IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sean Alimdjanov
September 4, 2023

Outline

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

Executive Summary

Summary of Methodologies

Data Collection

Data Wrangling

Exploratory Data Analysis

Interactive Visuals

Predictive Analysis with Classification Models

Summary of all results

Results of Exploratory Data Analysis

Analysis of Maps

Results of Predictive Analysis

Introduction

- In this capstone project, various tools were used to predict whether the first stage of SpaceX's Falcon 9 will land successfully
  - According to SpaceX:
    - A launch of Falcon 9 costs $62M
    - Competitors cost up to $165M
    - They save so much in comparison due to the ability to 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 SpaceX for a rocket launch.

Section 1

Methodology

Methodology

- Data collection methodology:
  - Requested rocket launch data from SpaceX's API
  - Performed Webscraping to extract data
- Performed data wrangling
  - Obtained the number of launches on each site
  - Amount and occurrences of each orbit type (e.g., LEO)
  - Displayed mission outcomes and occurrences with orbit types
  - Assigned success/failure outcomes to values 1/0

Methodology

- Performed exploratory data analysis (EDA) using visualization and SQL
  - Utilized Matplotlib to visualize data and observe patterns/trends
  - Queried data with SQL to evaluate the dataset
- Created interactive visual analytics using Folium and Plotly Dash
  - Used Folium to perform Geospatial Analysis
  - Plotly Dash was used to create an interactive dashboard

Methodology

- Performed predictive analysis using classification models
  - The following was performed through use of Scikit-learn:
    - Preprocess the data
    - Create test and training data for the model
    - Train various classification models
    - Obtain hyperparameters

Data Collection – SpaceX API

1. Made a Request to the SpaceX API as follows:

```
response = requests.get(spacex_url)
```

2. Converted Data into a Pandas DataFrame:

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

3. Cleaned up the requested data. This involved using custom functions to clean and retrieve data to fill lists pertaining to rocket parameters:

- The resulting lists were then used to create a dictionary `launch_dict`
- The dictionary was then used to create a new DataFrame called `df`

Data Collection – SpaceX API (Continued)

4. Filtered the DataFrame to include Falcon 9 data only:

```
data_falcon9 = data[data['BoosterVersion'] != 'Falcon 1']
```

5. Fixed the flight Number column due to the removal of values:

```
data_falcon9.loc[:, 'FlightNumber'] = list(range(1, data_falcon9.shape[0] + 1))
```

6. Replaced missing Payload mass values with the mean value:

```
mean_payload_mass = data_falcon9['PayloadMass'].mean()
data_falcon9['PayloadMass'].replace(np.nan, mean_payload_mass, inplace=True)
```

Notebook URL: https://github.com/ssa72/Applied-Data-Science-Capstone/blob/main/1-lab-spacex-data-collection-api.ipynb

Data Collection – Scraping

1. Requested the Falcon9 Launch Wiki page from its URL and created a Beautiful Soup object from the response:

```
response = requests.get(static_url)
soup = BeautifulSoup(response)
```

2. Extracted all column/variables names from the HTML table header:

```
html_tables = soup.find_all('table')
```

3. Created a DataFrame by parsing the HTML tables through creating a dictionary from the column names:

```
launch_dict = dict.fromkeys(column_names)
```

Iterating through the data with a for loop allowed us to add data from each launch to the data frame with the following line at the end of the loop:

```
df = df.append(launch_dict, ignore_index=True)
```

Notebook URL: https://github.com/ssa72/Applied-Data-Science-Capstone/blob/main/2-lab-spacex-webscraping.ipynb

Data Wrangling

1. Obtained the number of launches on each site, number and occurrence of each orbit, and mission occurrence of each mission based on orbit type with the value_counts() method:

```
df['LaunchSite'].value_counts()
df['Orbit'].value_counts()
df['Outcome'].value_counts()
```

2. Created the booster landing outcome label. The value is 1 if the landing outcome is successful and 0 if unsuccessful:

- Iterated through the keys of the `landing_outcomes` list
- Created a set `bad_outcomes` that contains all bad outcomes
- If the i-th landing outcome is in `bad_outcomes`, append 0 to a new list called `landing_class`. If not, append 1.
- Added a column to the DataFrame with this list.

Notebook URL: https://github.com/ssa72/Applied-Data-Science-Capstone/blob/main/3-lab-spacex-data-wrangling.ipynb

EDA with Data Visualization

- Created Scatter Plots to illustrate the following relationships:
  - Flight Number vs. Launch Site
  - Payload Mass vs. Launch Site
  - Flight Number vs. Orbit Type
  - Payload Mass vs. Orbit Type
- Plotted the Yearly Success Rate on a Line Chart.
- A Bar Chart showed the Success Rates with respect to Orbit Type.
- GitHub URL: https://github.com/ssa72/Applied-Data-Science-Capstone/blob/main/5-lab-eda-data-visualization.ipynb

EDA with SQL

Summary of SQL Queries Performed:

- Displayed the names of the unique launch sites in the space mission
- Displayed 5 records where launch sites begin with the string 'CCA'
- Displayed the total payload mass carried by boosters launched by NASA (CRS)
- Displayed the average payload mass carried by booster version F9 v1.1
- Listed the date when the first successful landing outcome in ground pad was achieved.
- Listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- Listed the total number of successful and failure mission outcomes
- Listed the names of the Booster Versions which have carried the maximum payload mass
- Listed the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
- Ranked the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

GitHub URL:
https://github.com/ssa72/Applied-Data-Science-Capstone/blob/main/4-lab-eda-sql-sqllite.ipynb

Build an Interactive Map with Folium

- Created a Folium Map with the following features:
  - Circles to highlight launch site locations
  - Markers to highlight the successful/failed launches with green/red sub markers
  - Displayed proximity to nearest coastlines, railroads, highways, and cities.
- GitHub URL: https://github.com/ssa72/Applied-Data-Science-Capstone/blob/main/6-lab-launch-site-locations.ipynb

Built a Dashboard with Plotly Dash

- Created a Plotly Dash Application with the following plots:
  - Pie Charts: Successful Launch Rate vs. Launch Site and Successful/Unsuccessful Launches per selected Launch Site (selection from dropdown-menu)
  - Scatter Plot: Payload Mass vs. Mission Success within a selected payload mass interval, via an interactive slider
- GitHub URL: https://github.com/ssa72/Applied-Data-Science-Capstone/blob/main/7-spacex_dash_app.py

Predictive Analysis (Classification)

1. Model Development:

Performed exploratory data analysis and determined training labels. Created a column for the classification, standardized the test data, and split data into training and test sets for model evaluation.

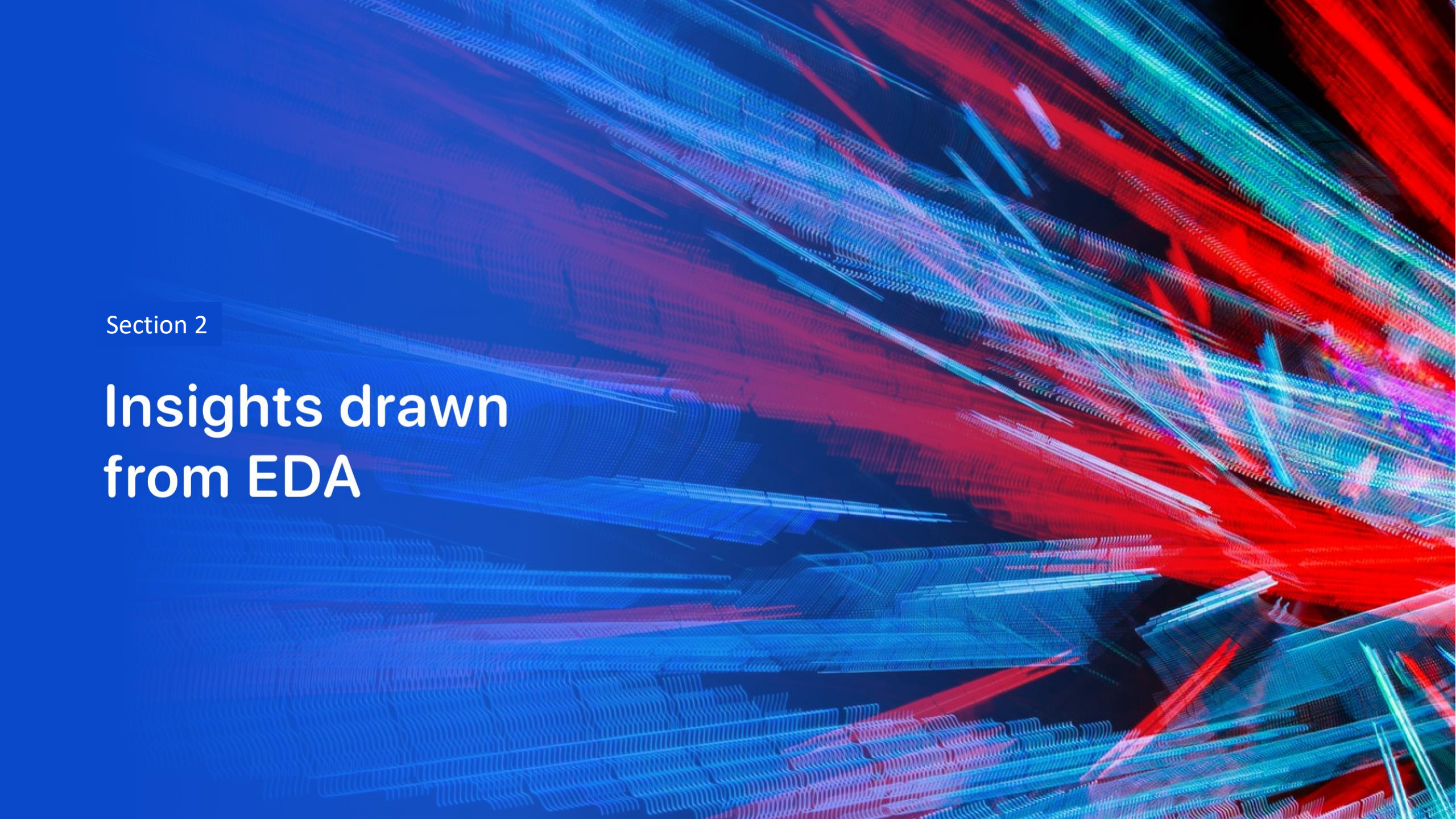
2. Model Evaluation:

Utilized Support Vector Machines (SVMs), Classification Trees, and Logistic Regression. Calculated accuracy on the test data using the `.score()` method and found the best hyperparameters for each model.

3. Conclusions:

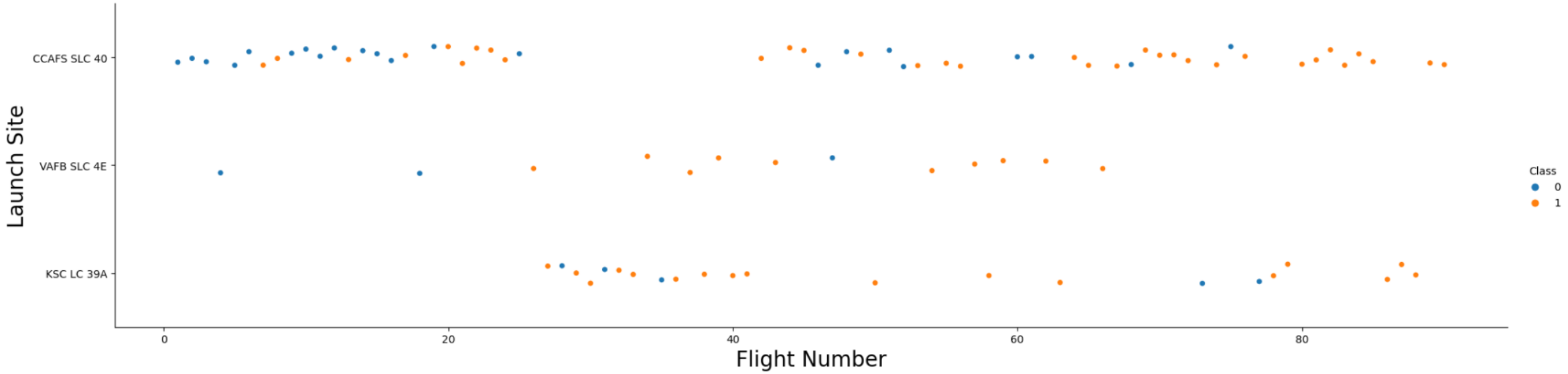
Which Model Performs the Best on the given data?

Notebook URL: https://github.com/ssa72/Applied-Data-Science-Capstone/blob/main/8-spacex-machine-learning-prediction.ipynb


Section 2
Insights drawn from EDA

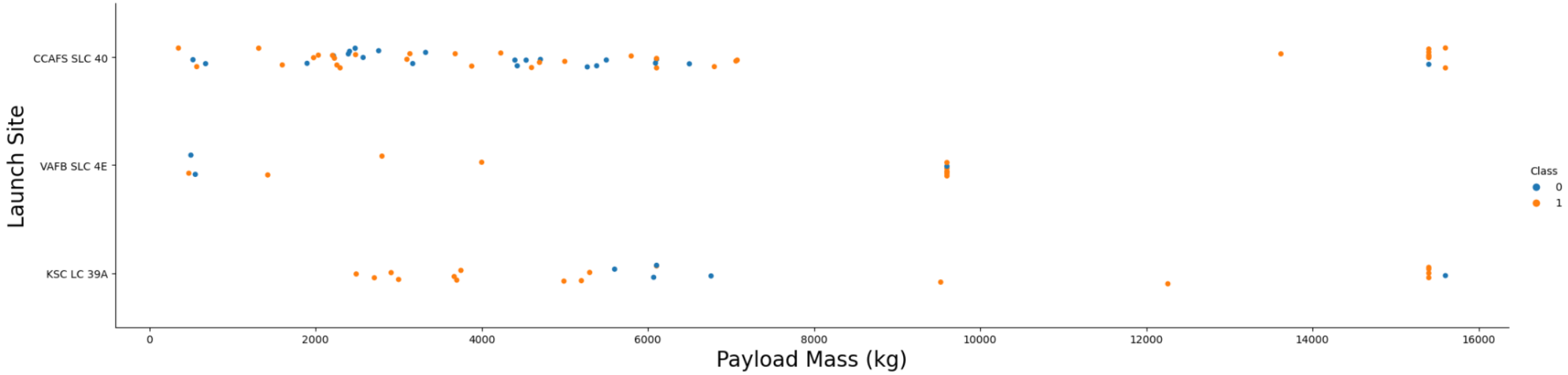
Results

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



Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site
- As the flight number increases, the number of successful launches also increase on average

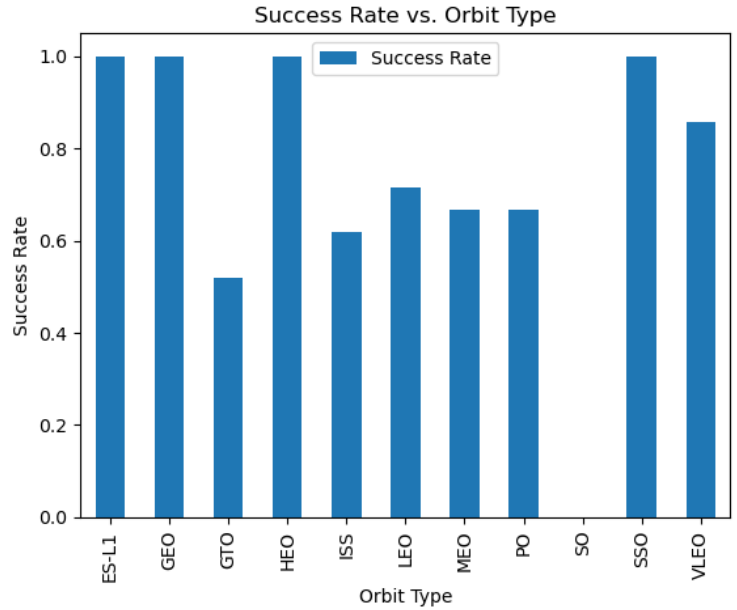


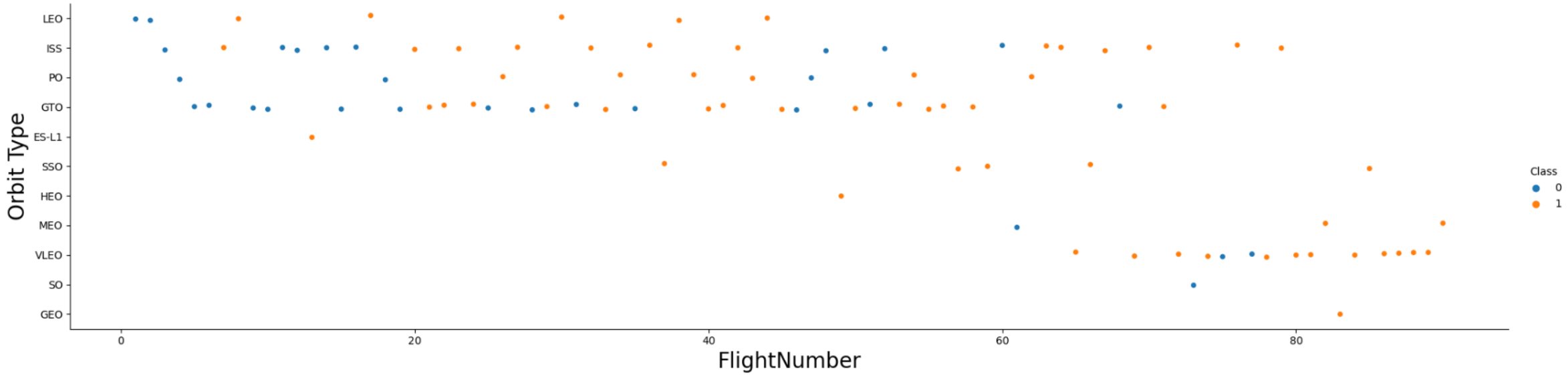
Payload Mass vs. Launch Site

- Scatter plot of Payload Mass vs. Launch Site
- In general, success rates are higher for each launch site when the payload mass exceeds 7,000 kg

Success Rate vs. Orbit Type

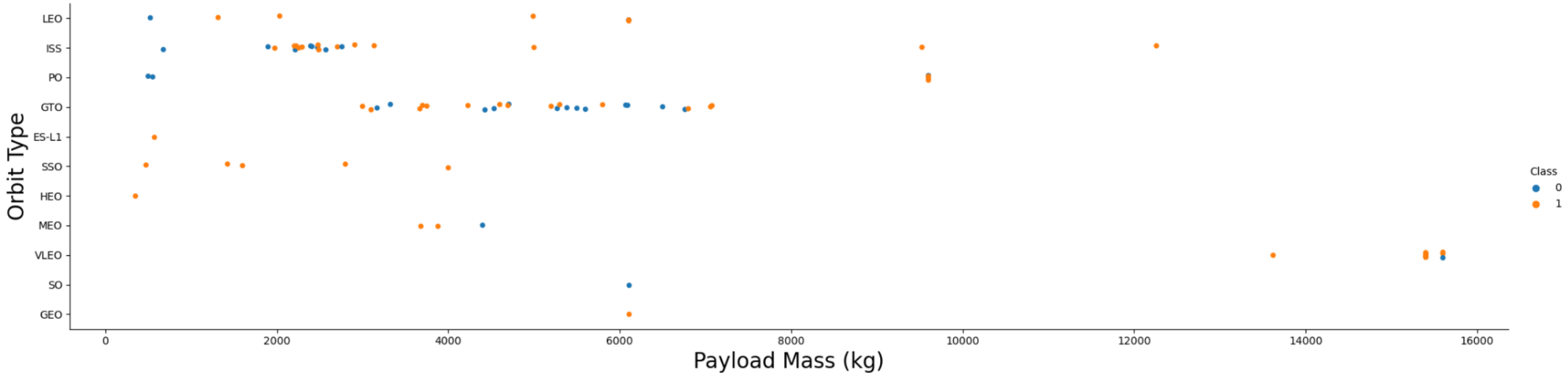
- The most successful Orbit Types are ES-L1, GEO, HEO, and SSO with 100% success rates
- Orbit Type SO always failed
- The other orbits have success rates between 50% and 75%





Flight Number vs. Orbit Type

- Scatter Plot of Flight number vs. Orbit type
- In general, success Rate increases as flight number increases.

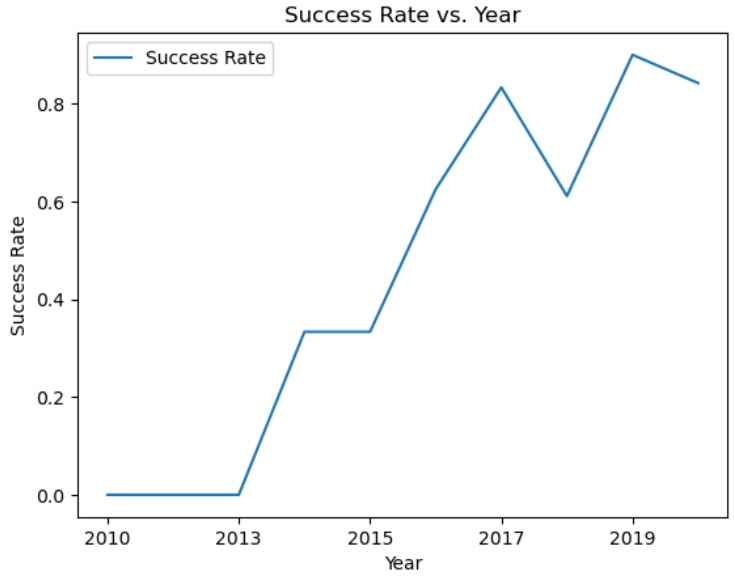


Payload vs. Orbit Type

- Scatter Plot of Payload Mass vs. Orbit Type
- LEO, ES-L1, SSO, and HEO have lower success rates at masses greater than 7000 kg
- ISS and PO have higher success rates at heavier masses

Launch Success Yearly Trend

- All launches were unsuccessful up to 2013
- Success rates sky-rocketed from 2013-2017
- A decline was observed in 2018 but rates recovered for 2019



All Launch Site Names

```
%sql SELECT DISTINCT("Launch_Site") FROM SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE '%CCA%' LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
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

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

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1%'
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Success%ground pad%'
```

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

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000 kg

```
%%sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT Mission_Outcome, COUNT(Mission_Outcome) FROM SPACEXTBL GROUP BY Mission_Outcome
```

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

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

Boosters that Carried Maximum Payload

```
%%sql

SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL
)
```

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

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

```
%%sql

SELECT Date, substr(Date, 0, 5) , substr(Date, 6, 2) AS "Month", Landing_Outcome, Booster_Version, Launch_Site
FROM SPACEXTBL WHERE substr(Date, 0, 5) = '2015' AND Landing_Outcome LIKE 'Failure%drone%ship%'
```

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

Date	substr(Date, 0, 5)	Month	Landing_Outcome	Booster_Version	Launch_Site
2015-10-01	2015	10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	2015	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
%%sql

SELECT Landing_Outcome, COUNT(*) AS Total
FROM SPACEXTBL
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
    GROUP BY Landing_Outcome ORDER BY COUNT(*) DESC
```

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

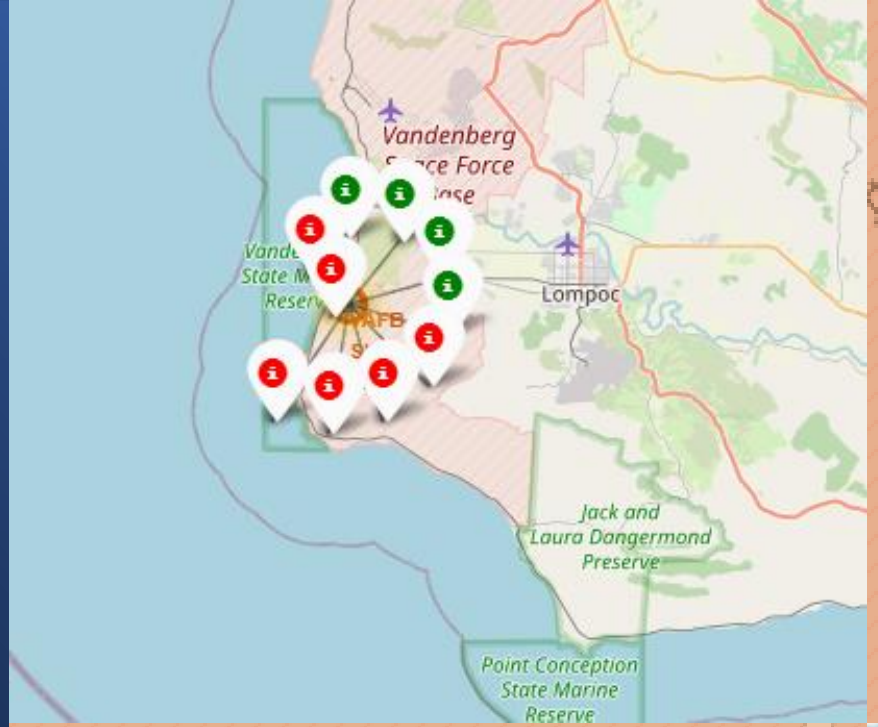
Landing_Outcome	Total
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

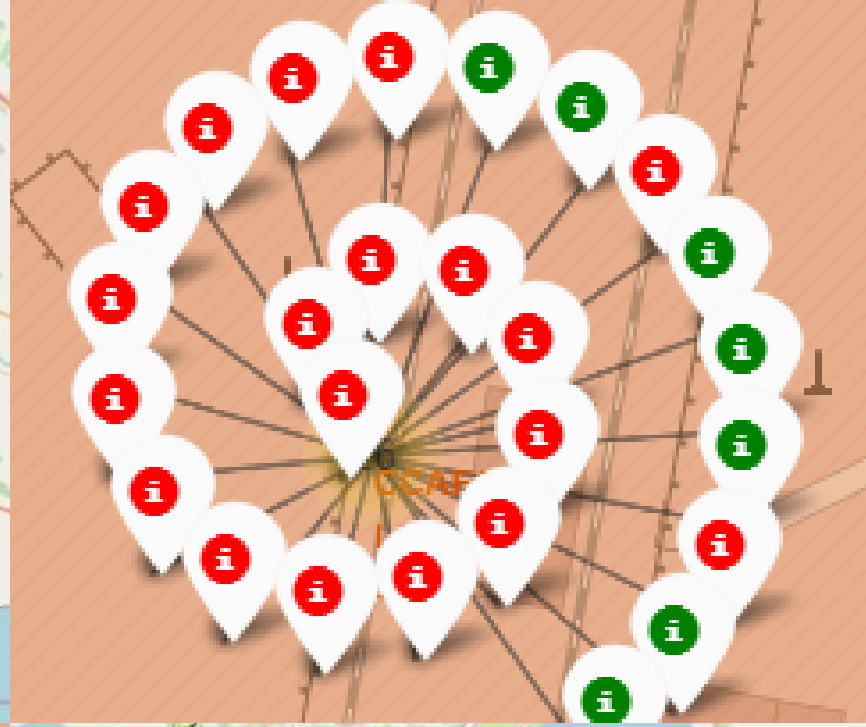
Section 3

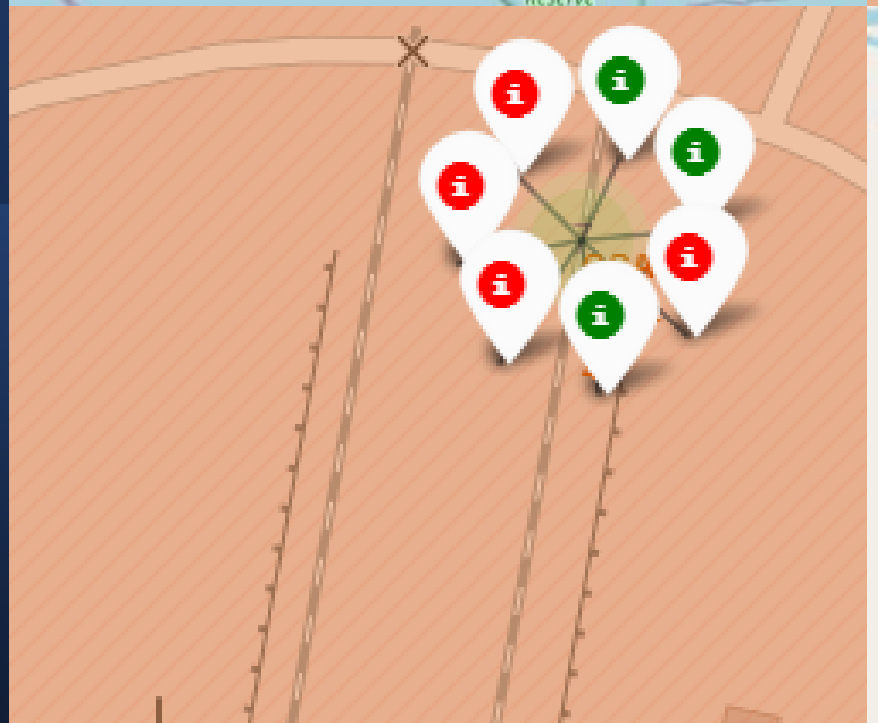
Launch Sites Proximities Analysis

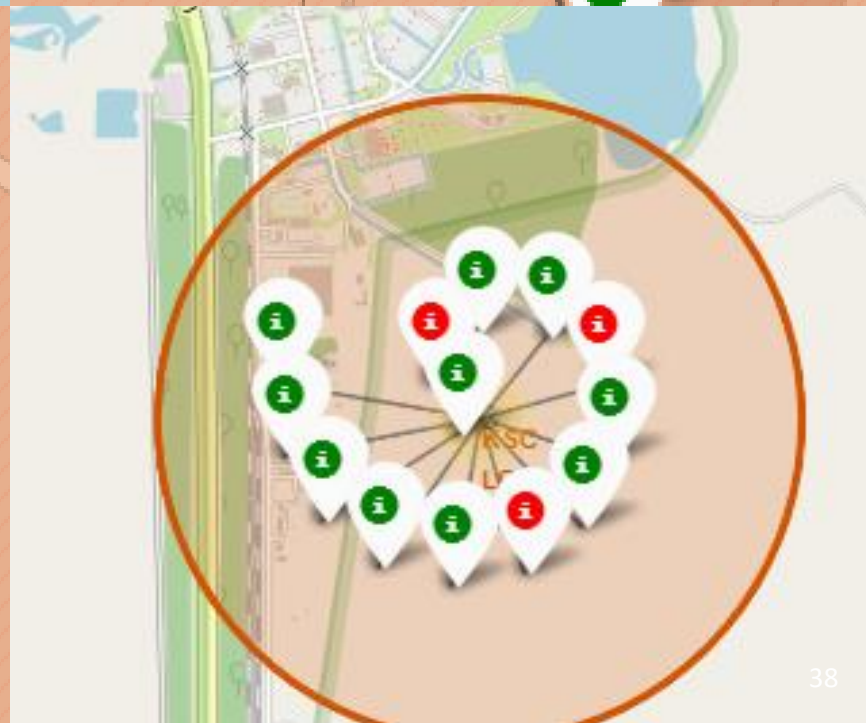
Marked All Launch Sites on the Map

Marked the Successful/ Failed Launches at Each Site



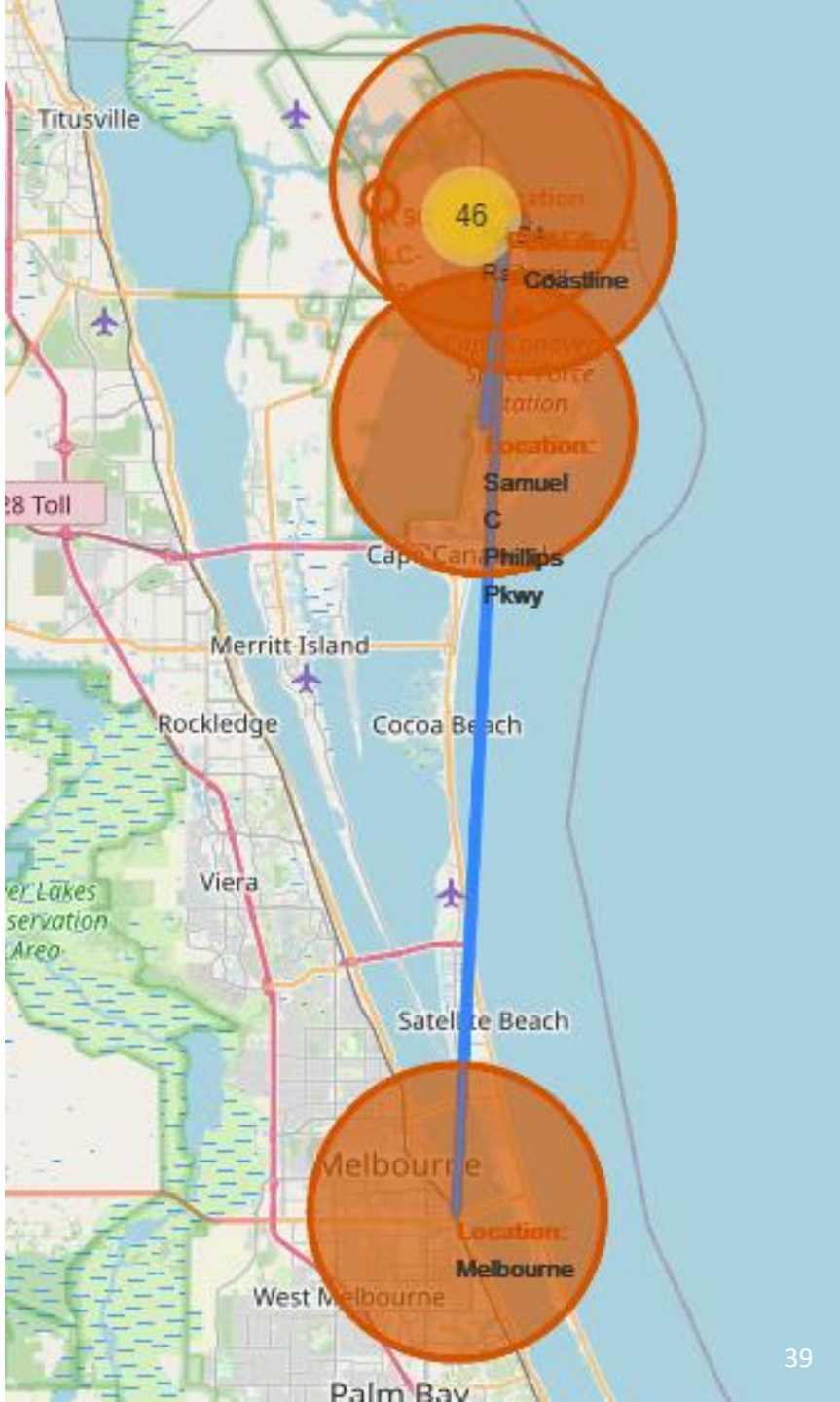


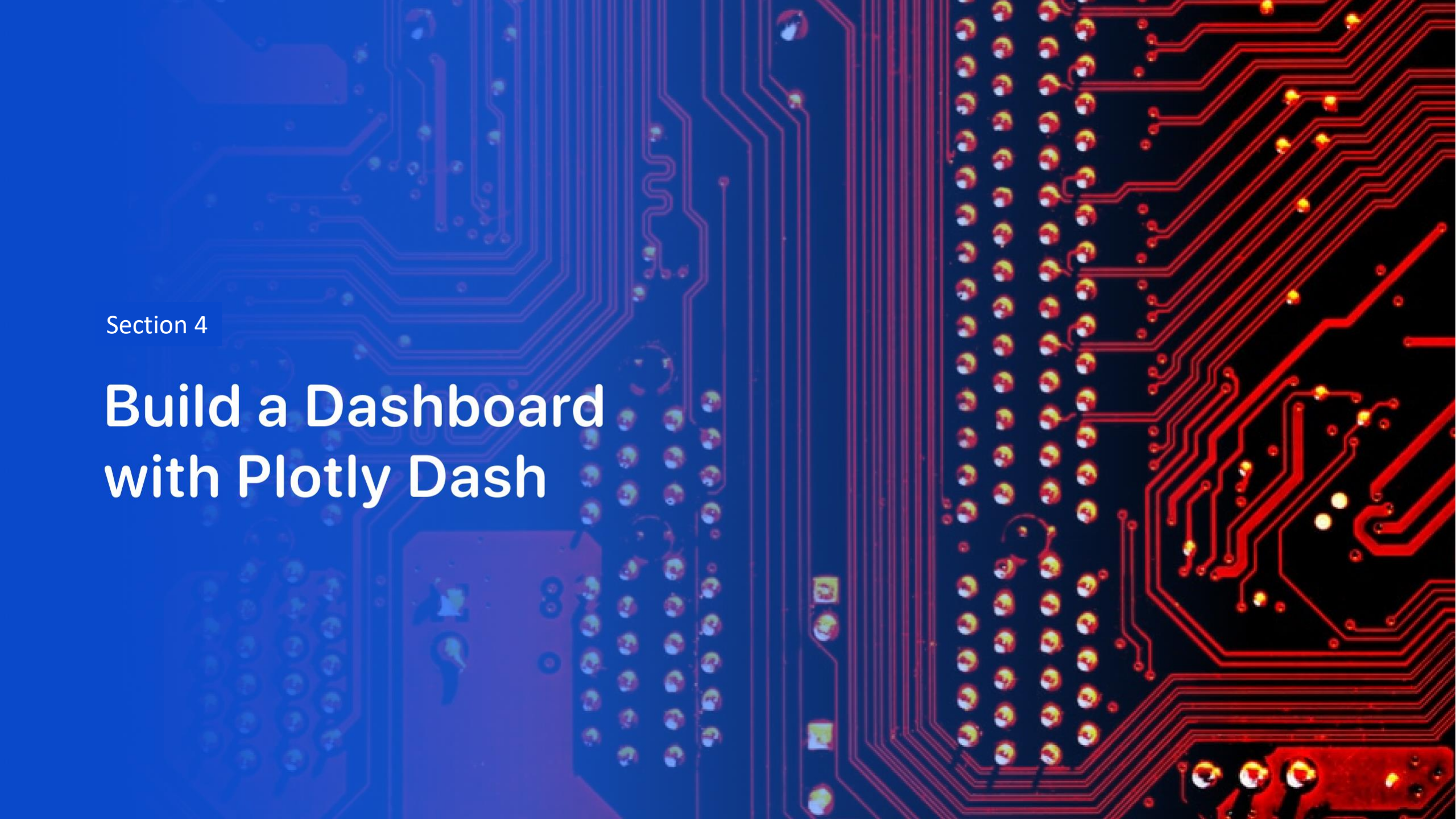




Proximities to Landmarks

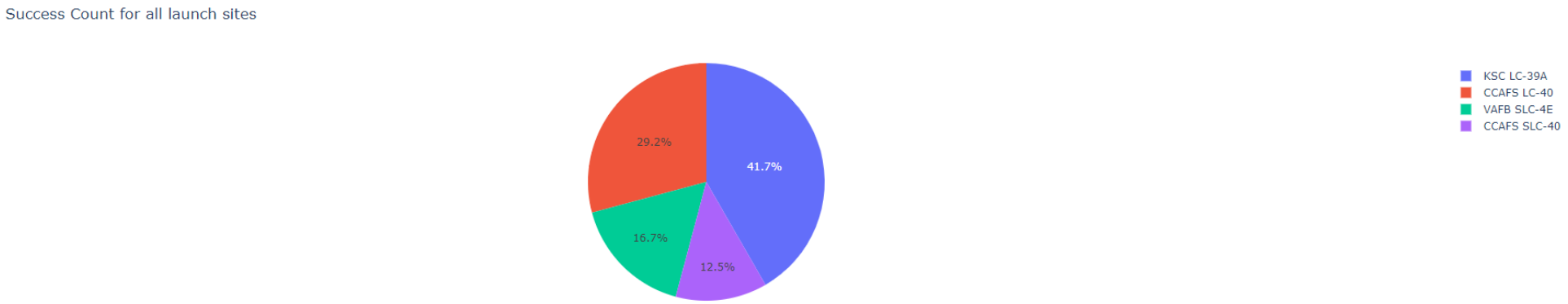
- Plotted lines connecting the launch site to the nearest highway (Samuel C. Phillips Pkwy), nearest coastline, nearest City (Melbourne), and nearest railway (NASA Railway)



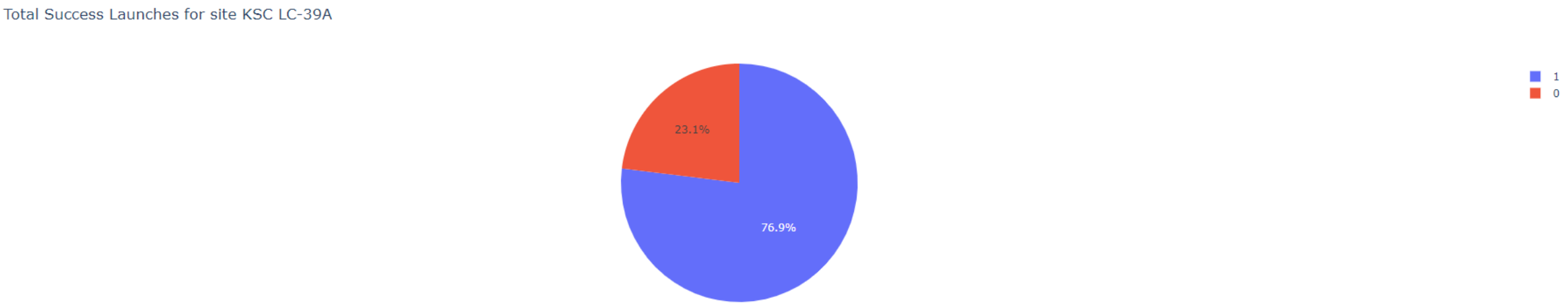

Section 4
Build a Dashboard with Plotly Dash

Pie Chart: Success Rates for All Launch Sites

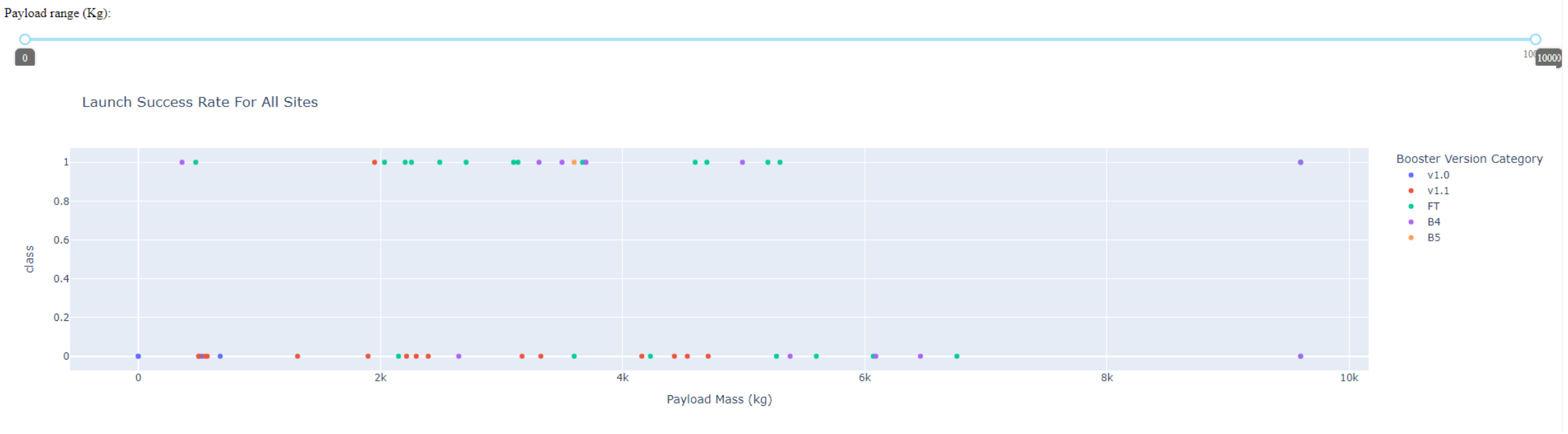
- Launch Site KSC LC-39A had the highest success rate:



Successful Launches for KSC LC-39A



Scatter Plot: Launch Success Rate (All sites) vs. Payload Mass



Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Decision Tree was the model with the higher ‘Best Score’ of 87.5%

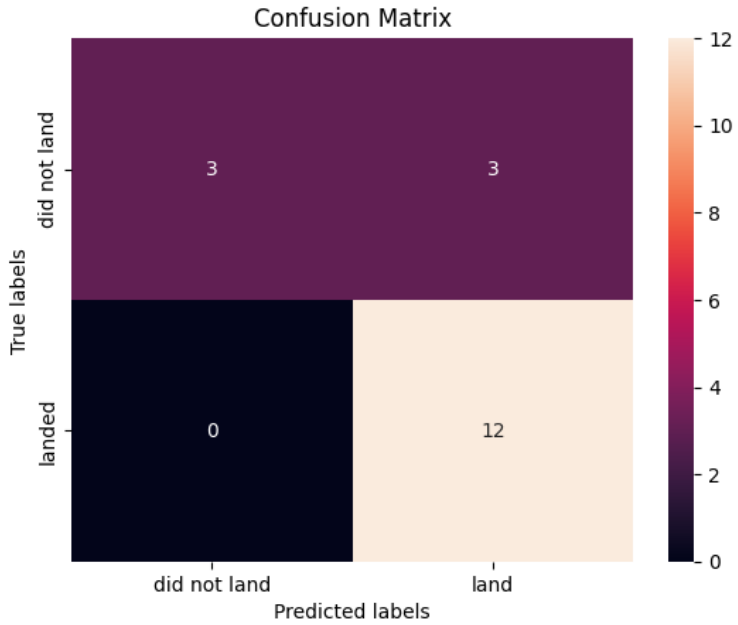
```
results_dict = {
    'Logistic Regression': {'Accuracy Score': lr_score, 'Best Score': logreg_cv.best_score_},
    'Support Vector Machine': {'Accuracy Score': svm_score, 'Best Score': svm_cv.best_score_},
    'Decision Tree': {'Accuracy Score': tree_score, 'Best Score': tree_cv.best_score_},
    'k Nearest Neighbors': {'Accuracy Score': knn_score, 'Best Score':knn_cv.best_score_}
}

score_df = pd.DataFrame(results_dict)
score_df
```

	Logistic Regression	Support Vector Machine	Decision Tree	k Nearest Neighbors
Accuracy Score	0.833333	0.833333	0.833333	0.833333
Best Score	0.846429	0.848214	0.875000	0.848214

Confusion Matrix

- The confusion matrix for the Decision Tree model.
- Shows that there are only three false positives and zero false negatives



Conclusions

- In general, success rate increases as the flight number increases.
- Launch sites VAFB SLC 4E and KSC LC 39A have the highest success rates.
- For every launch site, a payload mass over 7,000 kg yields a better result. However, at VAFB SLC 4E and KSC LC 39A, payload masses less than 6,000 kg are more successful
- There were 99 successful missions and 1 unsuccessful mission between 2010 and 2019
- The highest performing classification model was the Decision Tree model

Thank you!