IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jorge Solis
10/12/2023

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

Introduction

- Project background and context

SpaceX promotes Falcon 9 rocket launches on its website for a price of US$62M, significantly cheaper than other providers, which charge over US$165M per launch. A substantial portion of this cost savings stems from SpaceX's ability to reuse the initial rocket stage. As a result, assessing the successful landing of this first stage becomes crucial in determining the overall launch cost. This information can prove invaluable when competing with SpaceX for a rocket launch contract. The objective of this project is to establish a machine learning pipeline capable of predicting the first stage's successful landing.

- Problems you want to find answers

- What elements influence the success of a rocket landing?

- The interplay of different factors that establish the success rate of a landing.

- What conditions must be met to guarantee the success of a landing program?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was obtained from SpaceXAPI(https://api.spacexdata.com/v4/rockets/) and WebScrapping (https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches)
- Perform data wrangling
  - Additional data was enhanced by generating a landing result category through the examination and synthesis of 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

Methodology

Executive Summary

- Perform predictive analysis using classification models

The data gathered up to this point was standardized, split into training and testing datasets, and subjected to assessment by four distinct classification models. The performance of each model was assessed using varying parameter combinations to determine accuracy.

Data Collection

- Describe how data sets were collected.

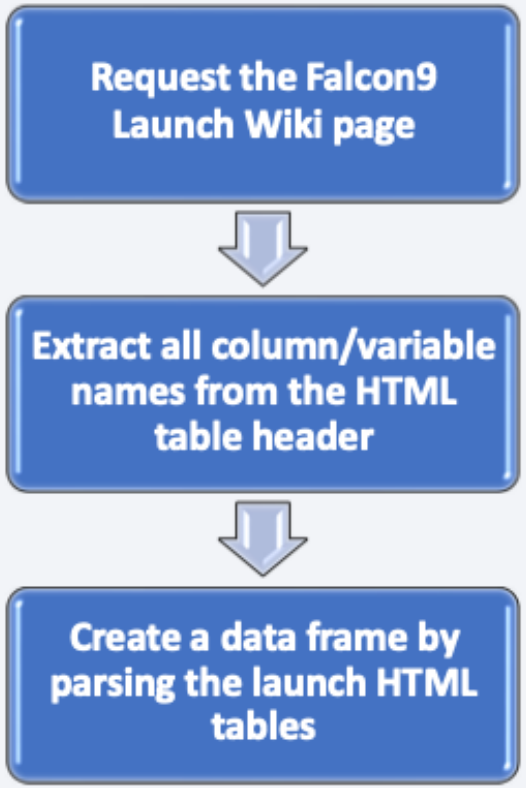
Data sets were obtained through web scraping techniques from both the SpaceX API (https://api.spacexdata.com/v4/rockets/) and Wikipedia (https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches).

Data Collection – SpaceX API

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

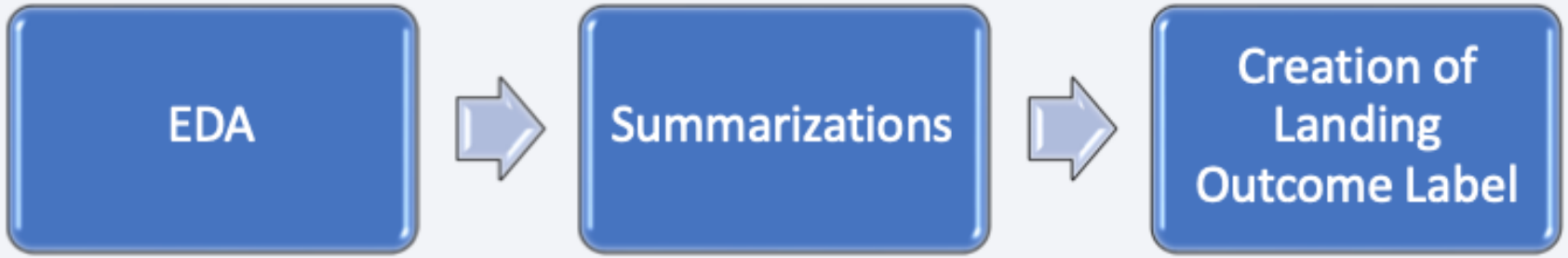
Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts.



Data Wrangling

At first, an exploratory data analysis (EDA) was conducted on the dataset. Subsequently, the dataset was used to generate summaries regarding the number of launches per site, the frequency of each orbit type, and the occurrence of mission outcomes for each orbit category. Lastly, the landing outcome label was derived from the Outcome column.

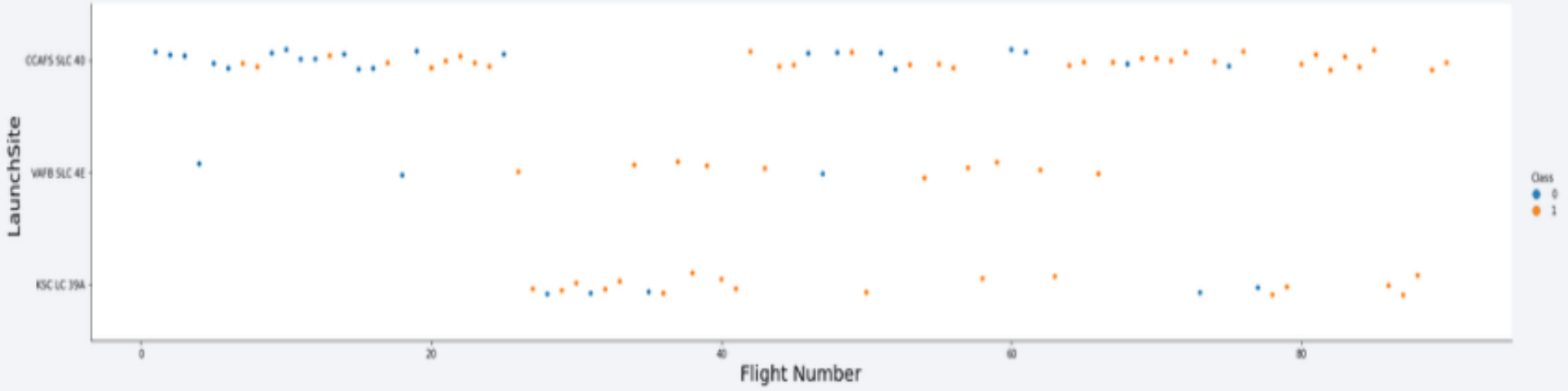


EDA with Data Visualization

1. Retrieving the unique launch site names in the space mission.
2. Identifying the top 5 launch sites whose names start with the 'CCA' string.
3. Calculating the total payload mass carried by boosters launched by NASA (CRS).
4. Determining the average payload mass carried by booster version F9 v1.1.
5. Finding the date when the first successful landing occurred on a ground pad.
6. Listing the names of boosters that achieved success on a drone ship and had a payload mass ranging from 4000 to 6000 kg.
7. Calculating the total count of successful and failed mission outcomes.
8. Identifying the booster versions that carried the maximum payload mass.
9. Retrieving information about failed landing outcomes on a droneship in the year 2015, including their booster versions and launch site names.
10. Ranking the count of landing outcomes, such as failures on droneships or successes on ground pads, between June 4, 2010, and March 20, 2017.

EDA with SQL

- In the process of data exploration, scatterplots and bar plots were employed to visually represent the connections between various pairs of features, including Payload Mass and Flight Number, Launch Site and Flight Number, Launch Site and Payload Mass, Orbit and Flight Number, and Payload and Orbit.



Build an Interactive Map with Folium

In Folium Maps, a variety of visual elements were employed:

- Markers, which signify specific points such as launch sites.
- Circles, used to delineate highlighted areas around particular coordinates, as seen with NASA Johnson Space Center.
- Marker clusters, which group multiple events or launches at each specific coordinate.
- Lines, utilized to represent distances between two sets of coordinates.

Build a Dashboard with Plotly Dash

- Created an interactive dashboard using Plotly Dash.
- Generated pie charts illustrating the cumulative launches from specific sites.
- Produced scatter plots depicting the connection between Outcome and Payload Mass (in kilograms) for various booster versions.

Predictive Analysis (Classification)

We imported and processed the data using numpy and pandas, conducted data transformation, and divided it into training and testing subsets.

We constructed various machine learning models and optimized different hyperparameters through GridSearchCV.

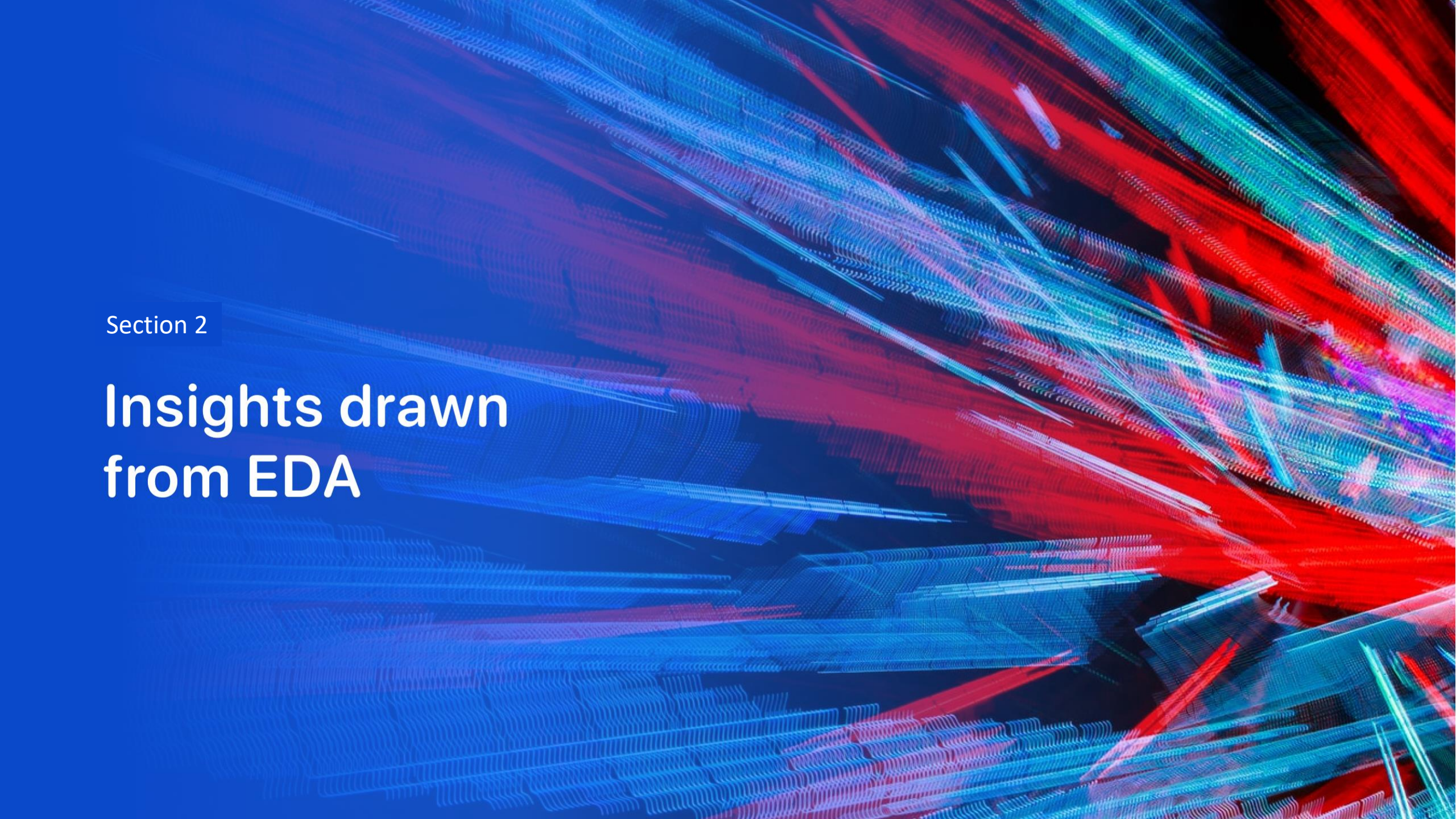
Accuracy served as our primary metric for model evaluation, and we enhanced the model through feature engineering and algorithm fine-tuning.

Ultimately, we identified the most effective classification model.

Results

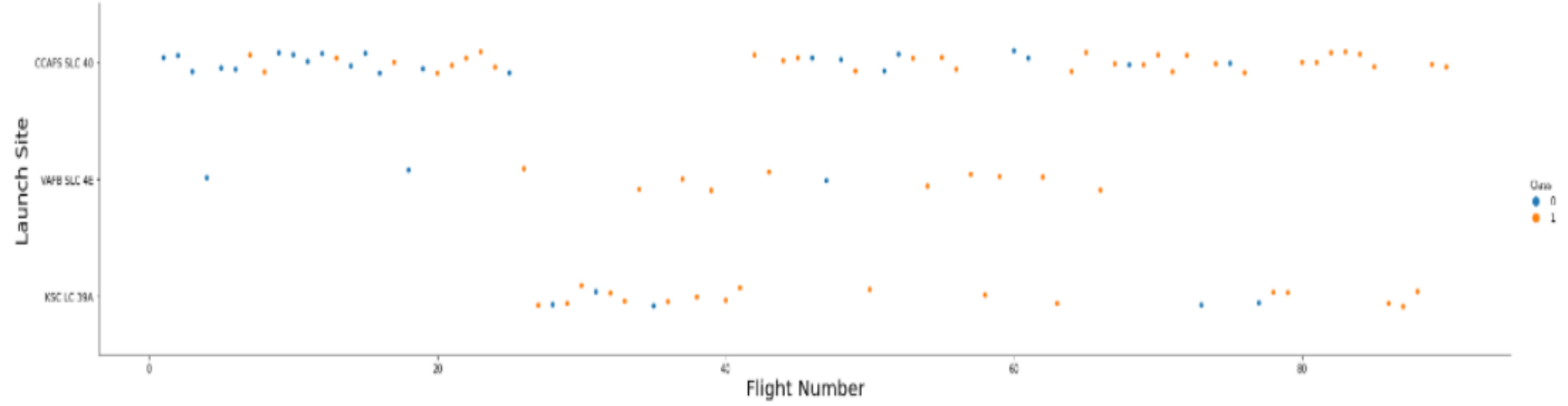
Key findings from the exploratory data analysis are as follows:

1. SpaceX operates from four different launch sites.
2. Initial launches were conducted for SpaceX itself and NASA.
3. The average payload mass for the F9 v1.1 booster is approximately 2,928 kg.
4. The first successful landing occurred in 2015, five years after the inaugural launch.
5. Several Falcon 9 booster versions achieved successful landings on drone ships, particularly when carrying payloads above the average.
6. Nearly 100% of mission outcomes were deemed successful.
7. In 2015, two booster versions, namely F9 v1.1 B1012 and F9 v1.1 B1015, experienced failed landing attempts on drone ships.
8. The success rate of landing outcomes improved over the years.


Section 2
Insights drawn from EDA

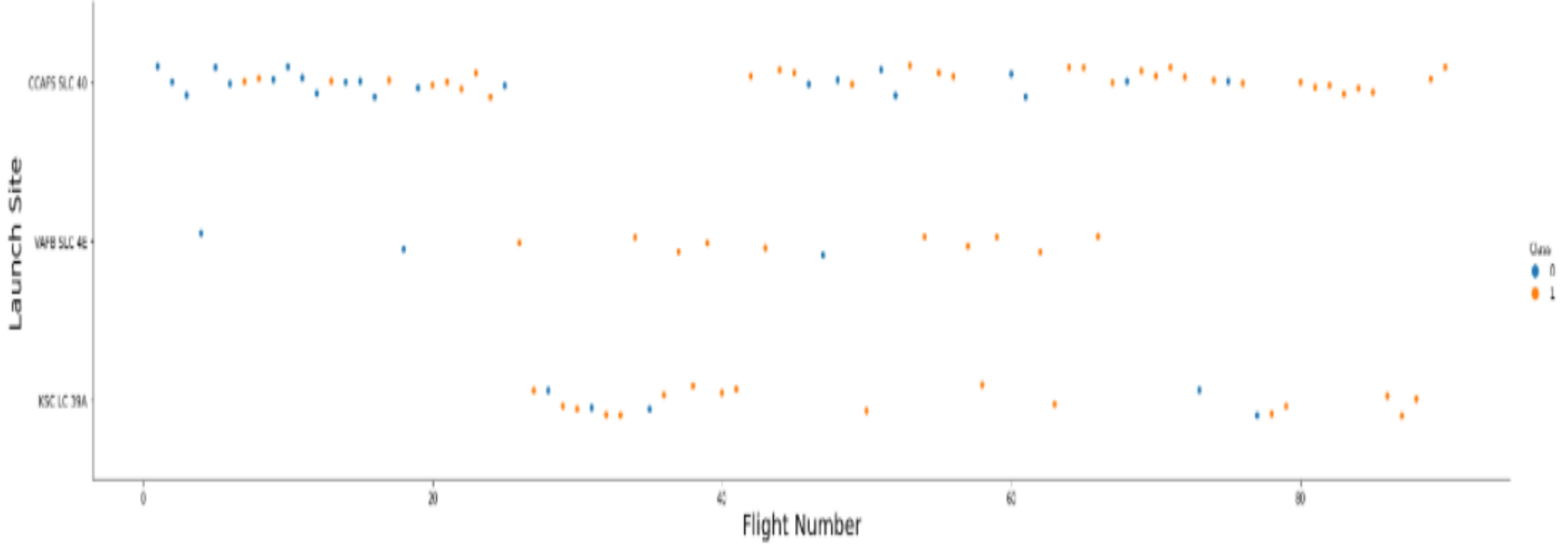
Flight Number vs. Launch Site

Based on the plot, we observed that there is a positive correlation between the number of flights at a launch site and the success rate at that site.



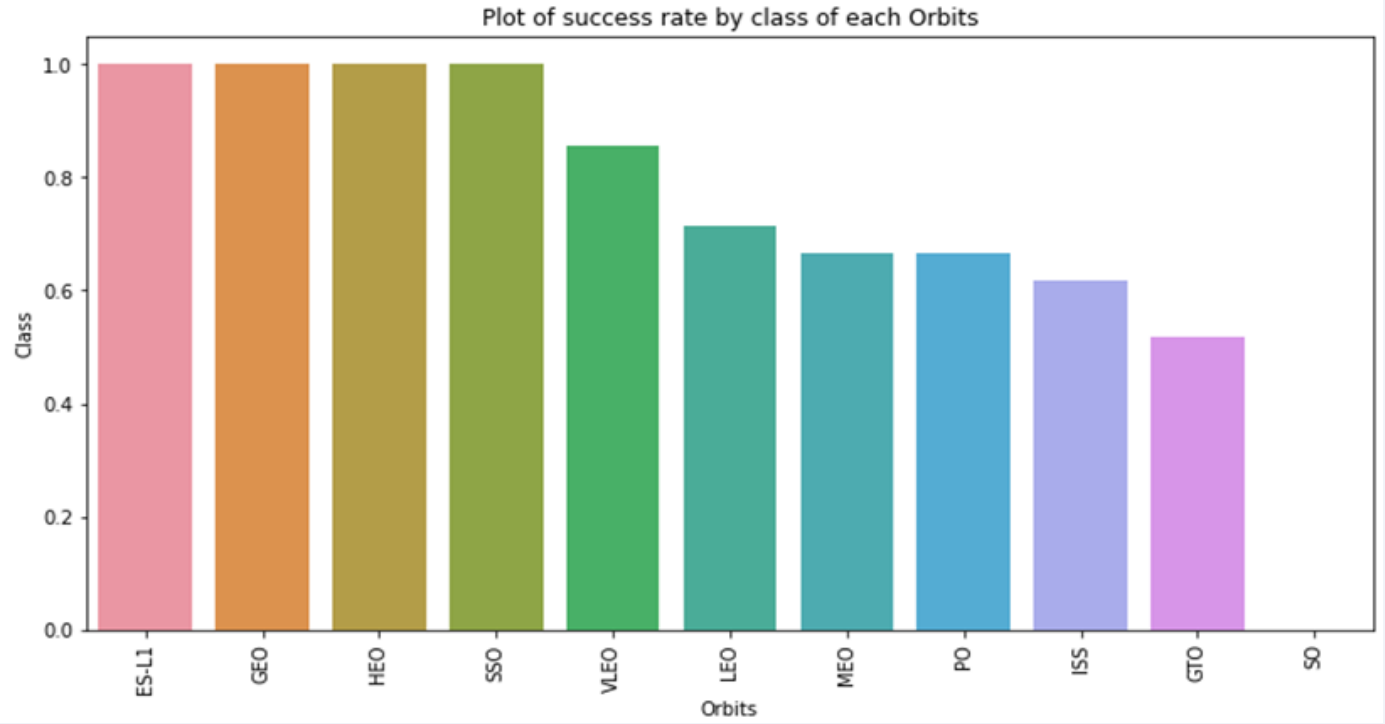
Payload vs. Launch Site

As the payload mass increases at launch site CCAFS SLC 40, the rocket's success rate also rises.



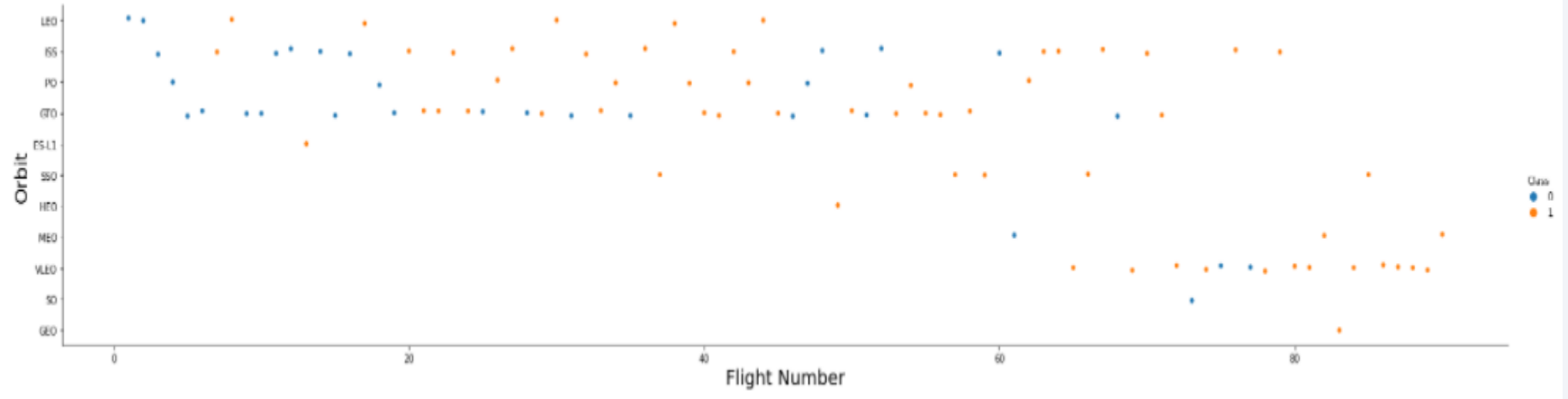
Success Rate vs. Orbit Type

- The plot reveals that ES-L1, GEO, HEO, SSO, and VLEO exhibited the highest success rates.



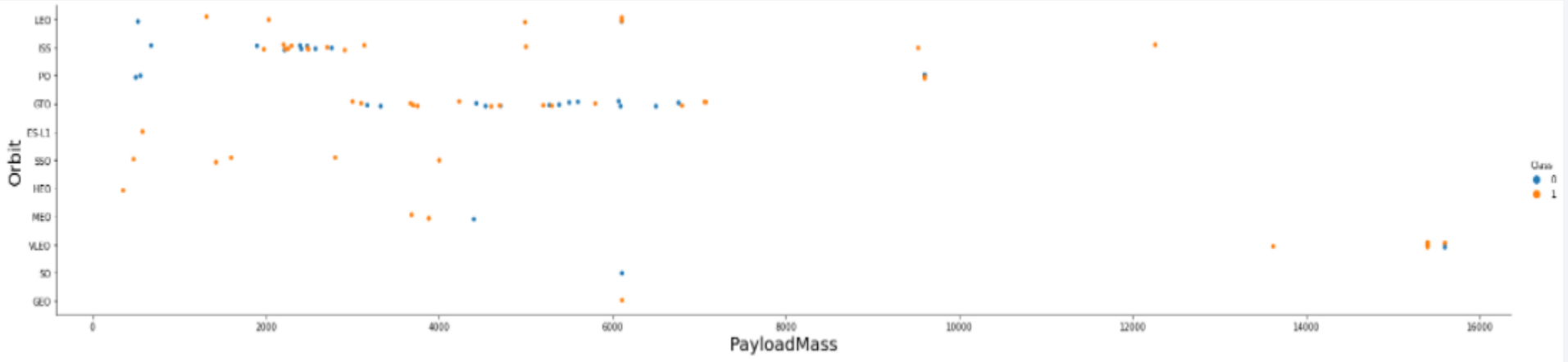
Flight Number vs. Orbit Type

- The chart provided displays the relationship between Flight Number and Orbit type. We can discern that, in the case of the LEO orbit, success is associated with the flight count, while for the GTO orbit, there is no apparent connection between the flight number and the orbit's success.



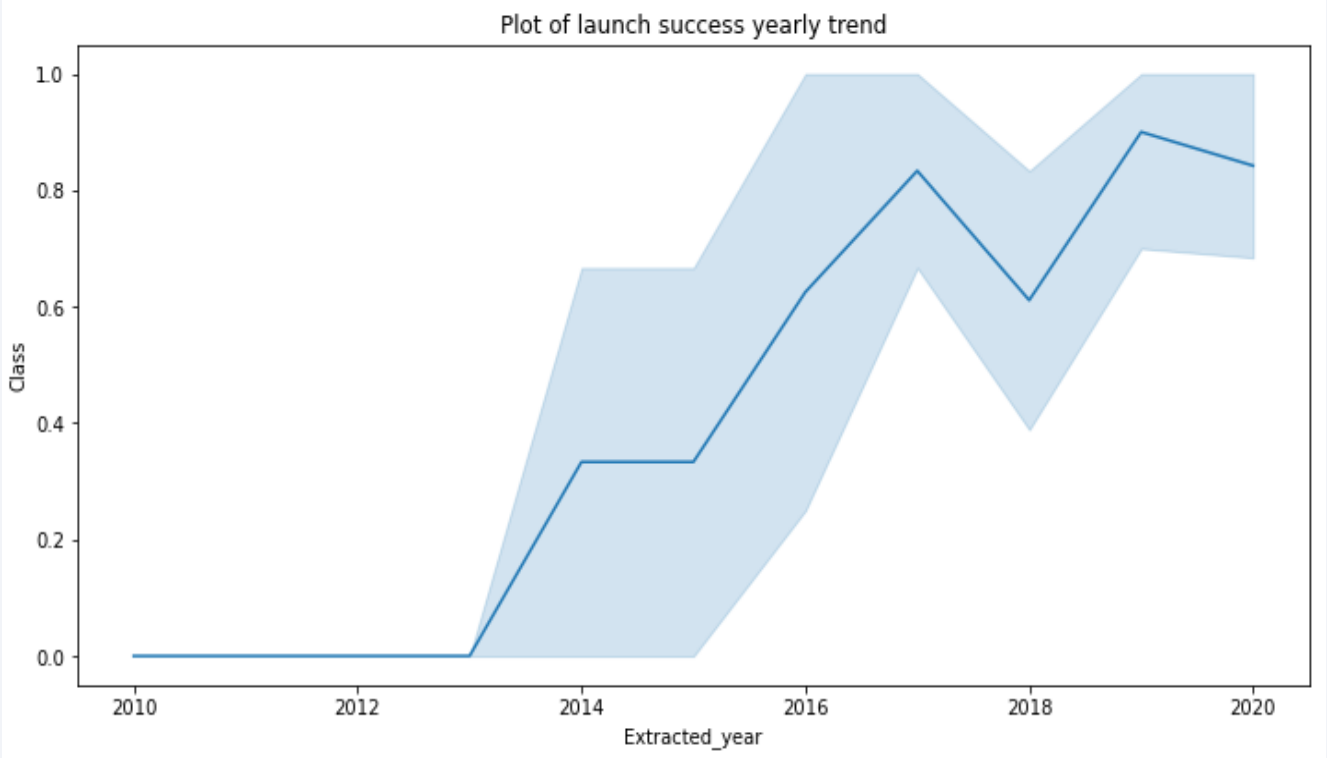
Payload vs. Orbit Type

- It's noticeable that for PO, LEO, and ISS orbits, there is an increased frequency of successful landings when heavy payloads are involved.



Launch Success Yearly Trend

The plot clearly indicates that the success rate has steadily risen from 2013 to 2020.



All Launch Site Names

- We employed the "DISTINCT" keyword to display exclusively the unique launch sites found in 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'

- The query mentioned above was utilized to retrieve five records where the launch sites start with the prefix '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

Total Payload Mass

- We determined that the cumulative payload carried by NASA's boosters amounted to 45,596 using the following query:

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

- The average payload mass carried by the booster version F9 v1.1 was computed to be 2,928.4.

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

- It was noted that the date of the initial successful landing outcome on a ground pad occurred on the 22nd of 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

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.

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

Total Number of Successful and Failure Mission Outcomes

We employed wildcard symbols, such as '%,' in the filtering process to select records where the MissionOutcome was either 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

By utilizing a subquery within the WHERE clause and the MAX() function, we identified the booster that had transported the 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

2015 Launch Records

- We employed a combination of the WHERE clause, the LIKE operator, AND condition, and BETWEEN condition to filter for failed landing outcomes on a drone ship, along with their corresponding booster versions and launch site names, specifically for the 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 extracted both Landing outcomes and the COUNT of landing outcomes from the dataset and then applied the WHERE clause to filter for landing outcomes between June 4, 2010, and March 20, 2017

Subsequently, we used the GROUP BY clause to group the landing outcomes and the ORDER BY clause to arrange the grouped landing outcomes 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

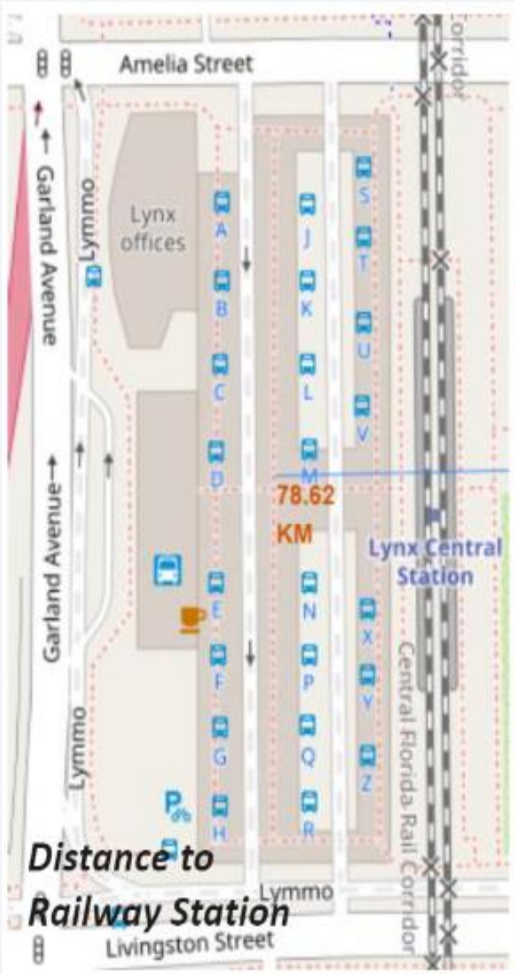
Section 3

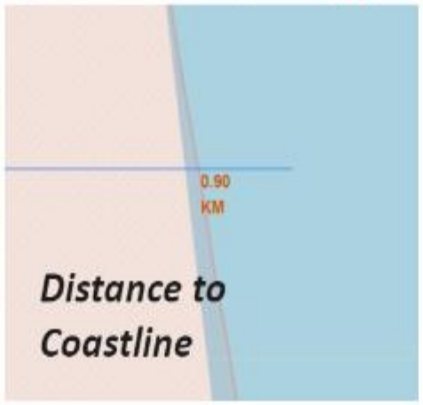
Launch Sites Proximities Analysis

Global launch sites map makers

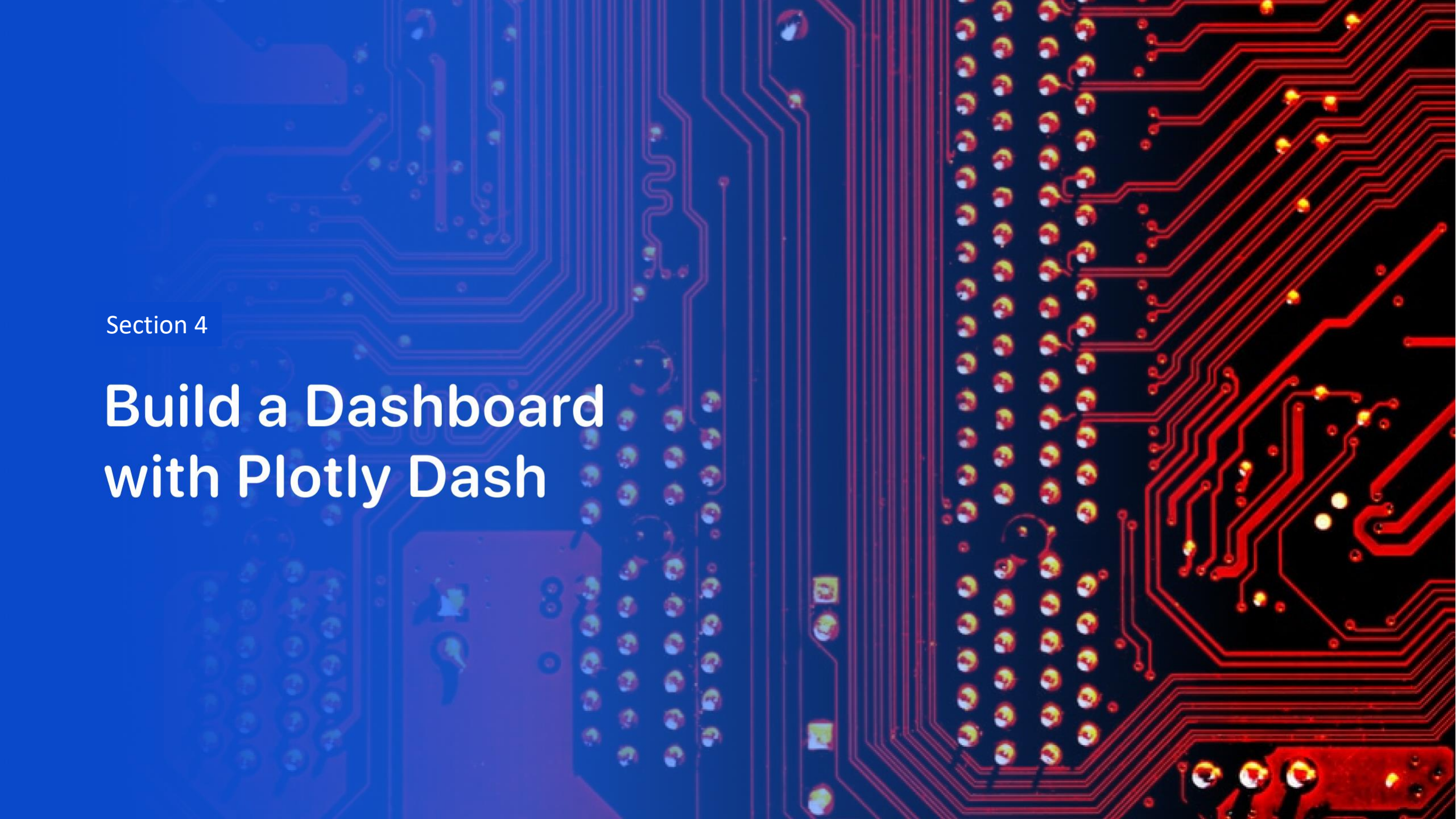
Launch sites markers and labels

Distance to launch sites landmarks

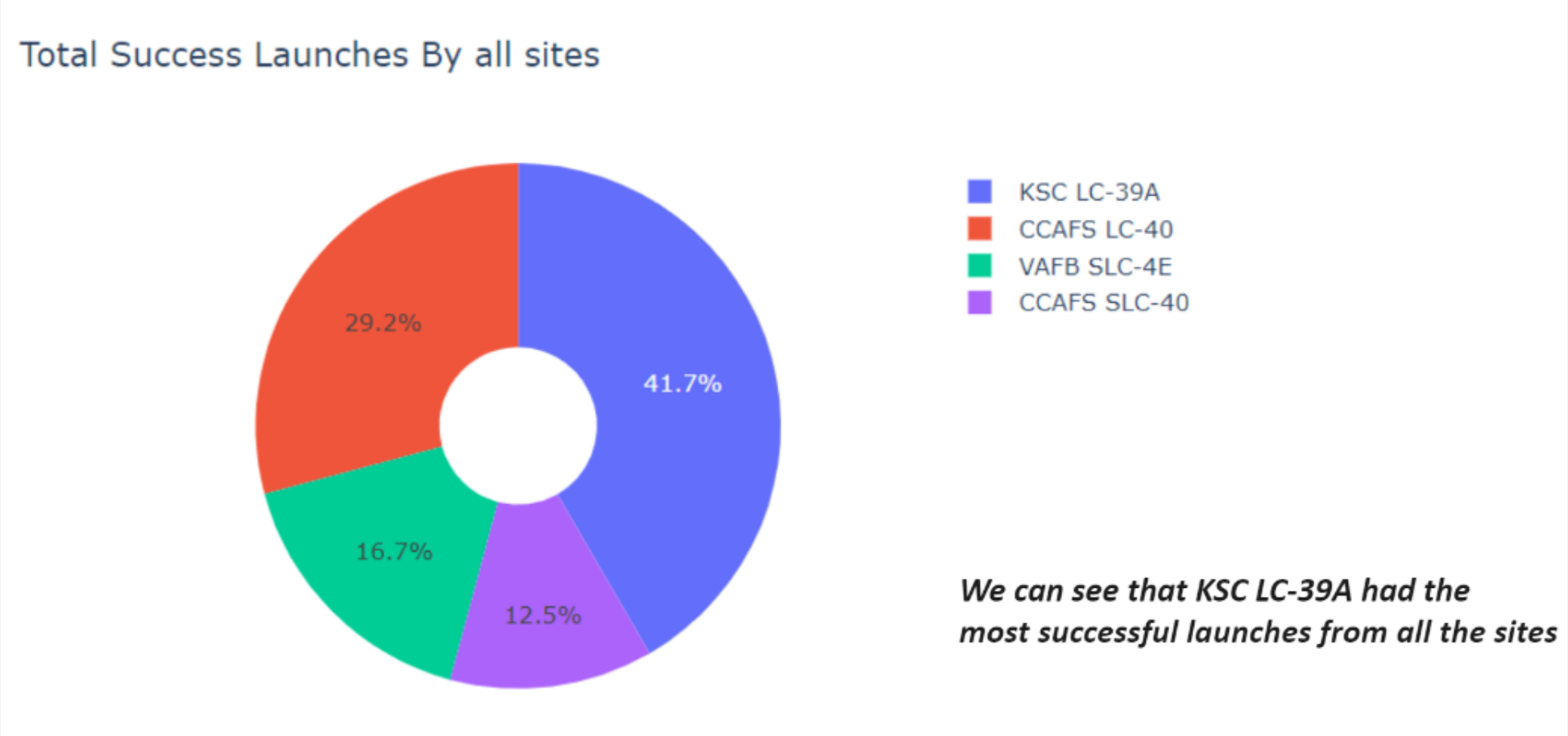




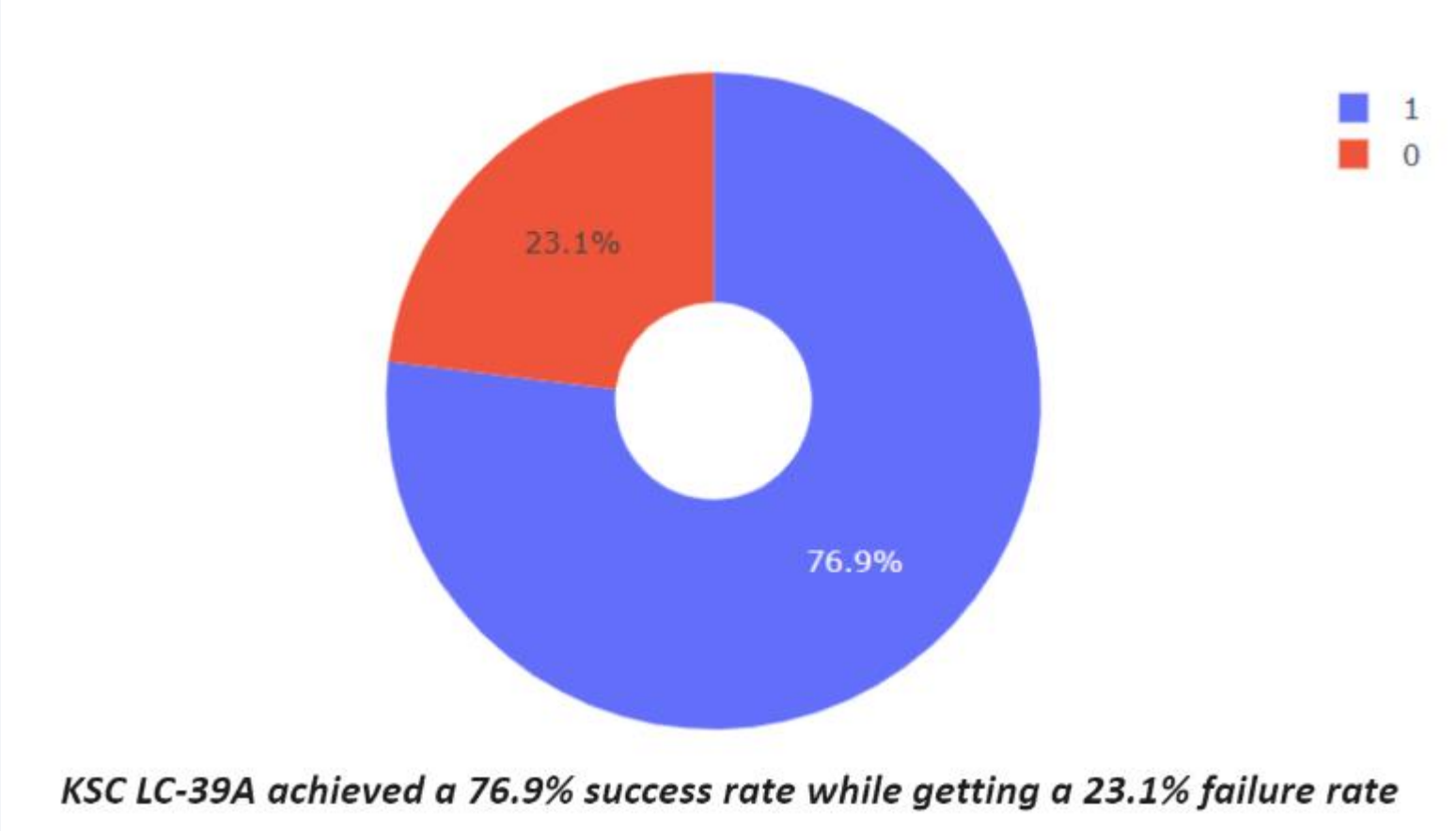
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

Total success lunches by all sites

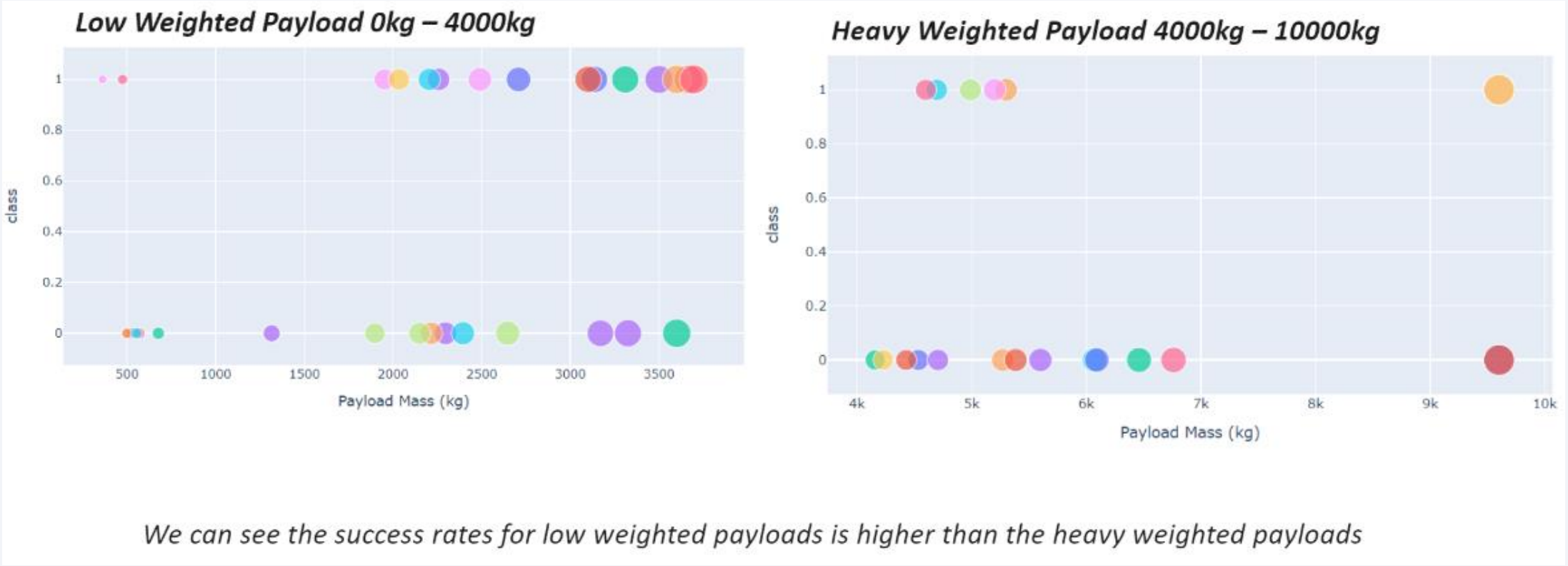


Launch site with the highest launch success ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Success rates for low and heavy weighted payloads



We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Among the models, the decision tree classifier exhibits the highest classification accuracy.

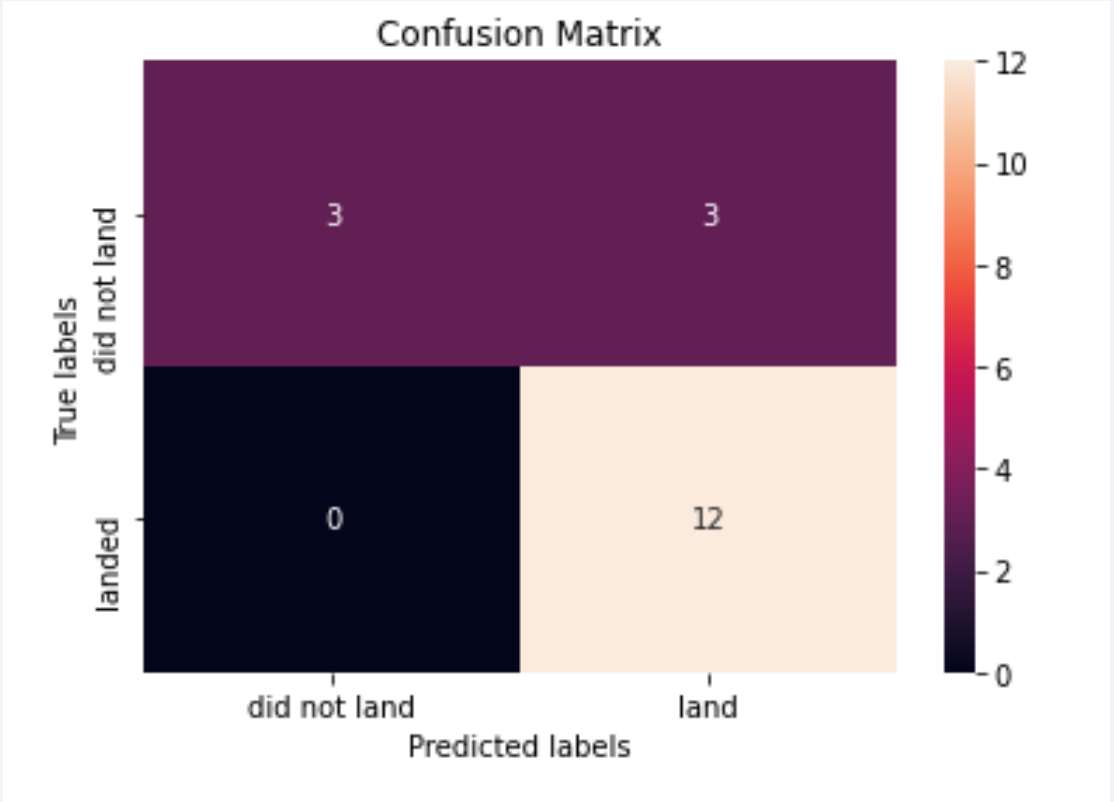
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

The confusion matrix generated for the decision tree classifier indicates the classifier's ability to differentiate between various classes. However, the primary issue lies in the false positives, which means that the classifier incorrectly identifies unsuccessful landings as successful landings.



Conclusions

In summary:

1. There is a positive correlation between the number of flights at a launch site and the success rate at that site.
2. The success rate began to rise in 2013 and continued to increase until 2020.
3. Orbits ES-L1, GEO, HEO, SSO, and VLEO exhibited the highest success rates.
4. KSC LC-39A had the highest number of successful launches among all the launch sites.
5. The Decision tree classifier stands out as the most suitable machine learning algorithm for this particular task.

Thank you!