IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Oscar de Miguel Nieves
20 June 2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

3

Introduction

- Project background and context

  Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers

  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

5

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- 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

- The data was collected using various methods
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection - SpaceX API

1. API Call

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Receive JSON File

```
data = pd.json_normalize(response.json())
```

3. Transform Data

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

4. Create Dictionary

```
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 }
```

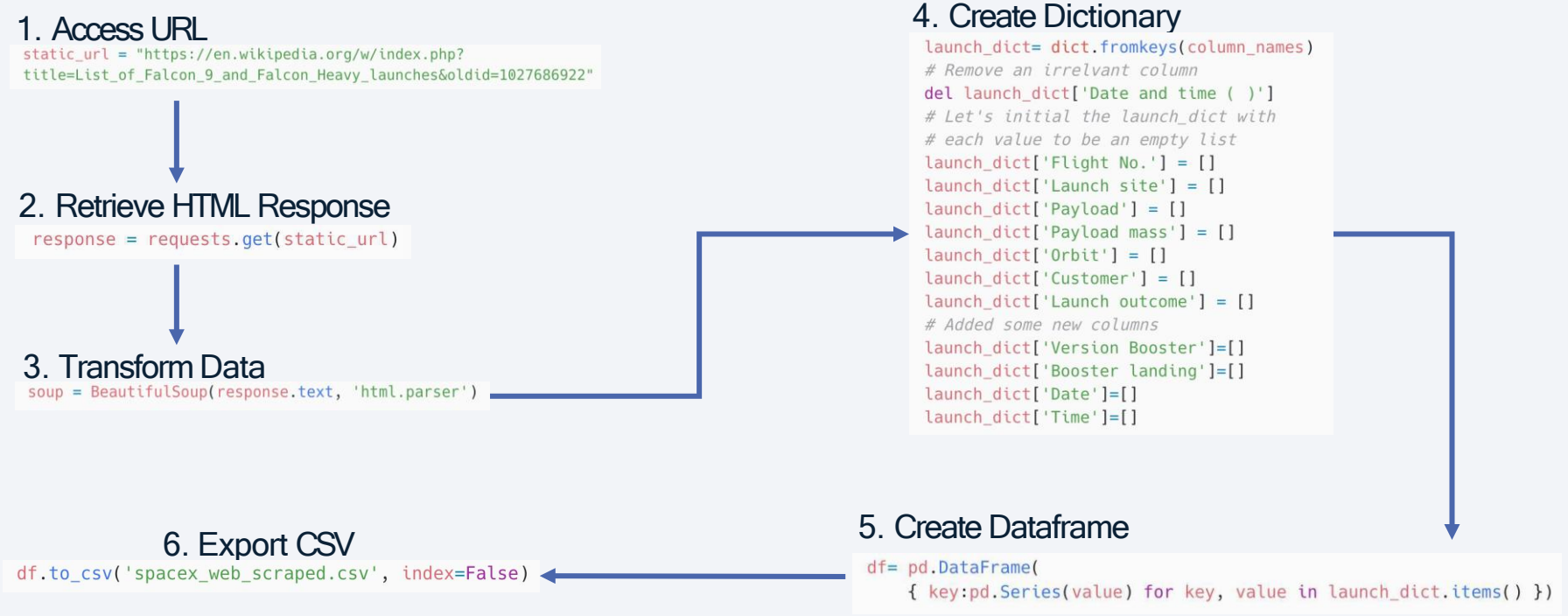
5. Create Dataframe

6. Filter Dataframe

7. Export CSV

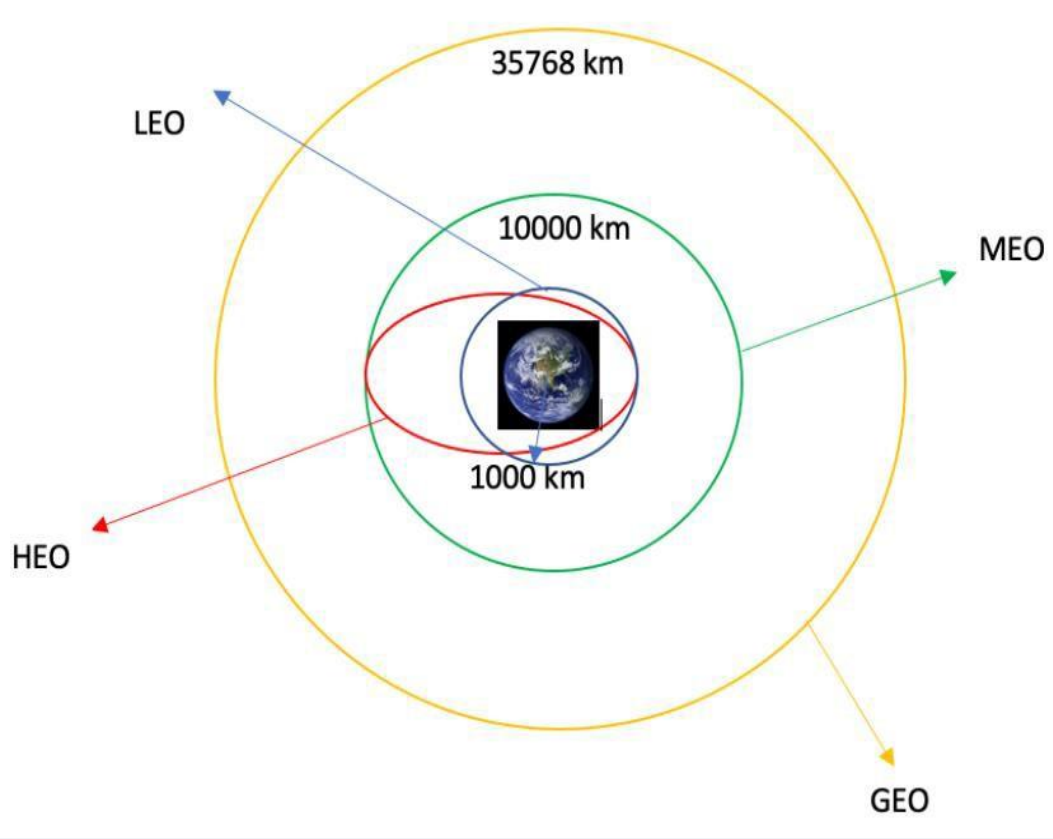
GitHub URL -> Data Collection API

Data Collection - Scraping



GitHub URL -> Data Collection Web Scraping

Data Wrangling



- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.

Data Wrangling

1. Calculating the number of launches at each site

```
launch_site_counts = df['LaunchSite'].value_counts()
print(launch_site_counts)
```

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

2. Counting occurrences of each orbit type

```
orbit_counts = df['Orbit'].value_counts()
print(orbit_counts)
```

```
Orbit
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
HEO       1
ES-L1     1
SO        1
GEO       1
Name: count, dtype: int64
```

3. Analyzing mission outcomes by orbit

```
landing_outcomes = df['Outcome'].value_counts()
print(landing_outcomes)
```

```
Outcome
True ASDS       41
None None       19
True RTLS       14
False ASDS       6
True Ocean       5
False Ocean      2
None ASDS        2
False RTLS       1
Name: count, dtype: int64
```

4. Creating a new landing outcome label from the Outcome column

```
landing_class = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1).tolist()
```

```
print(landing_class)
```

```
[0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

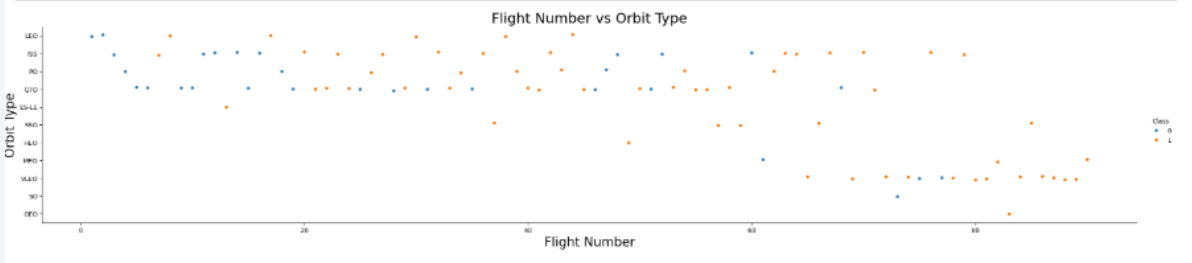
5. Export CSV

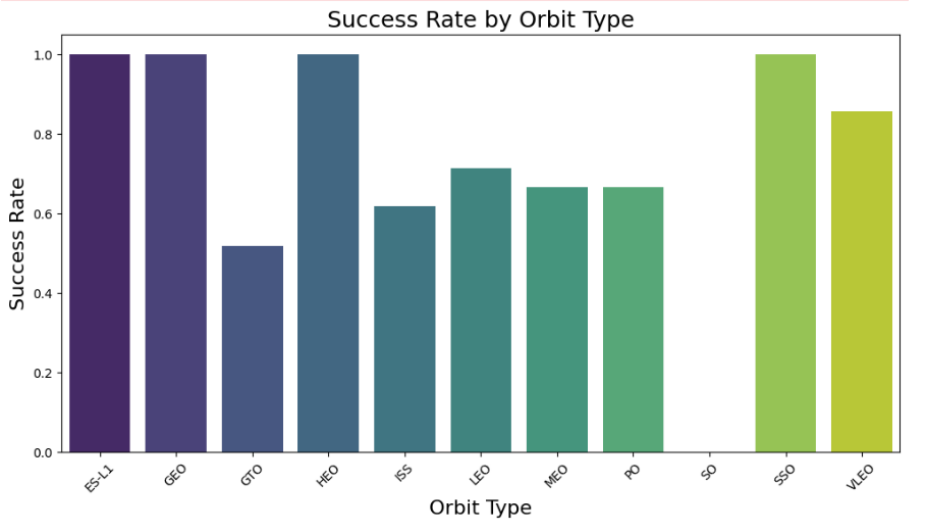
```
df.to_csv("dataset_part_2.csv", index=False)
```

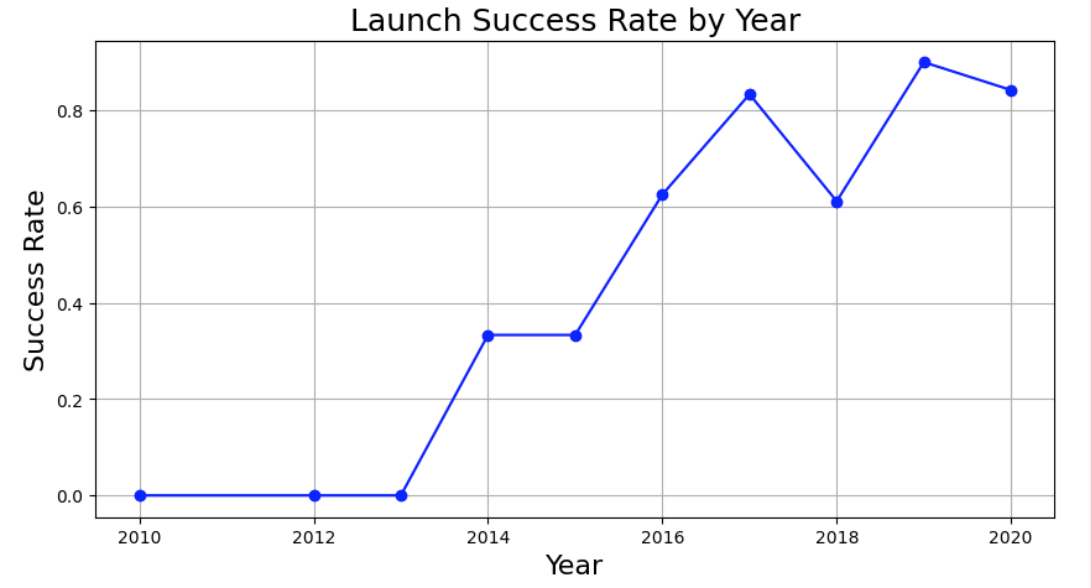
GitHub URL -> Data Wrangling

EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.







GitHub URL -> EDA with Data Visualization

EDA with SQL

The SQL queries performed were:

- Retrieved a distinct list of launch sites from the mission dataset.
- Selected five records where the launch site names begin with 'CCA'.
- Summed the payload mass carried by boosters launched by NASA (CRS).
- Calculated the average payload mass for booster version F9 v1.1.
- Determined the earliest date of a successful ground pad landing.
- Listed booster versions that achieved successful drone ship landings with payload masses between 4000 and 6000.
- Counted the total number of successful versus unsuccessful mission outcomes.
- Identified the booster versions that carried the maximum payload mass using a subquery.
- Extracted records for 2015 showing failed drone ship landings, along with corresponding booster versions and launch sites.
- Ranked landing outcome counts (e.g., Failure (drone ship), Success (ground pad)) between 2010-06-04 and 2017-03-20 in descending order.

GitHub URL -> EDA with SQL

Build an Interactive Map with Folium

Integrating diverse mapping elements, our interactive Folium map offers a comprehensive view of launch site distribution, performance outcomes, and geographical context–all within one dynamic map.

- Markers & Circles:

  Placed at each launch site (with labels/popups) to clearly identify locations.

- Marker Clusters:

  Grouped nearby markers to reduce clutter in dense areas.

- Colored Icons:

  Used **green** for successes and **red** for failures to quickly convey landing outcomes.

- Lines:

  Connected launch sites to nearby landmarks (e.g., railways, highways, cities) to show spatial relationships.

GitHub URL -> Interactive Map w/ Folium

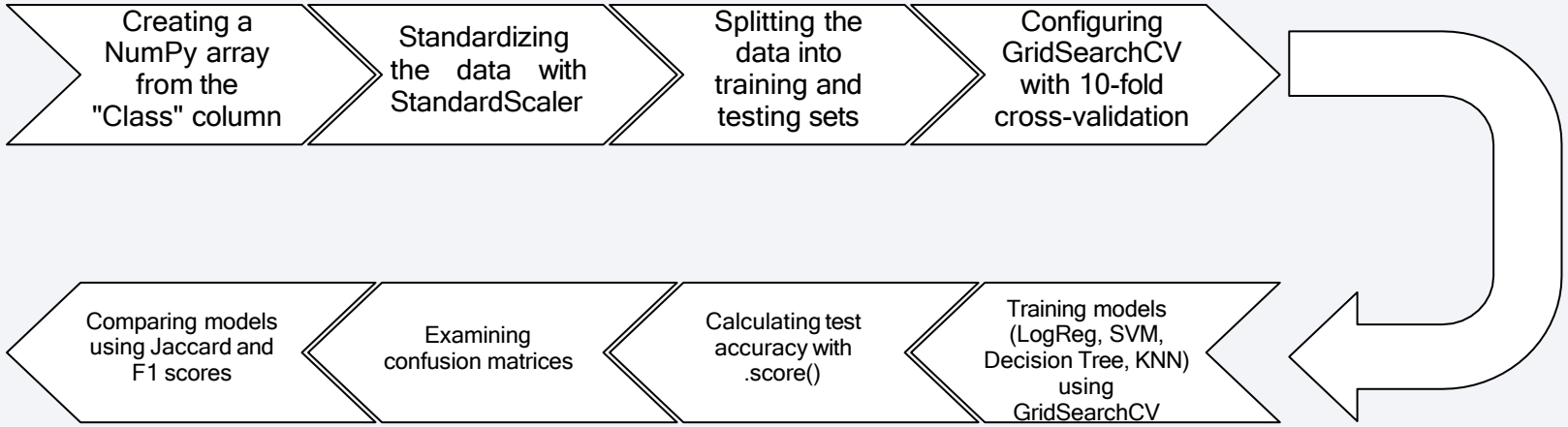
Build a Dashboard with Plotly Dash

An interactive dashboard was created to dynamically explore and analyze launch data:

- Launch Sites Dropdown:
  Provides selection of a specific launch site or all sites (using dash_core_components.Dropdown).
- Success Launches Pie Chart:
  Displays overall success counts and a breakdown of success versus failure for the chosen launch site (using plotly.express.pie).
- Payload Mass Range Slider:
  Allows filtering of data based on a specified payload mass range (using dash_core_components.RangeSlider).
- Payload Mass vs. Success Rate Scatter Chart:
  Visualizes the correlation between payload mass and launch success across different booster versions (using plotly.express.scatter).

GitHub URL -> Ploty Dash

Predictive Analysis (Classification)



GitHub URL -> Machine Learning Prediction

Results

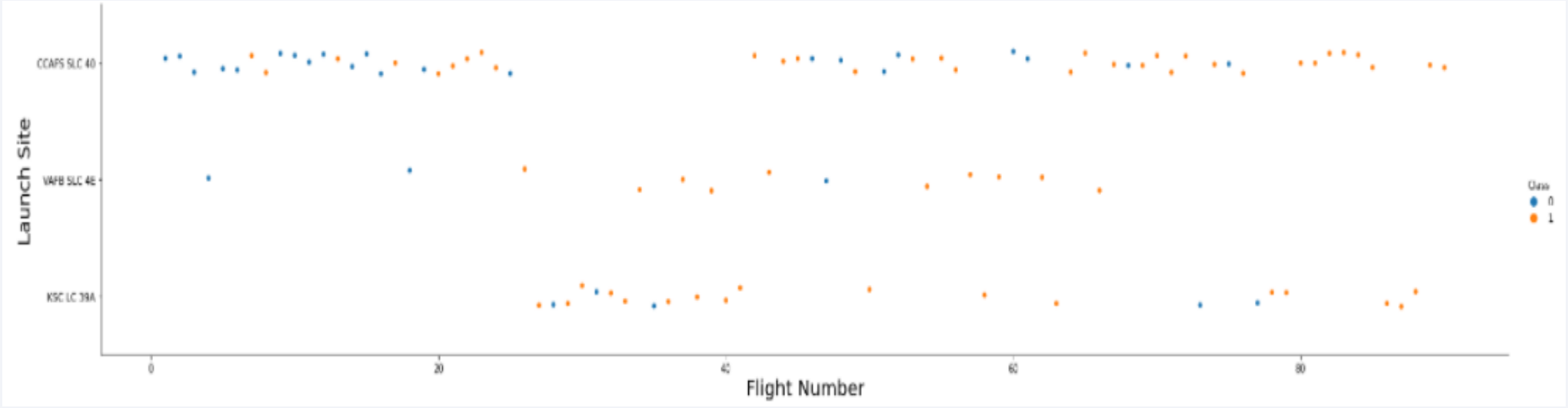
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

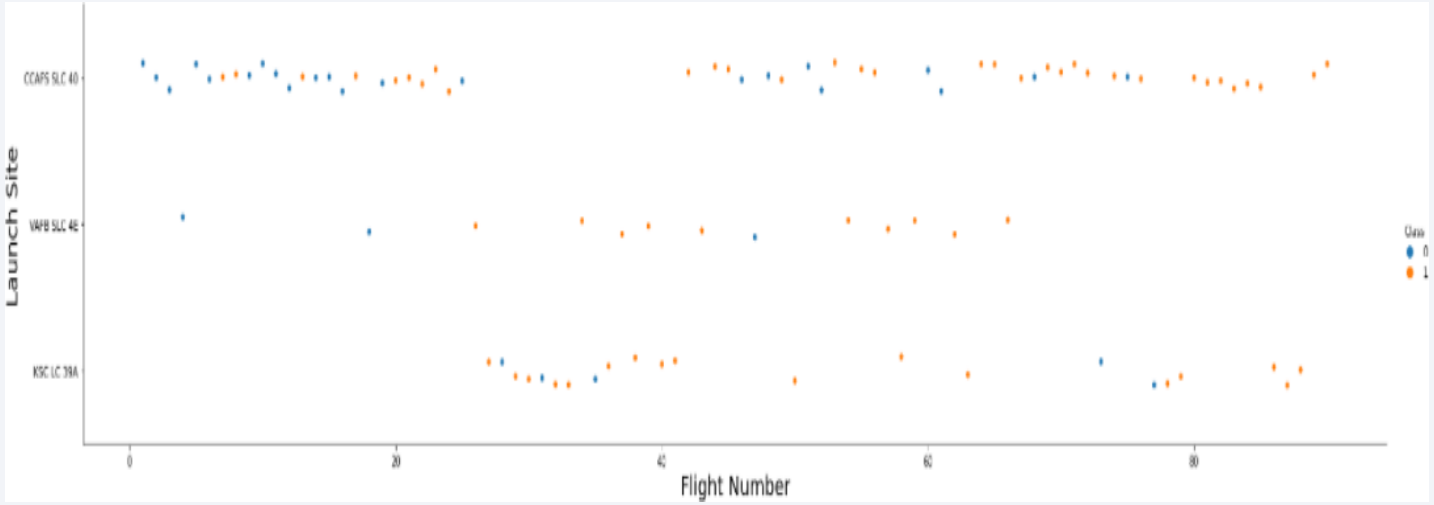
Flight Number vs. Launch Site

This plot shows flight numbers against launch sites. Early flights generally failed, while later flights succeeded more often. Also, CCAFS SLC 40 is the most frequently used launch site.



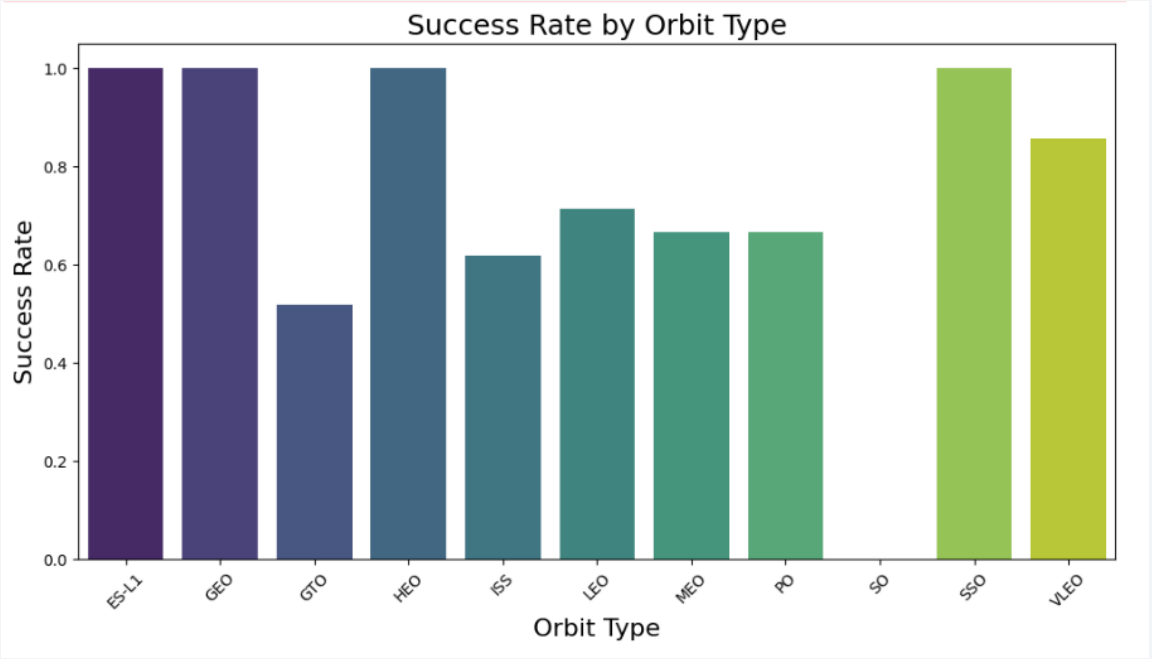
Payload vs. Launch Site

The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



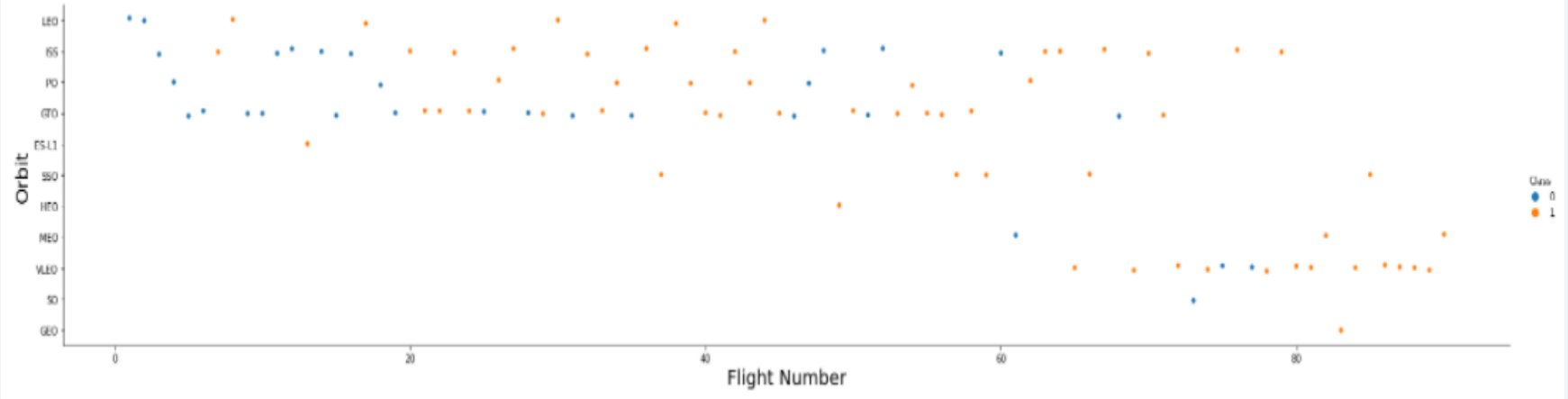
Success Rate vs. Orbit Type

From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



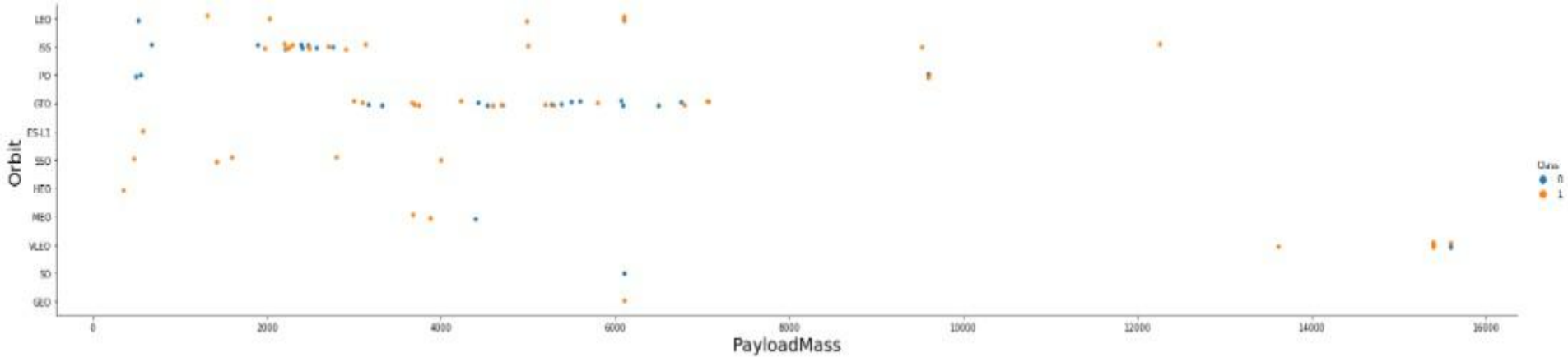
Flight Number vs. Orbit Type

This scatter plot examines flight numbers in relation to orbit types. In LEO, success appears to improve with more flights, but no clear trend is observed for orbits like GTO.



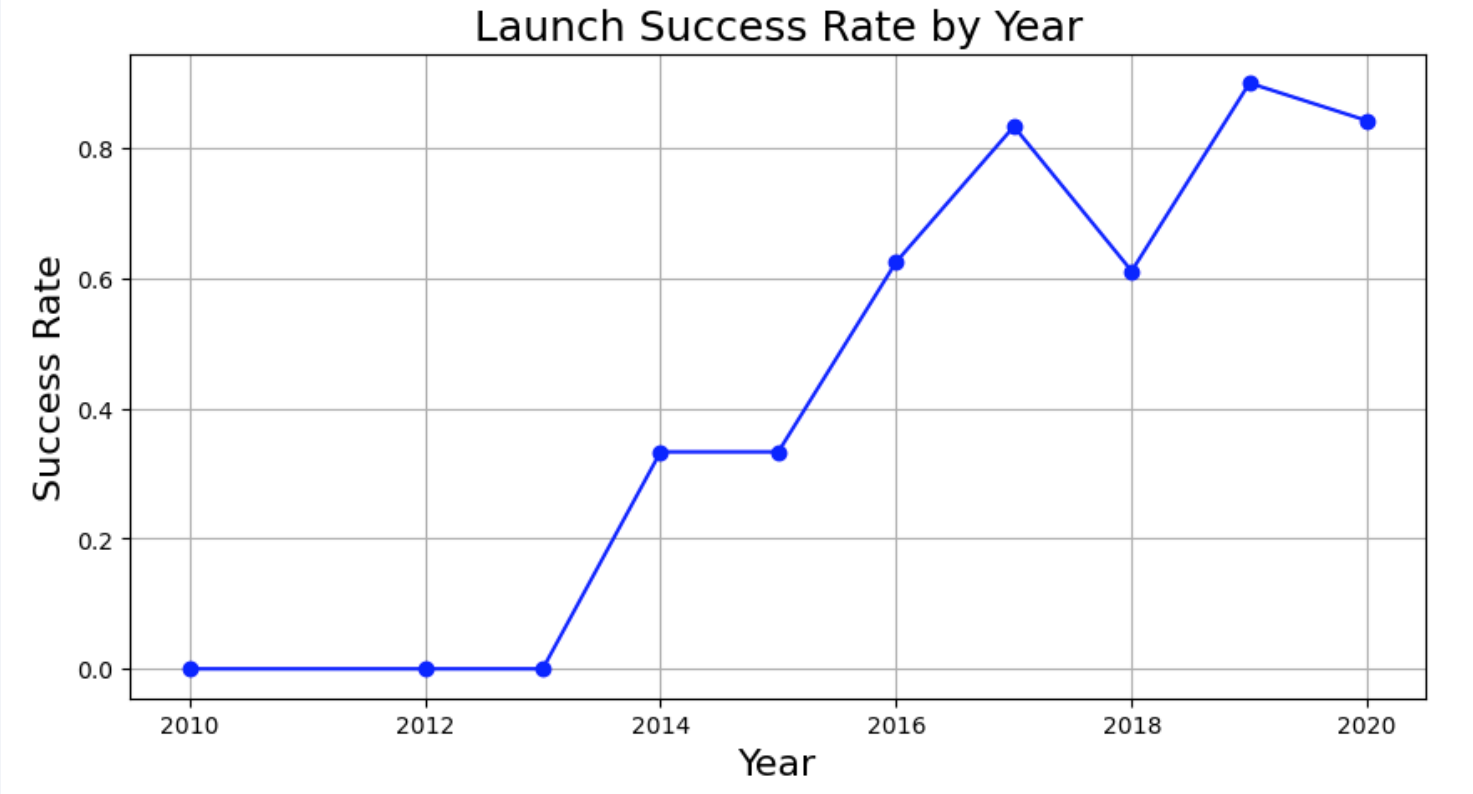
Payload vs. Orbit Type

This visualization relates payload mass to orbit type. It indicates that heavy payloads negatively impact GTO success, while they favor better outcomes in Polar and LEO (ISS) orbits.



Launch Success Yearly Trend

From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

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

In [10]:
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

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

Launch Site Names Begin with 'CCA'

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

In [11]:
```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

We used the query above to display 5 records where launch sites begin with `CCA`

Total Payload Mass

We calculated the total payload carried by boosters from NASA as 45596 using the query below

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

In [12]:
```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

Average Payload Mass by F9 v1.1

We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

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

In [13]:
```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

We observed that the dates of the first successful landing outcome on ground pad was **22nd December 2015**

In [14]:
```
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

Out[14]:

	firstsuccessfull_landing_date
0	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

In [15]:

```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

Out[15]:

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

We used wildcard like '%' to filter for **WHERE** MissionOutcome was a success or a failure.

List the total number of successful and failure mission outcomes

In [16]:
```
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

Out[16]:

The total number of successful mission outcome is:

	successoutcome
0	100

The total number of failed mission outcome is:

	failureoutcome
0	1

Boosters Carried Maximum Payload

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

In [17]:

```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

Out[17]:

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

We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

2015 Launch Records

We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

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

In [18]:

```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.
- We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

In [19]:
```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

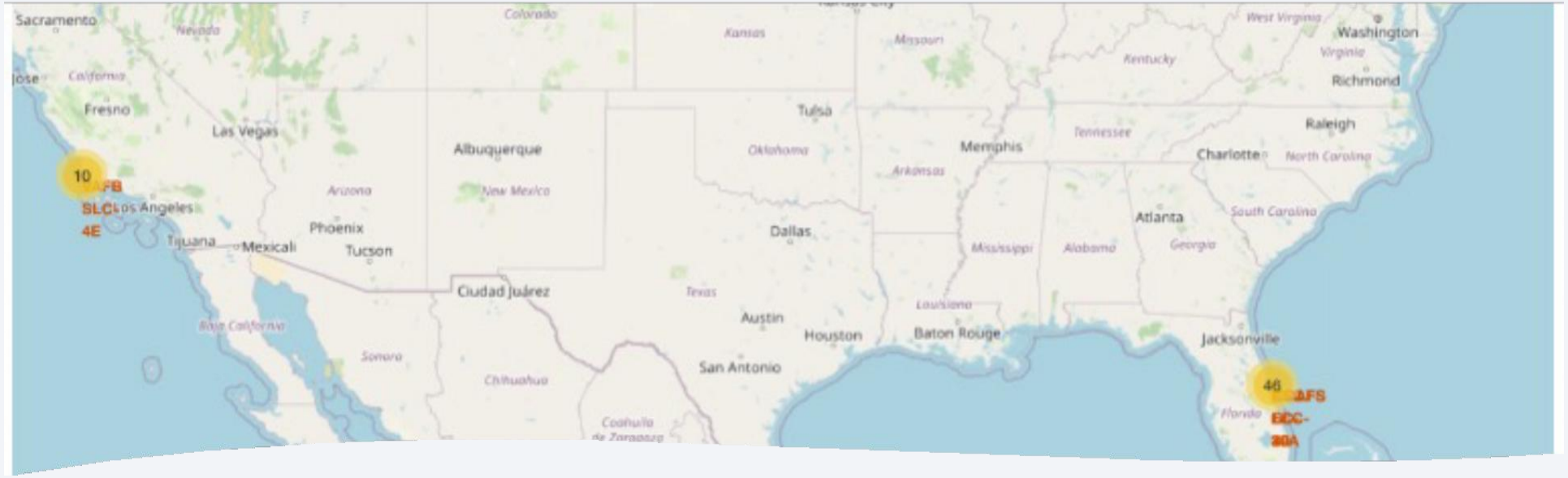
Out[19]:

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

Section 4

Launch Sites Proximities Analysis

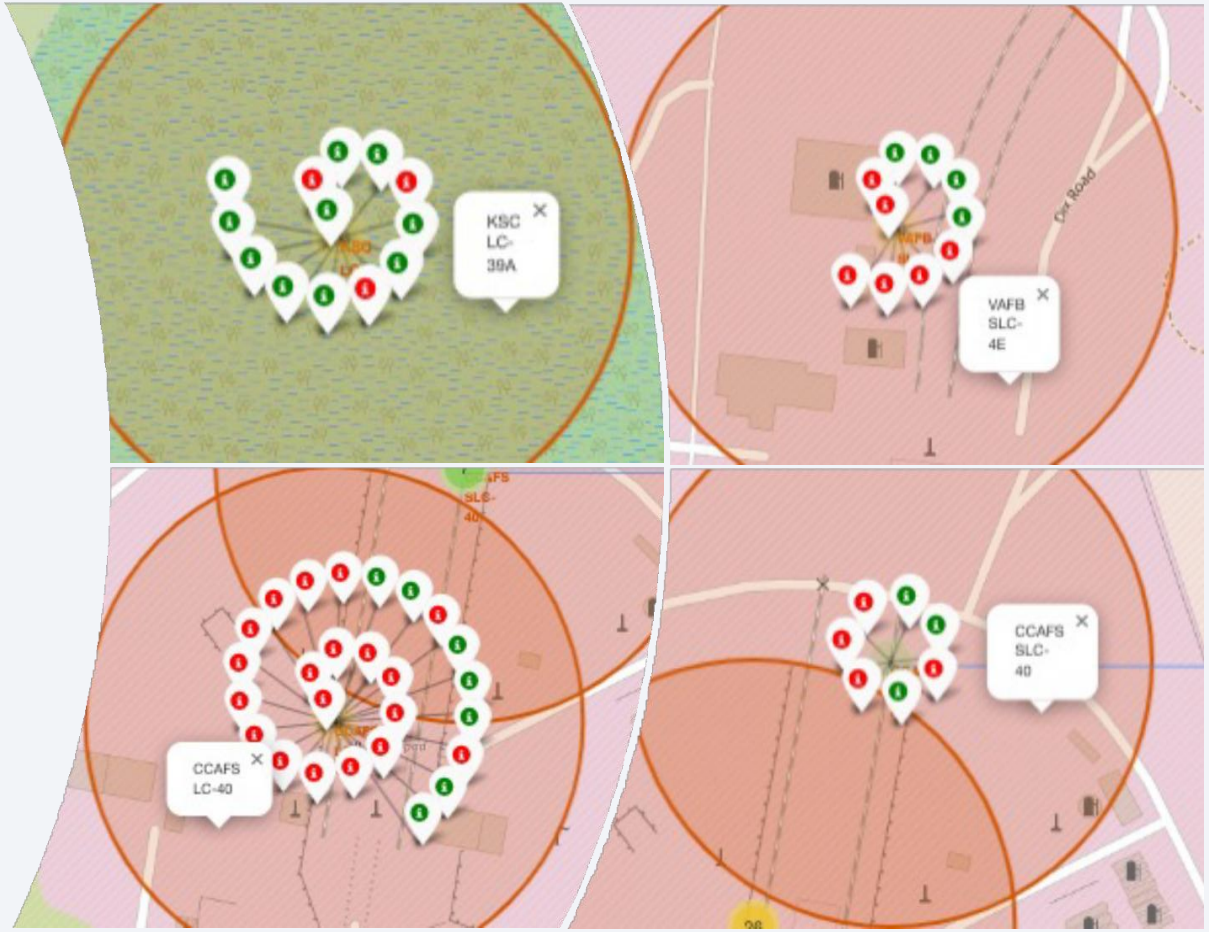
All launch sites global map markers



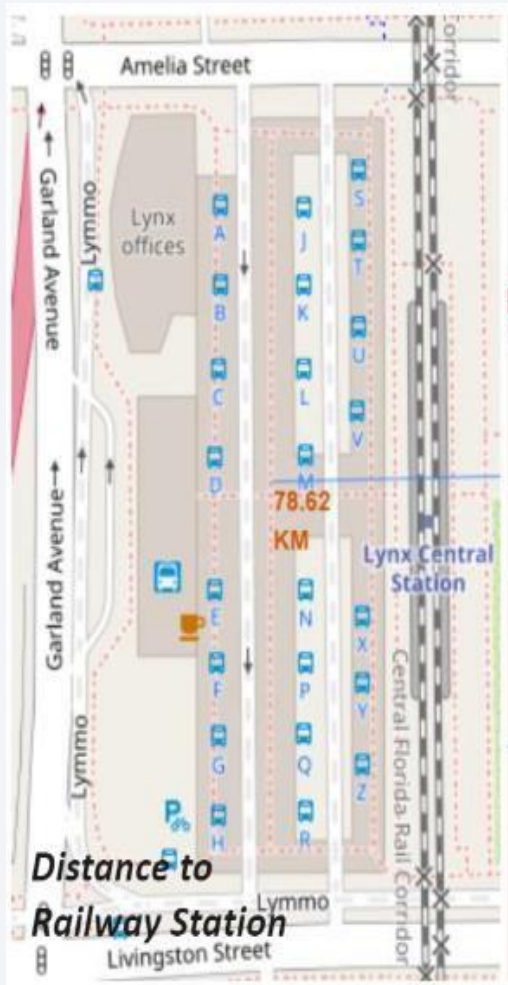
This global map displays all launch sites using markers and circles with labels and popups. Marker clusters help reduce clutter in dense regions, making it easy to identify the geographic distribution of launch sites.

Markers showing launch sites with color labels

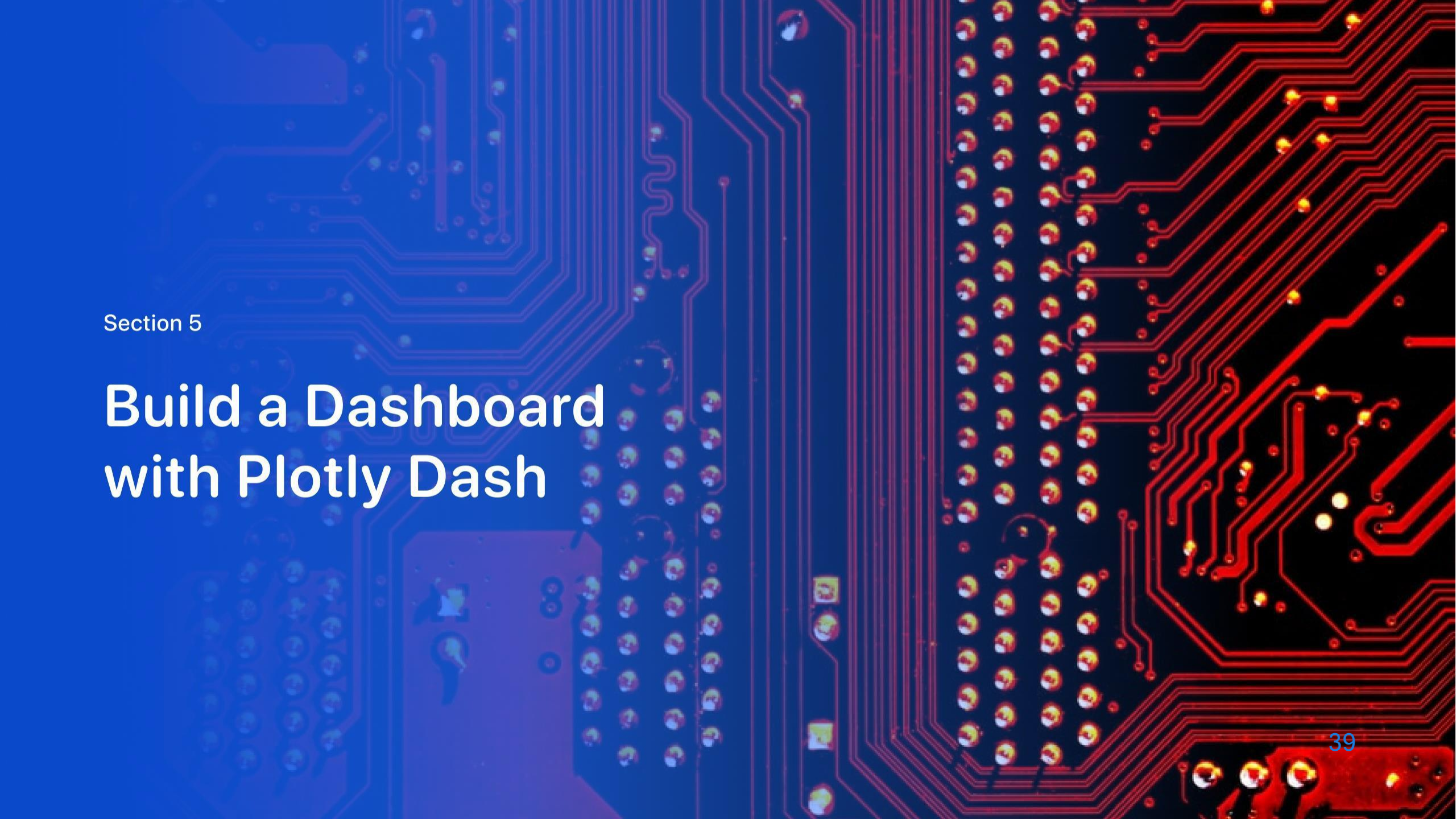
- This view uses colored markers to represent launch outcomes: green for successful landings and red for failures. This visual differentiation allows quick assessment of performance across different sites. Notably, the following sites show higher occurrences:
- **CCAFS SLC-40**: Exhibits the highest number of launches, with a mix of outcomes.
- **VAFB SLC-4E**: Displays significant activity with a strong success rate
- **KSC LC-39A**: Also shows high launch frequency and excellent success performance.



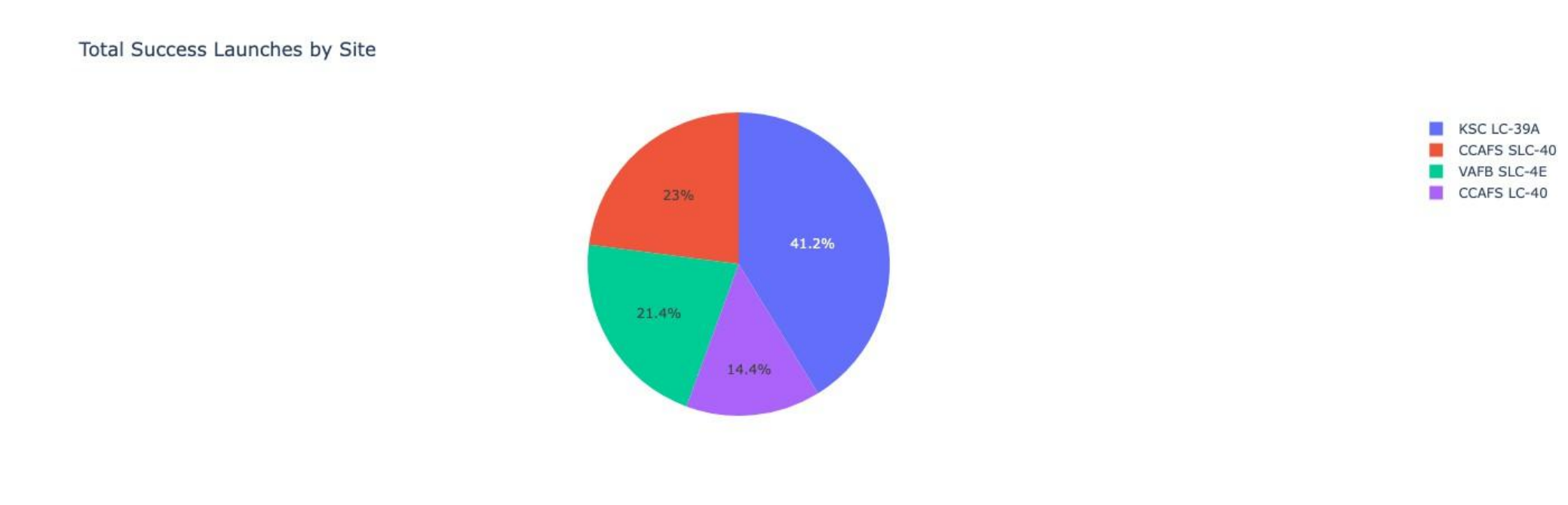
Launch Site distance to landmarks



1. Are launch sites in close proximity to railways?→NO
2. Are launch sites in close proximity to highways?→NO
3. Are launch sites in close proximity to coastline?→YES
4. Do launch sites keep certain distance away from cities?→YES

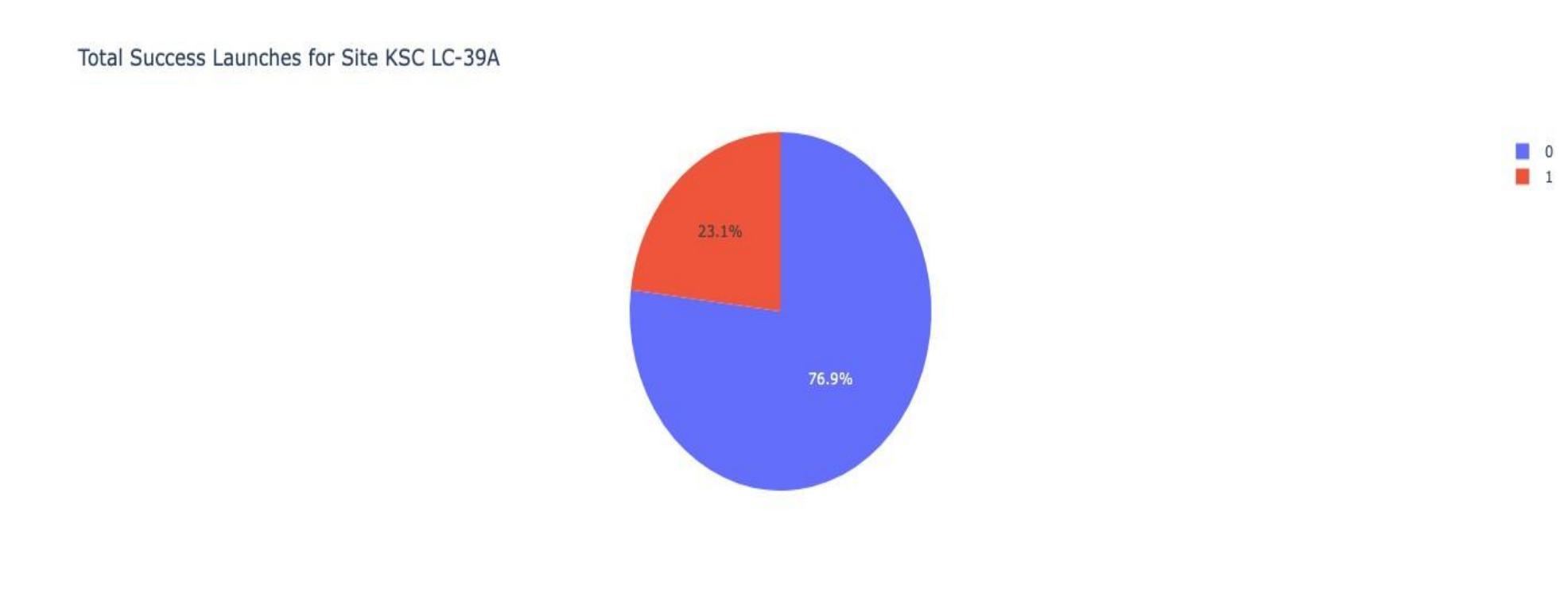

Section 5
Build a Dashboard with Plotly Dash

Dashboard – Overall Launch Success



Displays the total number of successful launches across all sites, highlighting which sites contribute most to the overall success count.

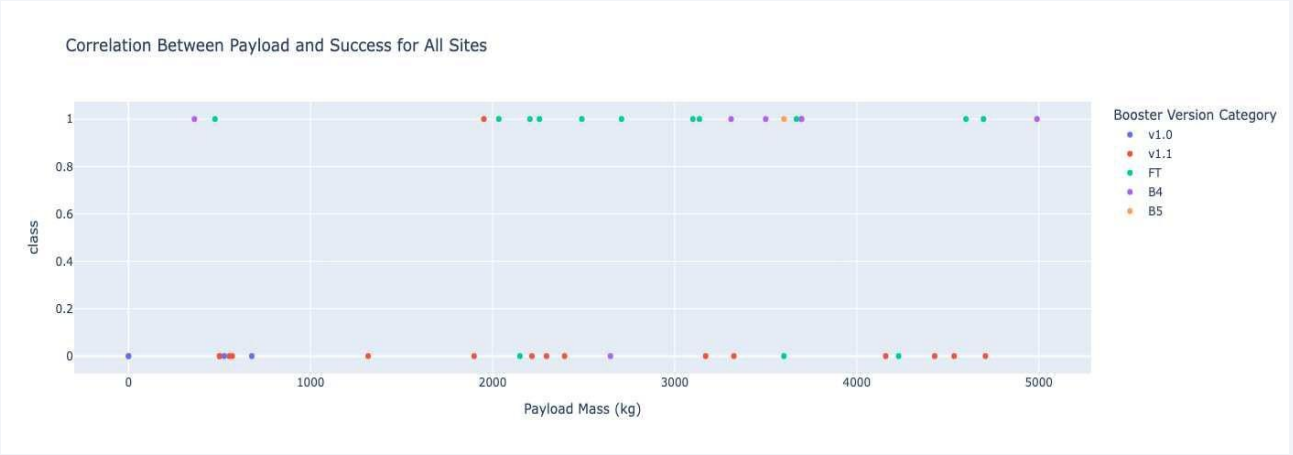
Dashboard – Top Site Success Ratio

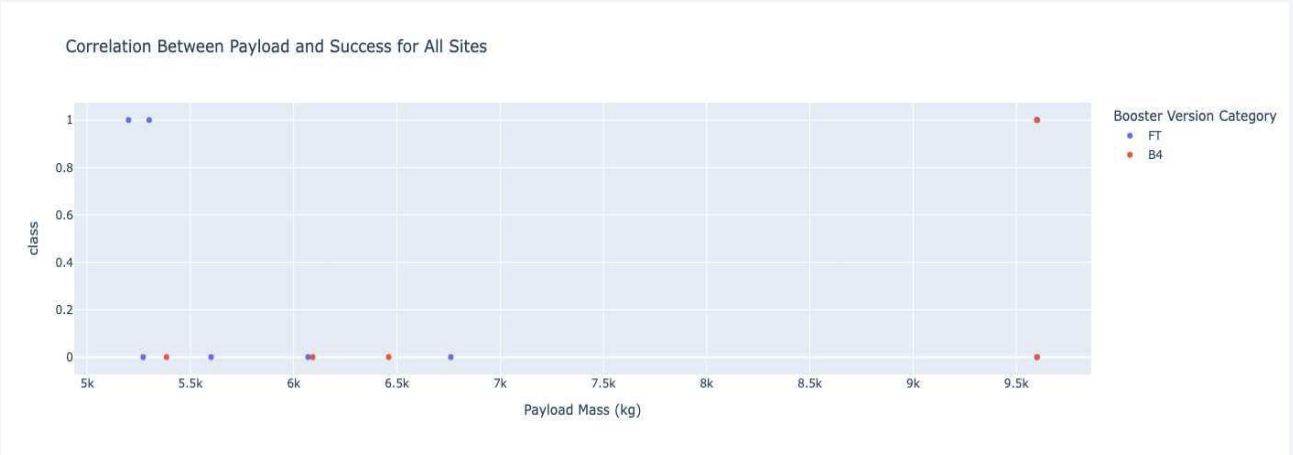


Focuses on KSC LC-39A, the site with the highest success rate, showing the breakdown of successes (76.9%) vs. failures (23.1%).

Dashboard – Payload vs. Launch Outcome

For payloads from 0 to 5000 kg, the success rate is high and consistent. In contrast, the 5000 to 10000 kg range shows greater variability, indicating heavier payloads pose more challenges for successful landings.



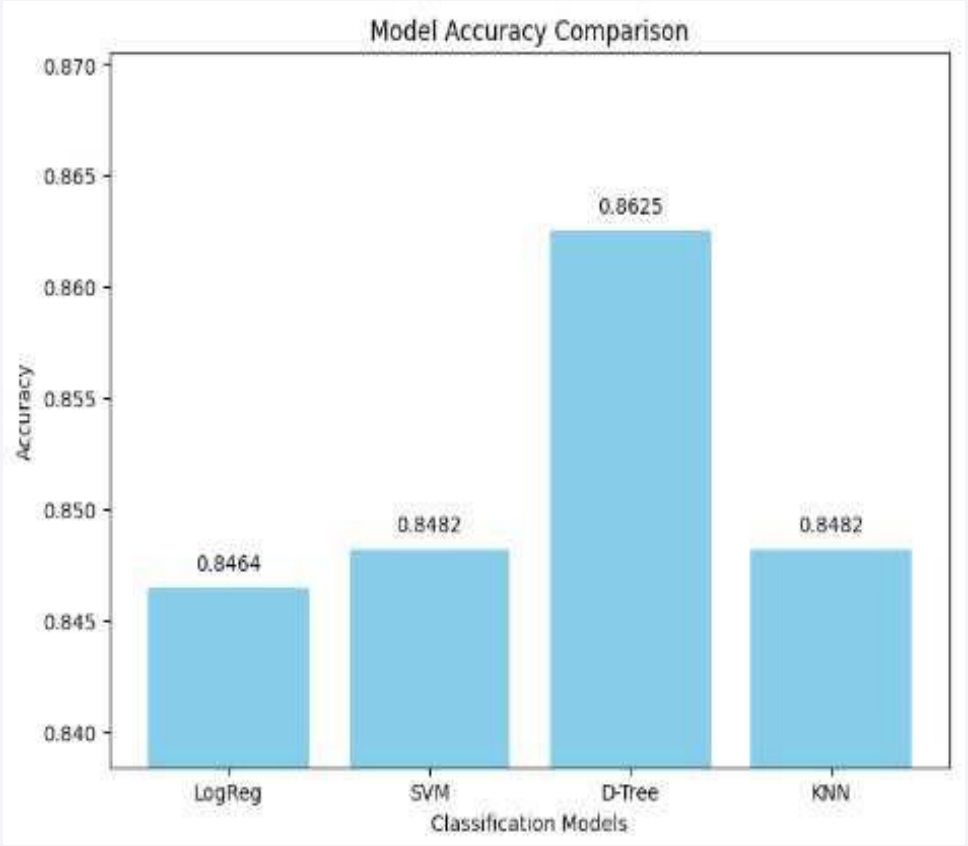


Section 6

Predictive Analysis (Classification)

Classification Accuracy

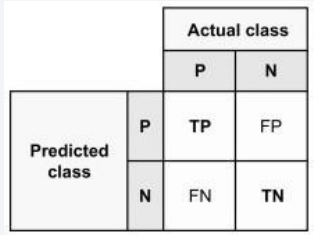
```
model_scores = {
    "Logistic Regression": logreg_cv.best_score_,
    "Support Vector Machine": svm_cv.best_score_,
    "Decision Tree": tree_cv.best_score_,
    "K-Nearest Neighbors": knn_cv.best_score_
}
best_model = max(model_scores, key=model_scores.get)
print("Model performances:", model_scores)
print(f"The best performing model is: {best_model} with accuracy {model_scores[best_model]:.4f}")

Ans.:
Model performances: {
  'Logistic Regression': 0.8464285714285713,
  'Support Vector Machine': 0.8482142857142856,
  'Decision Tree': 0.8625,
  'K-Nearest Neighbors': 0.8482142857142858}
The best performing model is: Decision Tree with accuracy 0.8625
```

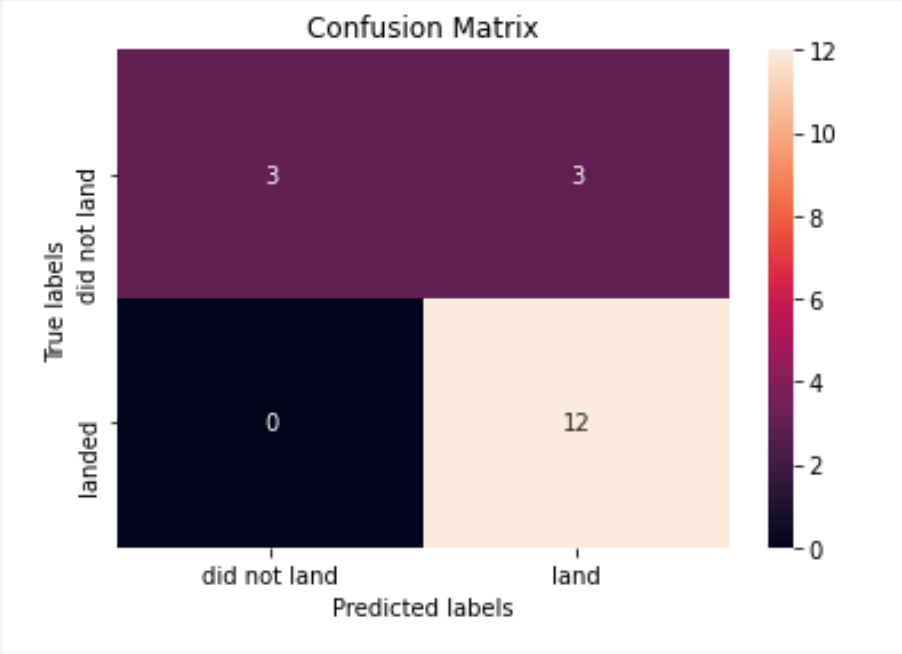


This bar chart compares the accuracy of the four classification models. The Decision Tree model achieved the highest accuracy at 86.25%, suggesting it is the best performer for predicting Falcon 9 first stage landing outcomes.

Confusion Matrix

The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.





Conclusions

Data Analytics Conclusions:

•**More launches, more success:** Launch sites with higher activity tend to achieve better success rates, suggesting that operational experience plays a key role.

•**Consistent improvement over time:** Between 2013 and 2020, launch success rates steadily increased, reflecting technological advancements and process enhancements,

•**Orbit selection matters:** Orbits such as ES-L1, GEO, HEO, and SSO consistently achieved 100% success, highlighting the importance of strategic planning.

•**KSC LC-39A leads the field**: This launch site stands out with the highest number of successful missions, thanks in part to its strategic coastal, equatorial location.

•**Accurate predictions with ML algorithms:** The Decision Tree model outperformed other algorithms, delivering the most accurate and reliable predictions.

•**Payload mass as a key factor:** Launches carrying between 2000 and 5000 kg showed higher success rates, whereas heavier payloads exhibited greater variability.

Thank you!