IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Dan Marks
December 26, 2023

# Outline

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

# Executive Summary

- Methodologies
  - Data collection via REST API calls and webscraping, storage in SQL DB
  - Data wrangling in Pandas data frame and EDA with SQL
  - Additional Visualization with Plotly & Folium
  - Predictive analysis using ML classification models (Scikit learn)
- Summary of results
  - The models predict with fairly high certainty, based on the parameters provided, the success of any given mission based on the payload mass, type of orbit, and other input features.  Logistic Regression selected.
  - The cost per successful mission is the quotient of the total cost (for all missions) and the number of successful missions.

# Introduction

- The commercial space age is here, companies are making space travel affordable for everyone
- SpaceX has reduced costs significantly by reusing first-stage launch components
  - SpaceX launch costs are ~$62M; other are upwards of ~$165M
- Goals:
  - Determine if the first stage will land based on launch data
  - Determine the cost of a launch.
- Approach:
  - Use Machine Learning to predict success of a given launch (and thus the reuse of components)
  - Aggregate data to determine per-launch costs

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - Data frames read from SpaceX Launch Data API and scraped from SpaceX Wikipedia page
- Perform data wrangling
  - Data frames in pandas were analyzed and manipulated to clean, transform, load data into SQL DB
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Scikit-learn utilized on transformed Pandas data frames and data series

# Data Collection

- Data frames read from SpaceX Launch Data API
  - https://api.spacexdata.com/v4/launches/past
  - Utilized Python ‘requests’ package for REST calls
  - Response text converted from JSON file and converted to Pandas data frame
- Additional data scraped from SpaceX Wikipedia page
  - https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches
  - Utilized Python ‘requests’ package and ‘BeautifulSoup’ (bs4) packages
  - Parsed table information using bs4 methods

# Data Collection – SpaceX API

- GitHub URL of the completed SpaceX API calls notebook:
  - https://github.com/dmarks84/IBM-DS-Capstone/blob/main/1%20Data%20Collection%20API.ipynb

- GET (JSON data)
  - requests.get(spacex launch url)
- CONVERT JSON
  - response.json()
- NORMALIZE
  - pd.json_normalize(response)
- INITIAL TRANSFORMATION
  - Multiple Pandas data frame manipulations
- CONVERT TO CSV
  - pd.to_csv(df)

# Data Collection - Scraping

- GitHub URL of the completed webscraping notebook:
  - https://github.com/dmarks84/IBM-DS-Capstone/blob/main/2%20Webscraping.ipynb

- GET (html data)
  - requests.get(spacex launch wiki)
- CONVERT TO BS4 OBJECT
  - BeautifulSoup(response.text)
- FIND TABLE DATA
  - soup.find_all('table')
- FIND LAUNCH ROWS
  - soup_table.find_all('th')
- FIND LAUNCH DATA
  - soup_table.row_all('tr')
- CONVERT TO CSV
  - pd.to_csv(df)

# Data Wrangling

- Data was reviewed for missing/null values and for data types
- Data was reviewed by launch site, orbit type, and mission success to identify meaningful metrics
- Class feature was engineered to record success of failure of each launch
- Data stored in SQL Database
- GitHub URL of the completed data wrangling notebook:
  - https://github.com/dmarks84/IBM-DS-Capstone/blob/main/3%20Data%20Wrangling.ipynb

# EDA with SQL

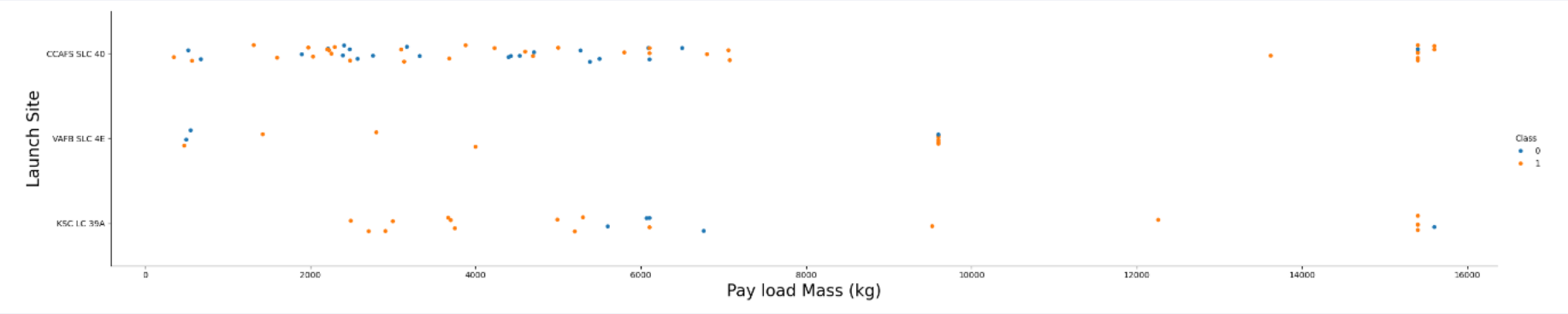
- Example SQL queries utilized and results:

| Query | Result |
|---|---|
| SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Customer" = "NASA (CRS)" | Total payload mass carried by boosters launched by NASA |
| SELECT MIN("Date") AS "First Successful Landing" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE "Success%" | Data of first successful landing outcome |
| SELECT "Landing_Outcome", COUNT("Landing_Outcome") AS "Count" FROM SPACEXTABLE WHERE ((substr("Date",0,5)*365 + substr("Date",6,2)*12 + substr("Date",9,2)) BETWEEN (2010*365+6*12+4) AND (2017*365+3*12+20)) GROUP BY "Landing_Outcome" ORDER BY COUNT("Landing_Outcome") DESC | Count of landing outcomes between the date 2010-06-04 and 2017-03-20. |

- GitHub URL of the completed EDA notebook:
  - https://github.com/dmarks84/IBM-DS-Capstone/blob/main/4%20EDA%20%26%20SQL.ipynb

# EDA with Data Visualization

- Visualizations focused on success of missions based on pairs of parameters.
- Example given: Success of Missions for Launch Site vs. Payload mass:




- GitHub URL of the completed EDA notebook:
  - https://github.com/dmarks84/IBM-DS-Capstone/blob/main/5%20EDA%20%26%20Data%20Visualization.ipynb

# Build an Interactive Map with Folium

- Objects created and their purpose:
  - Circles – Identify locations of launch sites
  - Markers – Individual launches and nearby landmarks
  - Clusters – Group launches together by mission success
  - Lines – Visualize distances between markers
- GitHub URL of the completed map notebook:
  - https://github.com/dmarks84/IBM-DS-Capstone/blob/main/5%20EDA%20%26%20Data%20Visualization.ipynb

# Build a Dashboard with Plotly Dash

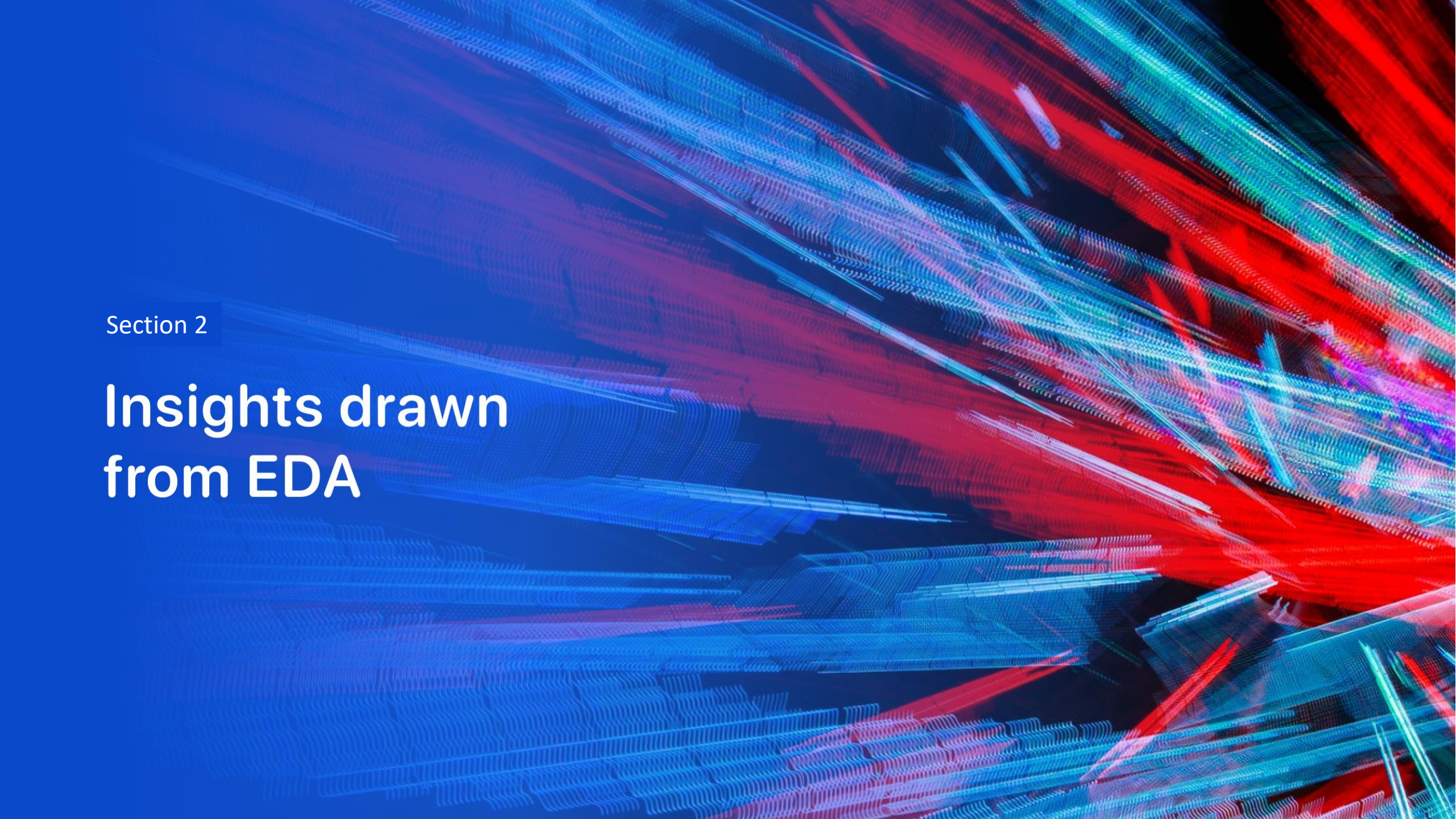
- Graphs/interactions and their purpose:
  - Dropdown – Allow selection of all launches from all launch sites, or filter by launch sites
  - Range Selector – Allow filter by payload range
  - Pie Chart – Show percentages of successful missions
  - Scatter Plot – Show mission success as a function of payload and launch site
- GitHub URL of the completed dashboard python (Plotly Dash) code:
  - https://github.com/dmarks84/IBM-DS-Capstone/blob/main/5%20EDA%20%26%20Data%20Visualization.ipynb

# Predictive Analysis (Classification)

- Data scaled prior to model creation
- Various models were fitted and trained on the data, splitting among training and testing sets, and performing a GridSearch on relevant parameters:
  - Logistic Regression, Support Vector Machine, Decision Tree, and K-Nearest Neighbor
- Models were scored and confusion matrix generated
  - Model selected with the highest score on testing data
- GitHub URL of the completed machine learning model evaluation notebook:
  - https://github.com/dmarks84/IBM-DS-Capstone/blob/main/8%20Machine%20Learning.ipynb

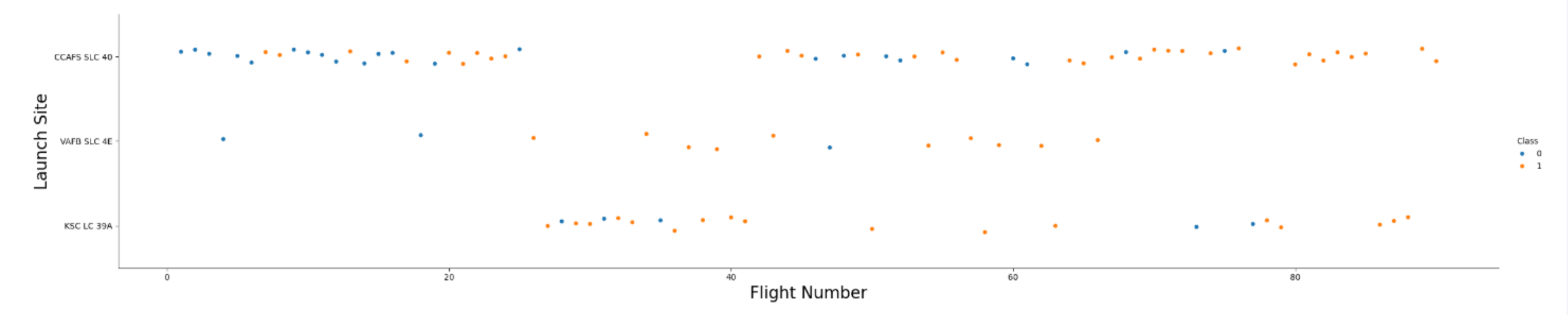
# Results

- EDA showed that:
  - Success rate is correlated with the launch site and payload mass
  - The likelihood of mission success has also increased with time
  - See following slides for this and interactive dashboard demo
- Predictive analysis results
  - Results showed that all models performed equally well.  Logistic Regression Model selected.
  - The models predict with fairly high certainty, based on the parameters provided, the success of any given mission based on the payload mass, type of orbit, and other input features
  - The cost per successful mission is the quotient of the total cost (for all missions) and the number of successful missions.


Section 2
Insights drawn from EDA

# Flight Number vs. Launch Site

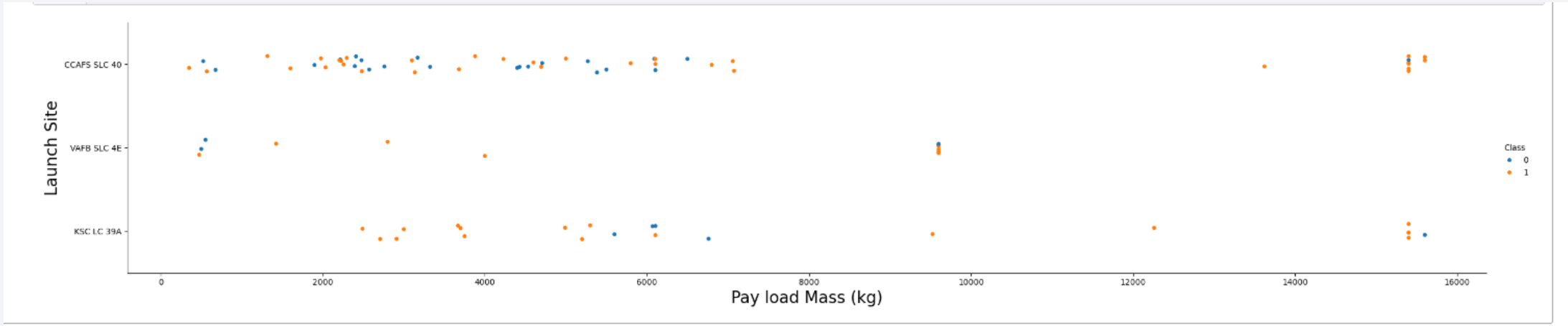
- Flight Number vs. Launch Site




For all launch sites, the mission success tends to increases with the flight number

# Payload vs. Launch Site

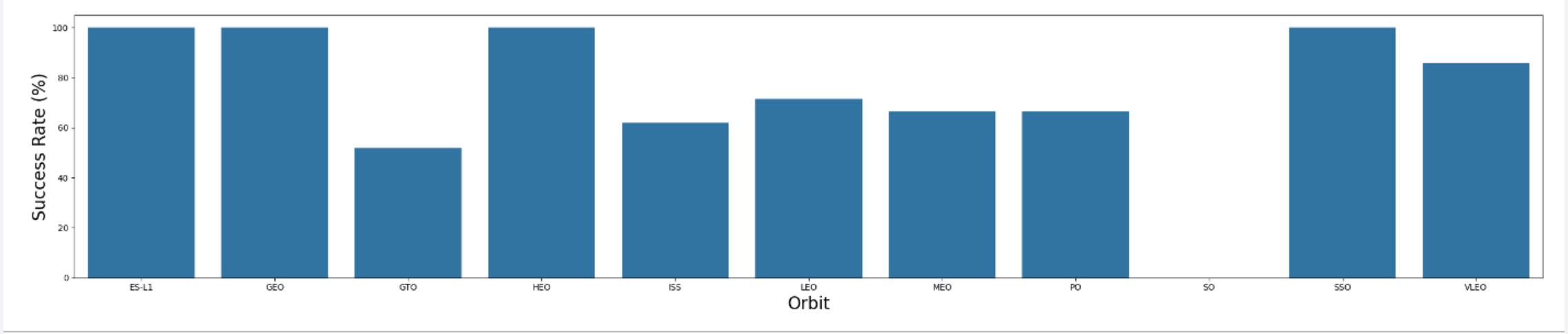
- Payload vs. Launch Site




Mission success tends to increase with payload, but the VAFB site has not done any large-payload launches

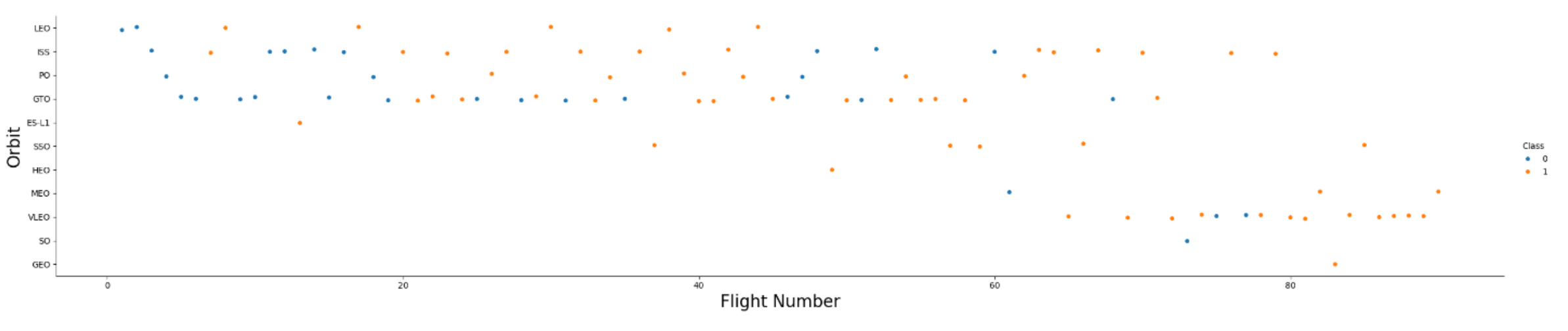
# Success Rate vs. Orbit Type

- Success for orbit type



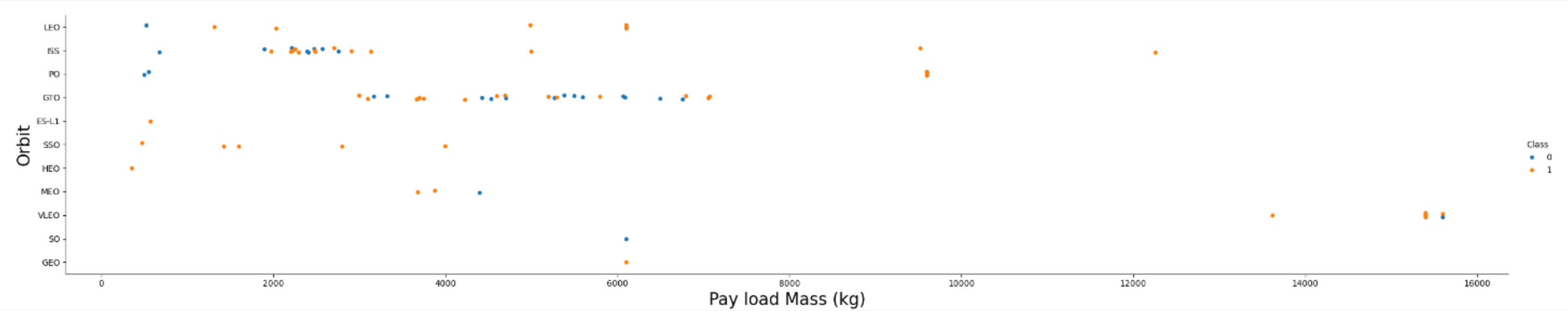
# Flight Number vs. Orbit Type

- Flight number vs. Orbit type



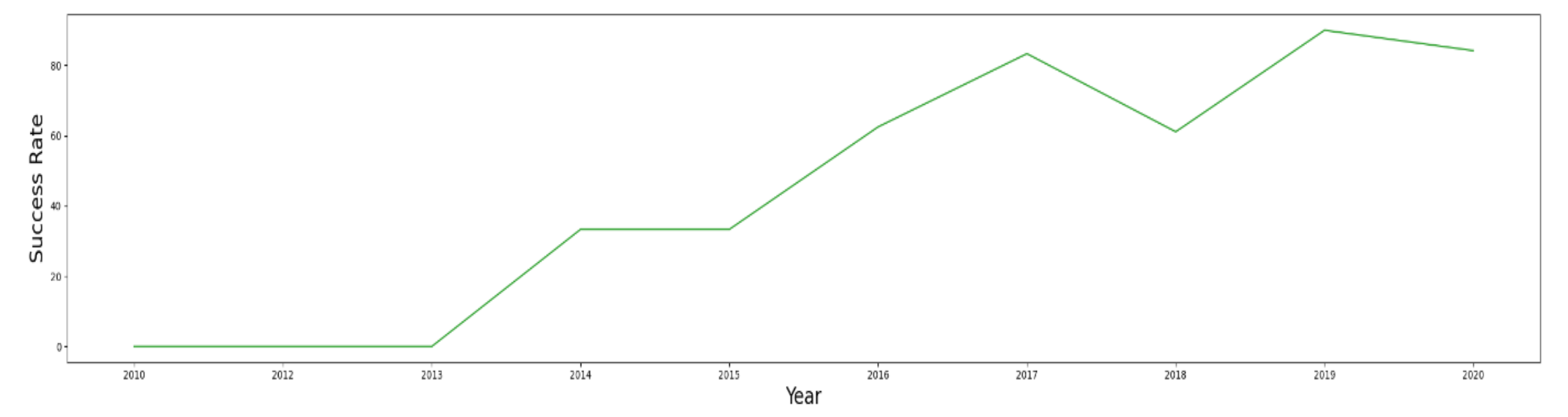
# Payload vs. Orbit Type

- Payload vs. Orbit type



# Launch Success Yearly Trend

- Yearly Ave Success Rate



# All Launch Site Names

```
%sql SELECT Distinct "Launch_Site" FROM SPACEXTABLE
 * 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 SPACEXTABLE 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 |
|---|---|---|---|---|---|---|---|---|---|
| 2010-06-04 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 2010-12-08 | 15:43:00 | F9 v1.0 B0004 | CCAFS LC-40 | Dragon demo flight C1, two CubeSats, barrel of Brouere cheese | 0 | LEO (ISS) | NASA (COTS) NRO | Success | Failure (parachute) |
| 2012-05-22 | 7:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 2012-10-08 | 0:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 2013-03-01 | 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 SPACEXTABLE 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 SPACEXTABLE 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") AS "First Successful Landing" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE "Success%"
```

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

| First Successful Landing |
| --- |
| 2015-12-22 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE ("PAYLOAD_MASS__KG_" > 4000) AND ("PAYLOAD_MASS__KG_" < 6000) AND ("Landing_Outcome" LIKE "%d
```

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

Done.
```

| Booster_Version |
|---|
| F9 FT B1020 |
| 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(*) AS "Count" FROM SPACEXTABLE GROUP BY "Mission_Outcome"
```

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

| Mission_Outcome | Count |
|---|---|
| Failure (in flight) | 1 |
| Success | 98 |
| Success | 1 |
| Success (payload status unclear) | 1 |

# Boosters Carried Maximum Payload

```
%sql SELECT "Booster_Version", "PAYLOAD_MASS__KG_" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)
```

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

| Booster_Version | PAYLOAD_MASS__KG_ |
|---|---|
| F9 B5 B1048.4 | 15600 |
| F9 B5 B1049.4 | 15600 |
| F9 B5 B1051.3 | 15600 |
| F9 B5 B1056.4 | 15600 |
| F9 B5 B1048.5 | 15600 |
| F9 B5 B1051.4 | 15600 |
| F9 B5 B1049.5 | 15600 |
| F9 B5 B1060.2 | 15600 |
| F9 B5 B1058.3 | 15600 |
| F9 B5 B1051.6 | 15600 |
| F9 B5 B1060.3 | 15600 |
| F9 B5 B1049.7 | 15600 |

# 2015 Launch Records

```
%sql SELECT substr("Date",6,2) AS "Month", substr(Date,0,5) AS "Year", "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE substr(
```

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

| Month | Year | Day | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|---|---|
| 01 | 2015 | 10 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | 2015 | 14 | 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("Landing_Outcome") AS "Count" FROM SPACEXTABLE WHERE ((substr("Date",0,5)*365 + substr("Date",6,2)*12 + substr("Date
```

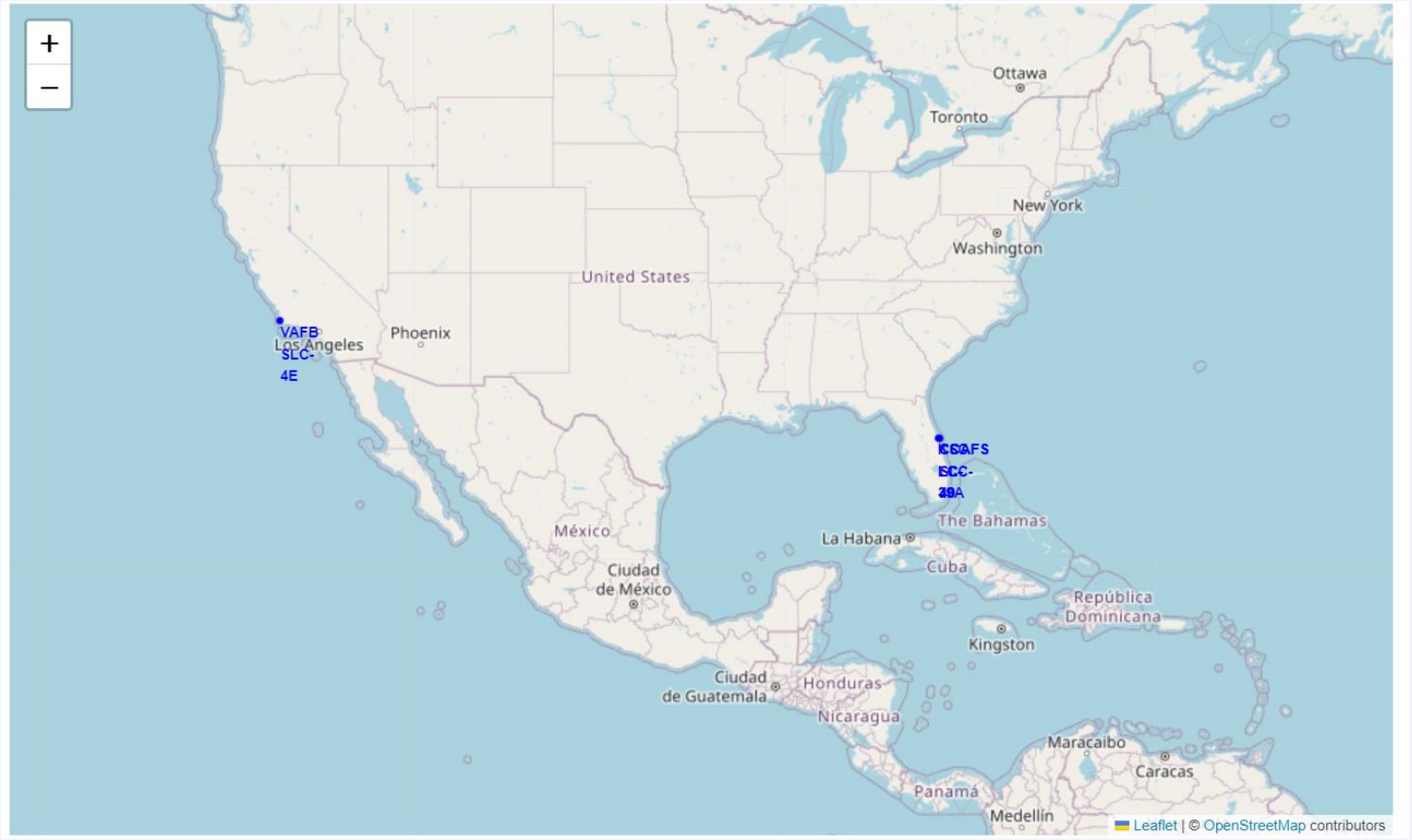
```
 * sqlite:///my_data1.db

Done.
```

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

Section 3

# Launch Sites Proximities Analysis

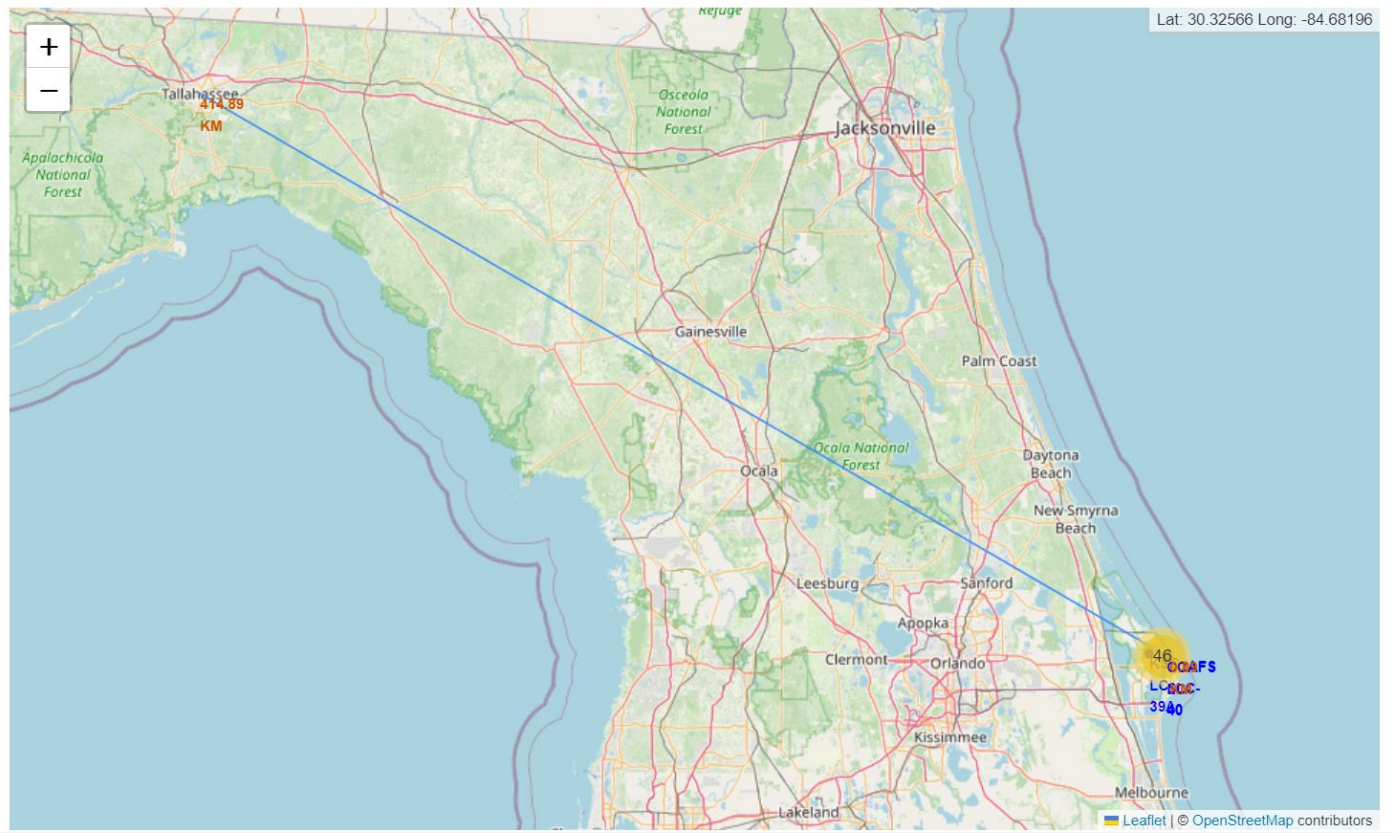
# All Launch Sites



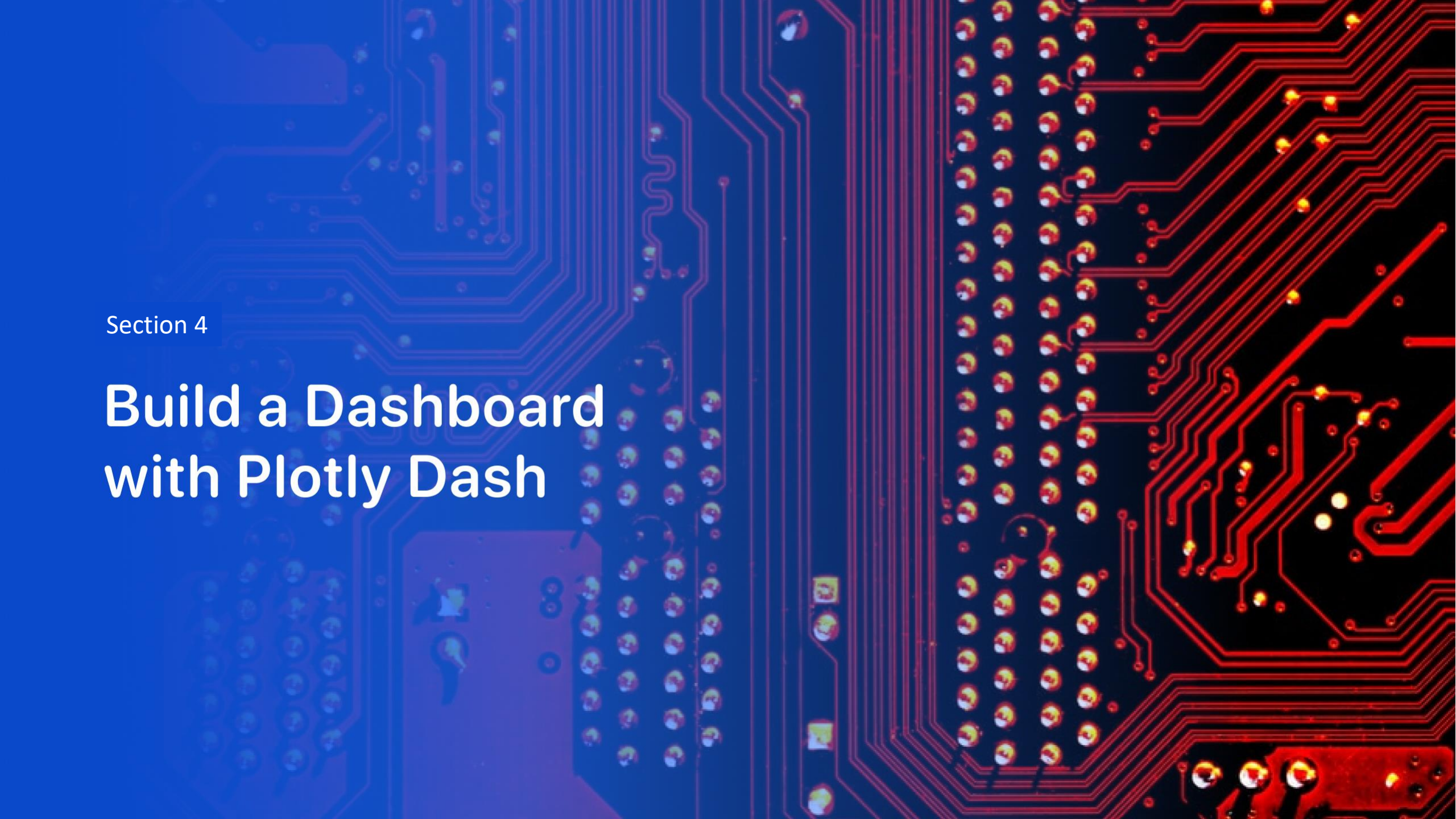
# Color-Labeled Launch Outcome Map


This is for the CCAFS LC-40 Launch Site

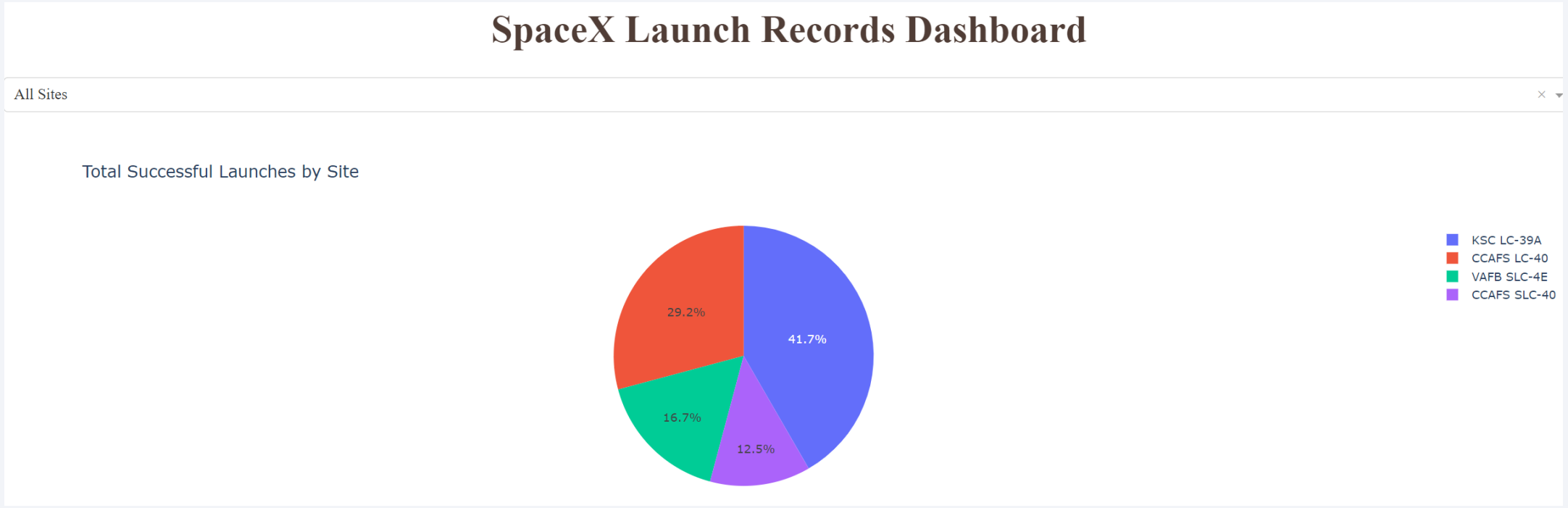
# Florida Launch Site Proximate to State Capitol




- The exact distance displayed is of the CCAFS LC-40 Launch Site

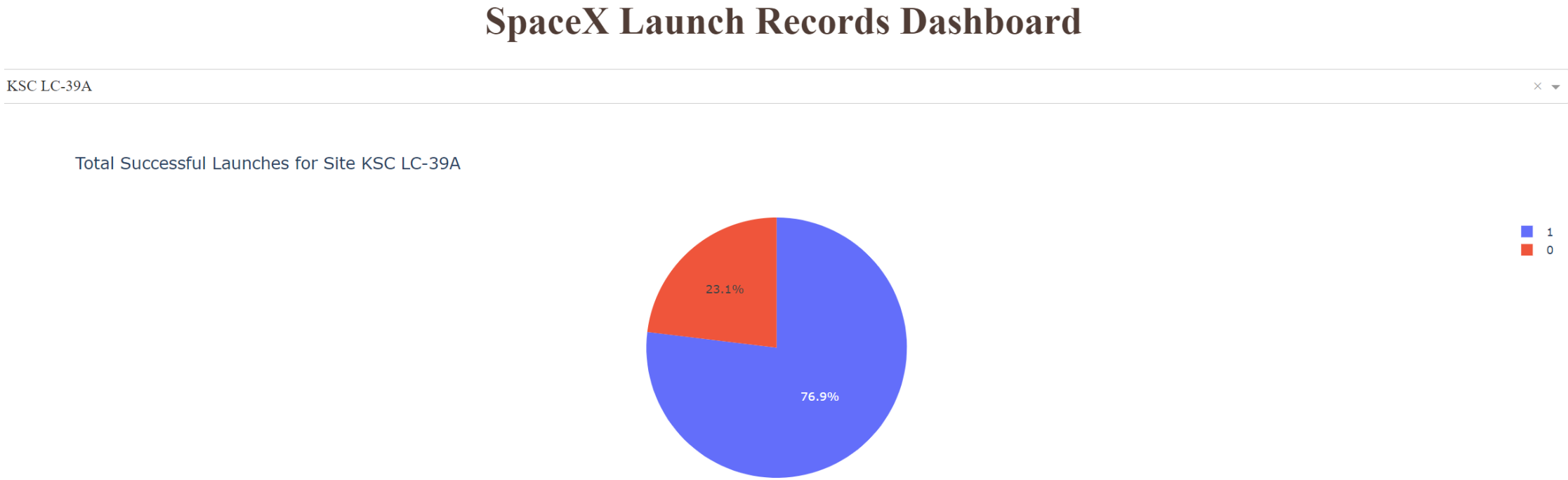

Section 4
Build a Dashboard with Plotly Dash

# SpaceX Launch Success by Site



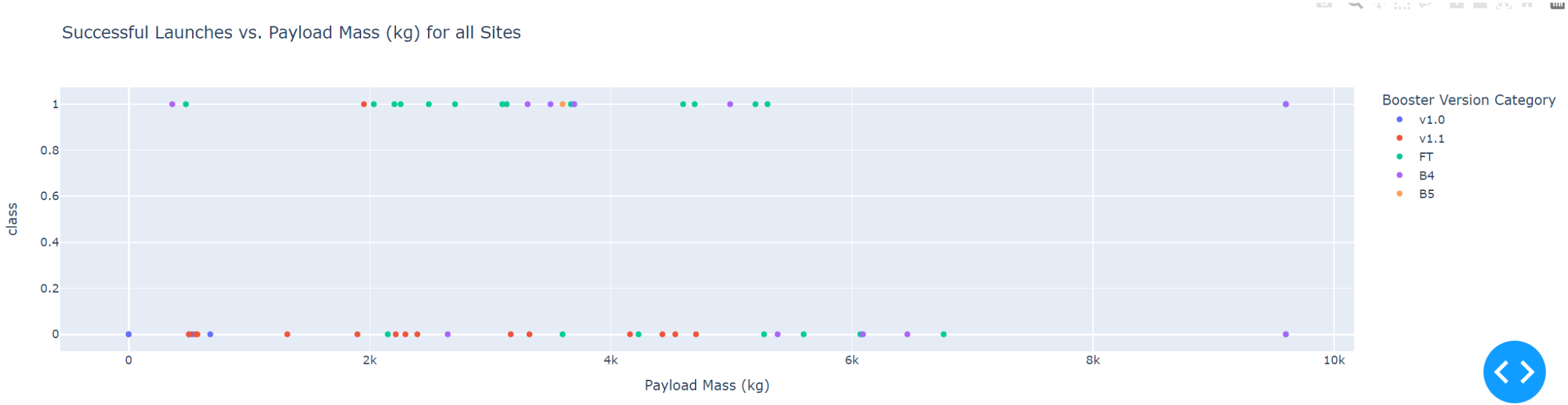

KSC LC-39A is the most consistent site, followed by (in order) CCAFS LC-40, VAFB SLC-4E, and CCAGS SLC-40

# SpaceX Launch Success at KSC LC-39A




KSC LC-39A has a success rate of 76.9%

# SpaceX Launch Success at KSC LC-39A



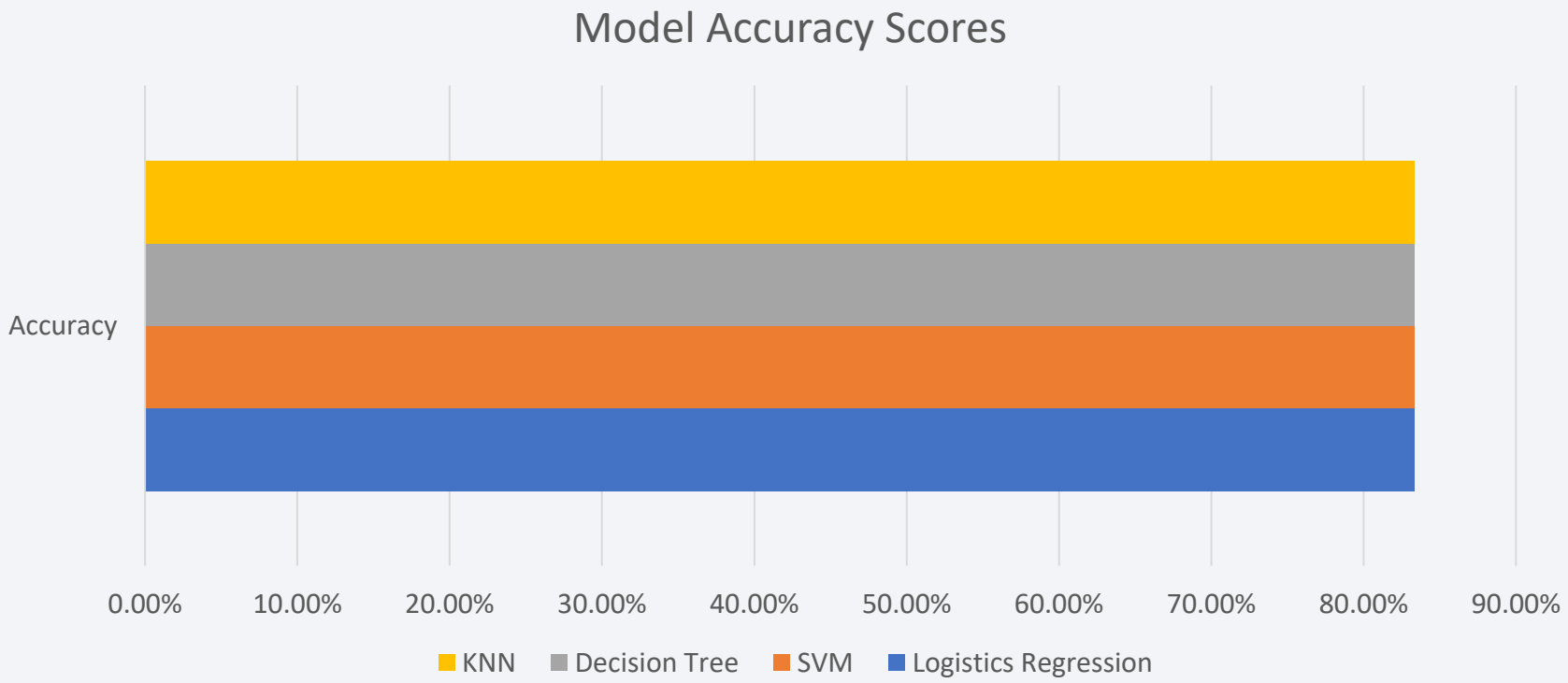

The F9 FT has a high success rate and numerous launches

V1.0 and v1.1 have low success rates, and B4 and B5 have moderate or high success rates but relatively few luanches

Section 5

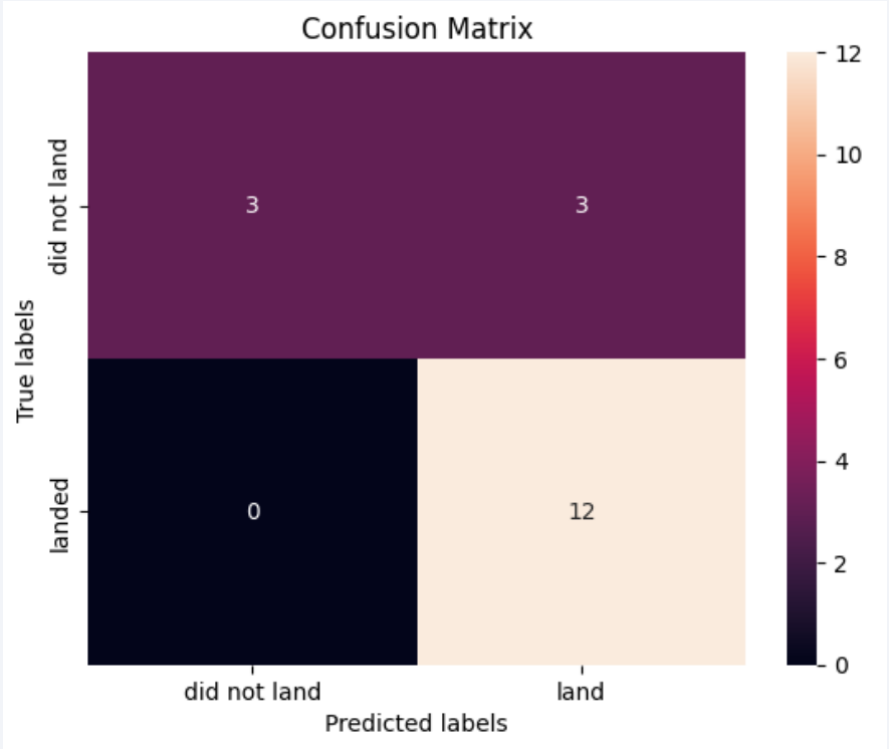
# Predictive Analysis (Classification)

# Classification Accuracy




- All models displayed the same accuracy score

# Confusion Matrix




- The model had no false positives but a few false negatives. It was very good at predicting landings but less good at predicting if it did not land

# Conclusions

- The EDA and dashboards can be used to visualize how different relevant features correspond to one another and to the success of the missions they represent.
- The models predict with fairly high certainty, based on the parameters provided, the success of any given mission based on the payload mass, type of orbit, and other input features
- The cost per successful mission is the quotient of the total cost (for all missions) and the number of successful missions.

# Thank you!