Winning Space Race with Data Science

Shubham Khairmode
August, 16, 2023

Outline

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

Executive Summary

- Summary of methodologies

Data was collected through APIs and web scraping. The collected data then underwent data wrangling to clean and transform it into a usable format. SQL facilitated exploratory data analysis to understand trends and patterns. Data visualizations provided further insights from the exploratory analysis. Tools like Folium enabled interactive visual analytics. Finally, machine learning models made predictions based on the collected, wrangled, analyzed, and visualized data. Overall, the data actively moved through different stages of collection, wrangling, analysis, visualization, and prediction.

- Summary of all results

The data underwent exploratory analysis which revealed key insights and trends. Screenshots showcased interactive analytics that enabled fluid data visualization. Predictive analytics models then generated results forecasting future outcomes. Overall, the data actively moved through exploration, visualization, and prediction to produce meaningful end results.

Introduction

- Project Background and Context
  - SpaceX Falcon 9 Landing Prediction

    - Goal: Build machine learning pipeline to predict first stage landing success

    - Significance:
      - Enables determining SpaceX launch costs
      - Allows other providers to competitively bid against SpaceX
      - Currently no automated prediction system exists
- Problems you want to find answers
  - What factors determine if the rocket will land successfully?
  - How do different features interact to influence the success rate of landing?
  - What operating conditions need to be in place to ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

- Data Collection Methodology
  - Data from SpaceX obtained from two sources:
  - SpaceX API (https://api.spacexdata.com/v4/rockets/)
  - Web scraping (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches)
- Data Wrangling
  - Collected data enriched by:
    - Creating landing outcome label based on outcome data
    - Summarizing and analyzing 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
  - The collected data was normalized and split into training and test sets. Four different classification models were evaluated by training them on the data using various parameter combinations. The accuracy of each model was then evaluated.

Data Collection

- Describe how data sets were collected.
  - The datasets were collected from two sources:
    - The SpaceX API (https://api.spacexdata.com/v4/rockets/) which provides launch data directly from SpaceX.
    - Wikipedia (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches) which was web scraped to extract Falcon 9 and Falcon Heavy launch data.

Data Collection – SpaceX API

API Request	Data Selection	Data Wrangling
An API request to the SpaceX launch data	Filtering the returned API response to select only Falcon 9 launch data	Preprocessing the dataset, i.e., handling missing values

- SpaceX API Data Collection
  - SpaceX offers a public API that provides launch data.
  - The API was used to obtain the data following the process outlined in the accompanying flowchart.
  - The retrieved data was then persisted/stored.
  - See source code for implementation details.

Data Wrangling

EDA	Feature Engineering	Creation of class label
Performed initial Exploratory Data Analysis (EDA) on the dataset to understand trends.	Calculated summary statistics such as launches per site, orbit occurrence, and mission outcome per orbit type.	Generated the landing outcome label from the Outcome column to enable modeling.

See source code for implementation details.

EDA with Data Visualization

- Exploratory SQL Queries
  - Identified unique launch sites
  - Analyzed top 5 sites starting with 'CCA'
  - Calculated total payload mass for NASA (CRS) boosters
  - Determined average payload for F9 v1.1 boosters
  - Found date of first successful ground pad landing
  - Retrieved boosters with drone ship success and 4000-6000 kg payload
  - Counted total successful and failed outcomes
  - Identified boosters with max payload mass
- Examined failed drone ship outcomes, booster versions, sites in 2015
  - Ranked landing outcomes from 2010-06-04 to 2017-03-20

- See source code for implementation details.

EDA with Data Visualization

- Catplots, barplots, scatterplot and line plots were used to to visualization the data to study various trends and insights.
- See source code for implementation details.

EDA with SQL

- The following SQL queries were performed:
  - SELECT, Aggregate functions (SUM, AVG, MIN, etc)
  - WHERE clause
  - GROUP BY clause
  - And subqueries
- See source code for implementation details.

Build an Interactive Map with Folium

Interactive Maps with Folium:

- Markers indicate specific points like launch sites.
- Circles highlight areas around coordinates like NASA Johnson Space Center.
- Marker clusters show grouped events at each coordinate, such as launches per site.
- Lines display distances between two coordinates.
- See source code for implementation details.

Build a Dashboard with Plotly Dash

Interactive Dashboards with Plotly Dash:

- Built an interactive dashboard using Plotly Dash.
- Created pie charts showing total launches by site.
- Generated scatter plots exhibiting the relationship between outcome and payload mass for each booster version.
- See source code for implementation details.

Predictive Analysis (Classification)

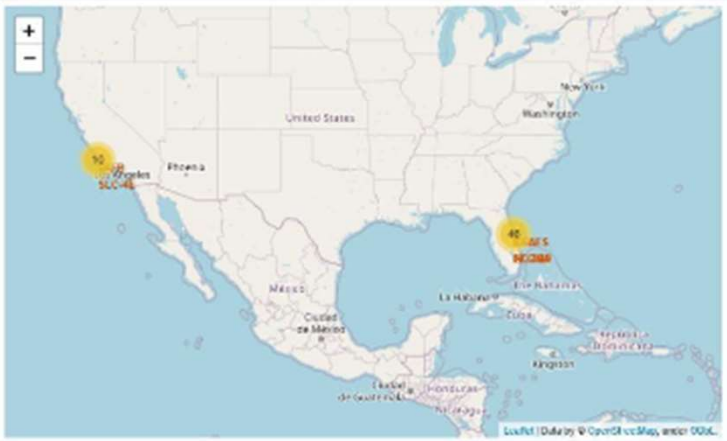
Machine Learning Modeling

- Loaded data using Pandas, transformed features, and split into training and test sets.
- Built machine learning models and tuned hyperparameters using GridSearchCV.
- Used accuracy as evaluation metric and improved models via feature engineering and tuning.
- Identified best performing classification algorithm: Decision Tree classifier.
- See source code for implementation details.

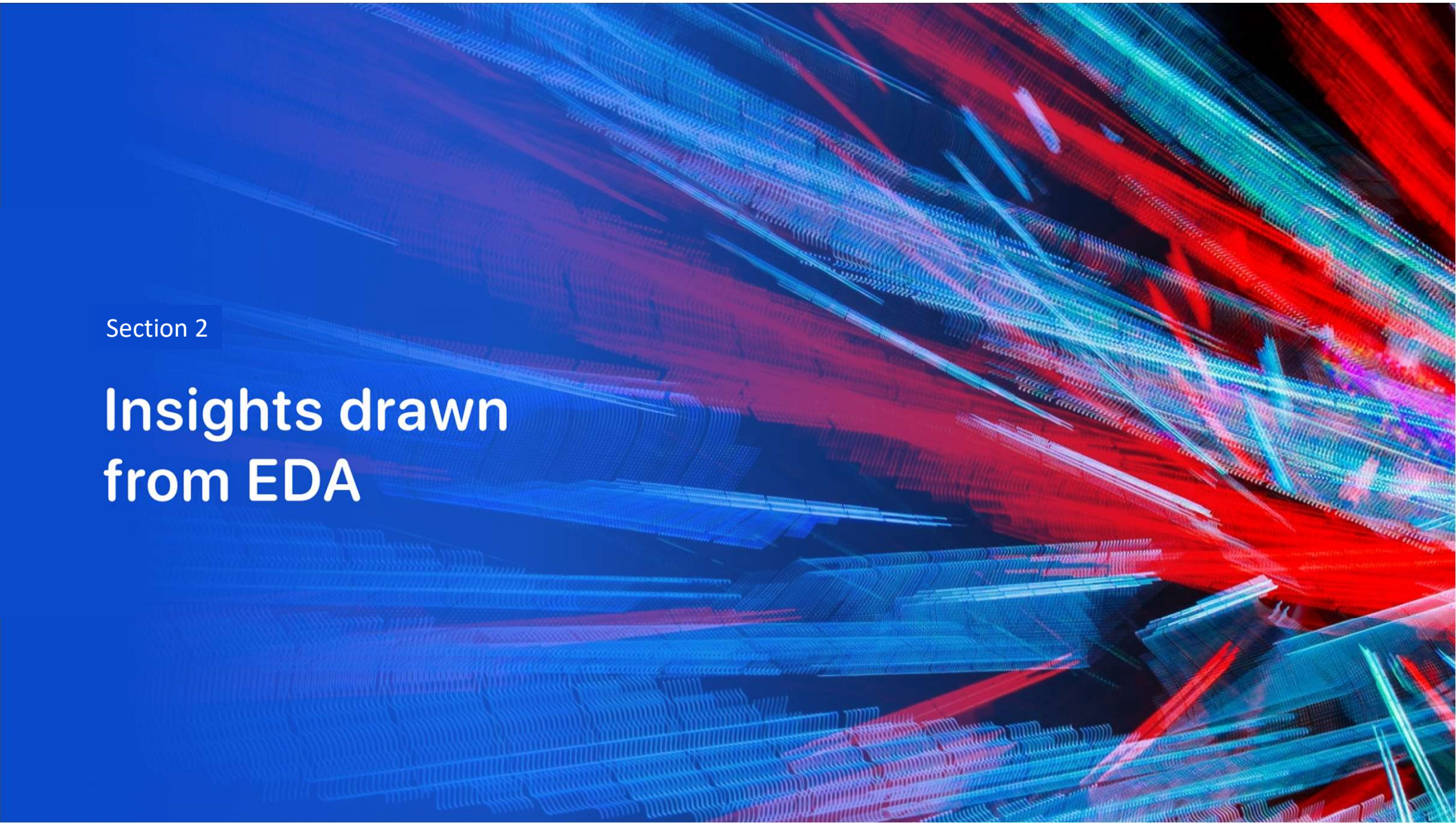
Results

- SpaceX uses 4 different launch sites
- Initial launches were for SpaceX and NASA
- Average payload of F9 v1.1 is 2,928 kg
- First successful landing was in 2015, 5 years after first launch
- Many boosters succeeded landing with above average payload
- ~100% mission success rate
- 2 booster versions failed drone ship landing in 2015
- Landing outcomes improved over time

Results

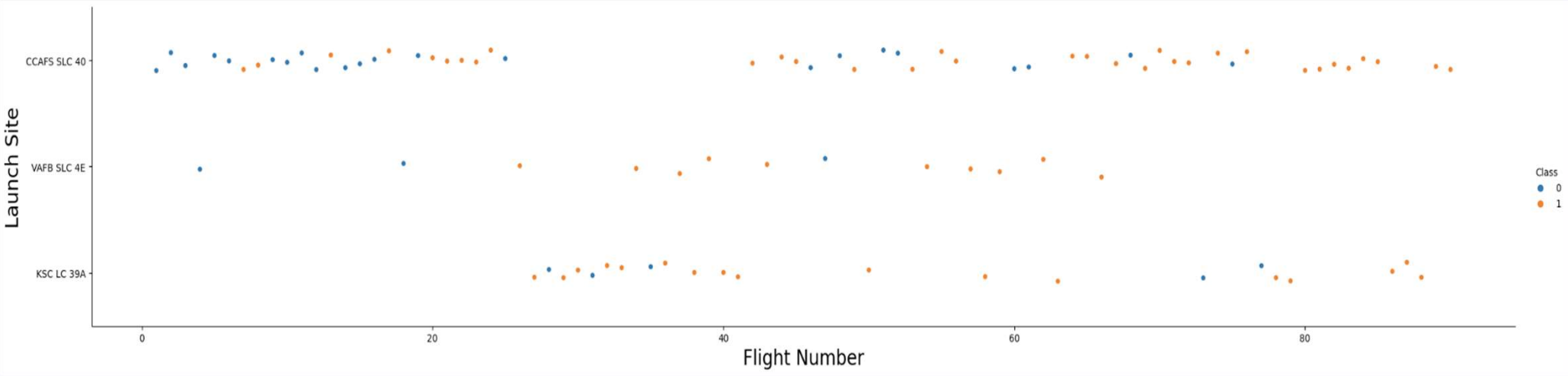


- Launch sites are typically located in safe coastal areas with robust surrounding infrastructure.
- The majority of launches occur at East Coast sites.

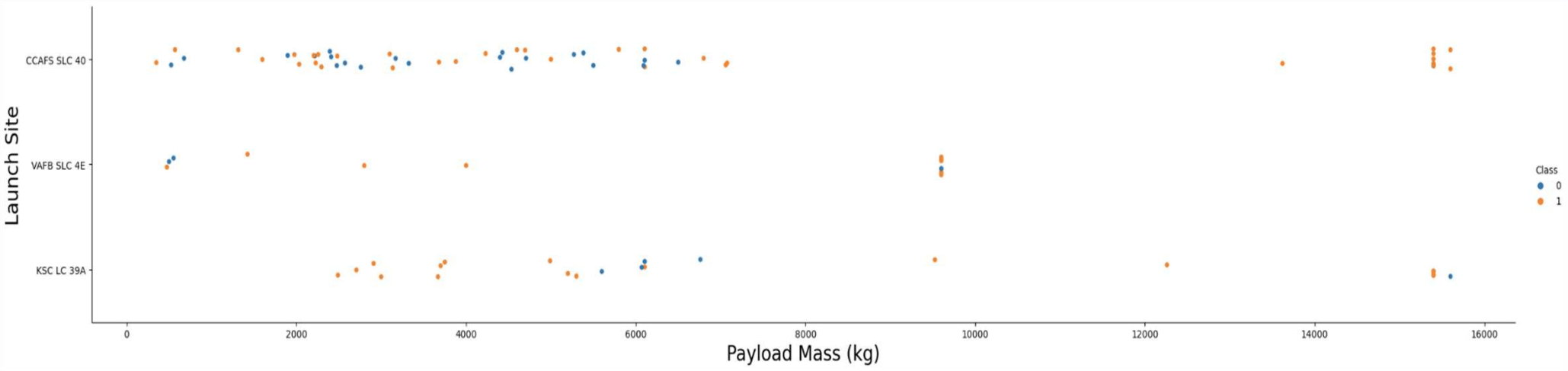

Section 2
Insights drawn
from EDA

Flight Number vs. Launch Site

The plot revealed that launch sites with a higher number of flights tend to have greater launch success rates.

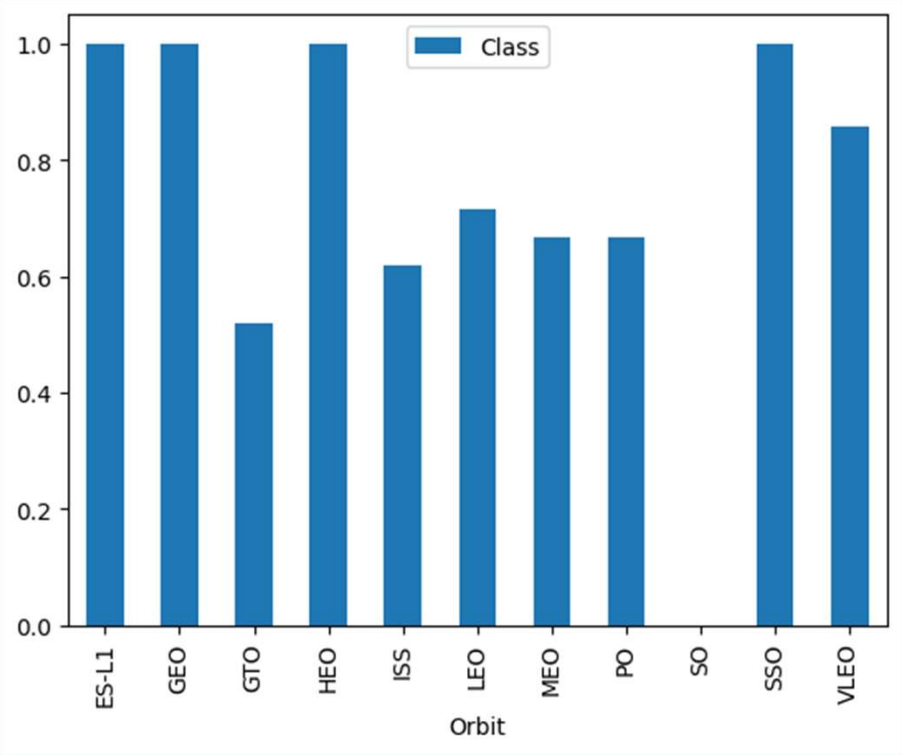


Payload vs. Launch Site



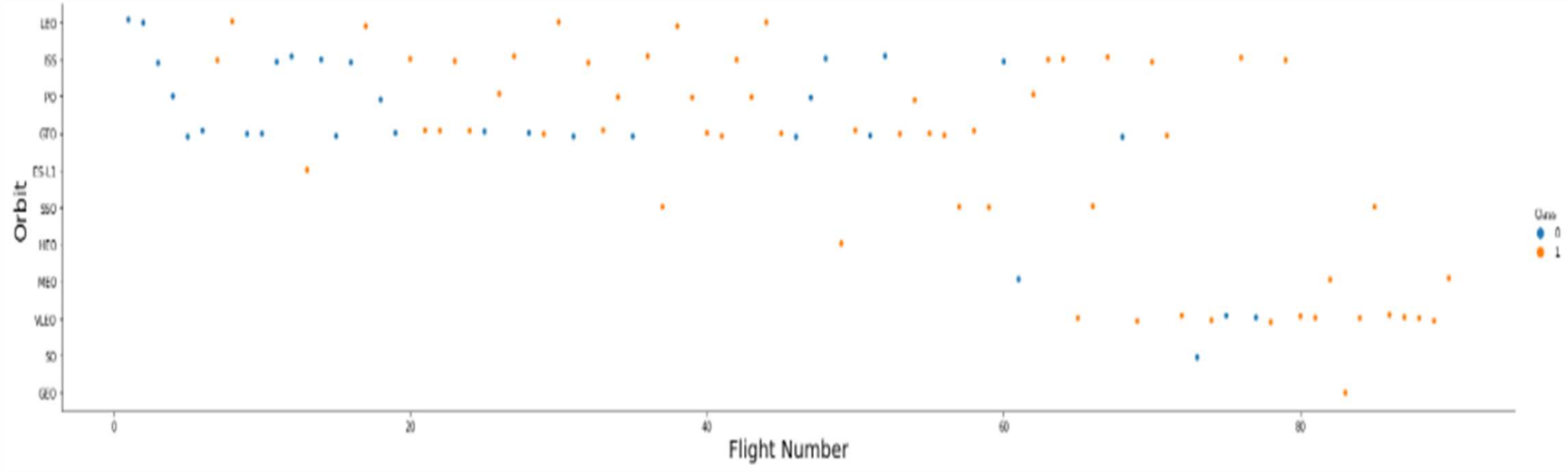
The Payload vs Launch Site plot shows no rockets with heavy payloads (greater than 10,000 kg) have been launched from the VAFB-SLC site.

Success Rate vs. Orbit Type



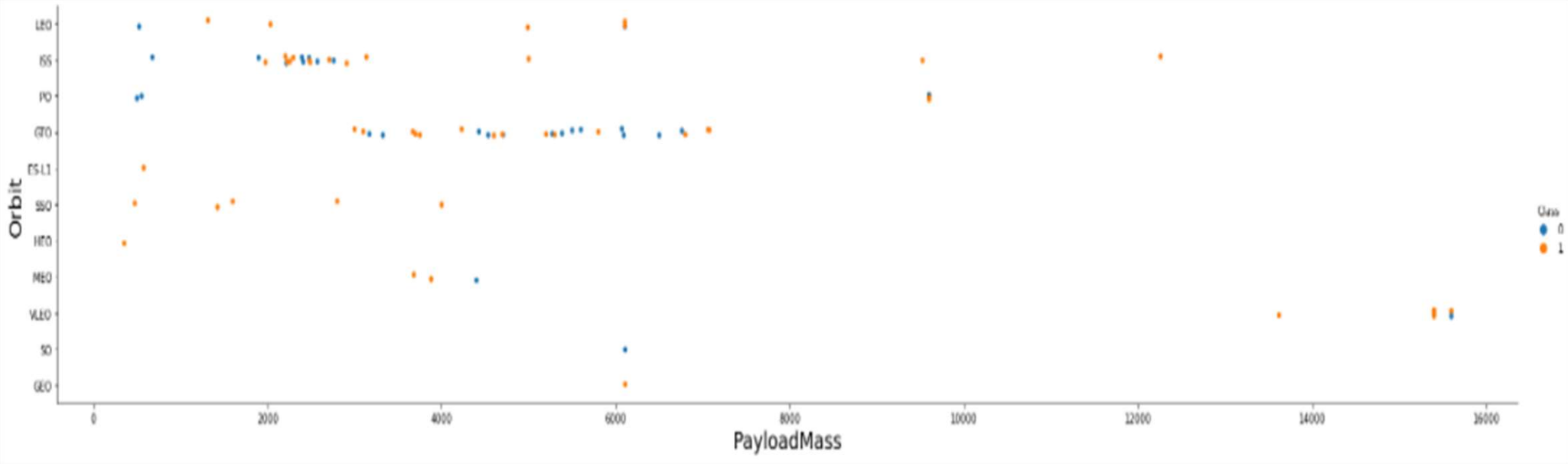
The plot shows the orbits with the highest success rates are ES-L1, GEO, HEO, SSO, and VLEO.

Flight Number vs. Orbit Type



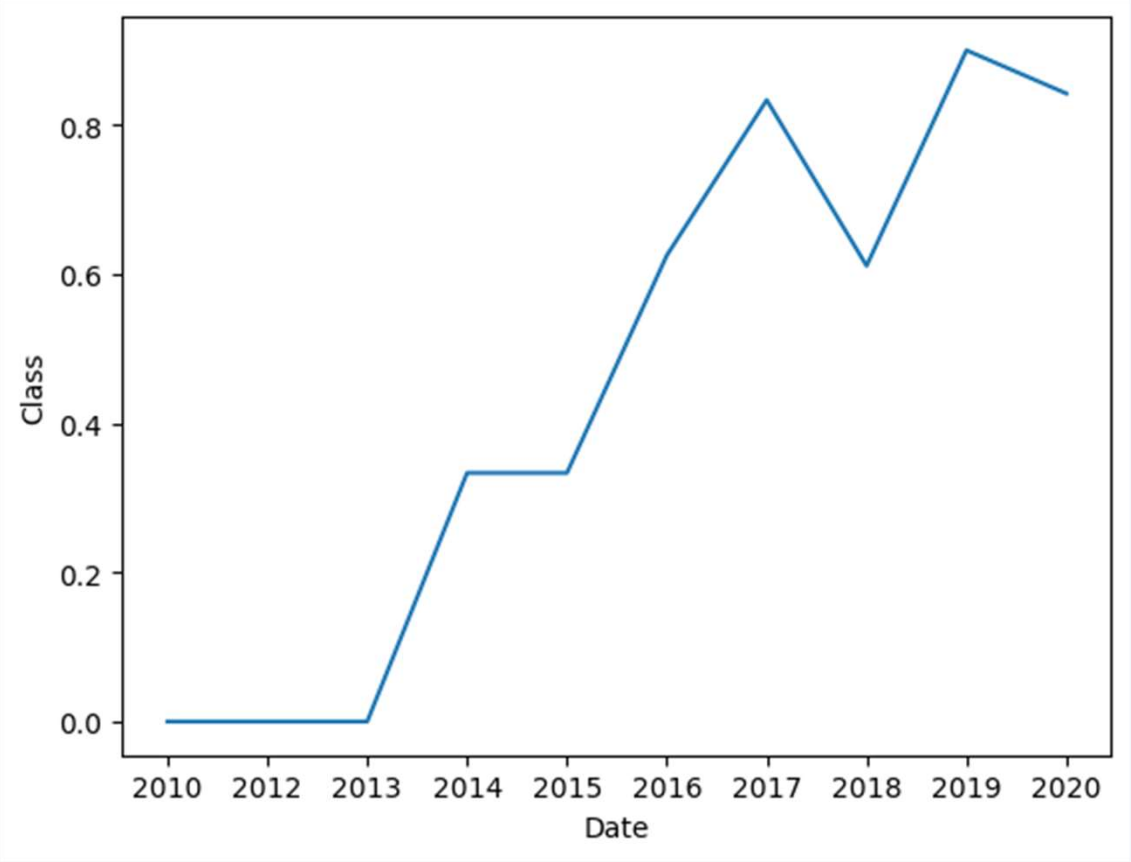
The Flight Number vs Orbit Type plot shows launches to LEO orbit have increasing success rates with more flights, whereas flights to GTO orbit display no clear trend between flight number and success.

Payload vs. Orbit Type



The plot shows successful landings with heavy payloads are more frequent for payloads to PO, LEO, and ISS orbits.

Launch Success Yearly Trend



The plot shows the success rate continually increased from 2013 to 2020.

All Launch Site Names

- We used the keyword **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

The query displays the top 5 launch site records where the site name begins with "CCA".

Total Payload Mass

The query calculates the total payload mass carried by NASA boosters as 45,596 kg.

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 query determines the average payload mass for the F9 v1.1 booster version is 2,928.4 kg.

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

The query shows the first successful ground pad landing occurred on December 22, 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

The query filters for boosters with successful drone ship landings having payloads between 4,000 kg and 6,000 kg using WHERE and AND clauses.

Total Number of Successful and Failure Mission Outcomes

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

The query uses a wildcard '%' with the WHERE clause to filter for all MissionOutcome values containing "success" or "failure".

Boosters Carried Maximum Payload

The query identifies the boosters carrying maximum payload using a subquery with the MAX() function in the WHERE clause.

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

The query filters for failed drone ship landings in 2015 using WHERE, LIKE, AND, and BETWEEN clauses to specify booster versions and launch sites.

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

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

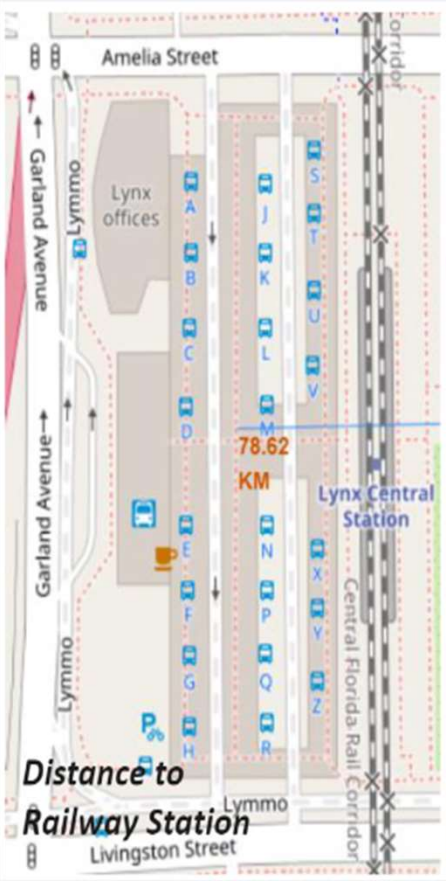
The query selects Landing outcomes and counts the outcomes WHERE the date is BETWEEN 2010-06-04 and 2017-03-20. It groups outcomes using GROUP BY and orders the groups in descending count with ORDER BY.


Section 3
Launch Sites
Proximities Analysis

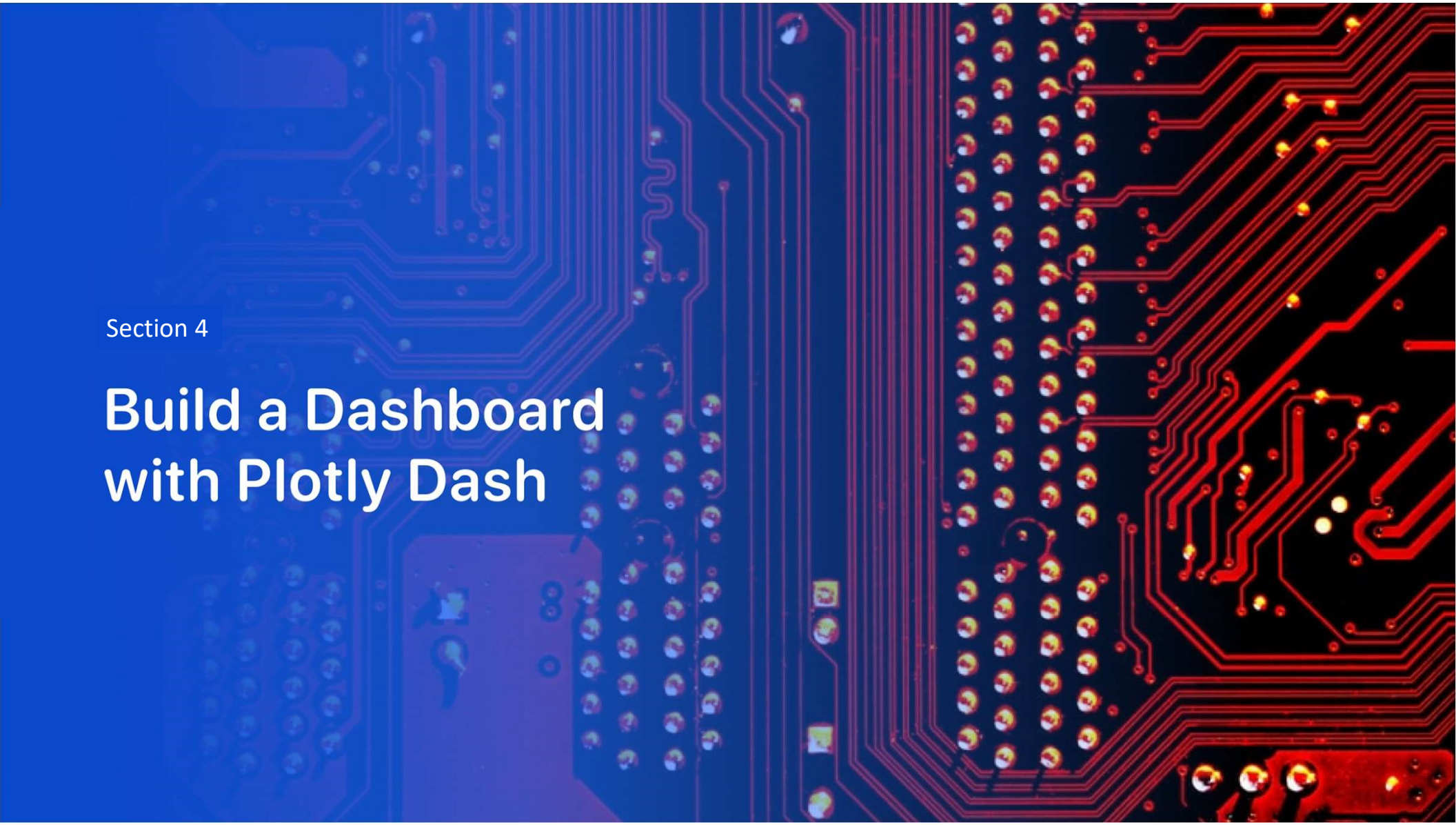
All launch sites global map markers

Markers showing launch sites with color labels

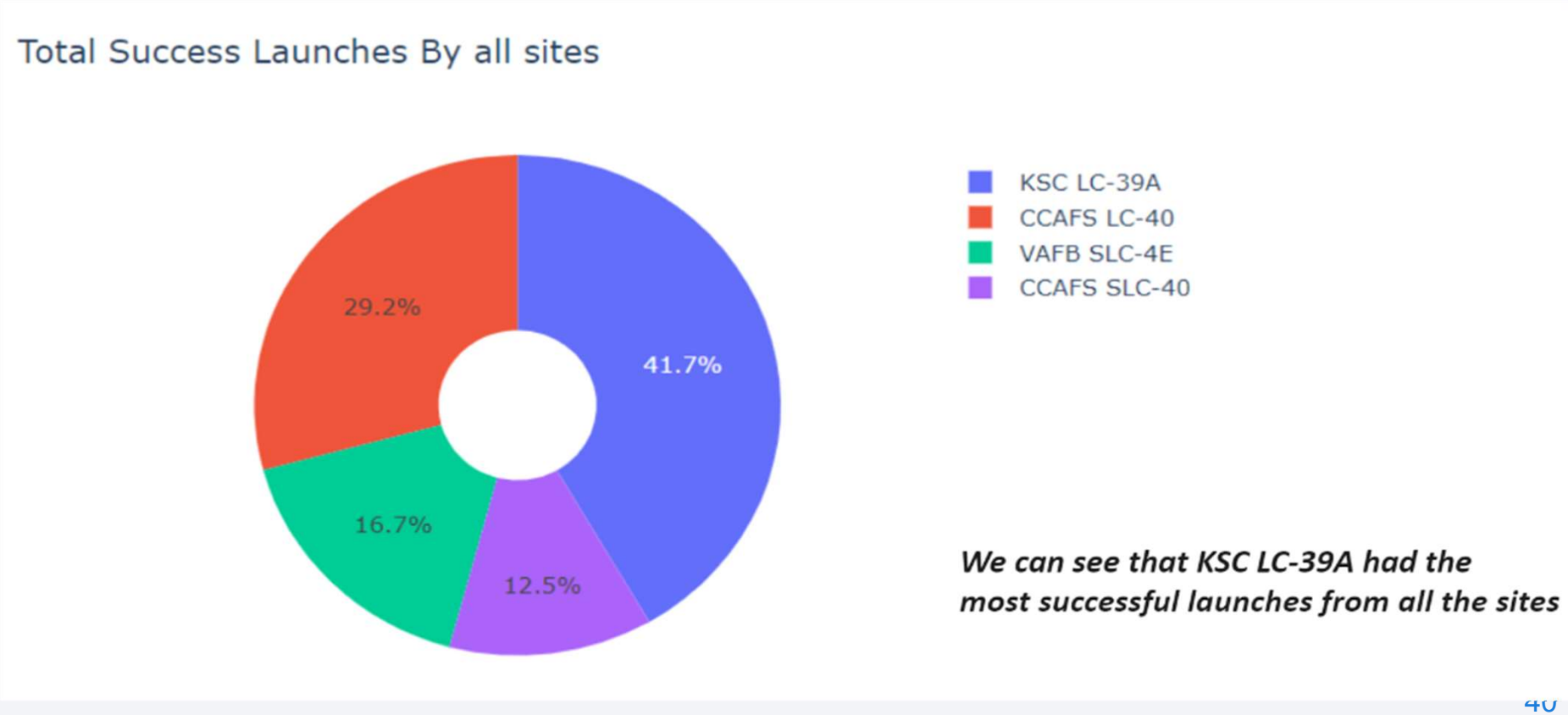
Launch Site distance to landmarks



- 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

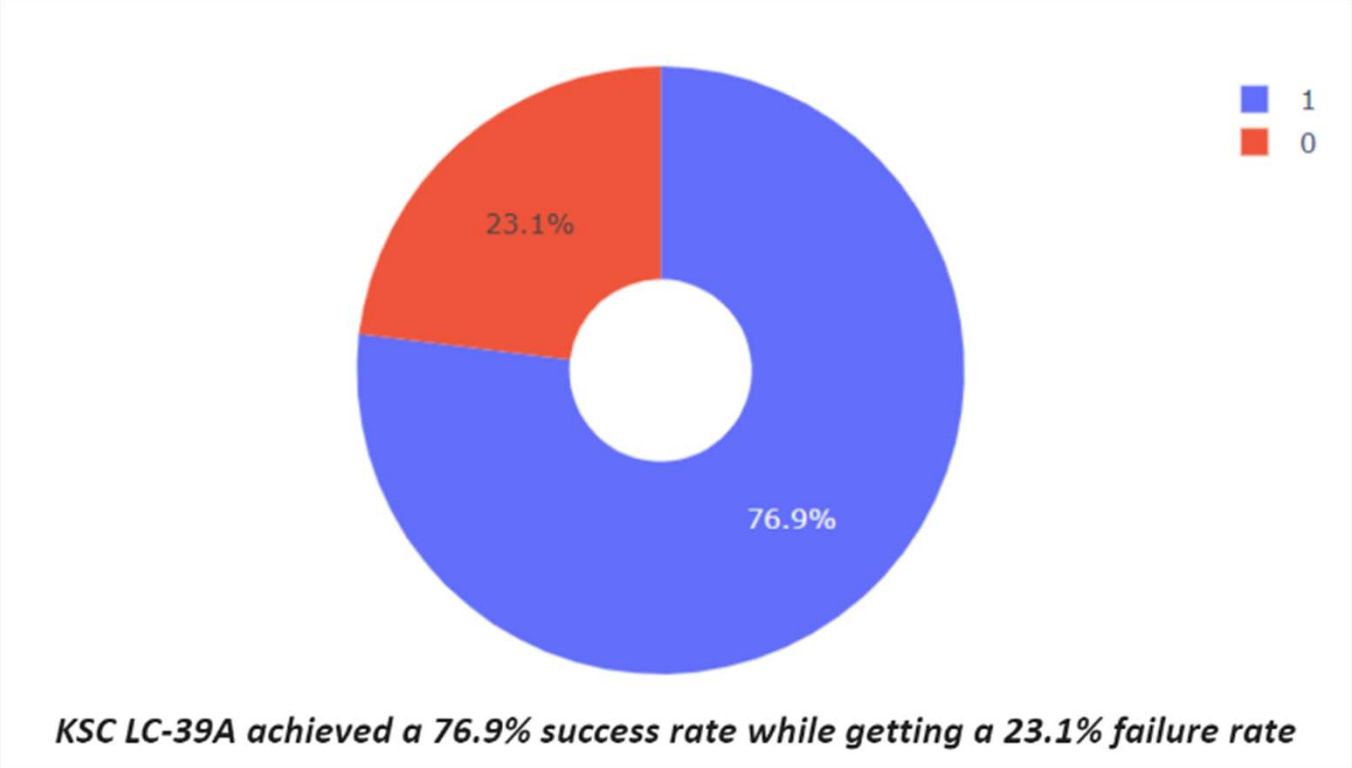

Section 4
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



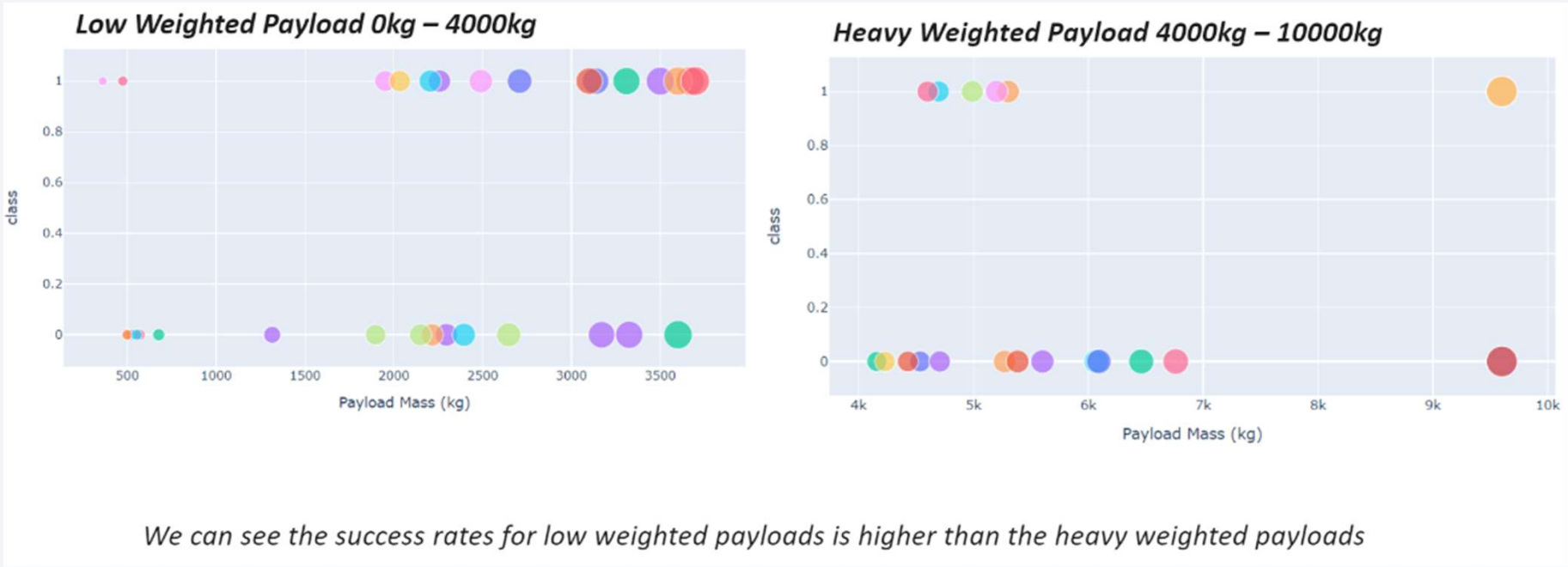
We can see that KSC LC-39A had the most successful launches from all the sites

Pie chart showing the Launch site with the highest launch success ratio



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

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider



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

Section 5

Predictive Analysis (Classification)

Classification Accuracy

The decision tree classifier is the model with the highest classification accuracy

Find the method performs best:

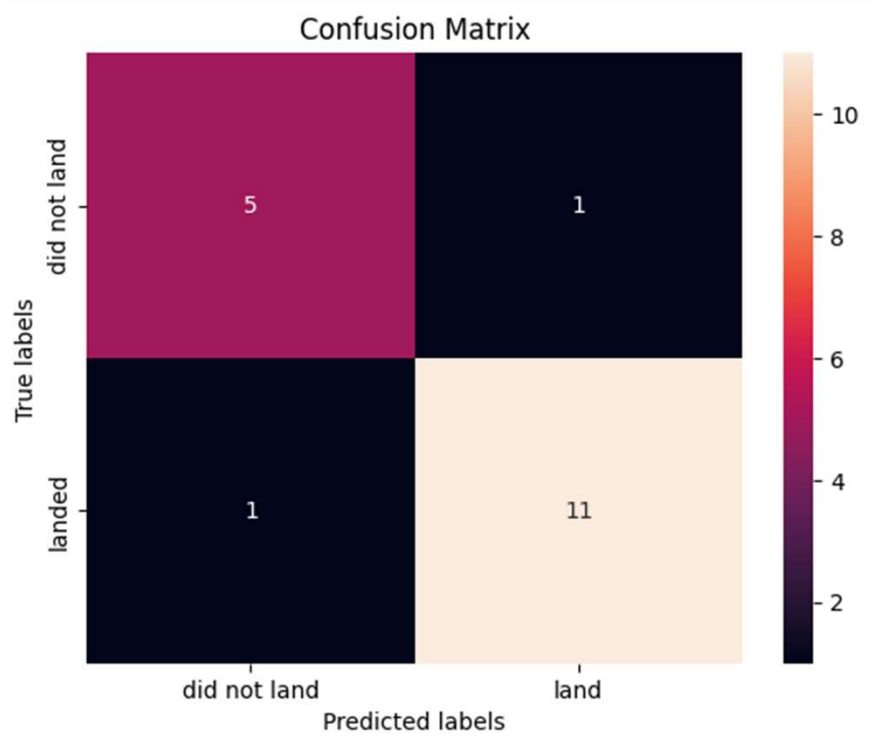
```
# Find the best method

test_results  = {'model_names': ['logreg_cv','svm_cv','tree_cv','knn_cv'], 'test_scores': []}

for model in test_results['model_names']:
    test_results['test_scores'].append(eval(model).score(X_test, Y_test))

test_results = pd.DataFrame(test_results)

# get the max score index
max_score_index = test_results['test_scores'].idxmax()

# print the model name with the max score
print("Best model ",test_results['model_names'][max_score_index]," with score=", test_results['test_scores'][max_score_index]   )
```

Python

```
Best model  tree_cv  with score= 0.8888888888888888
```

Confusion Matrix

The Decision Tree confusion matrix demonstrates the classifier can differentiate between classes. However, a key issue is false positives, where unsuccessful landings are incorrectly classified as successful.



Conclusions

- Launch sites with more flights have greater success rates
- Success rate increased steadily from 2013 to 2020
- Highest success rates achieved for orbits ES-L1, GEO, HEO, SSO, VLEO
- Most successful launches occurred at site KSC LC-39A
- Decision tree classifier is the optimal ML algorithm

Thank you!