IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Bilal BOUDJEMA
06/01/2024

Outline

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

Executive Summary

Summary of methodologies :

- **Data Collection**: SpaceX REST API, web scraping with BeautifulSoup.
- **Data Wrangling**: Cleaning, imputing missing values in PayloadMass.
- **EDA**: SQL queries, statistical analysis of launch data.
- **Visual Analytics**: Interactive maps and dashboards using Folium and Plotly Dash.
- **Predictive Analysis**: ML models (Logistic Regression, SVM, Decision Trees, KNN) for first-stage landing prediction.

Key Results:

- **Data Insights**: Identified factors affecting launch success; trends in launch frequency and success rates.
- **Model Performance**: Highest accuracy model at [X]% for landing predictions.
- **Operational Insights**: Recommendations for launch planning, payload optimization.
- **Dashboards:** Interactive tools for real-time data exploration and decision-making.

Introduction

❑**Project Background and Context:**

- The evolution of commercial spaceflight has made access to space more affordable.
- SpaceX has led the industry with cost-effective launches, primarily due to their reusable rocket technology.
- In this project, we analyze SpaceX Falcon 9 launch data to gain insights into launch costs and success factors.

❑**Problems to Address:**

- What factors contribute to the successful landing and reuse of Falcon 9's first stage?
- How does payload mass affect launch outcomes and costs?
- Can we predict the likelihood of a first stage landing based on historical launch data?

Section 1

Methodology

Methodology

Data Collection Methodology:

- Utilized **SpaceX REST API** for launch data.
- Web scraping with **Python's BeautifulSoup** for additional records.

Data Wrangling:

- Cleaned and formatted data to ensure quality and consistency.
- Imputed missing values, particularly for **Payload Mass**.

Exploratory Data Analysis (EDA):

- Analyzed datasets using **SQL** for structured insights.
- Employed **visualization techniques** to identify patterns.

Interactive Visual Analytics:

- Created dynamic maps with **Folium** to display geographic data.
- Built interactive dashboards using **Plotly Dash** for real-time data interaction.

Predictive Analysis:

- Developed classification models to predict rocket landing success.
- Techniques included **Logistic Regression**, **SVM**, **Decision Trees**, and **KNN**.

Model Building and Evaluation:

- Built models and tuned hyperparameters for optimal performance.
- Evaluated models using metrics like **accuracy**, **precision**, and **recall**.

Data Collection

- Data was meticulously gathered through a two-pronged approach. Initially, we accessed the SpaceX REST API to retrieve comprehensive launch data, which included detailed information on rockets, payloads, and outcomes.
- Complementing this, we employed web scraping techniques using Python's BeautifulSoup to extract supplementary launch records from specialized spaceflight databases and Wikipedia.
- This dual-source strategy ensured a rich dataset, which was then rigorously checked for integrity and completeness to lay a strong foundation for subsequent analysis stages.

Data Collection – SpaceX API

- Data was sourced via SpaceX REST API, extracting comprehensive launch details.
- API endpoints /launches, /rockets, and /payloads were methodically queried.
- The collection process is detailed in a Jupyter Notebook, available on GitHub for review.
- Here is the GitHub URL of the completed SpaceX API calls notebook (https://github.com/BilalBoudjema/SpaceX_Falcon_9/blob/main/Data%20Collection%20with%20Web%20Scraping/Spacex%20Data%20Collection%20Api.ipynb)

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
[9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
[10]: response.status_code
```

```
[10]: 200
```

Data Collection - Scraping

- Leveraged BeautifulSoup for targeted web scraping.
- Extracted Falcon 9 launch data from multiple web sources.
- Processed and structured data into a clean dataset.

Here is the GitHub URL of the completed web scraping notebook,
(https://github.com/BilalBoudjema/SpaceX_Falcon_9/blob/main/Data%20Collection%20with%20Web%20Scraping/Web%20Scraping.ipynb)

TASK 1: Request the Falcon9 Launch Wiki page from its URL ¶

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
[6]: # use requests.get() method with the provided static_url
     response = requests.get(static_url)
     # assign the response to a object
     response.status_code

[6]: 200
```

Create a `BeautifulSoup` object from the HTML `response`

```
[11]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
      soup = BeautifulSoup(response.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
[12]: # Use soup.title attribute
      soup.title

[12]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Data Wrangling

- Refined raw data to a structured format suitable for analysis.
- Techniques included outlier detection, missing value imputation, and data type conversion.
- Performed some Exploratory Data Analysis to search some patterns in data and determine what data would be the label for training ML supervised models.
- Wrangling steps are documented and can be reviewed on GitHub: https://github.com/BilalBoudjema/SpaceX_Falcon_9/blob/main/Data%20Wrangling/Data%20wrangling.ipynb

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome`; otherwise, it's one. Then assign it to the variable `landing_class`:

[16]:
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = [0 if x in bad_outcomes else 1 for x in df['Outcome']]
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

[17]:
```
df['Class']=landing_class
df[['Class']].head(8)
```

[17]:

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

EDA with Data Visualization

- Utilized bar charts, line graphs, and scatter plots to reveal trends and correlations in launch data.
- Bar charts showcased frequency of launches, line graphs for temporal trends, and scatter plots for payload vs. outcome correlations.
- For a detailed walkthrough of our EDA, visit our GitHub repository: https://github.com/BilalBoudjema/SpaceX_Falcon_9/blob/main/Exploratory%20Analysis/Exploratory%20Analysis%20Using%20Pandas%20and%20Matplotlib.ipynb

EDA with SQL

- Executed SQL queries to filter launch data by year and success rate.
- Grouped data by payload mass and destination orbit to study launch characteristics.
- Analyzed correlations between launch outcomes and booster versions.
- Displaying average payload mass carried by booster version F9 v1.1
- GitHub repository for the SQL EDA notebook: https://github.com/BilalBoudjema/SpaceX_Falcon_9/blob/main/Exploratory%20Analysis/Exploratory%20Analysis%20Using%20SQL.ipynb

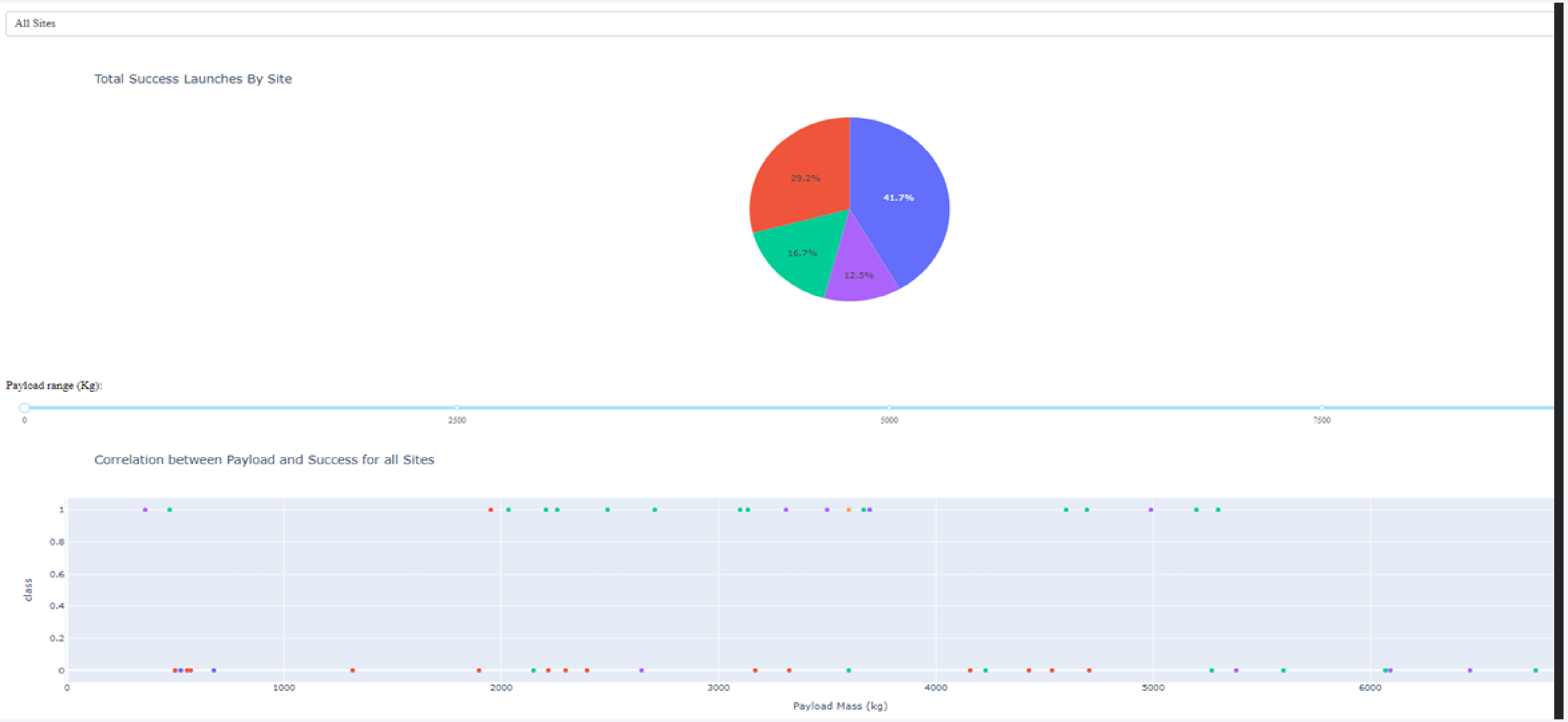
Build an Interactive Map with Folium

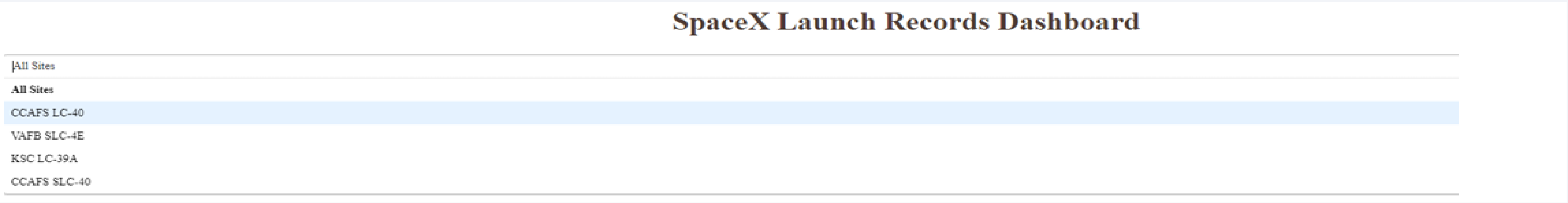
- Added markers to pinpoint launch sites for quick reference.
- Circles visualize the range and influence of each launch site.
- Lines connect launch sites to payload destinations, showing trajectories.
- Interactive features provide context and enhance user engagement.
- GitHub repository with Folium map: https://github.com/BilalBoudjema/SpaceX_Falcon_9/blob/main/Interactive%20Visual%20Analytics%20and%20Dashboard/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Incorporated dropdown lists and sliders for dynamic data exploration.
- Integrated pie charts to visualize categorical data distributions.
- Utilized scatter plots to investigate correlations with interactive hover details.
- These interactive components enable stakeholders to filter and find patterns in launch data effectively, aiding in strategic decision-making.
- GitHub repository with Plotly Dash dashboard: https://github.com/BilalBoudjema/SpaceX_Falcon_9/blob/main/Interactive%20Visual%20Analytics%20and%20Dashboard/Plotly%20Dash.py

Dashboard with Plotly Dash





Predictive Analysis (Classification)

- Developed models using Logistic Regression, Decision Trees, and SVM.
- Evaluated using accuracy, precision, recall, and F1 score metrics.
- Improved models with hyperparameter tuning and feature selection.
- Best model identified through cross-validation and performance metrics.
- GitHub repository with predictive analysis: https://github.com/BilalBoudjema/SpaceX_Falcon_9/blob/main/Machine%20Learning%20Prediction/SpaceX%20Machine%20Learning%20Prediction.ipynb

Results

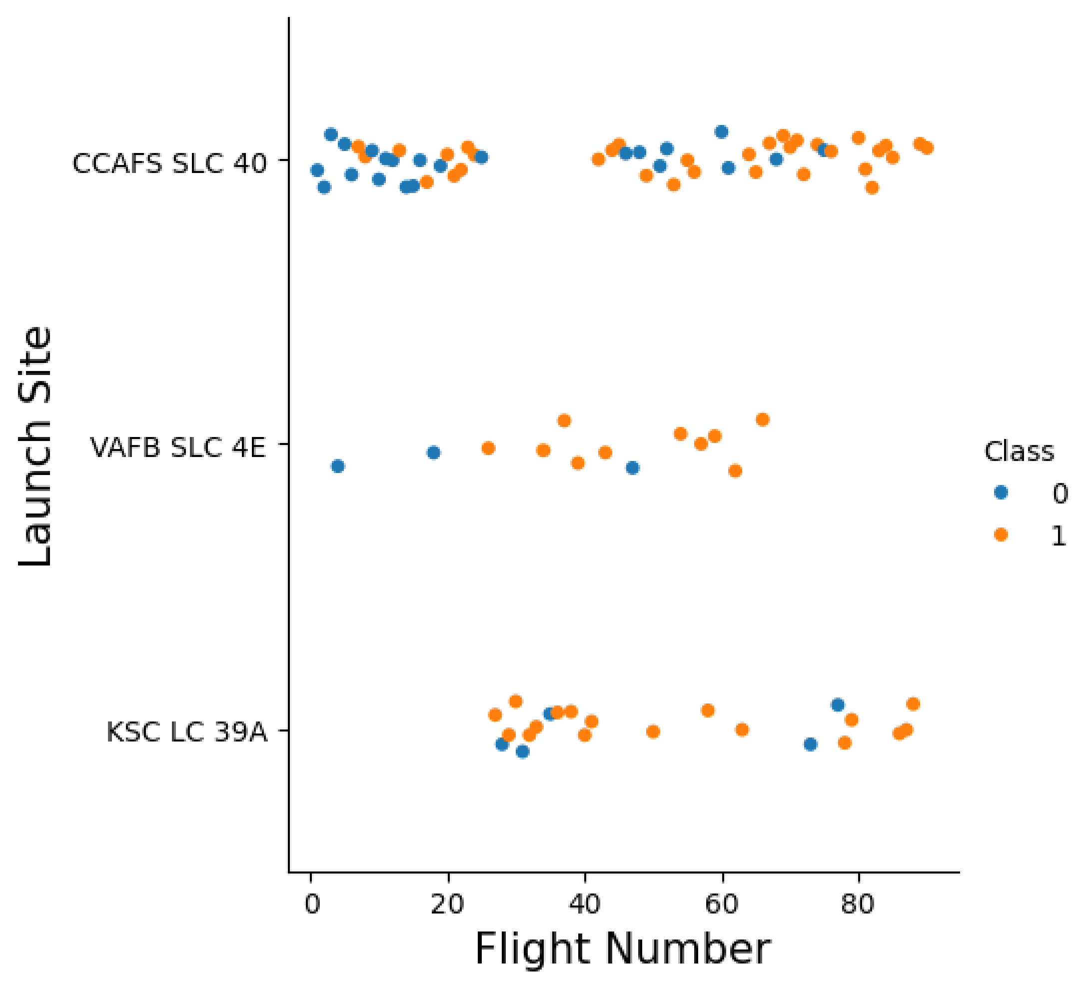
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

Scatter plot of Flight Number vs. Launch Site



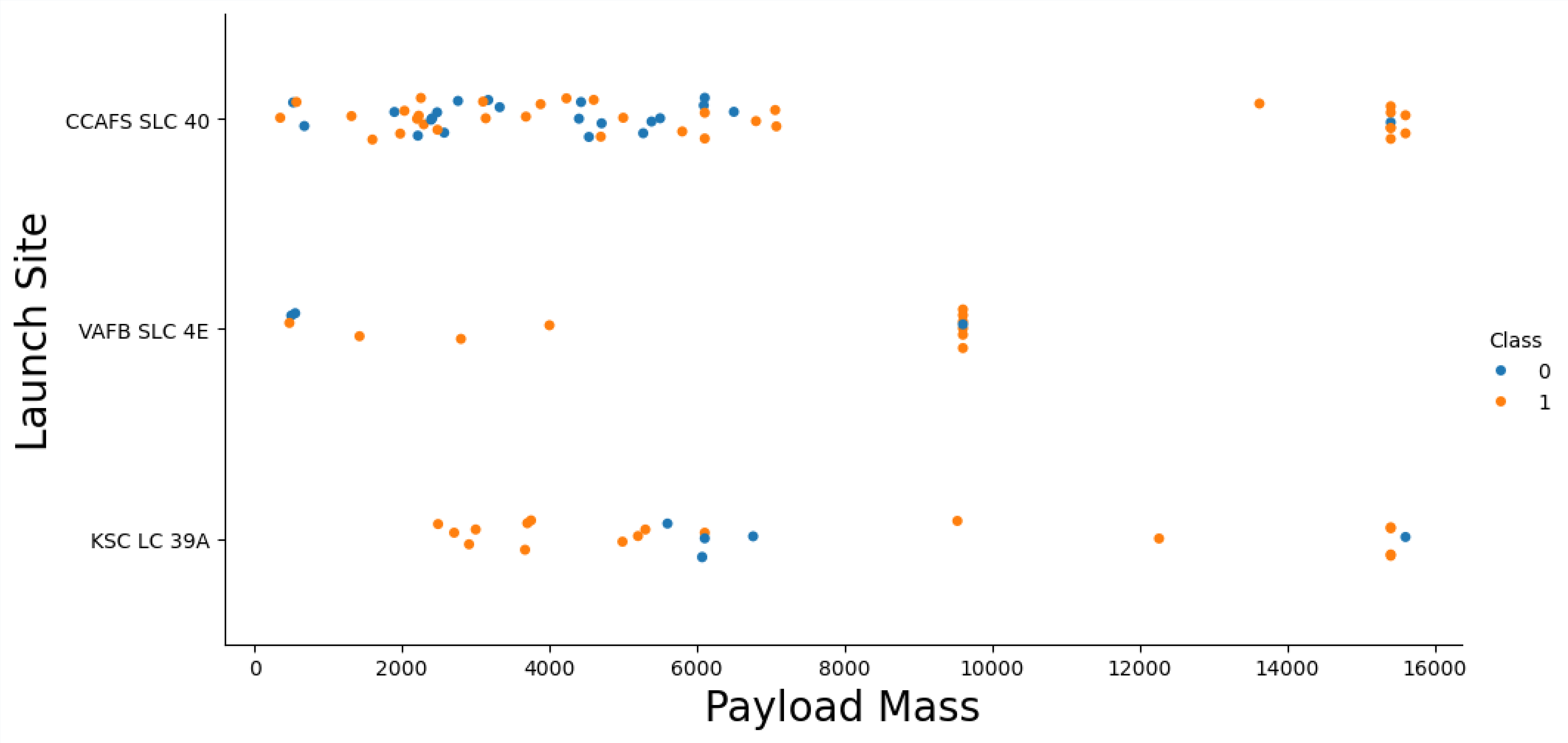
Flight Number vs. Launch Site

The explanations:

- This scatter plot provides insights into launch site usage and outcomes over time.
- Blue dots (Class 0) may represent unsuccessful landings, while orange dots (Class 1) indicate successful ones.
- The distribution of dots shows how launch frequency and success rates at each site have evolved across flights.
- The visual trend suggests an increasing success rate as the flight number increases, especially noticeable at the Kennedy Space Center (KSC LC 39A).

Payload vs. Launch Site

The scatter plot of Payload vs. Launch Site



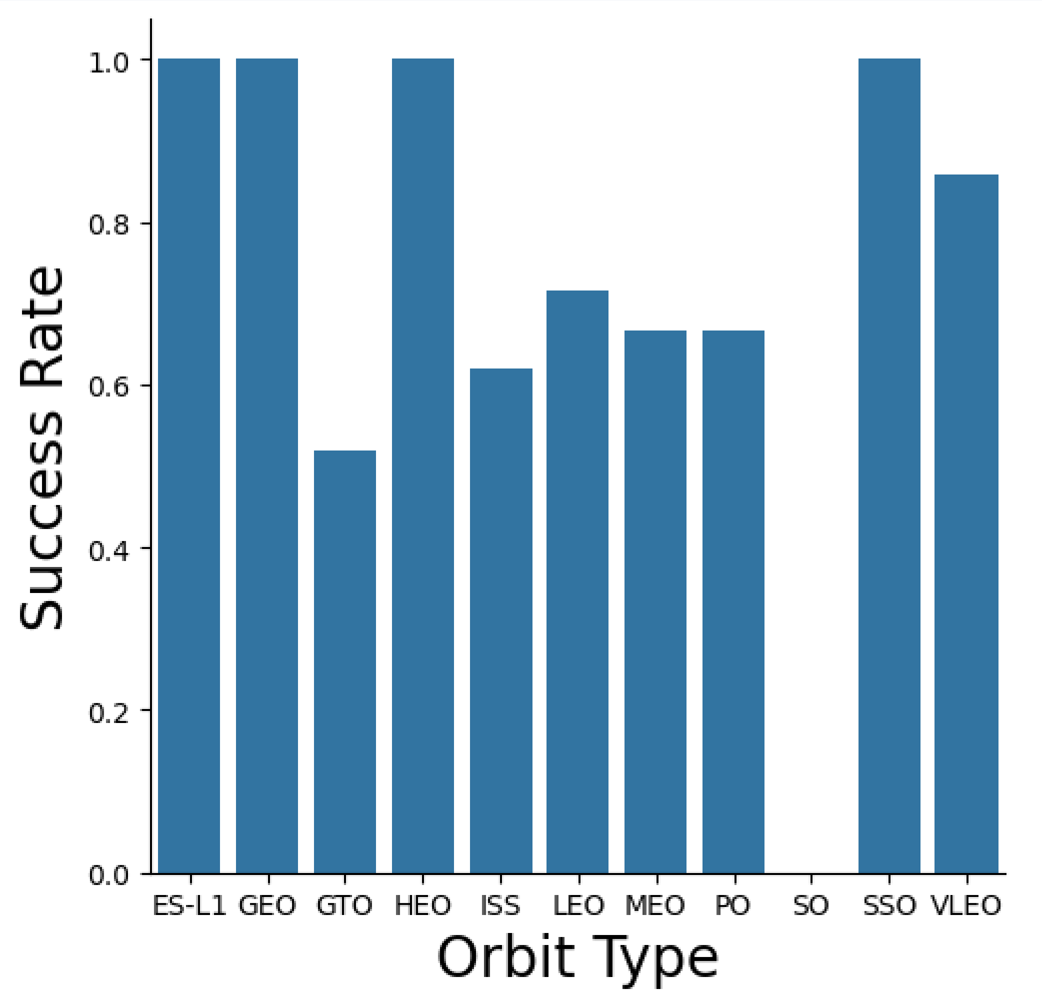
Payload vs. Launch Site

The explanations :

- This scatter plot illustrates the relationship between the mass of payloads and their corresponding launch sites.
- Each dot represents a specific launch, with its color indicating the class of the launch outcome: blue for class 0 (unsuccessful) and orange for class 1 (successful).
- The plot may show if there's a trend or pattern in the payload capacities that different launch sites can handle and how this correlates with the success of the launches.
- Analyzing the distribution of payload masses across different sites can provide insights into site specialization or preferred payload ranges.

Success Rate vs. Orbit Type

Bar chart for the success rate of each orbit type



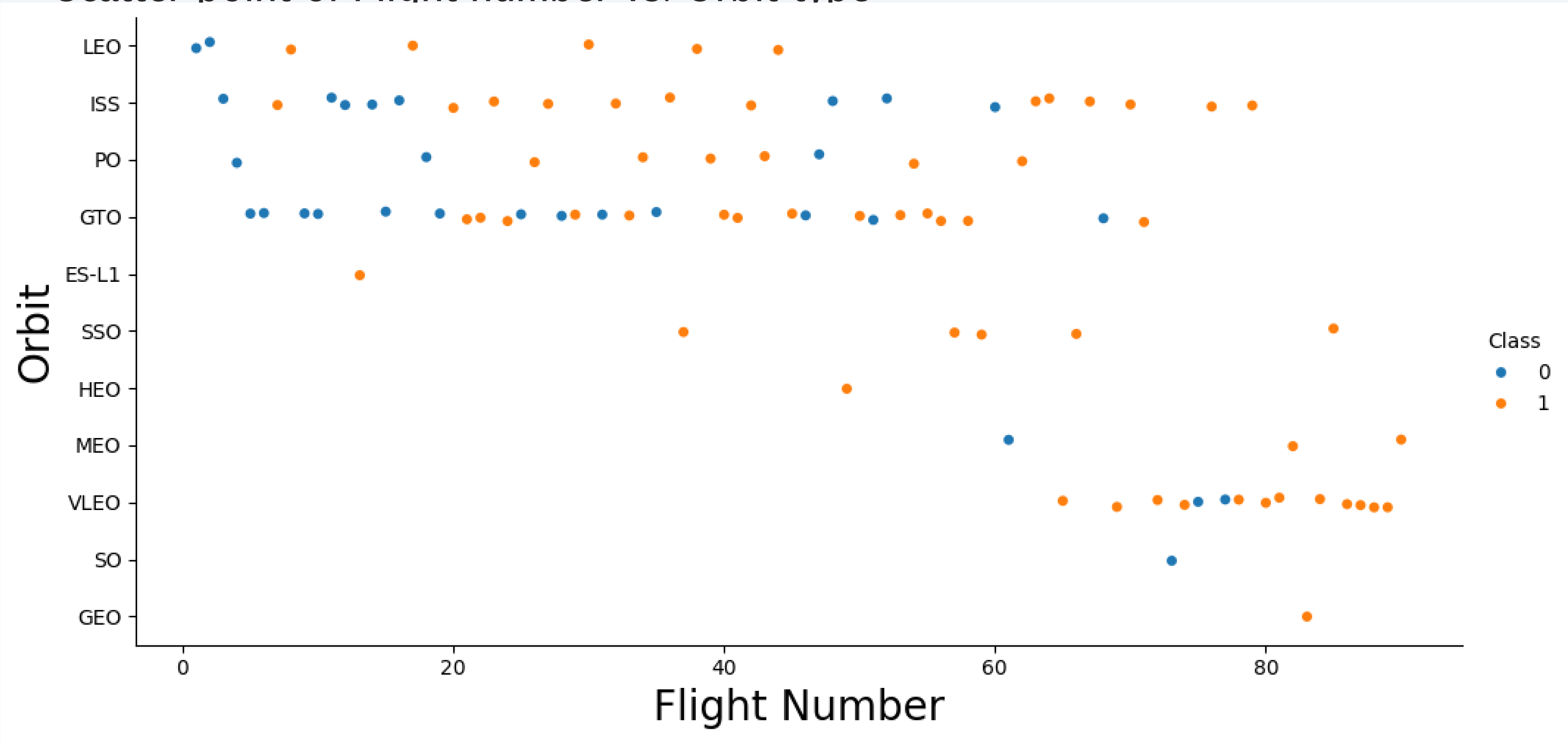
Success Rate vs. Orbit Type

The Explanations:

- This bar chart presents the success rate of SpaceX launches across different orbit types.
- Each bar represents the proportion of successful launches to total launches for orbits like GTO, ISS, LEO, etc.
- The highest success rates are observed for missions to SSO (Sun-Synchronous Orbit) and VLEO (Very Low Earth Orbit), indicating a proficiency in these mission profiles.
- The chart suggests that certain orbits may have more reliable launch outcomes, which could inform strategic decisions for mission planning.

Flight Number vs. Orbit Type

Scatter point of Flight number vs. Orbit type



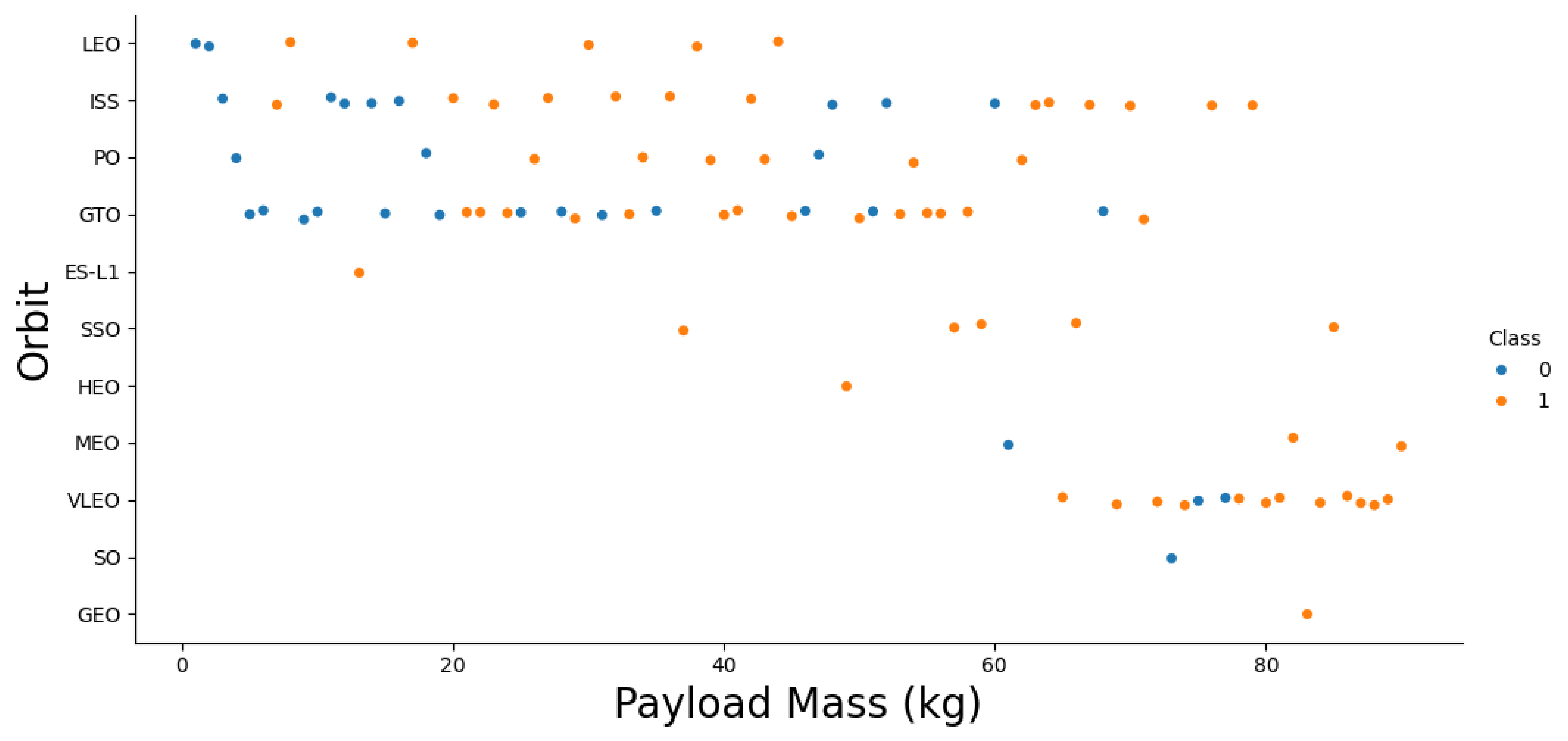
Flight Number vs. Orbit Type

The explanations:

- This scatter plot maps out each SpaceX flight against various orbit types, color-coded by mission outcome class.
- Blue dots (Class 0) may indicate unsuccessful missions, while orange dots (Class 1) represent successful ones.
- The spread of points across the flight numbers provides insight into the frequency of missions to each orbit type and their corresponding success rates.
- By analyzing the plot, we can deduce which orbits have seen more activity and which ones have a higher success frequency, which could be essential for planning future missions.

Payload vs. Orbit Type

Scatter point of payload vs. orbit type.



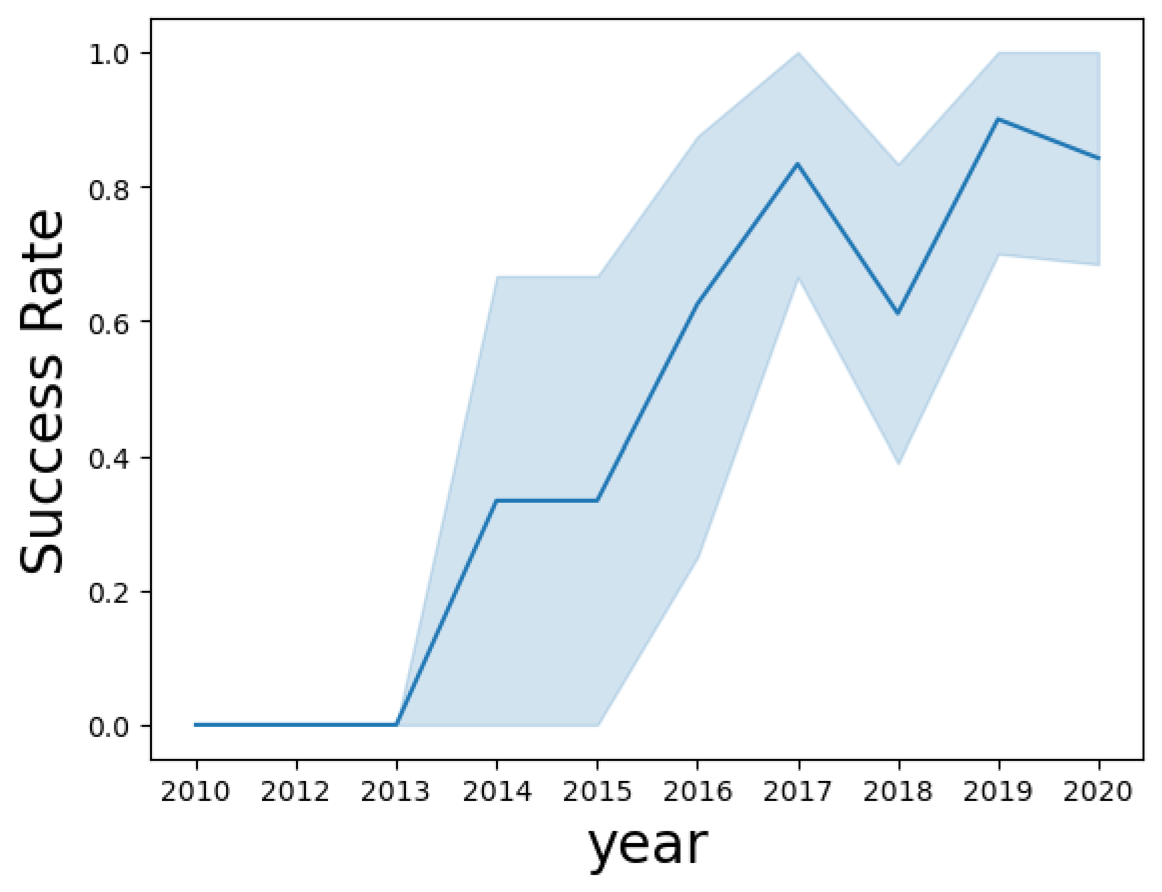
Payload vs. Orbit Type

The explanations

- This scatter plot depicts the progression of SpaceX flights across different orbit types.
- Each dot represents a launch, with the flight number on the x-axis and the orbit type on the y-axis.
- The color of the dots indicates the class of the outcome, with blue representing Class 0 (unsuccessful) and orange representing Class 1 (successful).
- Patterns in the scatter plot may suggest how frequently certain orbit types are targeted and how success rates vary with flight experience.

Launch Success Yearly Trend

Line chart of yearly average success rate



Launch Success Yearly Trend

The explanations

- This line chart illustrates the trend in SpaceX's launch success rates from 2010 through 2020.
- The y-axis represents the success rate, while the x-axis corresponds to the year.
- The shaded area around the line may represent the confidence interval or variance in the success rate data.
- The overall upward trend suggests improving reliability and success in SpaceX launches over time, with some fluctuations that may correspond to periods of technological change or operational scaling.

All Launch Site Names

An SQL query to extract unique launch site names from the SpaceX database.

- The query SELECT DISTINCT(Launch_Site) FROM SPACEXTBL LIMIT 5; lists distinct values from the 'Launch_Site' column.
- Results display major launch sites used by SpaceX, including CCAFS LC-40, VAFB SLC-4E, and KSC LC-39A.
- Note: The same site 'CCAFS LC-40' appears twice due to the LIMIT 5 clause in the query, which might not be necessary for distinct values.

```
[9]: %sql SELECT Distinct(Launch_Site) FROM SPACEXTBL limit 5;

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

[9]:

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

Launch Site Names Begin with 'CCA'

A Query result with a short explanation :

- The SQL command SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5; retrieves the first five records matching the criteria.
- The results table displays detailed launch information for these specific sites, including dates, payload masses, orbits, and outcomes.
- Notably, all displayed records pertain to the CCAFS LC-40 site, which has hosted a range of missions with varying payloads and outcomes.

Task 2 ¶

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

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

A short explanation :

The SQL query SELECT SUM(PAYLOAD_MASS_KG) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'; calculates the cumulative mass of payloads SpaceX has launched for NASA CRS missions.

The result, 45596 kg, represents the total mass of all payloads carried to space under the CRS program.

This metric highlights the significant cargo SpaceX has transported to support NASA's operations, underscoring their contribution to space resupply missions.

Task 3

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

```
%sql SELECT sum(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

A short explanation :

The SQL query SELECT AVG(PAYLOAD_MASS_KG) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1'; was executed to determine the average mass of payloads carried by the Falcon 9 version 1.1 booster.

The query result indicates that the average payload mass for this booster version is approximately 2984.4 kilograms.

This figure helps in understanding the lifting capabilities of the Falcon 9 v1.1 variant, contributing to payload planning and mission design for future launches.

Task 4

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

```
%sql SELECT avg(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1';
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

A short explanation :

- The executed SQL query SELECT MIN(Date) as 'First Successful Landing' FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'; is designed to find the earliest date on which SpaceX achieved a successful landing on a ground pad.
- The query result returned the date 2015-12-22, marking a significant achievement in the field of aerospace. It denotes the first time SpaceX managed to land a booster on solid ground, which is a key step towards their goal of reusability.
- This event not only demonstrated technical prowess but also paved the way for more economical space flight, as reusable boosters significantly reduce the cost of launches.

Task 5 ¶

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
%sql SELECT min(Date) as 'First Succeful Landing' FROM SPACEXTBL where Landing_Outcome = 'Success (ground pad)';
```

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

First Succeful Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

A short explanation here :

- The SQL query SELECT Payload FROM SPACEXTBL where (Landing_Outcome = 'Success (drone ship)') and (PAYLOAD_MASS_KG_ > 4000 and PAYLOAD_MASS_KG_ < 6000); was used to identify which SpaceX missions involved a payload mass within the specified range and concluded with a successful landing on a drone ship.
- The query results listed the payloads: JCSAT-14, JCSAT-16, SES-10, and SES-11/Echostar 105, indicating the specific missions where the booster not only carried a significant payload but also achieved a successful landing at sea.
- These results underscore SpaceX's capability to recover boosters even under challenging conditions with heavy payloads, marking a significant step towards their goal of reusability and cost reduction in space launches.

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

```
%sql SELECT Payload FROM SPACEXTBL where  ( Landing_Outcome = 'Success (drone ship)') and ( PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000 ) ;
```

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

Payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

A short explanation here:

- The SQL query executed was SELECT Mission_Outcome, count(Mission_Outcome) as total_number FROM SPACEXTBL GROUP BY MISSION_OUTCOME; to count the number of times each mission outcome occurred.
- According to the query results, there were 98 successful missions, one failure during flight, one mission with unclear payload status, and an additional entry that also states 'Success'. This discrepancy could be due to an input error or missing context in the data.
- This information highlights SpaceX's high success rate and provides a quantitative measure of mission outcomes, which is critical for evaluating the company's performance and reliability.

Task 7

List the total number of successful and failure mission outcomes

```
[15]: %sql SELECT Mission_Outcome, count(Mission_Outcome) as total_number FROM SPACEXTBL GROUP BY MISSION_OUTCOME ;
 * sqlite:///my_data1.db
Done.
```

[15]:

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

Boosters Carried Maximum Payload

A short explanation here:

The executed SQL query identifies the booster versions that have transported the heaviest payloads, highlighting the capacity of different booster iterations.

The SELECT statement with the subquery (SELECT MAX(PAYLOAD_MASS_KG) FROM SPACEXTBL) retrieves the maximum payload mass from the dataset.

The main query SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS_KG = uses the result of the subquery to find all instances where the payload mass matches this maximum value.

This data can be crucial for assessing the performance capabilities of various SpaceX boosters and their improvements over time. It also serves as an indicator of the company's progress in increasing its payload capacity to meet more demanding space missions.

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery ¶

```
[16]: %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
 * sqlite:///my_data1.db
Done.
```

[16]:

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

A short explanation :

- The slide shows the SQL command used to retrieve data for the year 2015 where the landing outcome was a failure on the drone ship.
- The substr function is employed to filter records based on the month and year extracted from the date column.
- The results listed are ordered by date and display the month (as '01' for January, '04' for April), booster version, launch site, and the outcome of the landing.
- This information is valuable for evaluating the performance of SpaceX landings over a given period, allowing for analysis of failure rates and identification of potential patterns or issues specific to certain times of the year or booster versions.

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 Month,
    Booster_Version,
    Launch_Site,
    Landing_Outcome
FROM
    SPACEXTBL
WHERE
    substr(Date, 0, 5) = '2015' AND
    Landing_Outcome = 'Failure (drone ship)'
ORDER BY
    Date;
```

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

Month	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)

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

A short explanation :

The SQL task to count and sort SpaceX landing outcomes from June 4, 2010, to March 20, 2017. The first SQL command is incorrect as it does not provide the desired ranking. The second query correctly ranks landing outcomes by their count 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.

[18]:
```
%sql SELECT * FROM SPACEXTBL limit 2;
```

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

[18]:

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)

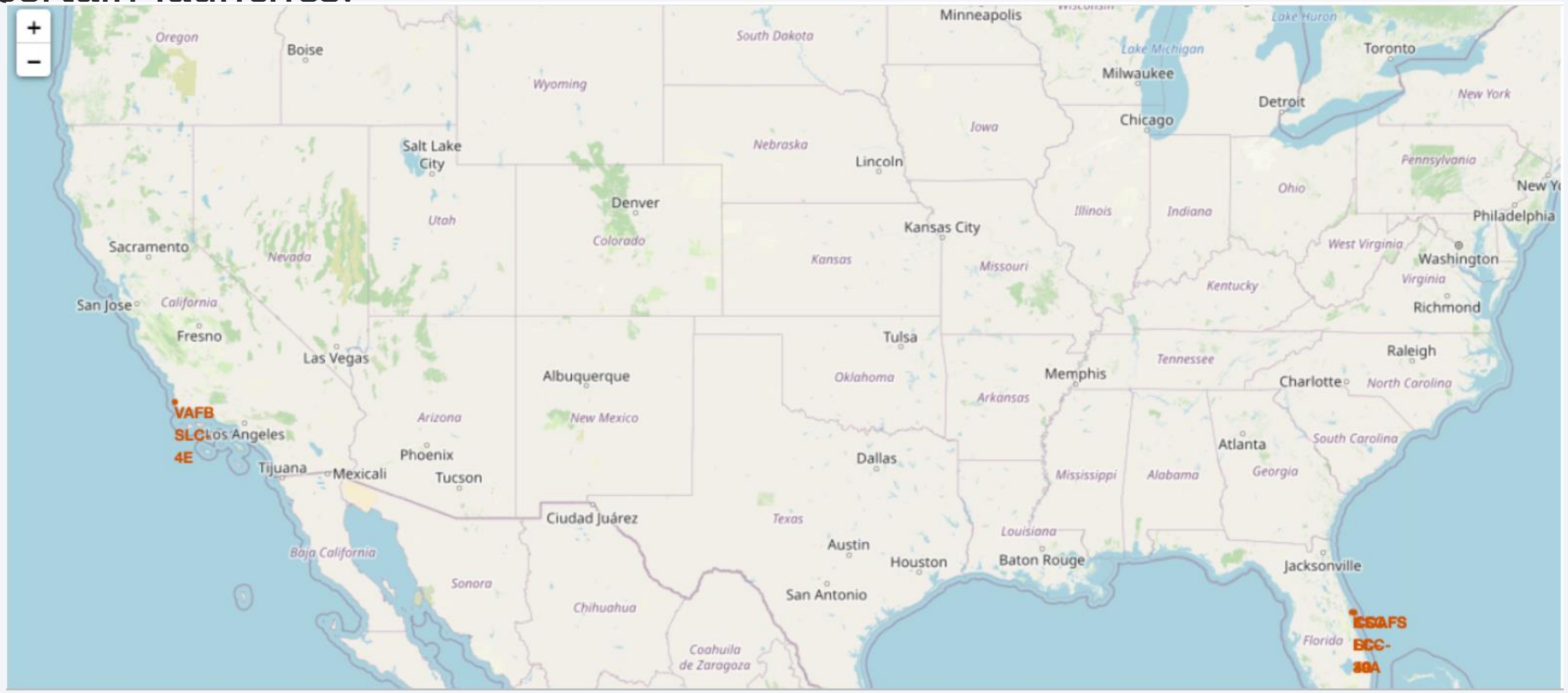
[19]:
```
%%sql SELECT Landing_Outcome, count(*) as count_outcomes
FROM SPACEXTBL
WHERE DATE between '2010-06-04' and '2017-03-20' group by Landing_Outcome order by count_outcomes DESC;
```

Section 3

Launch Sites Proximities Analysis

Overview of Key Launch Sites in the United States

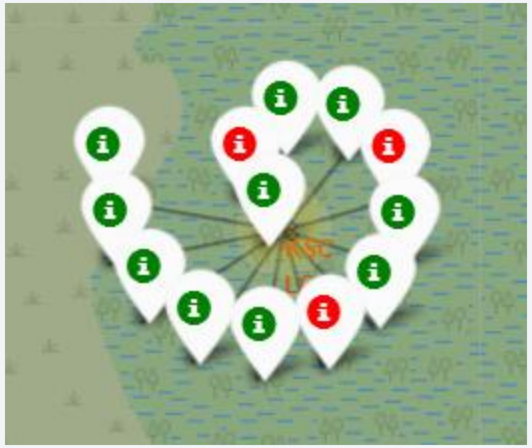
- Markers show the locations like "VAFB SLC 4E" and "CCAFS SLC 40", indicating Vandenberg Air Force Base and Cape Canaveral Air Force Station respectively. These sites are strategically positioned along the coast for safety and trajectory benefits, with proximity to the equator for certain launches.

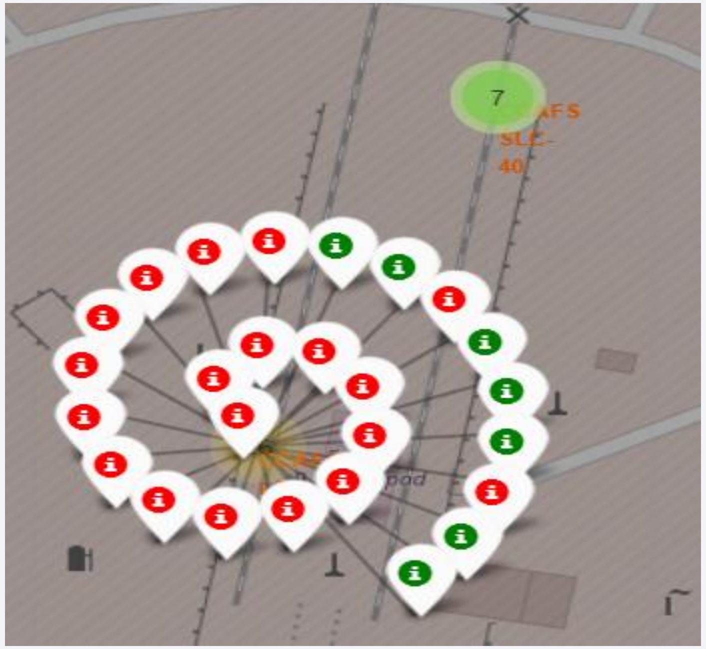


Analysis of Launch Outcomes at Key Sites

Site Of Florida:

- On the East Coast of Florida, the launch site at KSC LC-39A exhibits a higher success rate relative to that of CCAFS SLC-40 and CCAFS LC-40

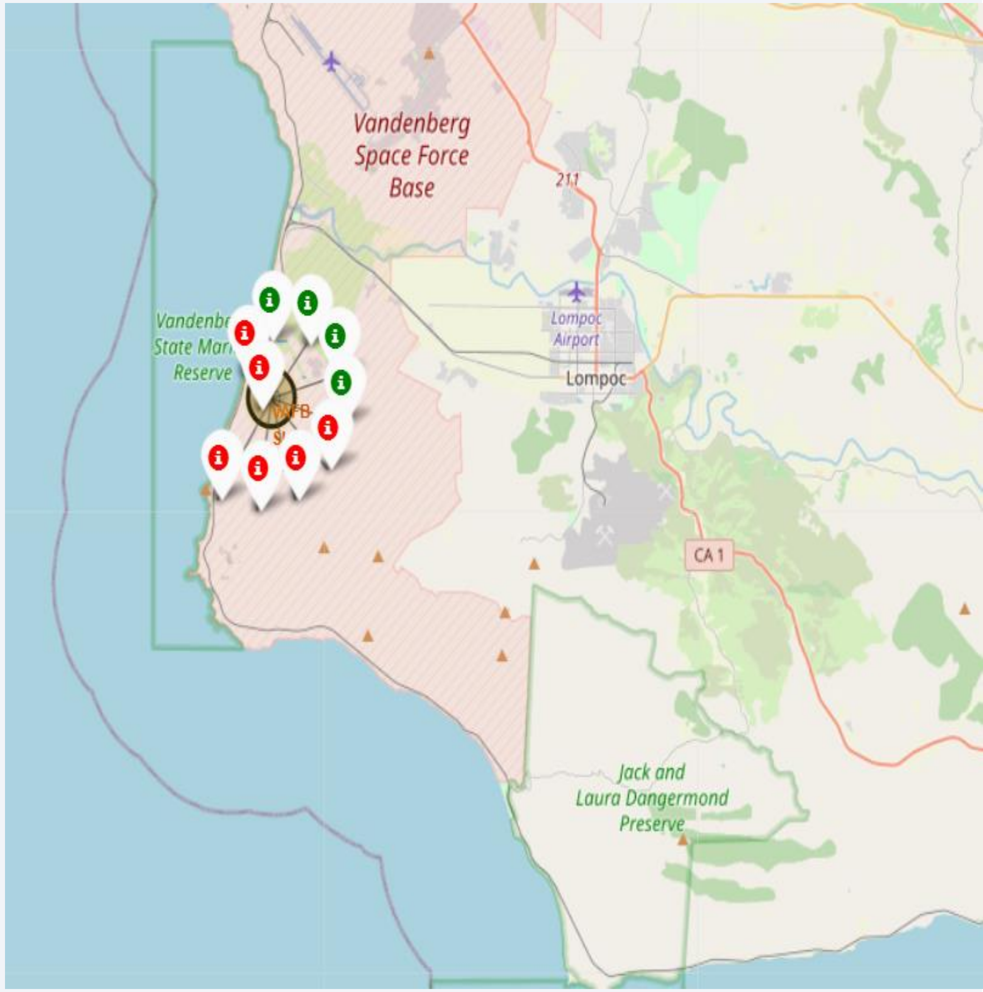




Comparative Success Rates of West and East Coast Launch Sites

The side of West Coast / Carlifonia:

- On the West Coast in California, the VAFB SLC-4E launch site shows a comparatively lower success rate of 40%, as opposed to the KSC LC-39A launch site on Florida's East Coast.

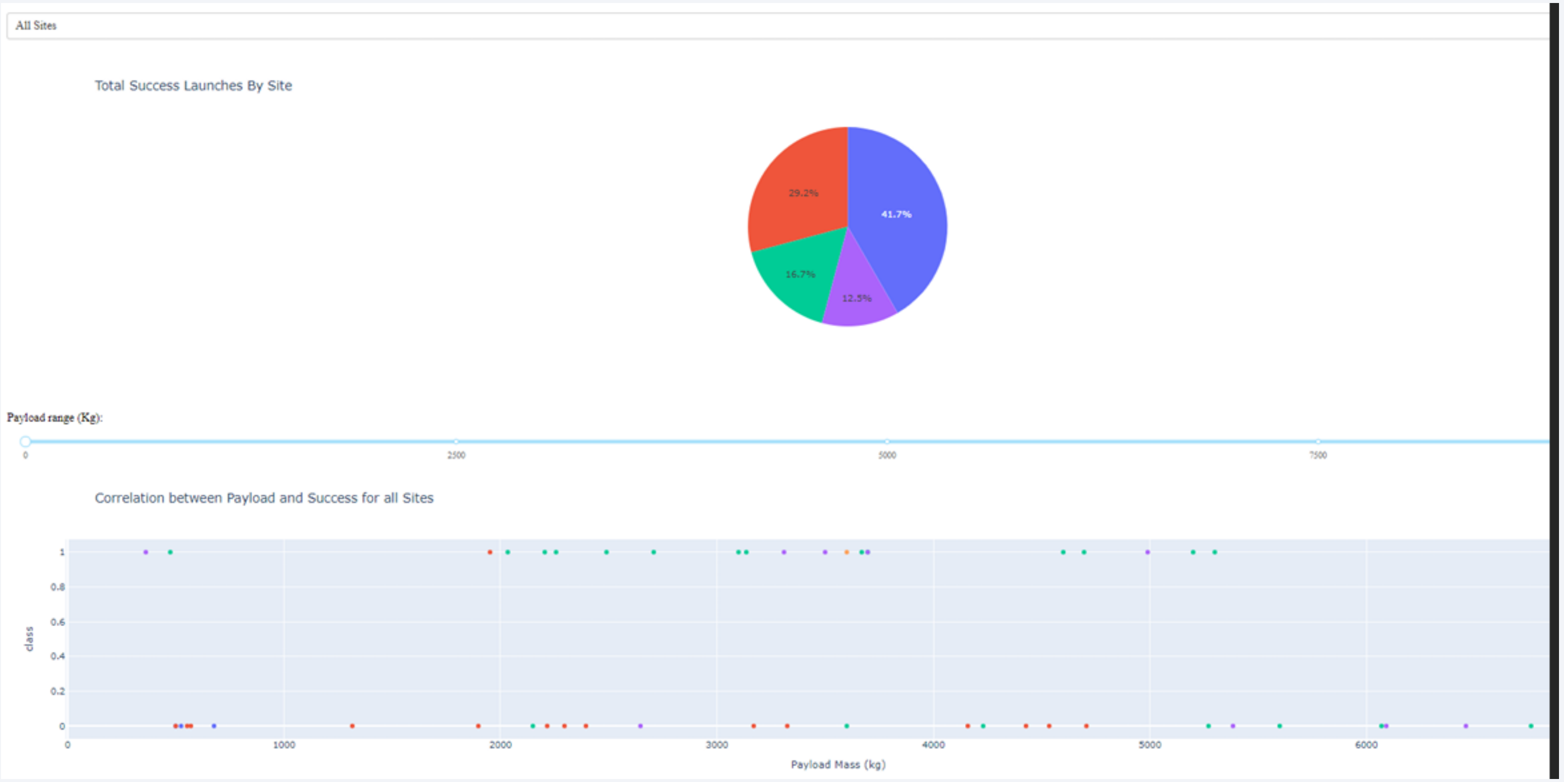


Section 4

Build a Dashboard with Plotly Dash

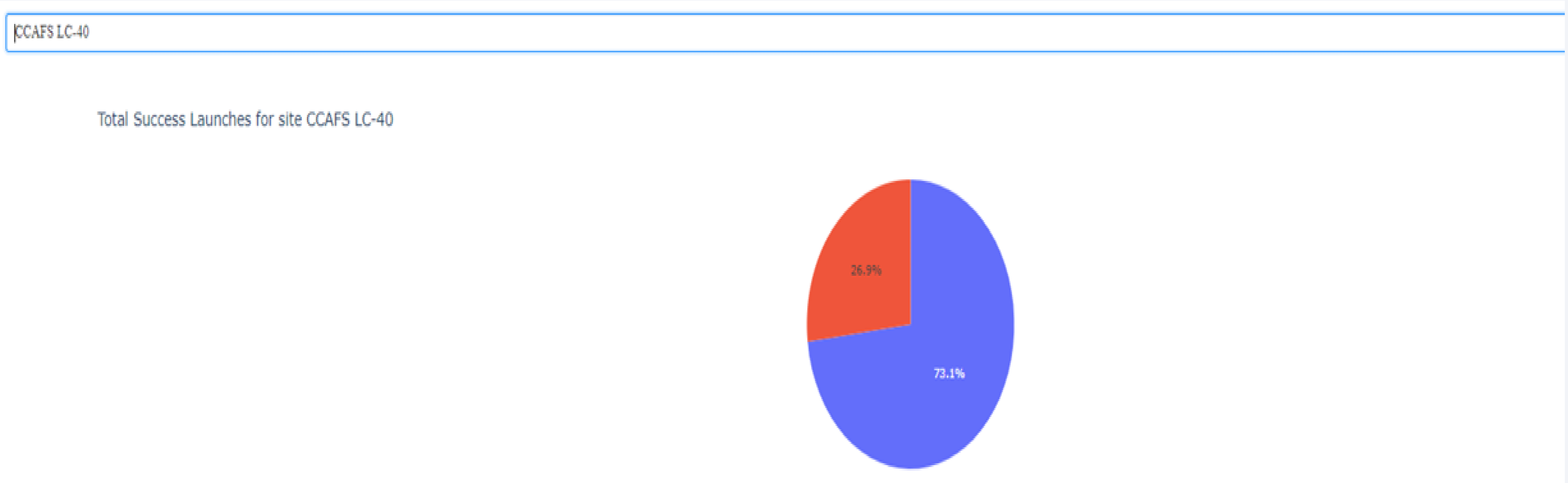
Pie-Chart for launch success count for all sites.

- KSC LC-39A leads with a 42% launch success rate, followed by CCAFS LC-40 at 29%, VAFB SLC-4E at 17%, and CCAFS SLC-40 at 13%



Pie chart for the launch site with 2ndhighest launch success ratio

- Launch site CCAFS LC-40 achieved the second-highest success ratio, with 73% of launches succeeding and 27% failing.



Payload vs. Launch Outcome scatter plot for the all sites

- At Launch site CCAFS LC-40, the booster version FT boasts the highest success rate for payloads exceeding 2000 kg.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- All algorithms perform good equally on the test data :

- They all have the same accuracy 83%

TASK 12

Find the method performs best:

[32]:

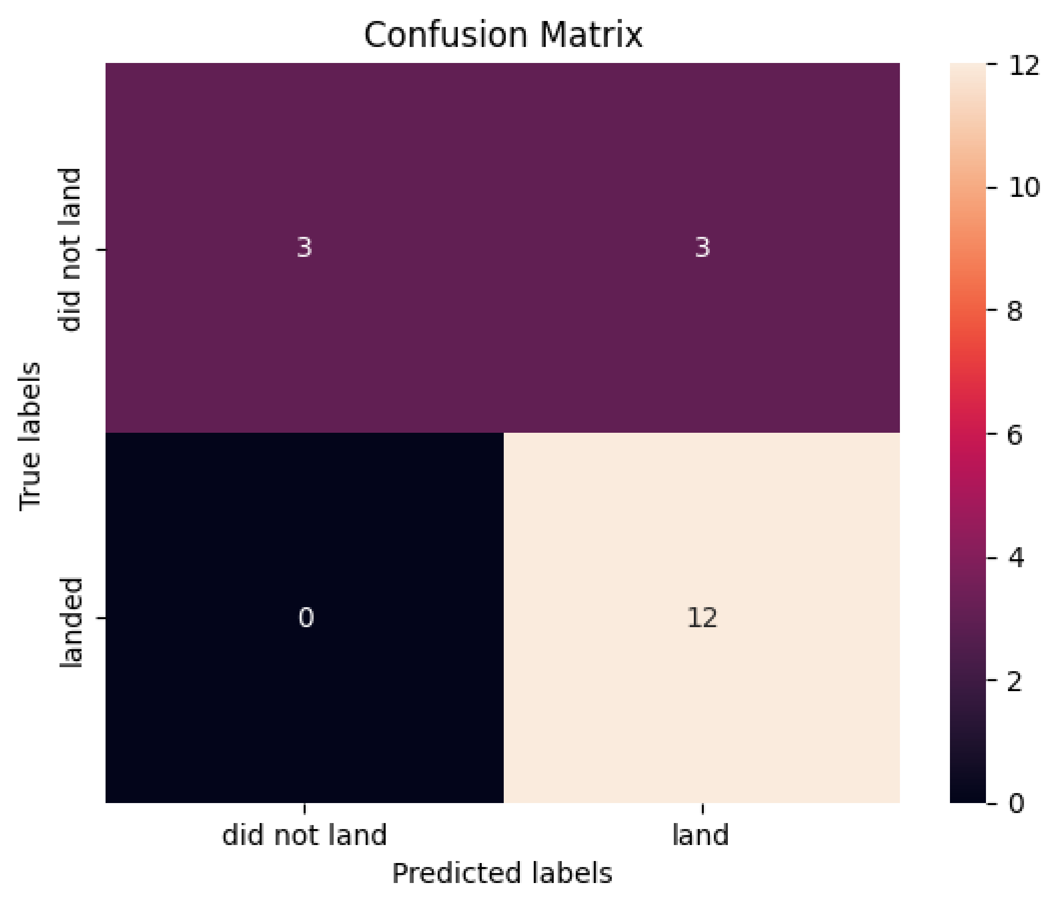
```
Report = pd.DataFrame({'Method' : ['Test Data Accuracy']})

knn_accuracy=knn_cv.score(X_test, Y_test)
Decision_tree_accuracy=tree_cv.score(X_test, Y_test)
SVM_accuracy=svm_cv.score(X_test, Y_test)
Logistic_Regression=logreg_cv.score(X_test, Y_test)

Report['Logistic_Reg'] = [Logistic_Regression]
Report['SVM'] = [SVM_accuracy]
Report['Decision Tree'] = [Decision_tree_accuracy]
Report['KNN'] = [knn_accuracy]


Report.transpose()
```

[32]:

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Confusion Matrix

The confusion matrix of the best performing model with an explanation



- Confusion Matrix for top model.
- False negatives: 3 (landed predicted as not landed).
- False positives: 3 (not landed predicted as landed).
- True positives/negatives: 12 each (accurate land/not land predictions).

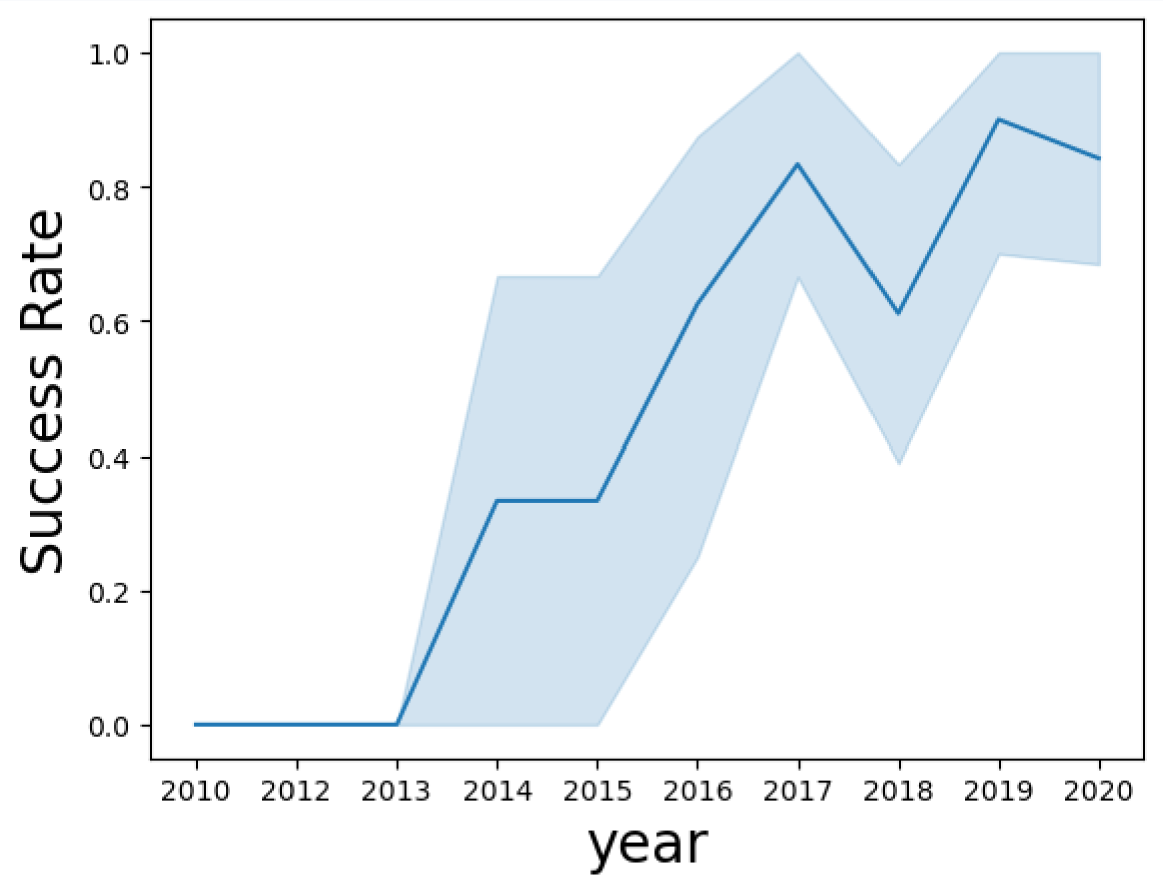
Conclusions

- High launch success rates are observed at KSC LC-39A, indicating operational efficiency.
- CCAFS LC-40 shows a significant success ratio, demonstrating reliability.
- VAFB SLC-4E has a lower success rate, suggesting possible improvements.
- Booster version FT is notably successful for payloads over 2000kg at CCAFS LC-40.
- The confusion matrix for the model indicates balanced prediction capabilities with equal false positives and negatives.
- Data analysis indicates geographical and technical factors influencing launch outcomes.
- Further optimization and analysis could enhance prediction models and operational success.

Appendix

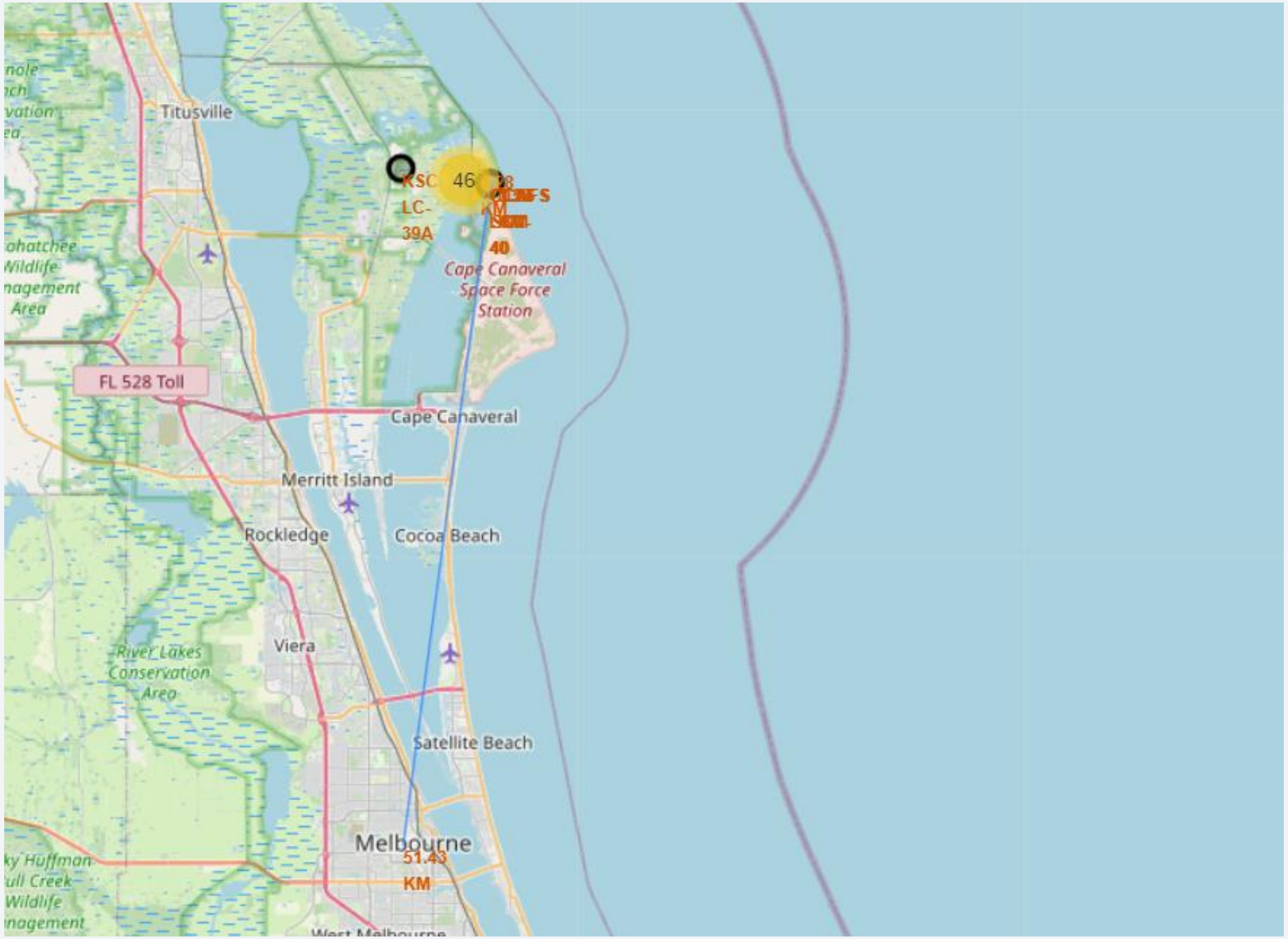
- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Success Rate by Year



The success rate since 2013 kept increasing till 2020

The distance between the launche site and the city of melbourne



Thank you!