IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dhanya Manam
4 July 2024

Outline

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

Executive Summary

- The project involved collecting SpaceX launch data using the SpaceX API and web scraping, followed by data wrangling to clean and prepare the dataset, including feature engineering. Exploratory Data Analysis (EDA) was performed using SQL and visualized with Python libraries Pandas and Matplotlib. Interactive visual analytics were created using Folium and Plotly Dash to analyze launch sites and outcomes. For predictive analysis, various machine learning models were developed, tuned, and evaluated, with Gradient Boosting achieving the highest accuracy in predicting Falcon 9 landing success.

- The EDA identified key features influencing landing success, and the interactive dashboards provided dynamic insights into geographical factors. The Gradient Boosting model stood out with around 90% accuracy, supported by robust evaluation metrics. These results highlight the effectiveness of machine learning and interactive analytics in understanding and predicting SpaceX Falcon 9 landings.

Introduction

SpaceX offers Falcon 9 rocket launches at a significantly lower cost of $62 million compared to other providers' costs of upwards of $165 million, primarily due to the reusability of the first stage. The ability to predict the success of the first stage landing is crucial as it directly impacts the cost-efficiency of these launches. By accurately forecasting whether the Falcon 9 first stage will land successfully using historical launch data, this capstone project aims to provide valuable insights that could be utilized by alternative companies bidding against SpaceX for rocket launches, thereby potentially leveling the competitive playing field in the aerospace market.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

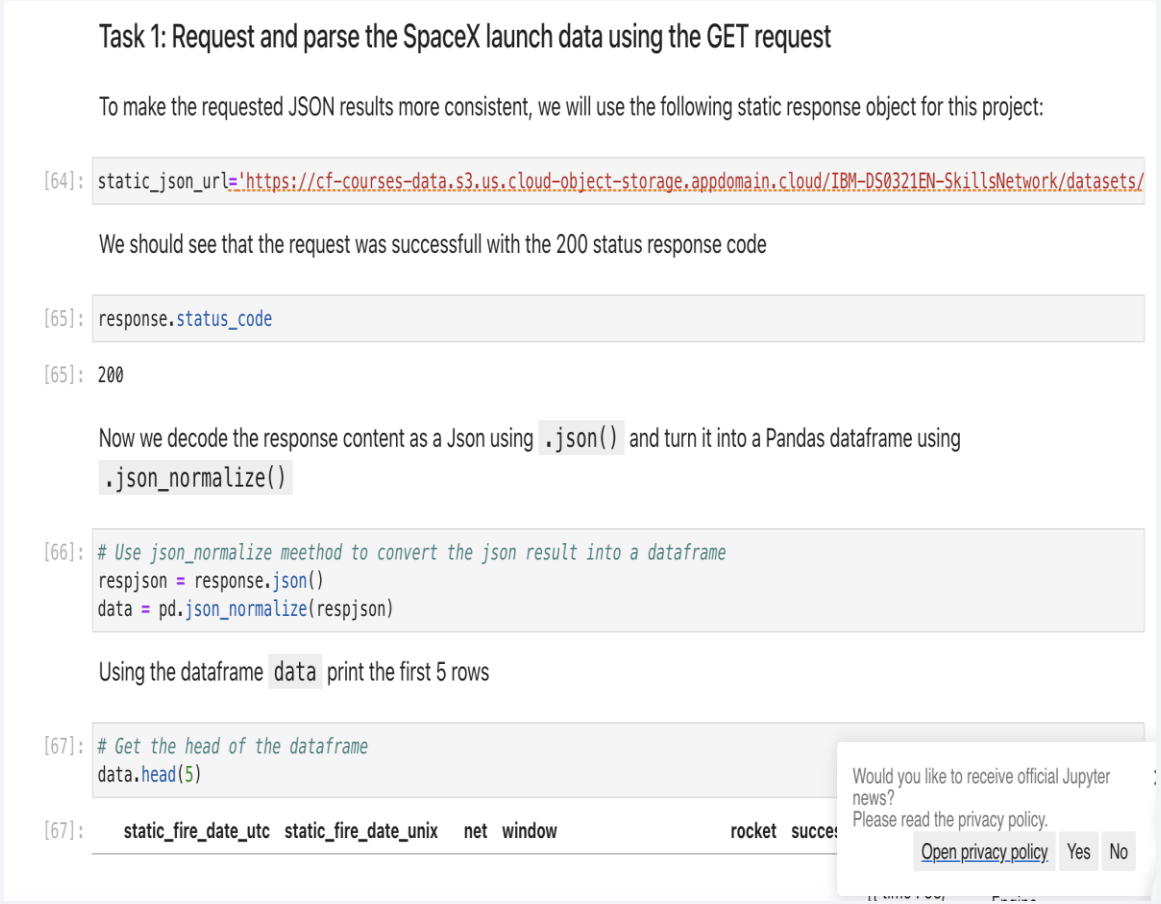
Data Collection

- **Using SpaceX API:** Automated retrieval of detailed and up-to-date launch data directly from SpaceX's API. This includes information on launch dates, rocket specifications, payload details, and landing outcomes.
- **Web Scraping:** Supplementary data collection from various space-related websites to gather additional information such as weather conditions, historical context, and other relevant factors that are not available through the API.

Data Collection – SpaceX API

- Data was collected by making GET requests to the SpaceX RESTful API. The API responses, which contain detailed launch information, were received in JSON format. These JSON responses were then parsed and converted into a Pandas Data Frame for further analysis.

- Here is the GitHub URL of the completed SpaceX API calls notebook (Github Link)



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:

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

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

```
[65]: response.status_code

[65]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
[66]: # Use json_normalize meethod to convert the json result into a dataframe
      respjson = response.json()
      data = pd.json_normalize(respjson)
```

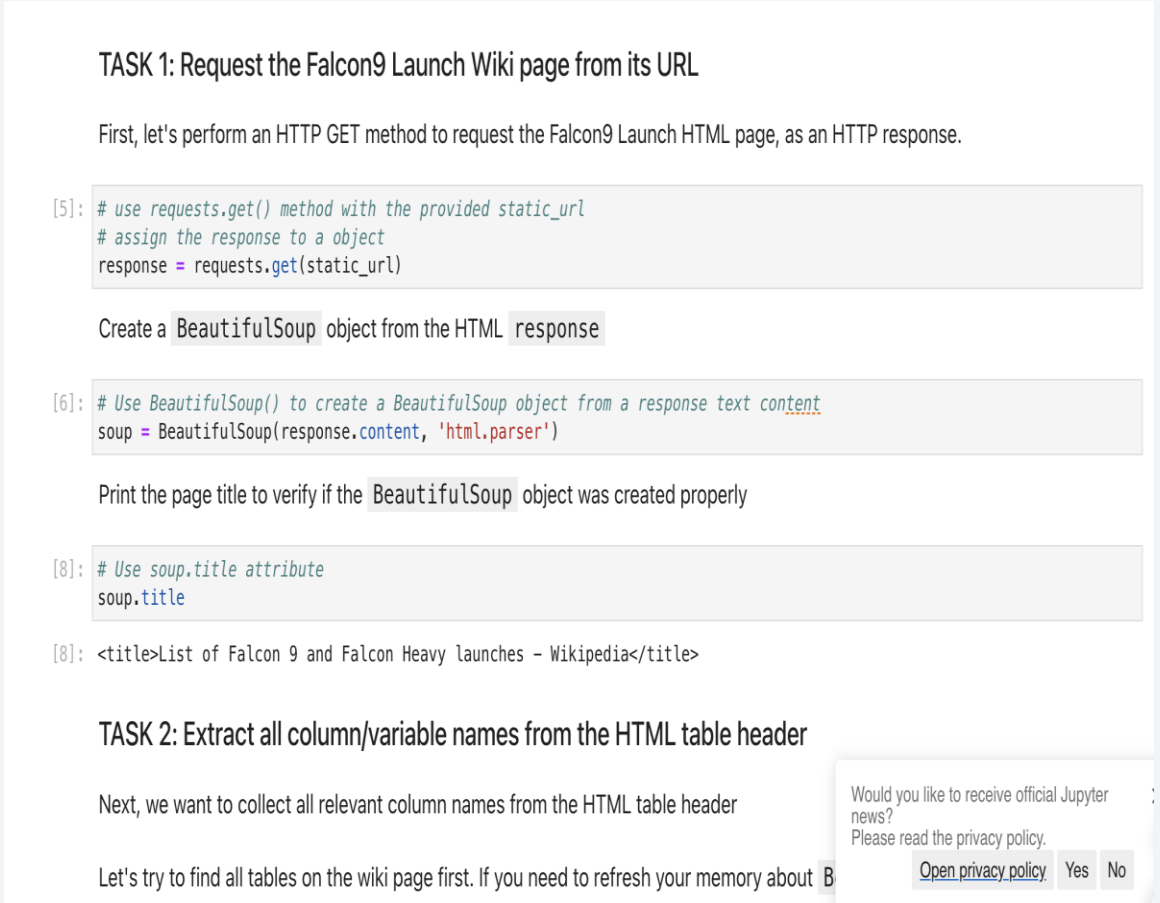
Using the dataframe `data` print the first 5 rows

```
[67]: # Get the head of the dataframe
      data.head(5)

[67]:   static_fire_date_utc  static_fire_date_unix  net  window  rocket  succes
```

Data Collection - Scraping

- Performed web scraping to collect Falcon 9 historical launch records from Wikipedia using Beautiful Soup and Requests. The HTML table containing Falcon 9 launch records was extracted and parsed, and the data was then converted into a Pandas Data Frame for further analysis.
- Here is the GitHub URL of the completed SpaceX web-scraping calls notebook (Github Link)



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.

[5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

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

[6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content, 'html.parser')
```

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

[8]:
```
# Use soup.title attribute
soup.title
```

[8]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

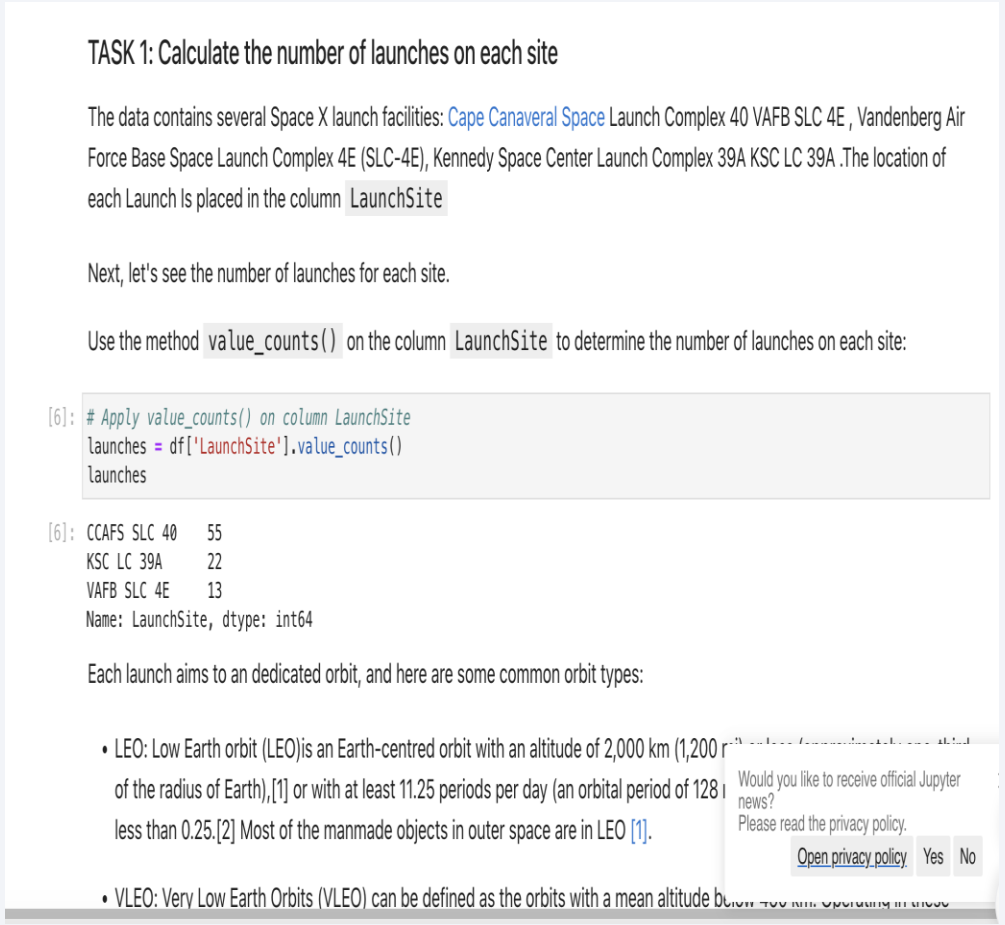
Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about B

Would you like to receive official Jupyter news?
Please read the privacy policy.
Open privacy policy Yes No

Data Wrangling

- After creating a Pandas Data Frame from the collected data, the dataset was filtered to include only Falcon 9 launches using the Booster Version column. Missing values in the Landing Pad and Payload Mass columns were handled, with the mean value used to replace missing entries in the Payload Mass column.
- Exploratory Data Analysis (EDA) was conducted to uncover patterns in the data and identify suitable labels for training supervised models.
- Here is the GitHub URL of the completed data wrangling related notebooks:[Github Link]



TASK 1: Calculate the number of launches on each site

The data contains several Space X launch facilities: Cape Canaveral Space Launch Complex 40 VAFB SLC 4E , Vandenberg Air Force Base Space Launch Complex 4E (SLC-4E), Kennedy Space Center Launch Complex 39A KSC LC 39A .The location of each Launch Is placed in the column `LaunchSite`

Next, let's see the number of launches for each site.

Use the method `value_counts()` on the column `LaunchSite` to determine the number of launches on each site:

```
[6]: # Apply value_counts() on column LaunchSite
     launches = df['LaunchSite'].value_counts()
     launches
```

```
[6]: CCAFS SLC 40    55
     KSC LC 39A      22
     VAFB SLC 4E     13
     Name: LaunchSite, dtype: int64
```

Each launch aims to an dedicated orbit, and here are some common orbit types:

- LEO: Low Earth orbit (LEO)is an Earth-centred orbit with an altitude of 2,000 km (1,200 mi) or less (approximately one third of the radius of Earth),[1] or with at least 11.25 periods per day (an orbital period of 128 minutes or less) and an eccentricity less than 0.25.[2] Most of the manmade objects in outer space are in LEO [1].

Would you like to receive official Jupyter news?
Please read the privacy policy.
Open privacy policy Yes No

- VLEO: Very Low Earth Orbits (VLEO) can be defined as the orbits with a mean altitude below 400 km. Operating in these

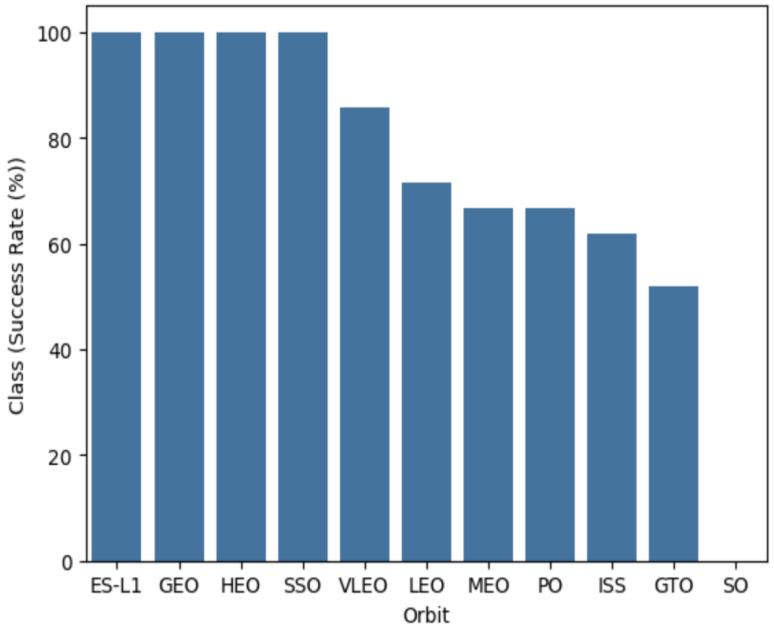
EDA with Data Visualization

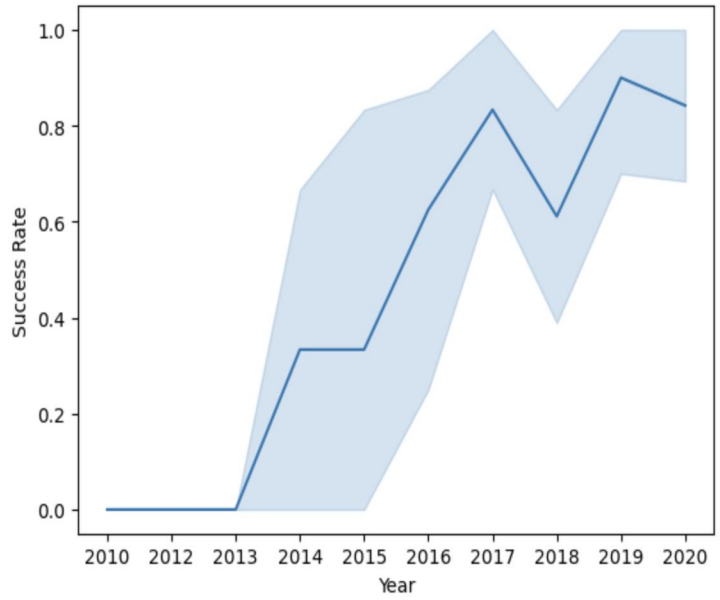
- Data analysis and feature engineering were conducted using Pandas and Matplotlib. This process included Exploratory Data Analysis (EDA) and preparing data through feature engineering. Various visualizations were created to explore relationships within the data, such as scatter plots showing the connections between Flight Number and Launch Site, Payload and Launch Site, Flight Number and Orbit Type, and Payload and Orbit Type.

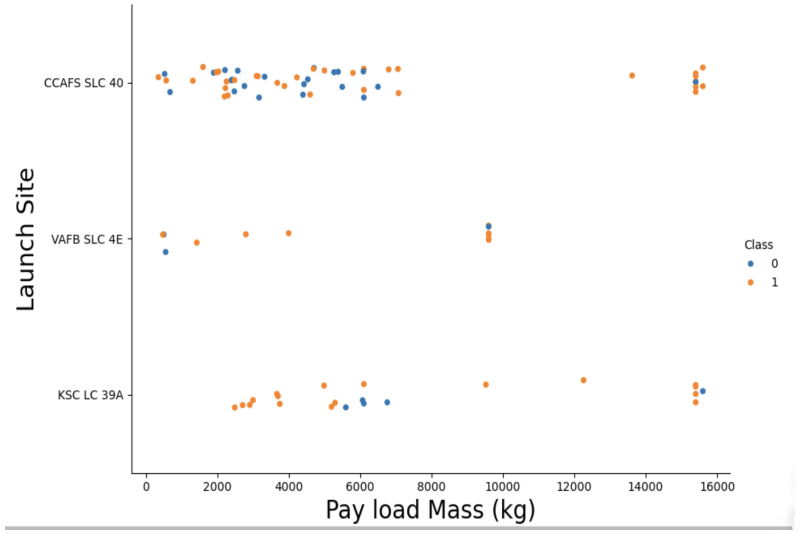
- Additionally, bar charts were used to visualize the success rates of different orbit types, while line plots illustrated the yearly trend of launch success. These visualizations helped uncover patterns and insights, providing a solid foundation for further predictive modeling and analysis.

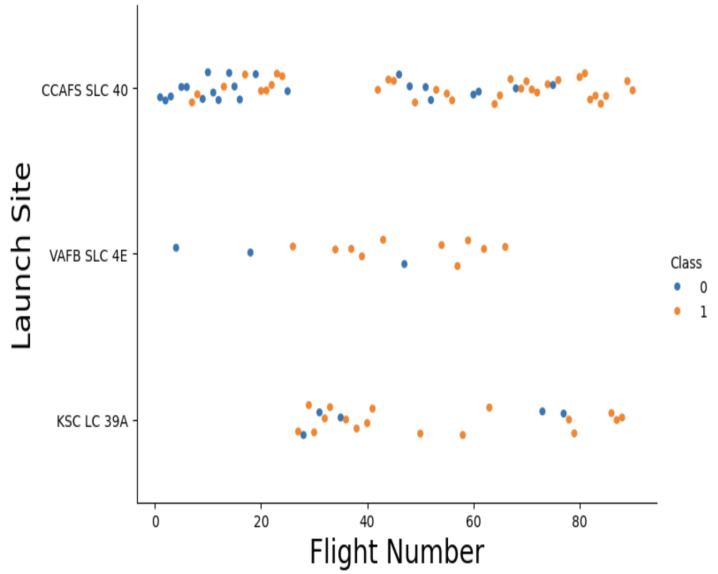
- Here is the GitHub URL of the completed data visualization related notebooks:[Github Link]

EDA with Data Visualization









EDA with SQL

During the exploratory data analysis (EDA) phase, several SQL queries were executed to extract valuable insights from the SpaceX mission dataset. The queries focused on revealing unique launch site names involved in the missions, identifying specific records where launch sites begin with the prefix 'CCA', calculating the total payload mass carried by boosters launched under NASA's Commercial Resupply Services (CRS), and determining the average payload mass for the booster version F9 v1.1. These queries helped in understanding the distribution of launch activities across different sites, analyzing mission specifics related to launch site naming conventions, assessing payload contributions by a key customer (NASA), and comparing payload characteristics across different booster versions, thereby laying a foundation for deeper analysis and decision-making in the project.[Github Link]

Task 1

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

```
[9]: %sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

[9]:

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

Task 2

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

```
[11]: %sql SELECT * FROM 'SPACEXTBL' WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

Build an Interactive Map with Folium

- Created a Folium map to visualize all the launch sites, utilizing map objects like markers, circles, and lines to denote the success or failure outcomes of launches at each site. Additionally, a launch outcome indicator was created, where failure was represented as 0 and success as 1, providing a clear visual representation of mission outcomes across different launch locations.
- Here is the GitHub URL of the completed interactive map with folium related notebooks:[Github Link]

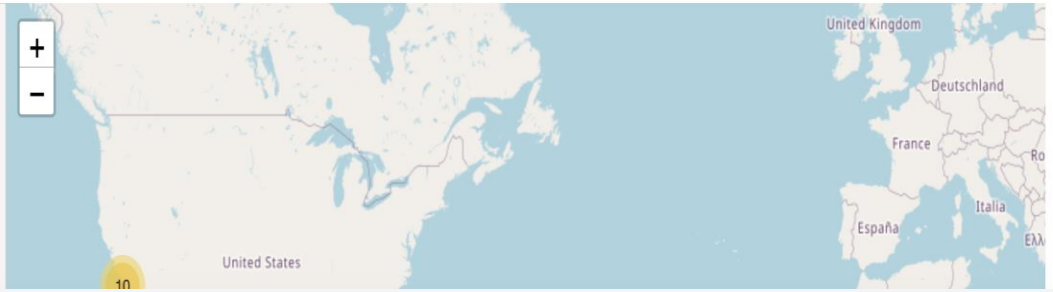
TODO: For each launch result in `spacex_df` data frame, add a `folium.Marker` to `marker_cluster`

[13]:
```
# Add marker_cluster to current site_map
site_map.add_child(marker_cluster)

# for each row in spacex_df data frame
# create a Marker object with its coordinate
# and customize the Marker's icon property to indicate if this launch was successed or failed,
# e.g., icon=folium.Icon(color='white', icon_color=row['marker_color']
for lat, lng, label, color in zip(spacex_df['Lat'], spacex_df['Long'], spacex_df['Launch Site'], spacex_df['marker_colo
    # TODO: Create and add a Marker cluster to the site map
    # marker = folium.Marker(...)
    coordinate = [lat, lng]
    marker = folium.Marker(
        coordinate,
        icon=folium.Icon(color='white', icon_color=color),
        popup=label
    )
    marker_cluster.add_child(marker)

site_map
```

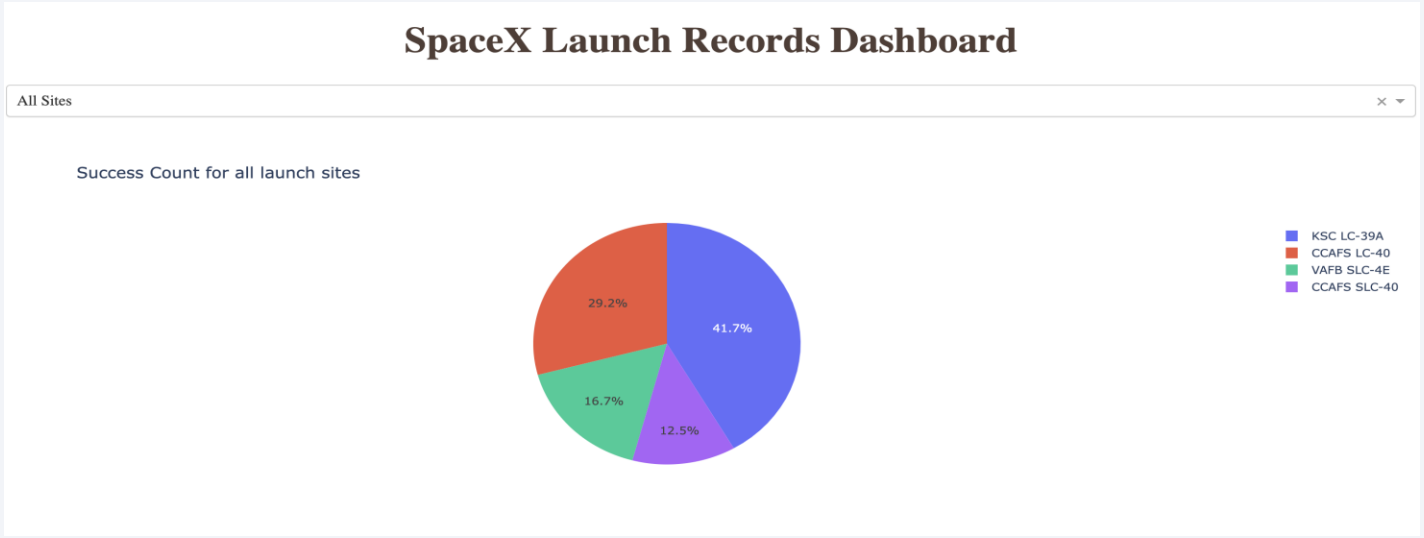
[13]:



Build a Dashboard with Plotly Dash

- An interactive dashboard application was developed using Plotly Dash, featuring a Launch Site drop-down input component for selecting specific launch sites. A callback function was implemented to dynamically render a success-pie-chart based on the selected launch site, providing a visual breakdown of launch success rates. Additionally, a Range Slider was incorporated to allow users to select payload ranges of interest. Another callback function was set up to generate a success-payload-scatter-chart scatter plot based on the chosen payload range, offering insights into the relationship between payload mass and launch success across different scenarios. These interactive features enhance user interaction and facilitate deeper exploration of SpaceX mission data directly through the dashboard interface.
- Here is the GitHub URL of the completed interactive map with folium related notebooks:[Github Link]

Build a Dashboard with Plotly Dash



Predictive Analysis (Classification)

The process began with loading the data into a Pandas DataFrame and conducting exploratory data analysis to define the training labels derived from the 'Class' column, which was converted into a NumPy array. To prepare the feature dataset (X) for modeling, standardization was applied using Scikit-learn's `StandardScaler`. Subsequently, the dataset was split into training and testing sets using `train_test_split`, ensuring 20% of the data was reserved for testing with a random state set to 2.

For predictive analysis, SVM, Classification Trees, k-nearest neighbors, and Logistic Regression models were evaluated. Each model underwent hyperparameter tuning using GridSearchCV with a 10-fold cross-validation. The best parameters and validation accuracy were identified using `best_params_` and `best_score_`, respectively. The performance of each model was then assessed on the test data using accuracy scores, and confusion matrices were generated to visualize model outcomes. This rigorous approach enabled comparison of model performances, determining the best classifier for the dataset based on test data accuracy.

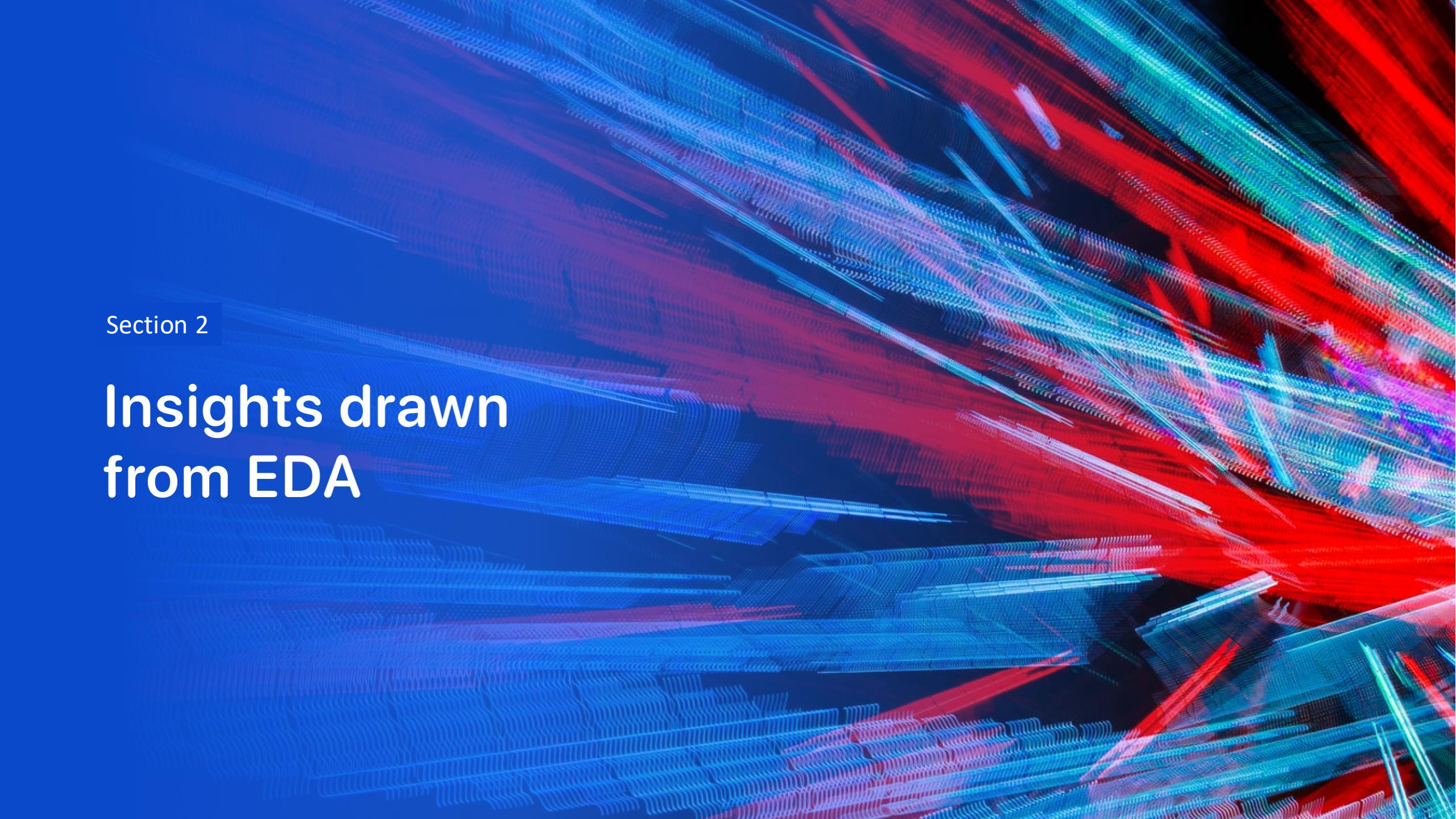
Predictive Analysis (Classification)

[37]:

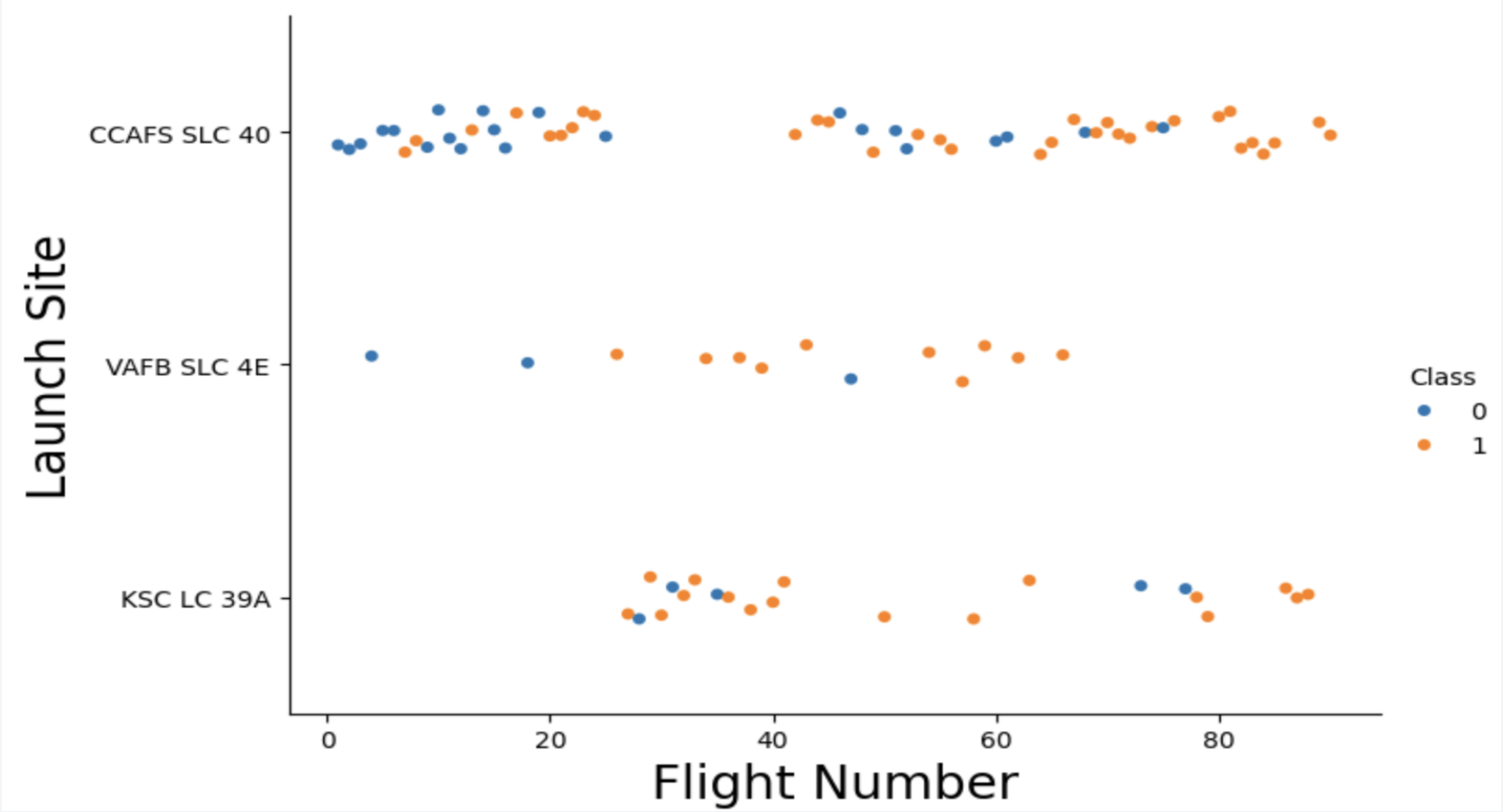
	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Results

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


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



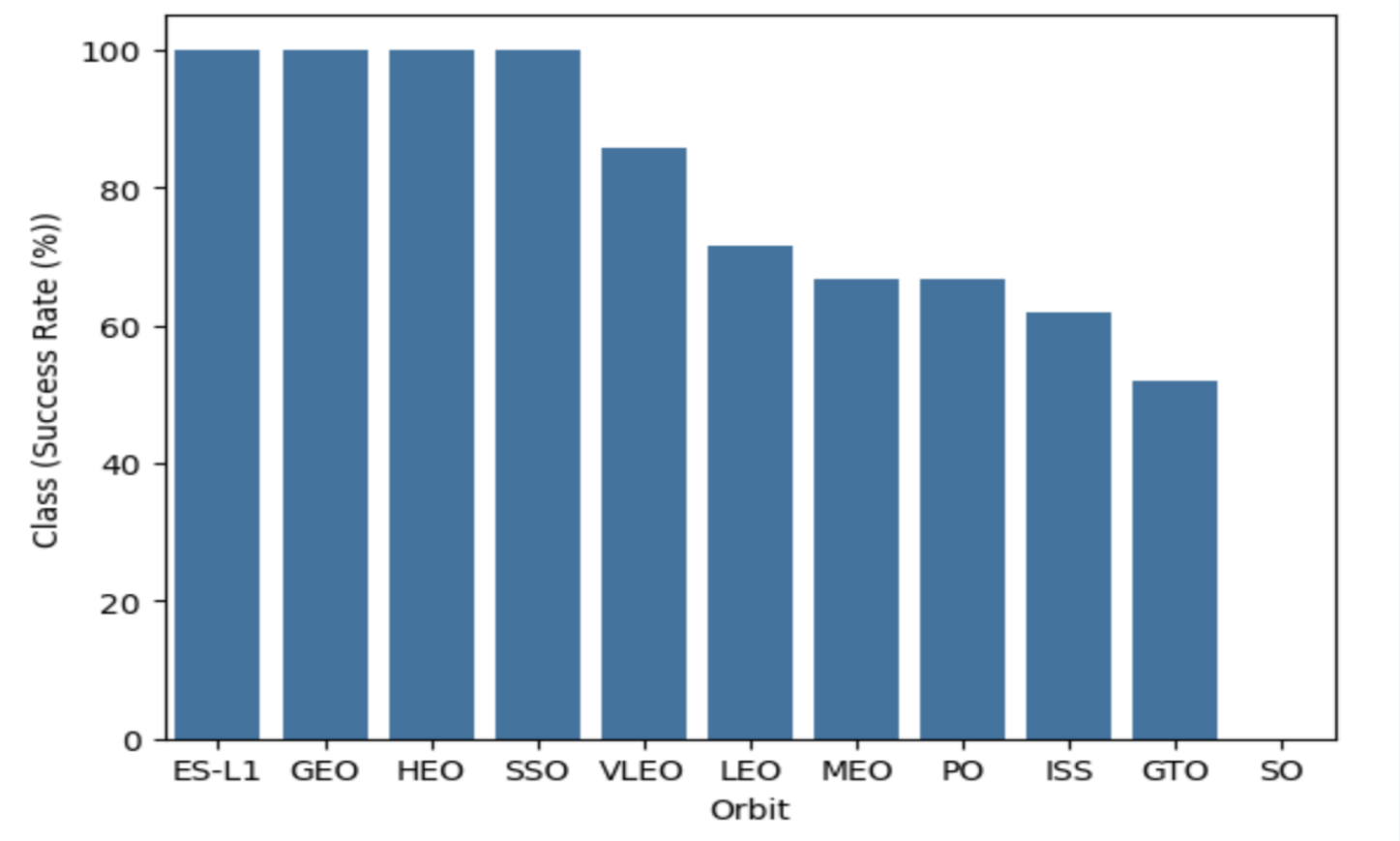
Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

We can deduce that, as the flight number increases in each of the 3 launcg sites, so does the success rate. For instance, the success rate for the VAFB SLC 4E launch site is 100% after the Flight number 50. Both KSC LC 39A and CCAFS SLC 40 have a 100% success rates after 80th flight.

Payload vs. Launch Site

Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

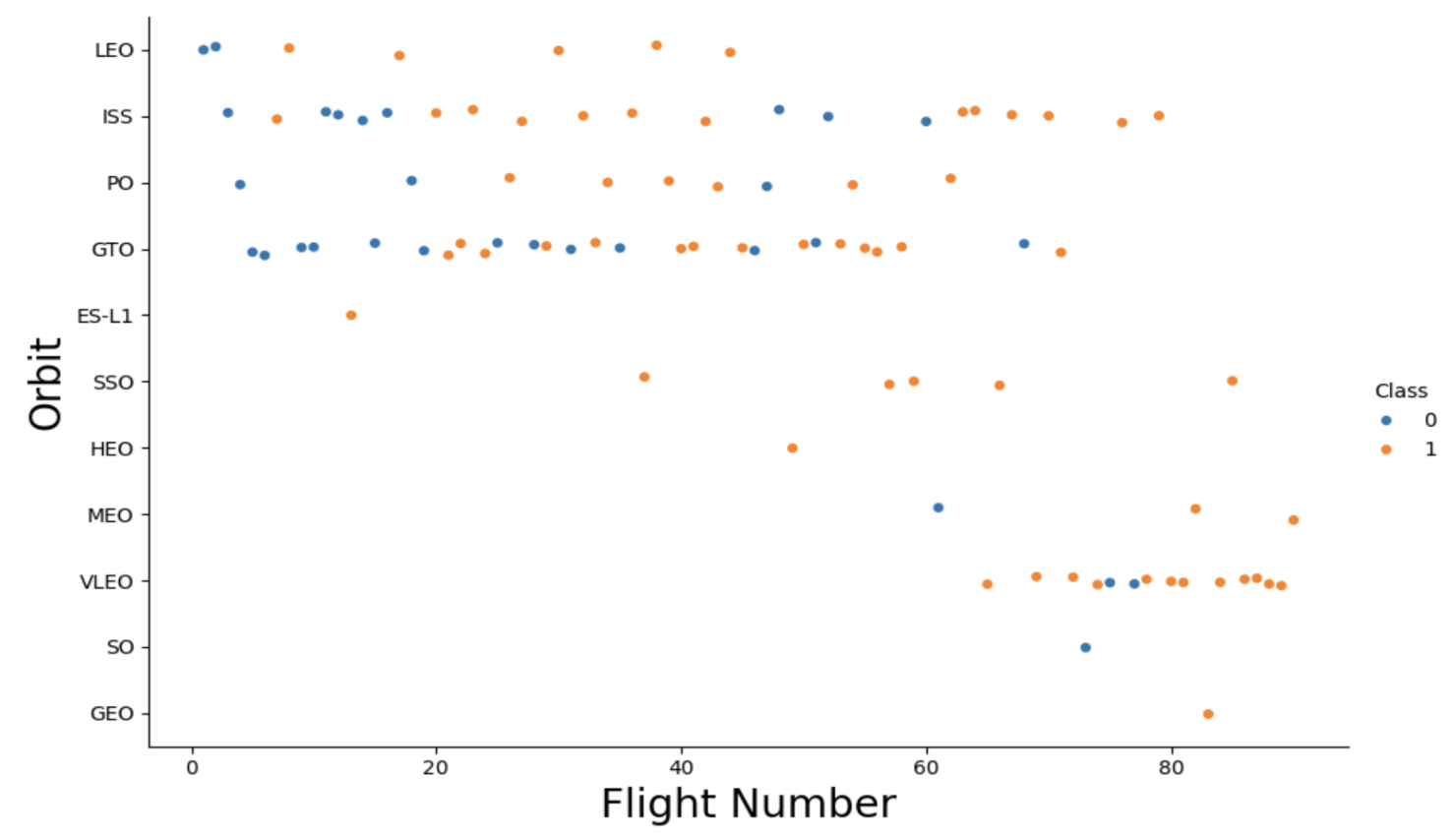
Success Rate vs. Orbit Type



Analyze the ploted bar chart try to find which orbits have high sucess rate.

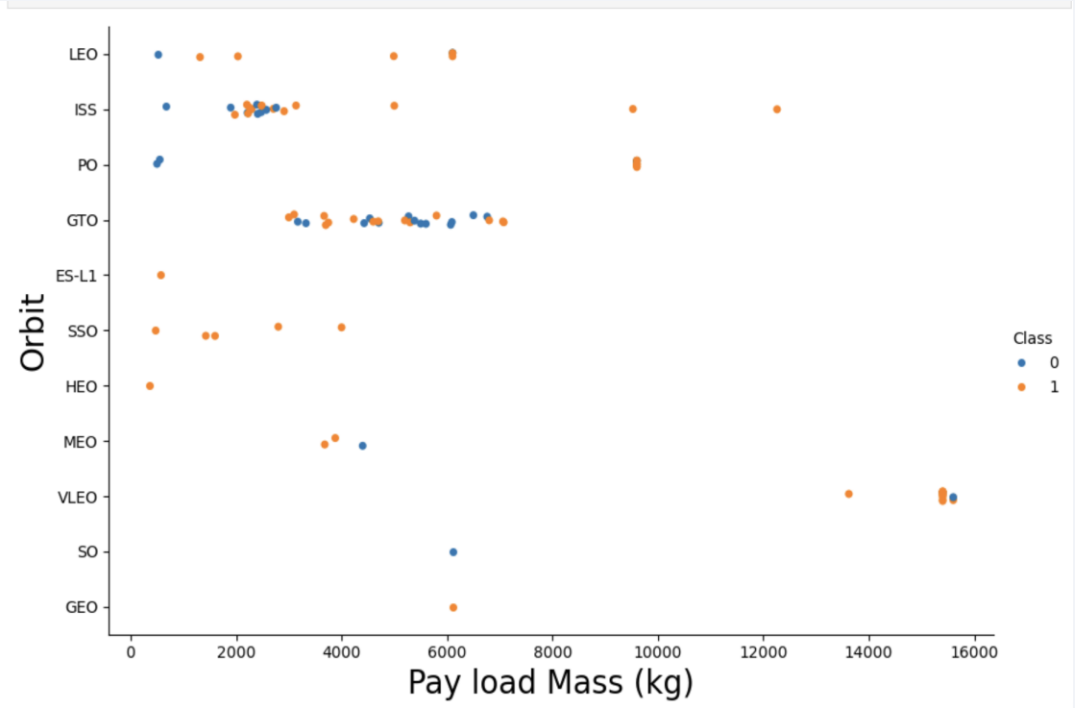
Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%, with SO orbit having the lowest success rate at ~50%. Orbit SO has 0% success rate.

Flight Number vs. Orbit Type



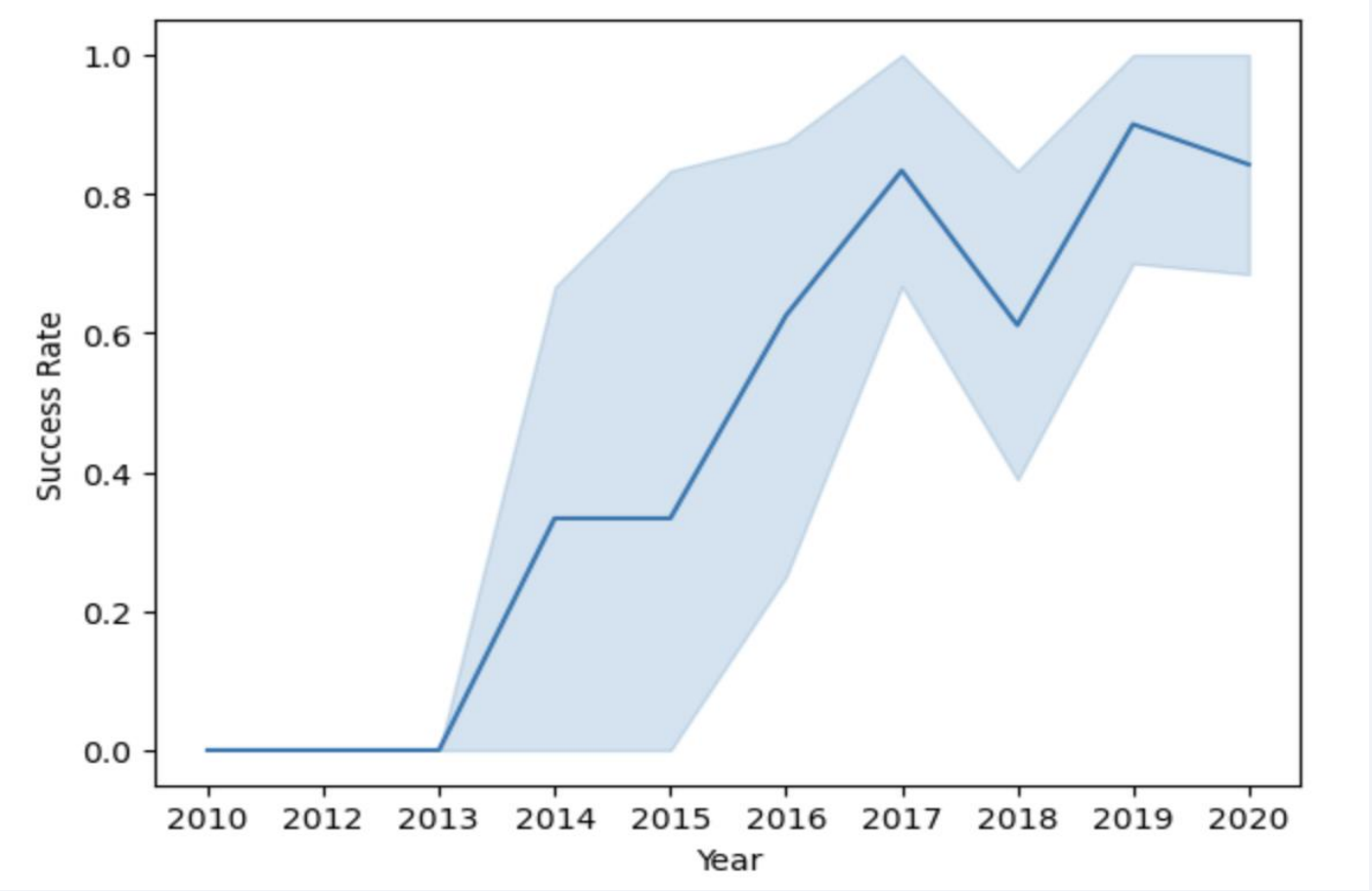
You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



Missions with heavy payloads aimed at Polar orbits, Low Earth Orbit (LEO), and the International Space Station (ISS) typically achieve higher success rates for landing. These orbits benefit from more predictable environmental conditions and operational factors conducive to successful landings. In contrast, missions targeting Geostationary Transfer Orbit (GTO) exhibit a more even distribution between successful and unsuccessful outcomes, indicating greater variability and complexity in achieving successful landings for payloads destined for higher orbits.

Launch Success Yearly Trend



Since 2013, the success rate kept going up till 2020

All Launch Site Names

To retrieve the names of unique launch sites from the `LAUNCH_SITE` column of the `SPACEXTBL` table, the SQL query utilized the `SELECT DISTINCT` statement. This query specifically returns only distinct values from the `LAUNCH_SITE` column, ensuring that each launch site is listed only once in the result set. This approach effectively identifies and lists all unique launch sites recorded in the dataset, providing a clear and concise summary of the available launch locations without duplication.

[9]:

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

Launch Site Names Begin with 'CCA'

This query selects all columns (`*`) from the `SPACEXTBL` table where the `LAUNCH_SITE` column begins with the string 'CCA'. The `LIKE` operator with the pattern 'CCA%' is used to filter rows where the launch site names start with 'CCA'. The `LIMIT 5` clause ensures that only the first 5 records meeting this criteria are returned in the query result. This approach efficiently retrieves and displays specific launch records based on the specified condition.

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

```
[11]: %sql SELECT * FROM 'SPACEXTBL' WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

[11]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcon
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachut
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 (parachut
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attem
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attem
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attem

Total Payload Mass

This SQL query calculates the sum of the `Payload_Mass__KG_` column from the `SPACEXTBL` table, filtering records where the `Customer` column begins with 'NASA'. The `LIKE 'NASA%'` condition ensures that all variations of NASA's customer names (e.g., NASA, NASA (CRS)) are included in the calculation. This query effectively computes the total payload mass carried by boosters for missions conducted by NASA, providing a consolidated figure for analysis or reporting purposes.

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

```
[12]: %sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';
```

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

[12]:

Total Payload Mass(Kgs)	Customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

This SQL query calculates the average (`AVG`) payload mass (`Payload_Mass__KG_`) carried by booster version 'F9 v1.1' from the `SPACEXTBL` table. The `WHERE` clause filters records where the `Booster_Version` column matches 'F9 v1.1', ensuring that only data related to this specific booster version is considered in the average calculation. This query provides a numerical average value representing the typical payload mass carried by the Falcon 9 version 1.1 booster across its recorded missions.

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

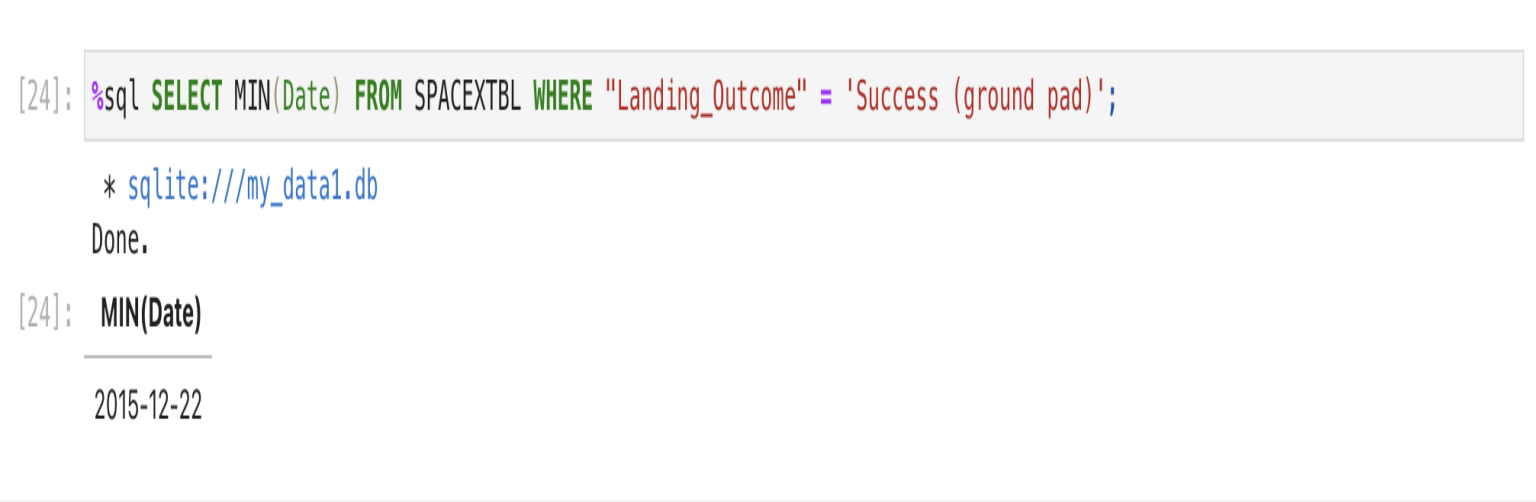
```
[13]: %sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Versi
 * sqlite:///my_data1.db
Done.
```

[13]:

Payload Mass Kgs	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

First Successful Ground Landing Date

This SQL query selects the minimum (`MIN`) date (`Date` column) from the `SPACEXTBL` table where the `Landing_Outcome` column contains the phrase 'Success (ground pad)'. The `LIKE '%Success (ground pad)%'` condition ensures that dates corresponding to successful ground pad landings are retrieved. This query effectively identifies the earliest date on which SpaceX achieved a successful landing of a Falcon 9 booster on a ground pad, providing historical context to the achievement.



Successful Drone Ship Landing with Payload between 4000 and 6000

This SQL query retrieves distinct (`DISTINCT`) values from the `Booster_Version` column of the `SPACEXTBL` table where the `Landing_Outcome` column indicates a successful landing on a drone ship (`LIKE '%Success (drone ship)%'`) and the `Payload_Mass__KG_` column value is between 4000 kg and 6000 kg (`> 4000 AND < 6000`). The query filters and lists only those boosters that meet both criteria, providing a clear list of boosters that successfully landed on a drone ship within the specified payload mass range.

```
[25]: %sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (ground pad)' AND PAYLOA
 * sqlite:///my_data1.db
Done.
```

[25]:

Booster_Version	Payload
F9 FT B1032.1	NROL-76
F9 B4 B1040.1	Boeing X-37B OTV-5
F9 B4 B1043.1	Zuma

Total Number of Successful and Failure Mission Outcomes

This SQL query retrieves the count of records (`COUNT(*)`) for each distinct `Mission_Outcome` value from the `SPACEXTBL` table. The `GROUP BY Mission_Outcome` clause groups the results by the `Mission_Outcome` column, separating records into categories based on their mission outcomes (e.g., 'Success', 'Failure'). This query provides a summary of the total number of successful and failure mission outcomes recorded in the dataset, offering insights into the overall success rates of SpaceX missions based on the available data.

```
[26]: %sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
 * sqlite:///my_data1.db
Done.
```

[26]:

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

Boosters Carried Maximum Payload

This SQL query first calculates the maximum payload mass (`MAX(Payload_Mass__KG_)`) from the `SPACEXTBL` table using a subquery. It then selects the `Booster_Version` from the `SPACEXTBL` table where the `Payload_Mass__KG_` matches the maximum payload mass identified in the subquery. This approach ensures that only the booster(s) associated with the maximum payload mass are retrieved. It provides a clear and direct way to identify which booster(s) have carried the heaviest payloads based on the dataset

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

```
[27]: %sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLO
```

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

[27]:

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

2015 Launch Records

This SQL query filters records from the `SPACEXTBL` table where the `Date` column year is 2015 (`YEAR(Date) = 2015`) and the `Landing_Outcome` column indicates a failure on a drone ship (`LIKE '%Failure (drone ship)%'`). It selects the `Landing_Outcome`, `Booster_Version`, and `Launch_Site` columns for these records. This approach retrieves specific details about failed landing attempts on drone ships during the year 2015, providing insights into booster versions and launch site locations associated with these unsuccessful missions.

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.

[44]:
```
%%sql SELECT
    strftime('%Y-%m', Date) AS month,
    Booster_Version,
    Launch_Site,
    Landing_Outcome
FROM
    SPACEXTBL
WHERE
    strftime('%Y', Date) = '2015'
    AND Landing_Outcome = 'Failure (drone ship)';
```

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

[44]:

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

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

This SQL query retrieves the count of each distinct `Landing_Outcome` from the `SPACEXTBL` table where the `Date` falls between '2010-06-04' and '2017-03-20' (`Date BETWEEN '2010-06-04' AND '2017-03-20'`). The `GROUP BY Landing_Outcome` clause groups the results by the `Landing_Outcome` column, calculating the number of occurrences for each landing outcome category (e.g., success or failure). The `ORDER BY Outcome_Count DESC` sorts the results in descending order based on the count of landing outcomes, presenting the most frequent outcomes first. This query provides a ranked list of landing outcomes during the specified date range, offering insights into the distribution and frequency of mission outcomes within that period.

[46]:
```
%%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.
```

[46]:

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

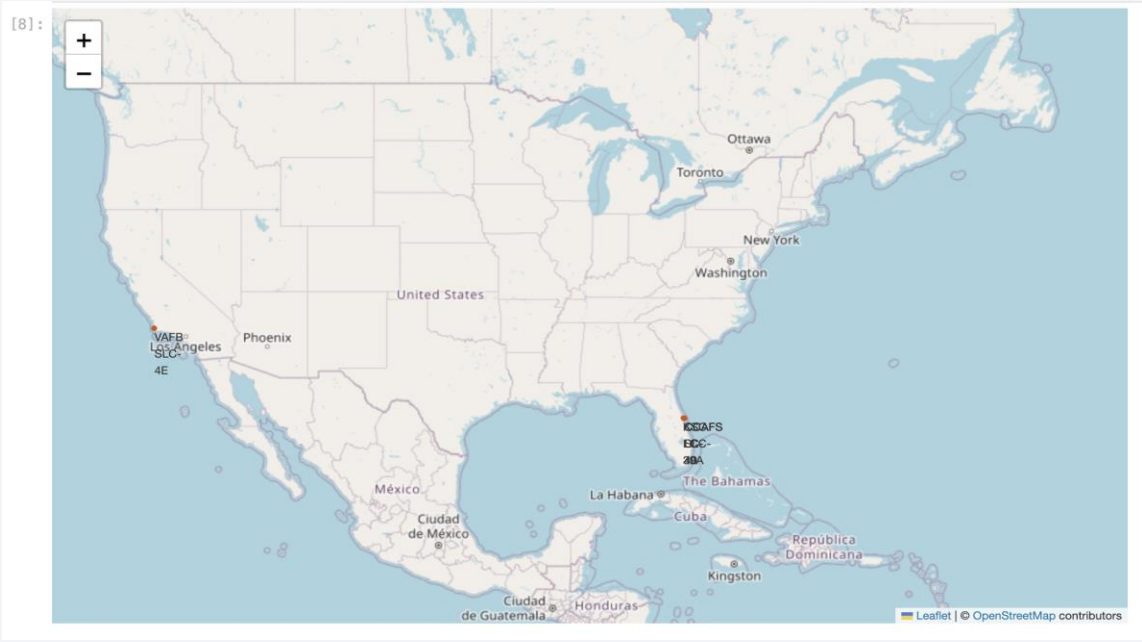
Would you like to receive official Jupyter

Section 3

Launch Sites Proximities Analysis

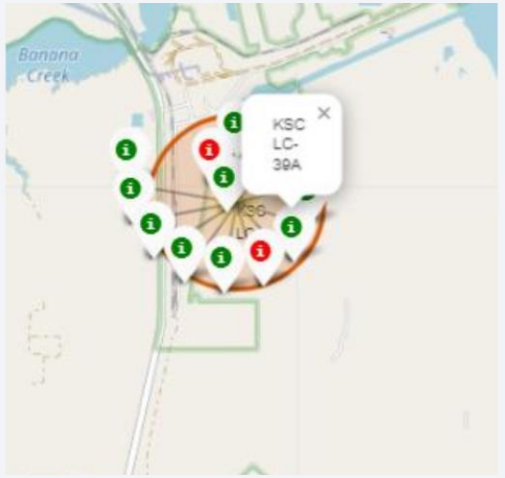
Markers of all launch sites on global map

The screenshot titled "Global Distribution of SpaceX Falcon 9 Launch Sites" displays a Folium-generated map showcasing markers representing locations where Falcon 9 rockets have been launched worldwide. This visualization offers insights into SpaceX's operational strategy, illustrating their capability to conduct missions from various geographic points to accommodate diverse orbital trajectories. It provides a clear geographical context of SpaceX's global footprint in the aerospace sector, highlighting both land-based and sea-based (drone ship) launch sites and emphasizing their strategic positioning for efficient and effective mission deployments.



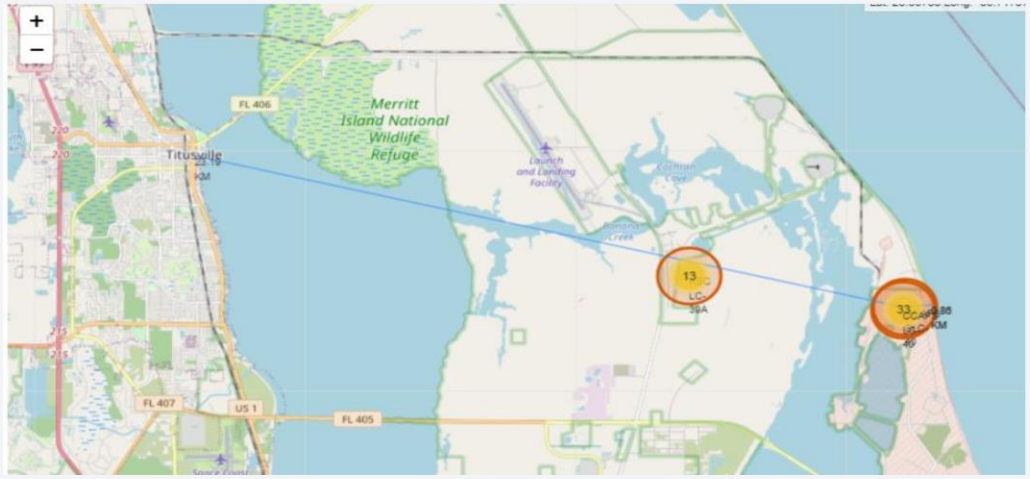
Launch Outcomes or Each Site on the Map with Color Mark

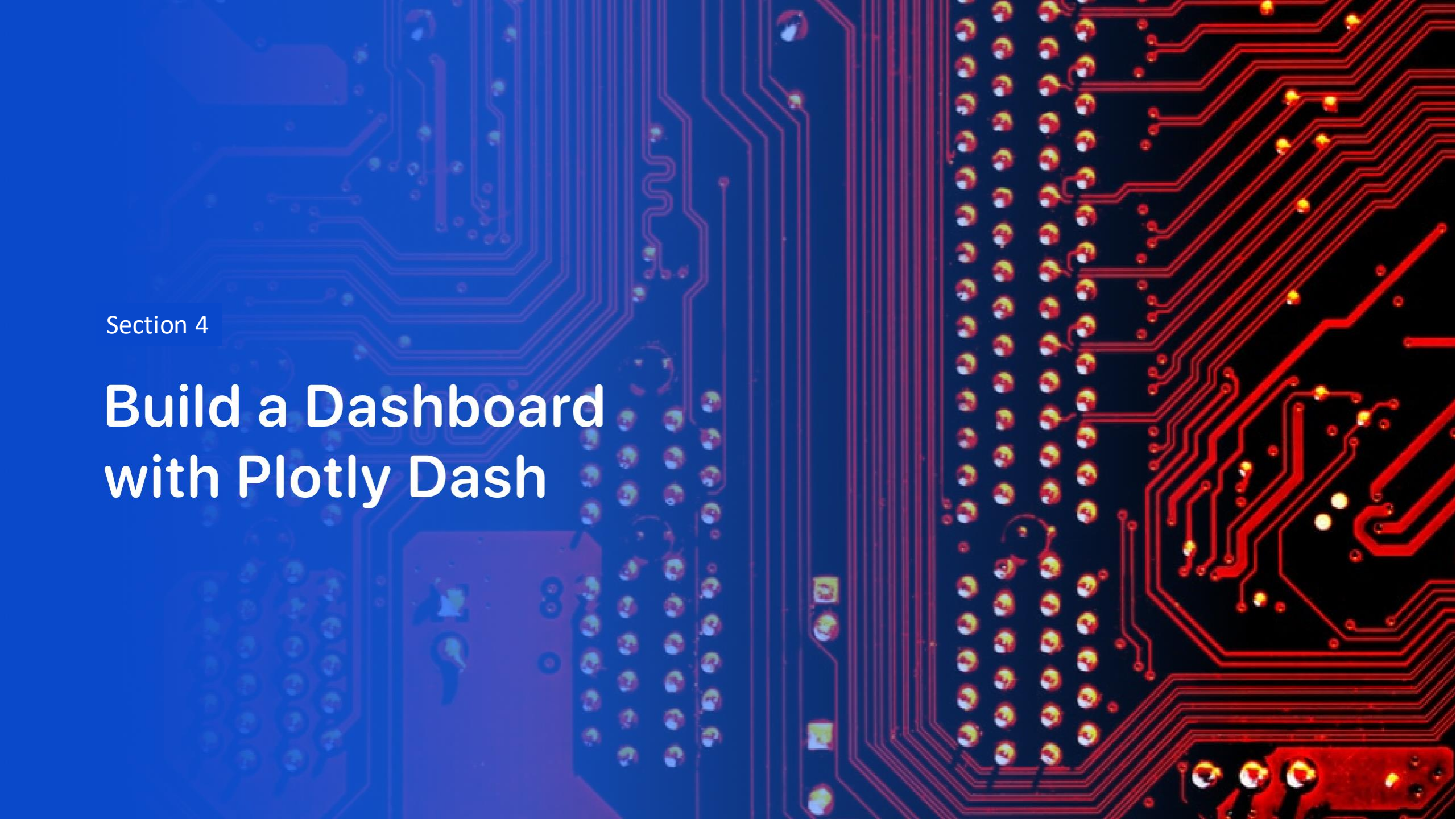
The screenshot titled "SpaceX Falcon 9 Launch Outcomes Worldwide" displays a Folium-generated map where markers represent SpaceX Falcon 9 launch sites globally, color-coded to indicate different mission outcomes. This visualization provides a clear overview of where SpaceX has achieved successful landings on ground pads and drone ships, as well as instances of mission failures. By visually mapping these outcomes, the screenshot highlights geographical patterns in SpaceX's operational performance, offering insights into success rates across various launch locations and areas for potential operational enhancements.



Distances Between a Launch Site to its Proximities

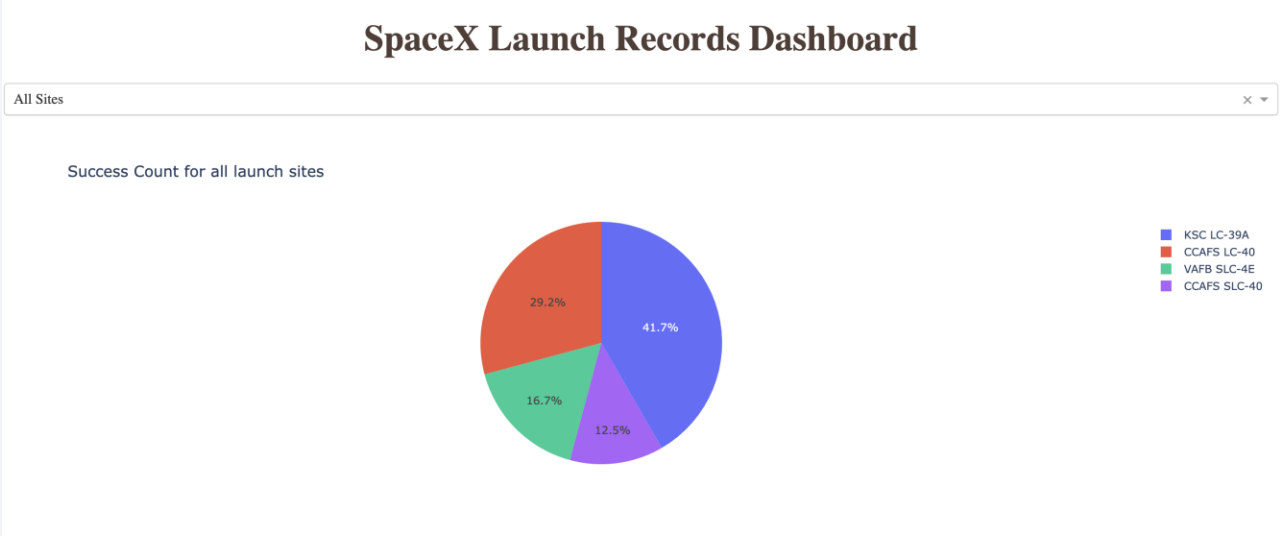
The screenshot titled "Proximity Analysis of Selected SpaceX Falcon 9 Launch Site" depicts a Folium map focusing on a specific launch site for SpaceX Falcon 9 missions. It visually highlights the site's proximity to nearby railways, highways, and coastlines, with distances calculated and displayed for each feature. This visualization provides essential spatial context, helping to assess logistical considerations and environmental factors that could influence mission planning and operational efficiency. It offers valuable insights into the site's accessibility and strategic positioning relative to critical infrastructure, aiding in understanding the operational dynamics of SpaceX's launch operations.




Section 4
Build a Dashboard with Plotly Dash

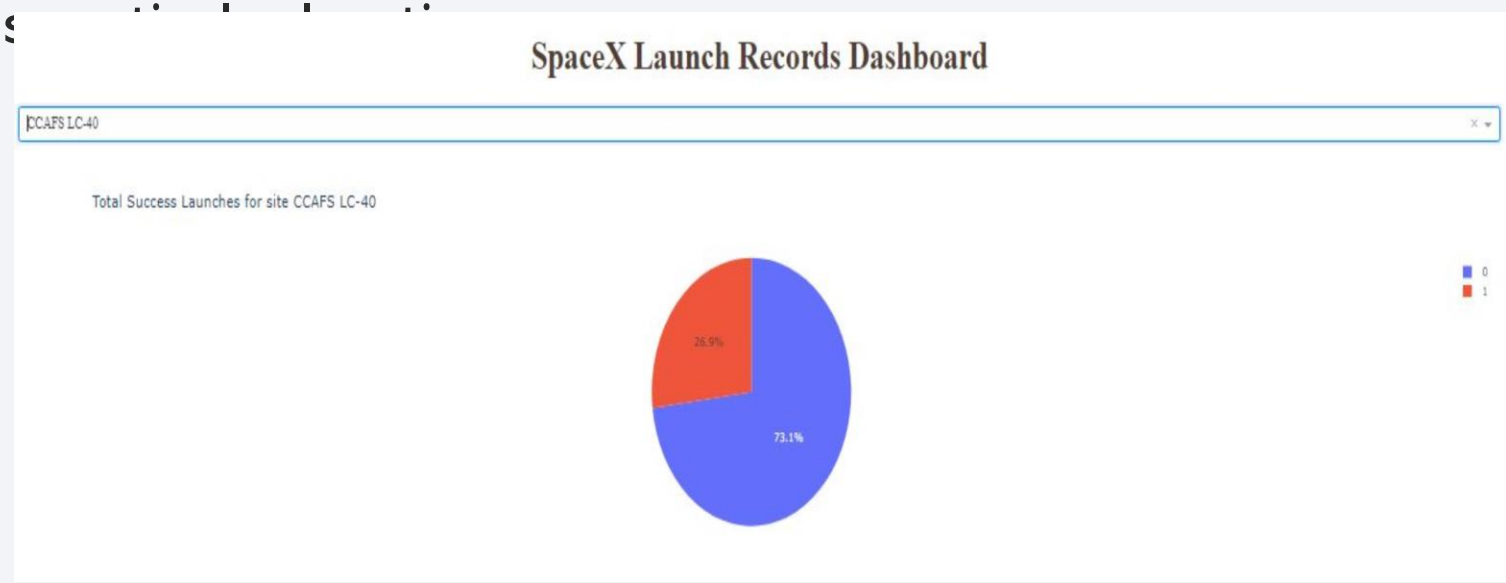
Pie-Chart for Launch Success Count for all Sites

The screenshot displays a pie chart from a dashboard, illustrating the distribution of SpaceX Falcon 9 launch success counts across all launch sites. Each segment of the pie represents a launch site, sized proportionally to its number of successful launches. This visualization offers a clear overview of success rates at different sites, highlighting locations with higher and lower success rates. It provides insights into the performance consistency across SpaceX's global launch infrastructure, aiding in strategic decision-making and operational planning for future missions.



Pie-chart for the Launch Site with 2nd Highest Launch Success Ratio

The screenshot features a pie chart from a dashboard showcasing the launch success ratio of the top-performing SpaceX Falcon 9 launch site. Each segment of the pie represents a specific outcome (success or failure) for missions conducted at this site, providing a visual representation of its success rate. This visualization highlights the site's exceptional performance in achieving successful launches, offering valuable insights into its operational efficiency and reliability compared to other launch sites. It serves as a key indicator for evaluating the effectiveness and consistency of SpaceX's operations at this



Payload vs Launch Outcome Scatter Plot for all Sites

The screenshot displays a scatter plot from a dashboard depicting the relationship between payload mass and launch outcomes (success or failure) across all SpaceX Falcon 9 launch sites. Utilizing a range slider, different payload ranges are selected to observe their impact on launch success rates. Important findings include identifying payload ranges or booster versions associated with the highest success rates, providing insights into factors influencing mission success. This visualization aids in understanding the correlation between payload characteristics and launch outcomes, informing strategic decisions regarding payload management and mission planning for future SpaceX miss

Section 5

Predictive Analysis (Classification)

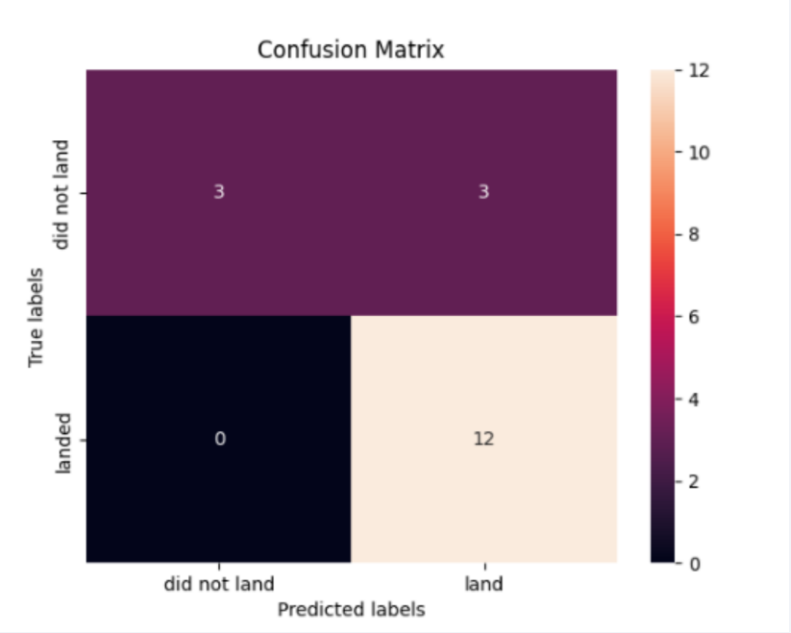
Classification Accuracy

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

All the methods perform equally on the test data: i.e. They all have the same accuracy of 0.833333 on the test Data

Confusion Matrix

All four classification models exhibited identical confusion matrices and demonstrated equal capability in distinguishing between different classes. However, a significant issue across all models was the prevalence of false positives. This common challenge indicates a tendency to incorrectly predict positive outcomes when they do not occur, highlighting a need for further model refinement or adjustment of decision thresholds to enhance accuracy in classification tasks.



Conclusions

- Launch sites exhibit varying success rates: CCAFS LC-40 achieves a 60% success rate, while KSC LC-39A and VAFB SLC-4E boast 77%. Notably, success rates tend to increase with flight numbers; for example, VAFB SLC-4E reaches 100% success after the 50th flight, and both KSC LC-39A and CCAFS SLC-40 achieve 100% after the 80th flight.

- Analyzing the Payload vs. Launch Site scatter plot reveals a distinct pattern for VAFB SLC-4E: there are no launches with heavy payload masses exceeding 10,000. This observation underscores strategic payload management practices specific to this launch site.

- Orbits like ES-L1, GEO, HEO, and SSO demonstrate impeccable success rates of 100%, whereas SO orbit records the lowest at approximately 50%, with some instances showing a 0% success rate. In LEO orbits, success rates appear correlated with the number of flights conducted, reflecting operational maturity and performance over time.

Thank you!