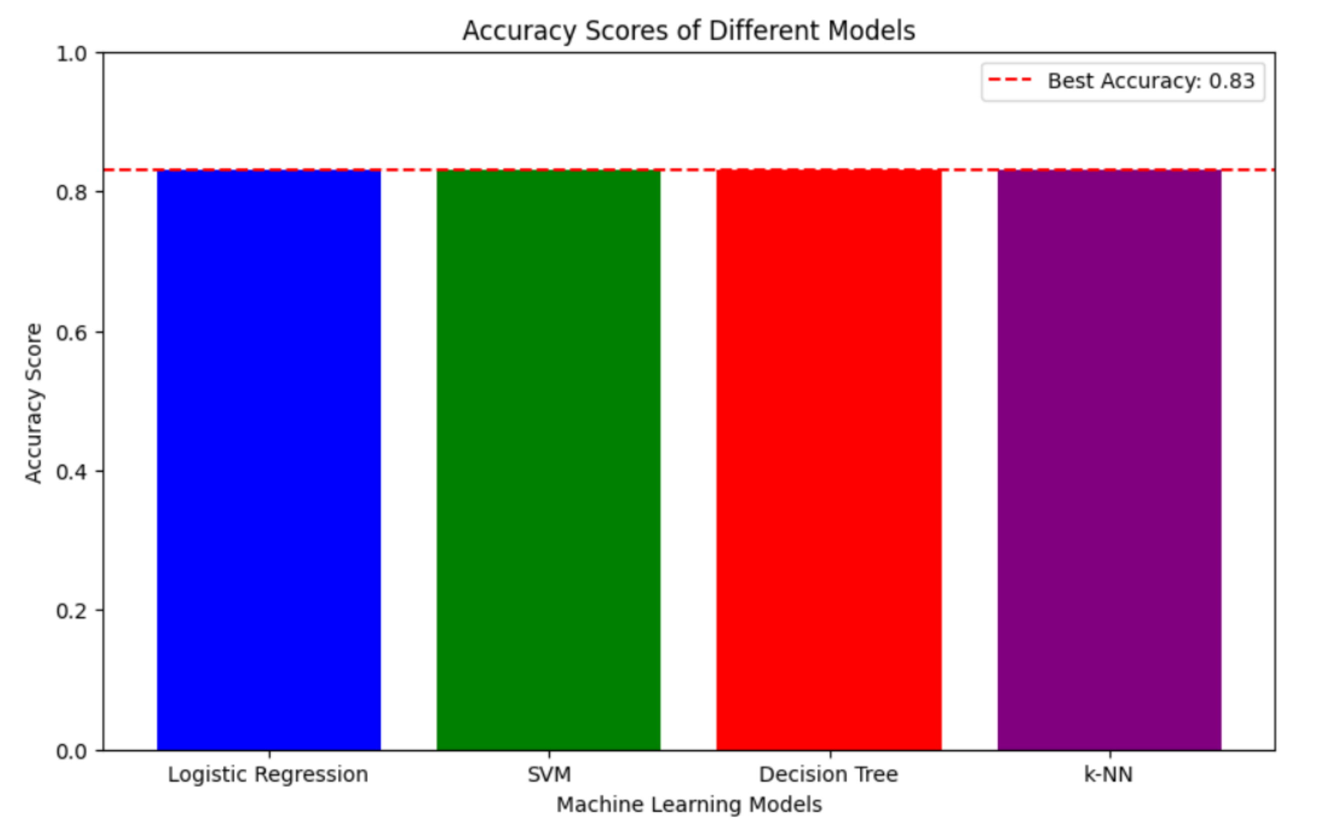
Success rate of launches is higher for low weighted payloads in comparison to heavy weighted payload.

Section 5

Predictive Analysis: Classification

Predictive Analysis (Classification)

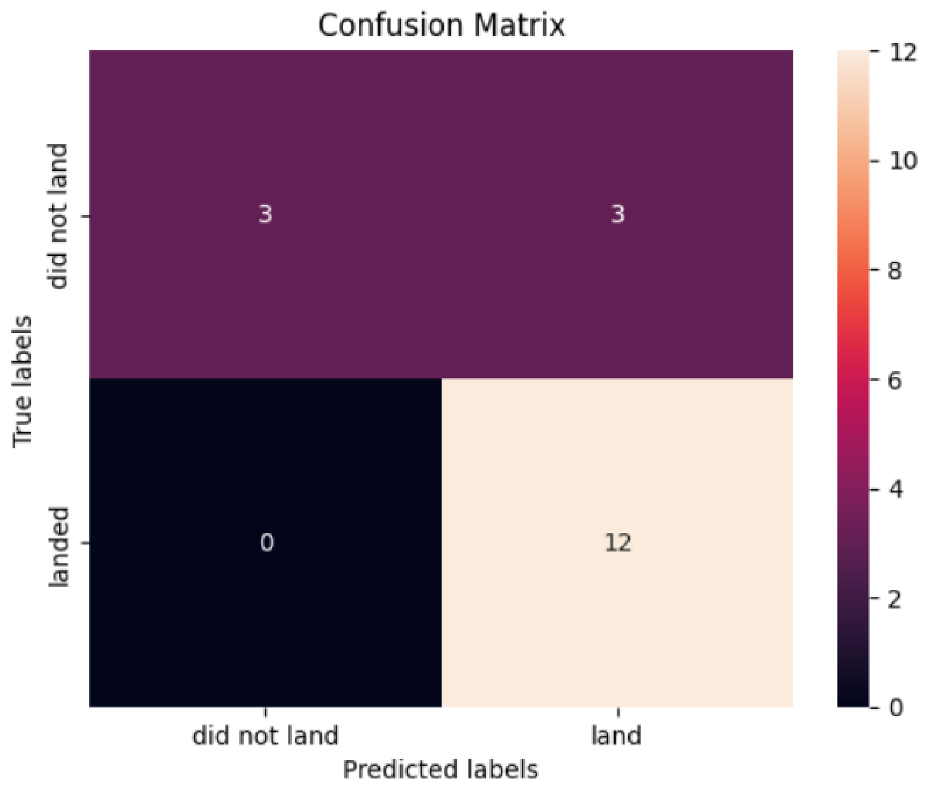
Model	Test set Accuracy	Precision	Recall	F1-Score	Best cross validated accuracy
Logistic Regression n	0.83	0.90	0.75	0.78	0.85
Decision Tree	0.83	0.90	0.75	0.78	0.89
Support Vector Machine	0.83	0.90	0.75	0.78	0.85
K-Nearest number	0.83	0.90	0.75	0.78	0.85



The predictive analysis results showed that all classification models had same accuracy, precision, recall, and F1-score in predicting the success or failure of SpaceX launches. So the final parameter used to decide the best model was Best cross validated accuracy which represents the average accuracy across the cross-validation folds and indicates how well the model is expected to perform on unseen data based on the training set. On the basis of best cross validated accuracy, the best model was found to be **Decision tree.**

Predictive Analysis -Confusion Matrix

- This confusion matrix shows that the model performs well in predicting landings (100% recall) but has some false positives (predicted landings that did not actually happen).
- The overall accuracy is 83%, and the model has a precision of 80% for predicting landings.
- The specificity is lower, at 50%, indicating that the model is less accurate in predicting non-landings.



Confusion Matrix Breakdown

True Negative	False positive
False Negative	True Positive

Conclusion

- The higher the number of flights at a launch site, the greater the success rate at that site.
- The launch success rate increased from 2013 to 2020.
- The orbits ES-L1, GEO, HEO, SSO, and VLEO had the highest success rates.
- KSC LC-39A had the most successful launches among all the sites.
- The Decision Tree classifier is the best machine learning algorithm for this task.

Appendix

DATA SETS

- SpaceX API (JSON): https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN SkillsNetwork/datasets/API_call_spacex_api.json
- Wikipedia (Webpage): https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
- SpaceX (CSV): https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321ENSkillsNetwork/labs/module_2/data/Spacex.csv?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetworkChannel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2022-01-01

Appendix

Jupyter Notebooks and Dashboard Python File

GitHub URL (Data Collection):https://github.com/suppijags/IBM-assignment/blob/main/1.%20jupyter-labs-spacex-data-collection-api.ipynb
GitHub URL (Web Scraping): https://github.com/suppijags/IBM-assignment/blob/main/2.%20jupyter-labs-webscraping.ipynb
GitHub URL (Data Wrangling): https://github.com/suppijags/IBM-assignment/blob/main/3.%20labs-jupyter-spacex-Data%20wrangling.ipynb
GitHub URL (EDA with SQL): https://github.com/suppijags/IBM-assignment/blob/main/5.%20jupyter-labs-eda-sql-coursera_sqllite.ipynb
GitHub URL (EDA with Data Visualization): https://github.com/suppijags/IBM-assignment/blob/main/4.%20jupyter-labs-eda-dataviz.ipynb
GitHub URL (Folium Maps): https://github.com/suppijags/IBM-assignment/blob/main/6.lab_jupyter_launch_site_location.ipynb
GitHub URL (Dashboard File): https://github.com/suppijags/IBM-assignment/blob/main/Spacex_dash.py
GitHub URL (Machine Learning): https://github.com/suppijags/IBM-assignment/blob/main/7.SpaceX_Machine%20Learning%20Prediction.ipynb

Thank you!