IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- Summary of methodologies
  - Web scraped Falcon 9 launches to obtain data on successful and unsuccessful launches
  - Performed data wrangling including one hot encoding for successful vs unsuccesful launches
- Summary of all results
  - SVM model is most accurate for ML
- For all code please refer to my GitHub: https://github.com/Joseph-Westover/IBM-Data-Science-Projects.git

Introduction

- Project background and context:
  - Gather info on Space X and build ML algorithms to determine if they will reuse the first stage. This knowledge will allow Space Y to remain competitive
- Problems you want to find answers
  - What machine learning model is best to use?
  - Can we use scraped data to learn more about Space X's launches?

Elon Tusk - Space Y CEO

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Use BeautifulSoup to scrape Wikipedia page on Falcon 9 launches
  - Iterate through table header <th> elements to extract column names using for loop
- Perform data wrangling
  - Import data from csv
  - Identify how many missing values for each column
  - Classify each launch as successful (1) or failure (0) with one hot encoding – these above steps are data wrangling
- 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

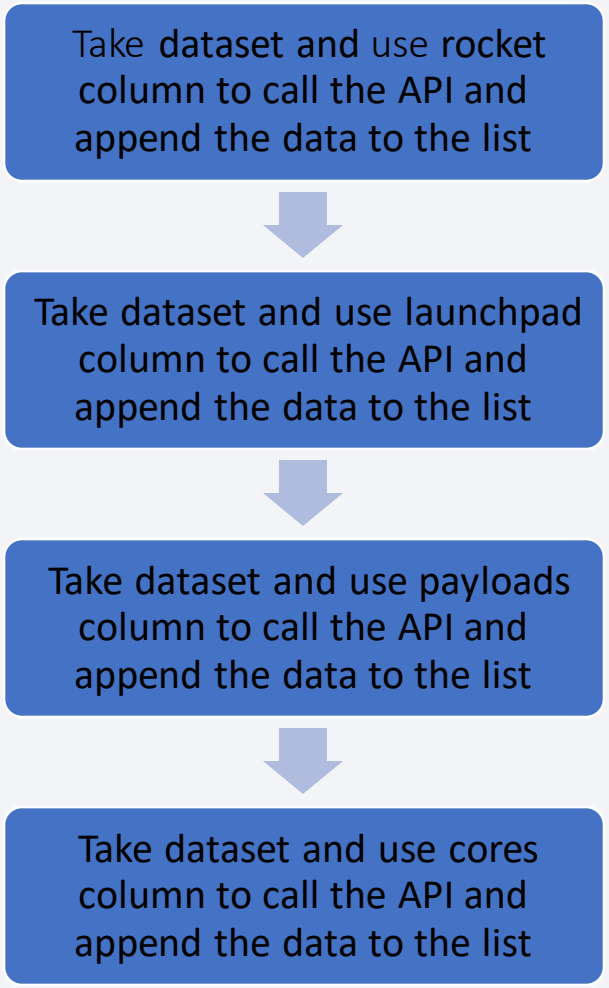
Data Collection

- Describe how data sets were collected.
  - Use BeautifulSoup to scrape Wikipedia page on Falcon 9 launches
  - Iterate through table header <th> elements to extract column names using for loop
  - Append data for each column by using for loop to append data from scraped data rows
  - Convert data into dataframe
- You need to present your data collection process use key phrases and flowcharts
  - Please see slide 9

Data Collection – SpaceX API

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

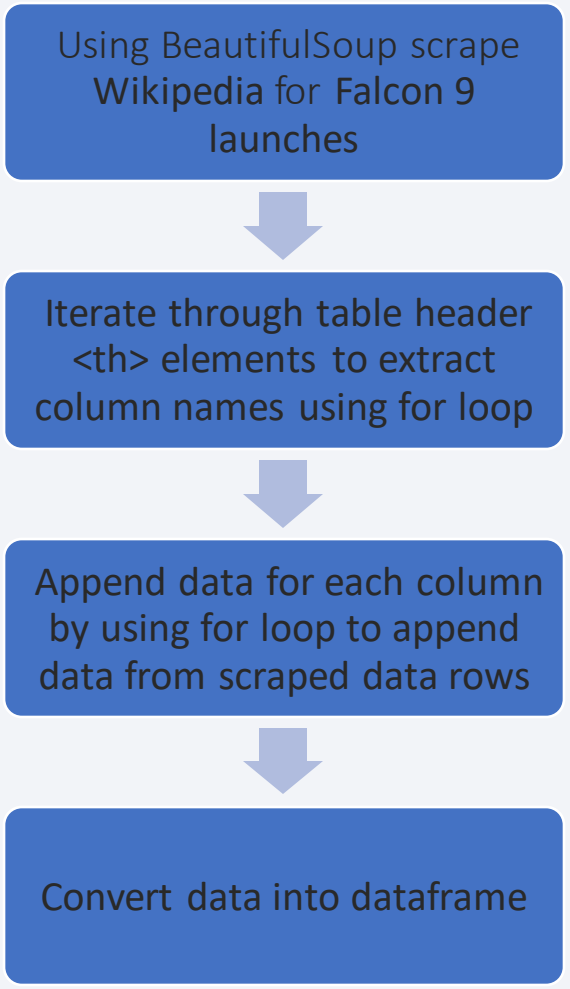
- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose
  - https://github.com/Joseph-Westover/IBM-Data-Science-Projects/blob/18ace41e285a730d4226b69ed9093258addb8877/IBM-DS-Capstone-Data-Collection-API.ipynb



Data Collection - Scraping

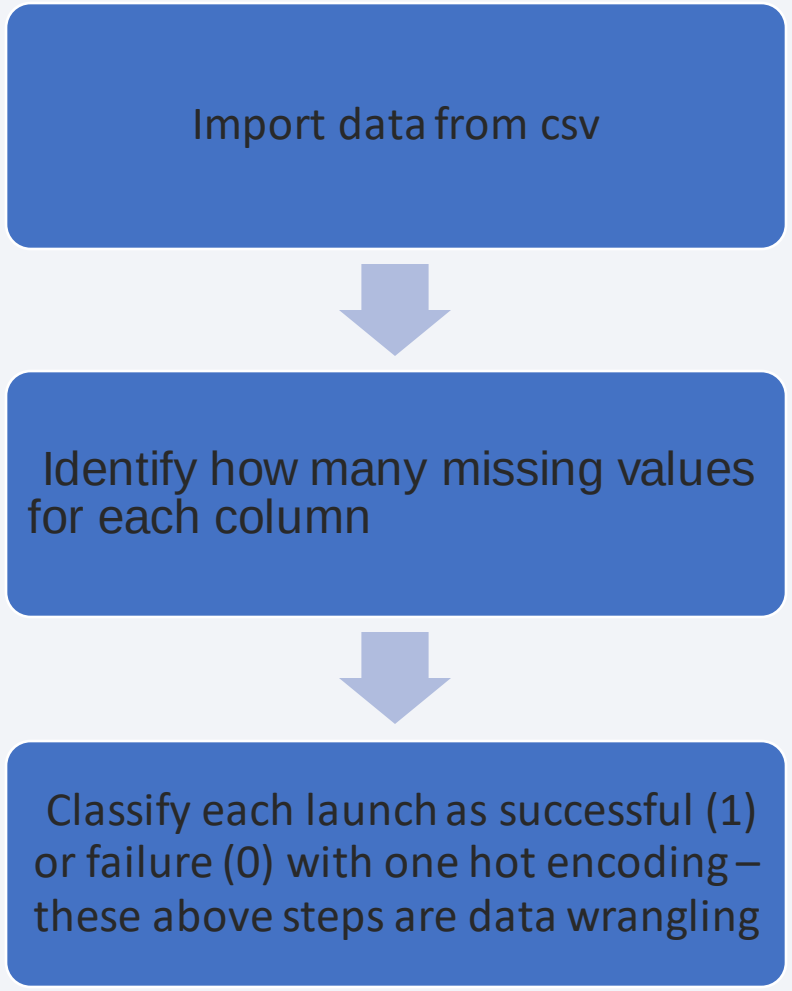
- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose
  - https://github.com/Joseph-Westover/IBM-Data-Science-Projects/blob/18ace41e285a730d4226b69ed9093258addb8877/IBM-DS-Capstone-webscraping.jupyterlite.ipynb



Data Wrangling

- Describe how data were processed
  - Please see flowchart
- You need to present your data wrangling process using key phrases and flowcharts
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose
  - https://github.com/Joseph-Westover/IBM-Data-Science-Projects/blob/18ace41e285a730d4226b69ed9093258addb8877/IBM-DS-Capstone-Data-Wrangling.jupyterlite.ipynb



EDA with Data Visualization

- Summary of charts plotted and why I used those charts
- Category plots showing class to see how these variables influence mission success:
  - Payload Mass vs Flight Number – as Number ↑, chance of success ↑; as Mass ↑, chance of success ↓
  - Launch Site vs Flight Number – as Number ↑, chance of success ↑ across all sites
  - Outcome vs Orbit – see which orbits have best chance of mission success
  - Payload Mass vs Orbit – see how increased mass influences success of each orbit type
  - Avg Success Rate vs Year – as Year ↑, average success ↑

- GitHub URL
  - https://github.com/Joseph-Westover/IBM-Data-Science-Projects/blob/18ace41e285a730d4226b69ed9093258addb8877/IBM-DS-Capstone-Plots.jupyterlite.ipynb

EDA with SQL

- Summary of SQL queries performed:
- Get unique launch sites
- Show launch sites beginning with 'CCA'
- Find total payload mass carried by boosters launched by NASA
- Display average payload mass carried by booster version F9 v1.1
- Find date when the first succesful landing outcome in ground pad was achieved
- Find names of the boosters with success in drone ship and payload mass between 4000 - 6000
- List total number of successful and failure mission outcomes
- List names of the booster_versions which have carried the maximum payload mass
- List records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
- Rank 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/Joseph-Westover/IBM-Data-Science-Projects/blob/18ace41e285a730d4226b69ed9093258addb8877/IBM-DS-Capstone-SQL-Analysis.jupyterlite.ipynb

Build an Interactive Map with Folium

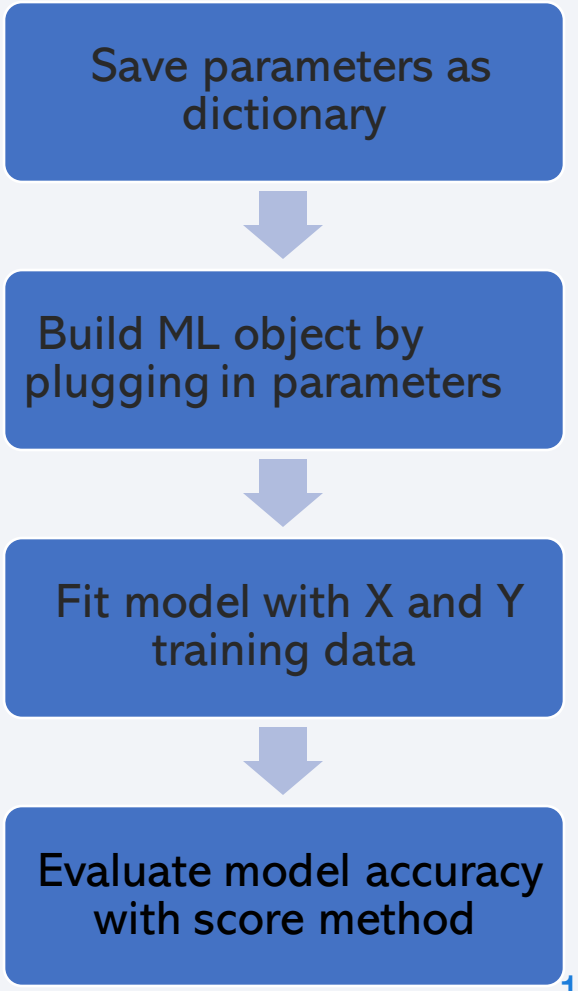
- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects
  - Marked each launch site and color coded succesful vs unsuccesful
    - This was used to visualize which site had the most successes to see if location influenced success
  - Showed how far one launch site was from a city, highway, railway, and coast
    - This is to indicate how close the launch site is to civilians
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose
  - https://github.com/Joseph-Westover/IBM-Data-Science-Projects/blob/18ace41e285a730d4226b69ed9093258addb8877/IBM-DS-Capstone-Folium-Map-Plots.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
  - Dashboard allows you to quickly view data on launches
- Explain why you added those plots and interactions
  - I added these specific plots so that folks could quickly evaluate launch data
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose
  - https://github.com/Joseph-Westover/IBM-Data-Science-Projects/blob/4faffd3ebb0477ccf77299a36bf5e490b1a3c28a/IBM-DS-Capstone-Plotly-Dash-App.py

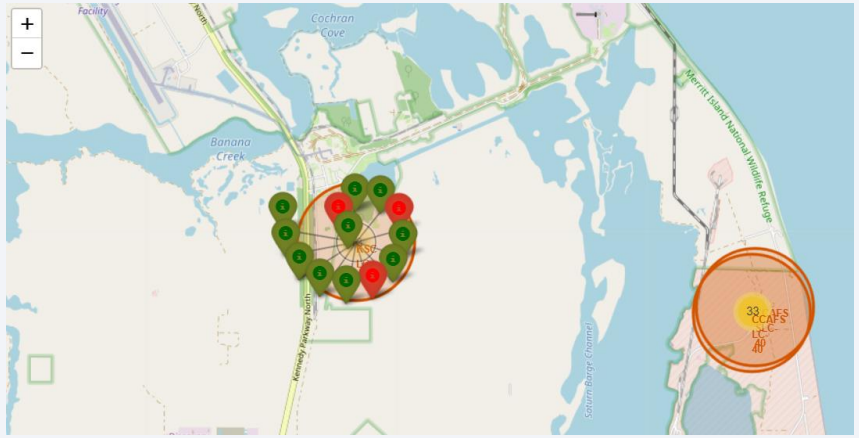
Predictive Analysis (Classification)

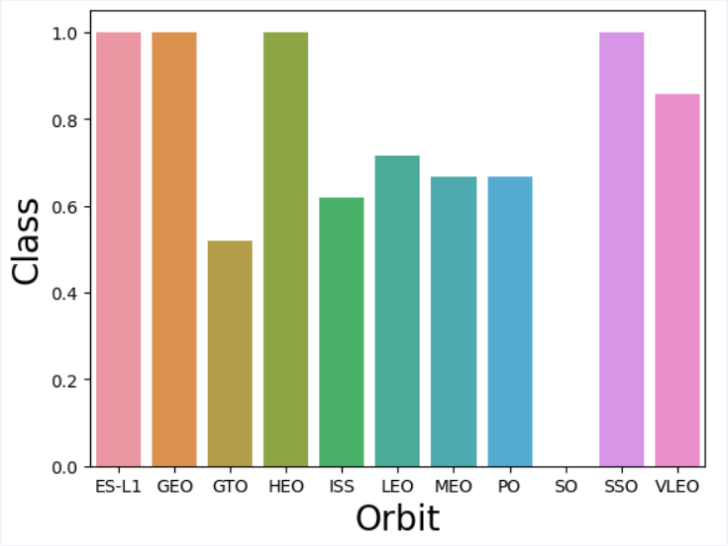
- Summarize how you built, evaluated, improved, and found the best performing classification model
  - Used GridSearch to evaluate the optimal parameters for each model
  - Tested several models including:
    - Logistic Regression
    - Decision Tree
    - SVM
    - KNN
  - Please refer to flowchart to see how I built and evaluated models
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose
  - https://github.com/Joseph-Westover/IBM-Data-Science-Projects/blob/4faffd3ebb0477ccf77299a36bf5e490b1a3c28a/IBM-DS-Capstone-Machine-Learning-Analysis.jupyterlite.ipynb



Results

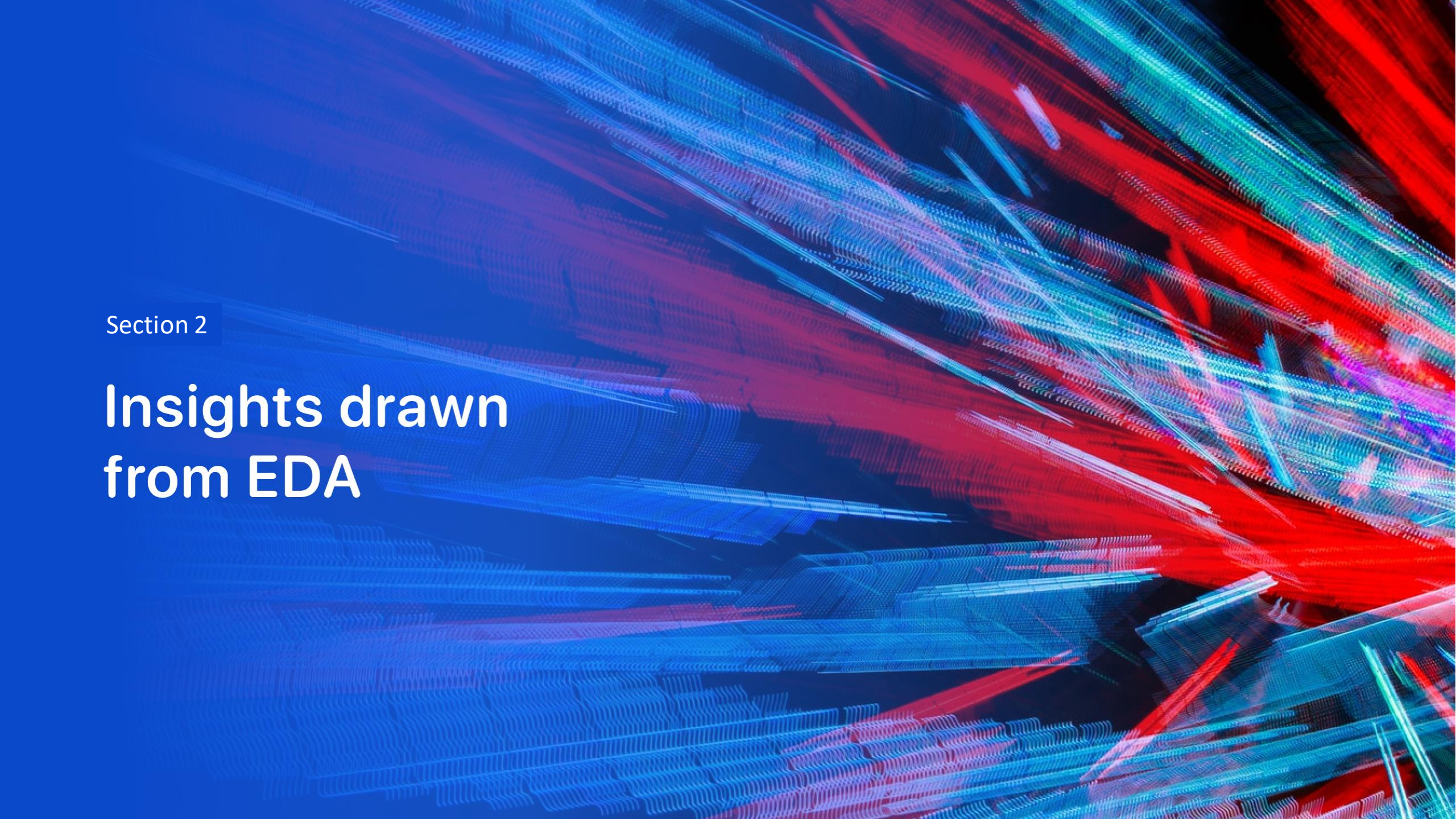
- Exploratory data analysis results
  - Was able to succesfully scrape data on launch sites and evaluate it
- Interactive analytics demo in screenshots
  - Was able to plot launch sites on a map to see which had the most successful (see map)
  - Also able to evaluate other metrics influence on launch success (see barchart)
- Predictive analysis results
  - SVM was the most effective ML model





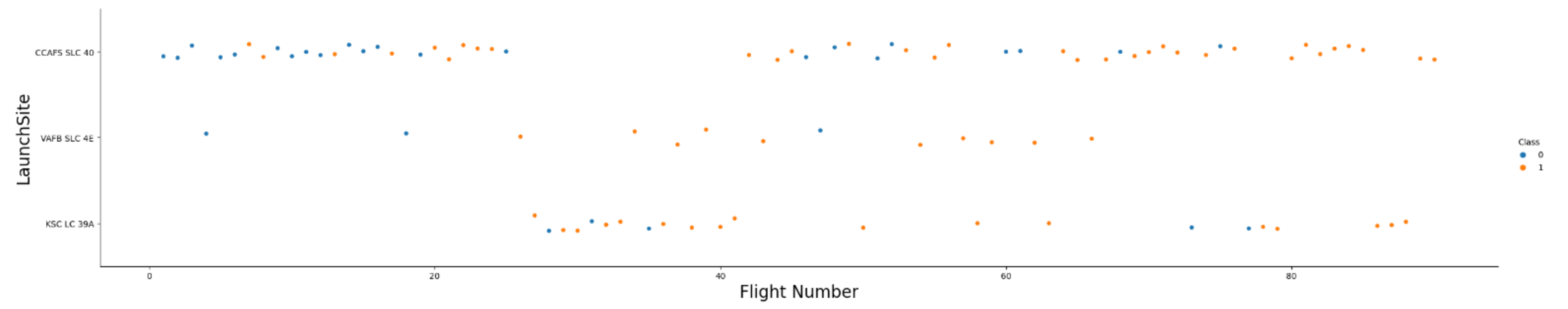
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

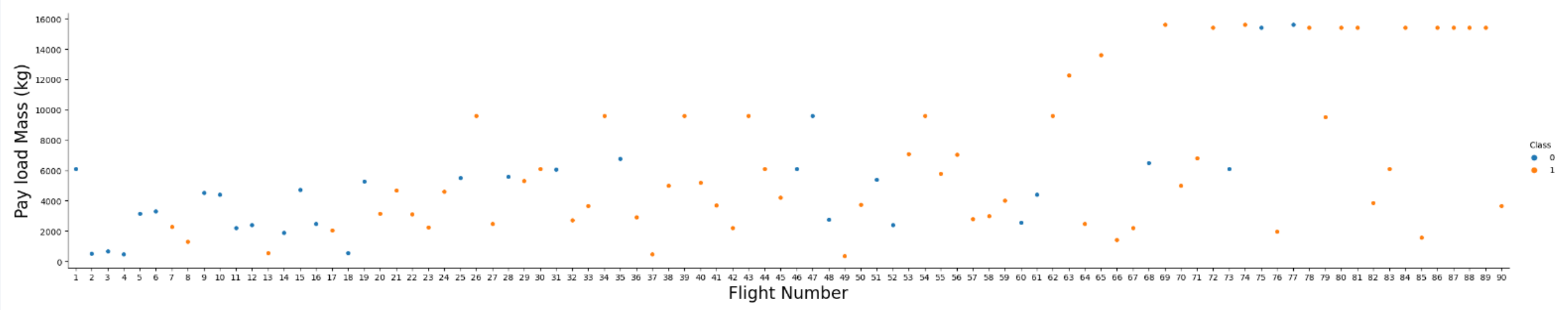
- Scatter plot of Flight Number vs. Launch Site



- Explanations
  - As Number ↑ chance of success ↑ across all sites

Payload vs. Launch Site

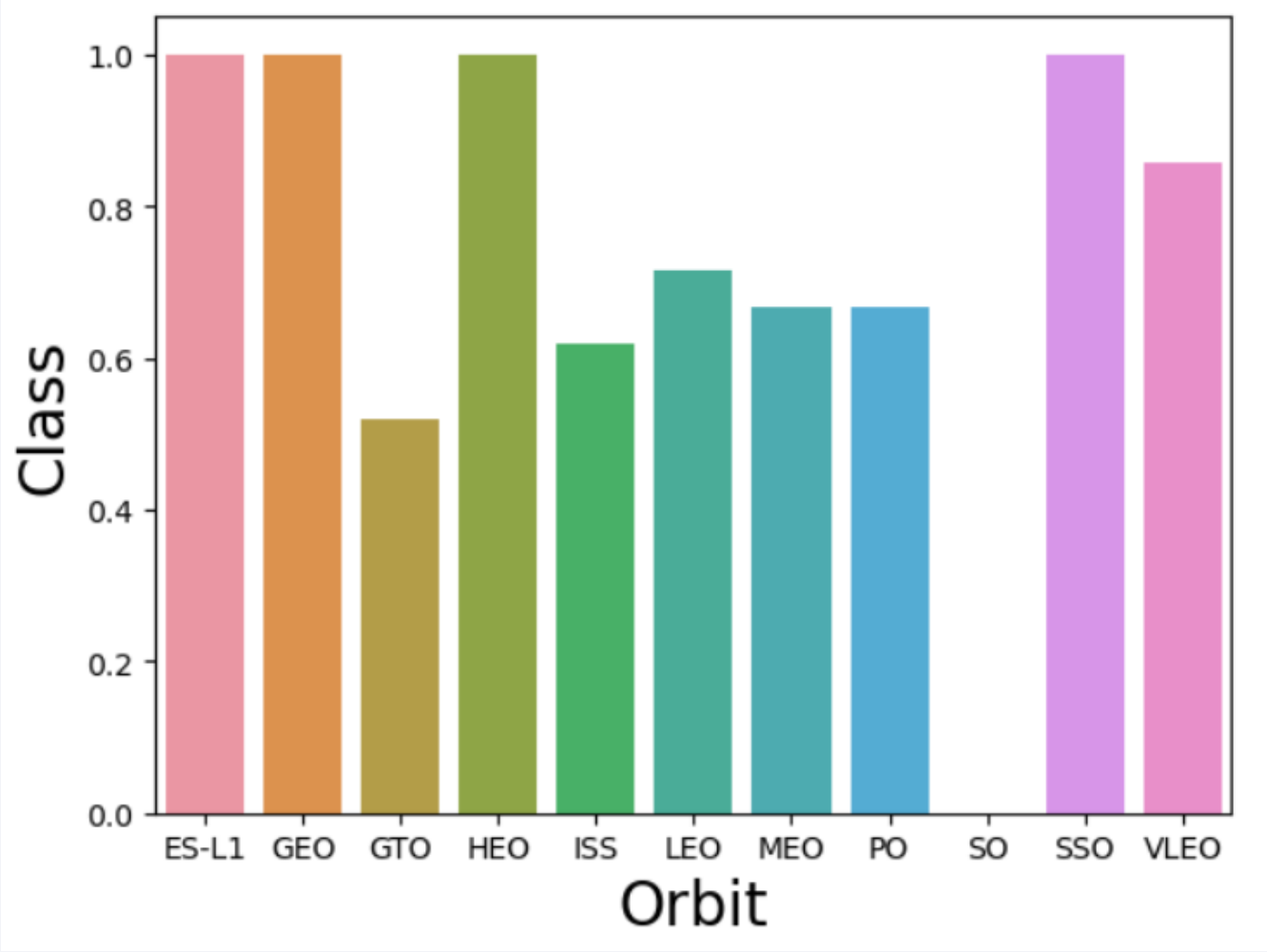
- Scatter plot of Payload vs. Launch Site



- Explanations:
  - As Number ↑, chance of success ↑
  - As Mass ↑, chance of success ↓

Success Rate vs. Orbit Type

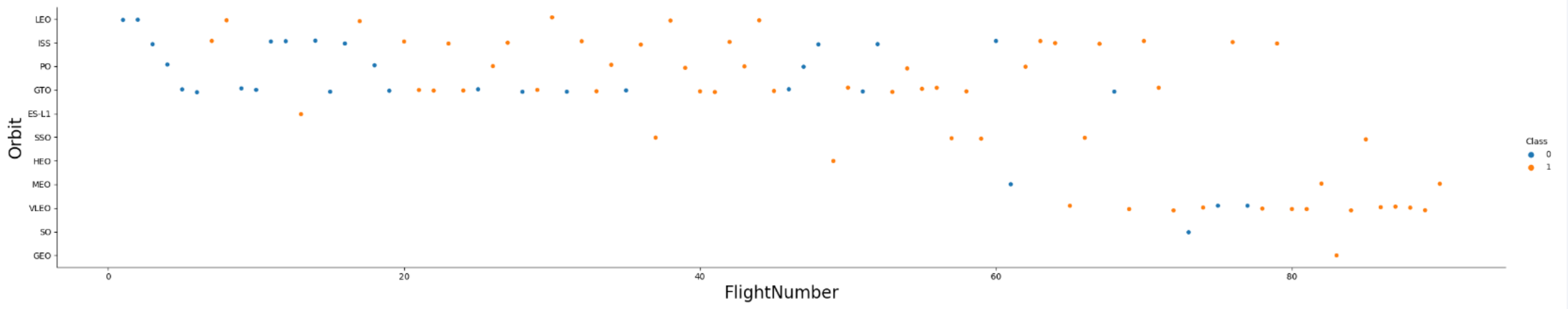
- Bar chart for the success rate of each orbit type



- Explanations:
- ES-L1, GEO, HEO, and SSO have the highest average success rate
- GTO has the lowest avg success rate

Flight Number vs. Orbit Type

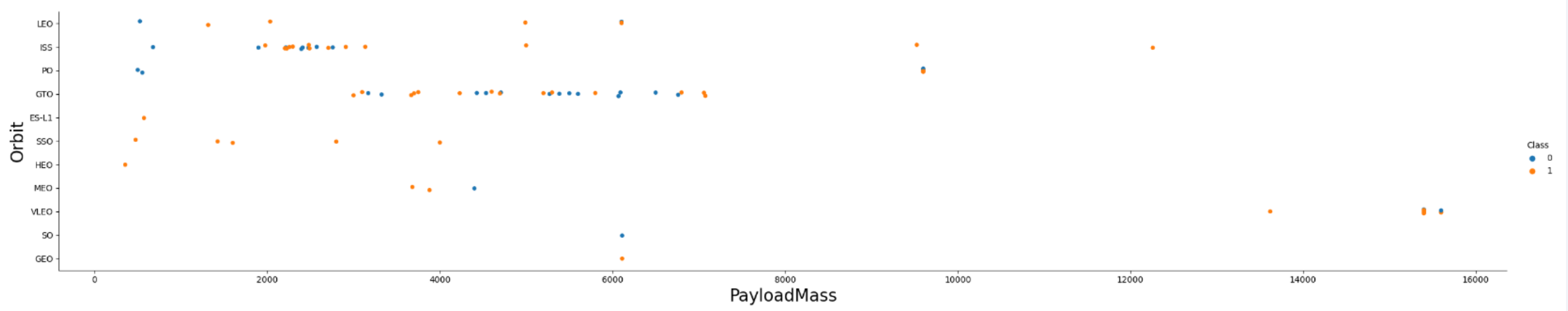
- Scatter point of Flight number vs. Orbit type



- Explanations
  - As Number ↑, chance of success ↑ for all Orbits

Payload vs. Orbit Type

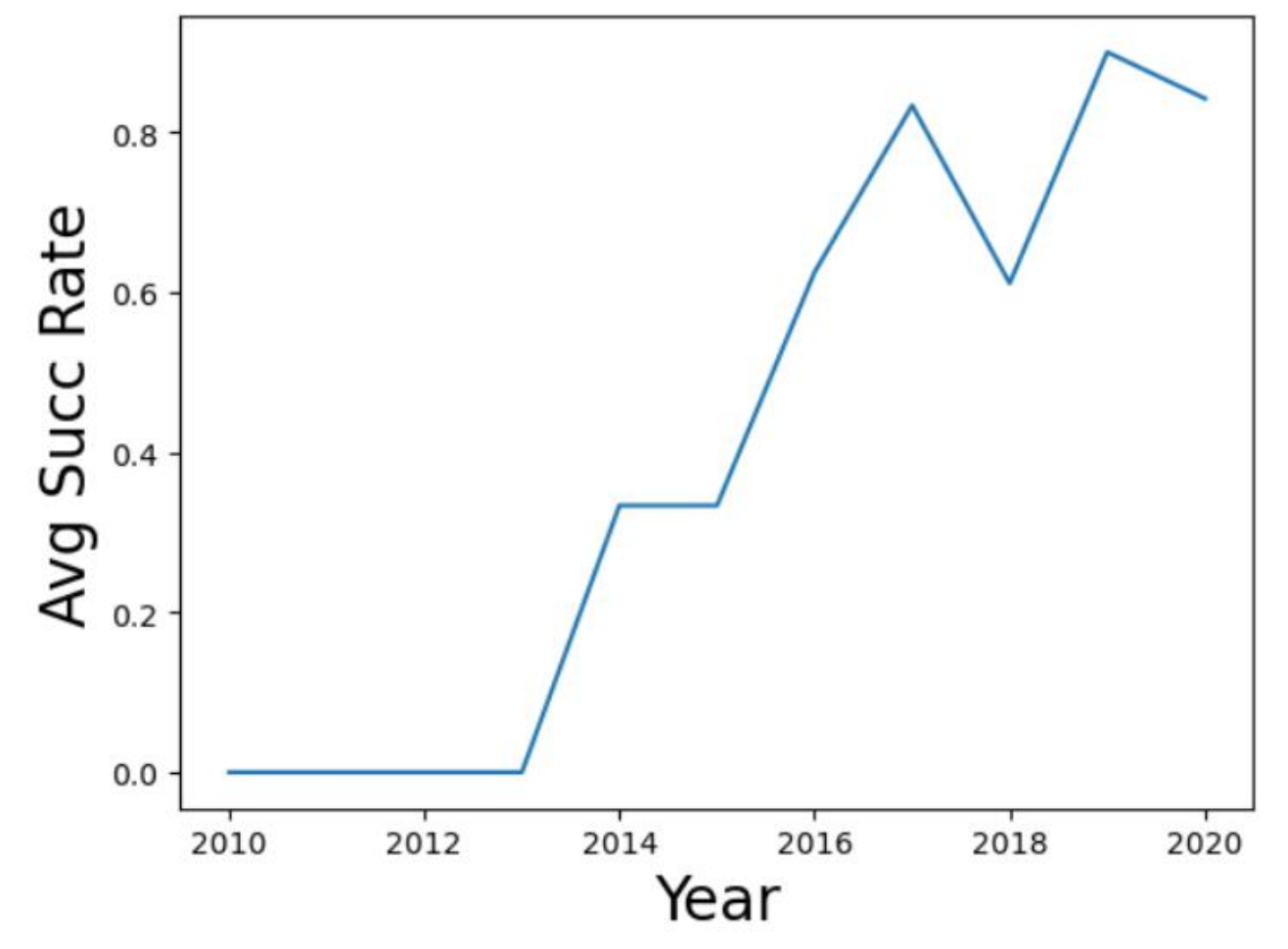
- Scatter point of payload vs. orbit type



- Explanations:
  - As Mass ↑, chance of success ↑ for all Orbits except for GTO (chance of success ↓)

Launch Success Yearly Trend

- Line chart of yearly average success rate



- Explanations
- As Year ↑, average success ↑ until 2019

All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here
  - Need to use DISTINCT to get unique launch sites

Task 1

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

```
%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
None

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here
  - Use WHERE statement with LIKE, % (Wildcard) and LIMIT

Task 2 ¶

Display 5 records where launch sites begin with the string '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
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here
  - SUM and LIKE allows us to sum every payload for NASA

Task 3

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

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") AS NASA_MASSA FROM SPACEXTBL WHERE "Customer" LIKE '%NASA (CRS)%';
```

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

NASA_MASSA
48213.0

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here
  - Use AVG and LIKE to get average for specified booster

Task 4

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

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") AS "F9_v1.1_Mass" FROM SPACEXTBL WHERE "Booster_Version" LIKE "F9 v1.1%";
```

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

F9_v1.1_Mass
2534.6666666666665

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here
  - LIKE shows only successes for ground pad and ORDER BY DESC shows first date

Task 5 ¶

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
%sql SELECT "Date", "Landing_Outcome" FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Success (ground pad)' ORDER BY "Date" DESC LIMIT 1
```

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

Date	Landing_Outcome
22/12/2015	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Present your query result with a short explanation here
  - Need to do LIKE and AND statements to filter to drone ship success and filter mass between 4000 and 6000

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql SELECT "Booster_Version", "Landing_Outcome", "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "Landing_Outcome" LIKE '%Success (drone ship)%'
AND ("PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000)
```

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

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696.0
F9 FT B1026	Success (drone ship)	4600.0
F9 FT B1021.2	Success (drone ship)	5300.0
F9 FT B1031.2	Success (drone ship)	5200.0

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here
  - COUNT takes each instance as 1 and filter with WHERE and LIKE

Task 7

List the total number of successful and failure mission outcomes

```
%sql SELECT count("Landing_Outcome") as "Sum_Failures" FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Failure%';

%sql SELECT count("Landing_Outcome") as "Sum_Successes" FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Success%';
```

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

Sum_Successes	Sum_Failures
61	10

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here
    - Use DISTINCT to not show repeats. Sub SELECT statement to only get max mass

Task 8

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

```
%sql SELECT DISTINCT("Booster_Version"), "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" IN (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600.0
F9 B5 B1049.4	15600.0
F9 B5 B1051.3	15600.0
F9 B5 B1056.4	15600.0
F9 B5 B1048.5	15600.0
F9 B5 B1051.4	15600.0
F9 B5 B1049.5	15600.0
F9 B5 B1060.2	15600.0
F9 B5 B1058.3	15600.0
F9 B5 B1051.6	15600.0
F9 B5 B1060.3	15600.0
F9 B5 B1049.7	15600.0

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Present your query result with a short explanation here
  - Sub SELECT statement to only show failures

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%sql SELECT "Date",substr(Date, 4, 2) as "Month", "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTBL WHERE substr(Date,7,4)='2015'
AND "Landing_Outcome" IN (SELECT "Landing_Outcome" FROM SPACEXTBL WHERE ("Landing_Outcome" LIKE 'Failure%'));
```

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

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

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)) between the date 2010-06-04 and 2017-03-20, in descending order
- Present your query result with a short explanation here
  - Sub SELECT to get launches in that date range

Task 10 ¶

Rank 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.

```
# Working code
#%sql SELECT "Landing_Outcome", substr("Date", 7, 4) || '-' || substr("Date", 4, 2) || '-' || substr("Date", 1, 2) AS "Reformatted_Date" FROM SPACEXTBL WH

%sql  SELECT "Landing_Outcome", COUNT("Landing_Outcome") AS "Count" FROM SPACEXTBL WHERE "Landing_Outcome" IN (SELECT "Landing_Outcome" FROM SPACEXTBL
WHERE substr("Date", 7, 4) || '-' || substr("Date", 4, 2) || '-' || substr("Date", 1, 2) > '2010-06-04'
AND substr("Date", 7, 4) || '-' || substr("Date", 4, 2) || '-' || substr("Date", 1, 2) < '2017-03-20') GROUP BY "Landing_Outcome" ORDER BY "Count" DESC;

#%sql SELECT COUNT("Landing_Outcome") as Count_Landing_Outcome, substr("Date", 7) || '-' || substr("Date", 4, 2) || '-' || substr("Date", 1, 2) AS "Reform
```

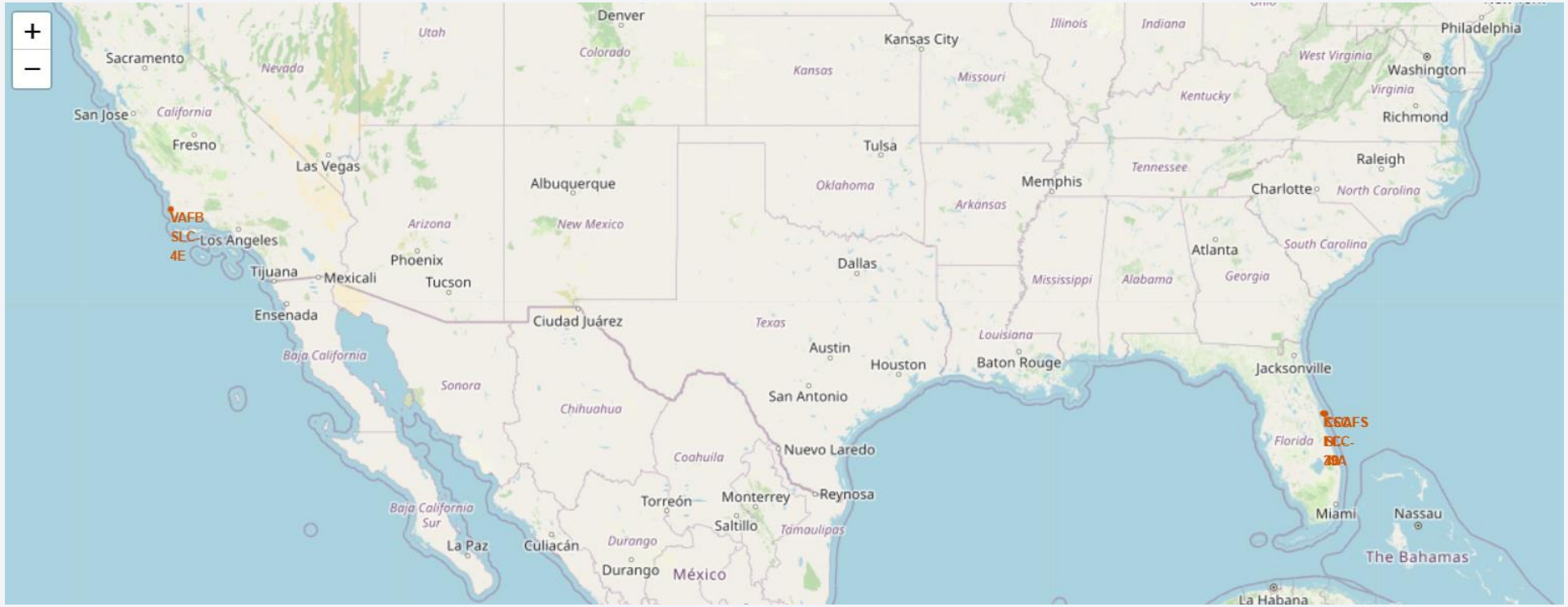
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	Count
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1


Section 3
Launch Sites
Proximities Analysis

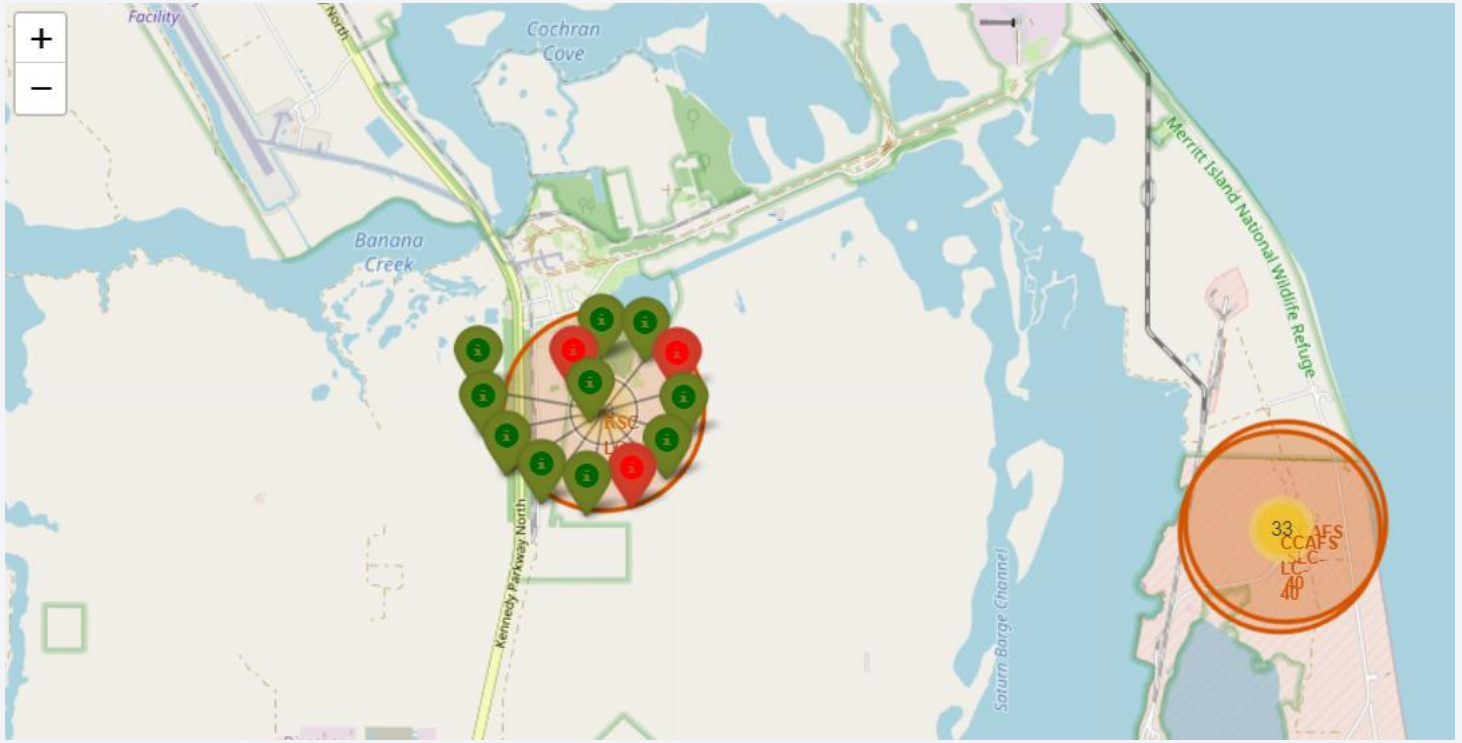
Map of Launch Site Locations

- Screenshot all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot
  - All launch sites are in CA or FL



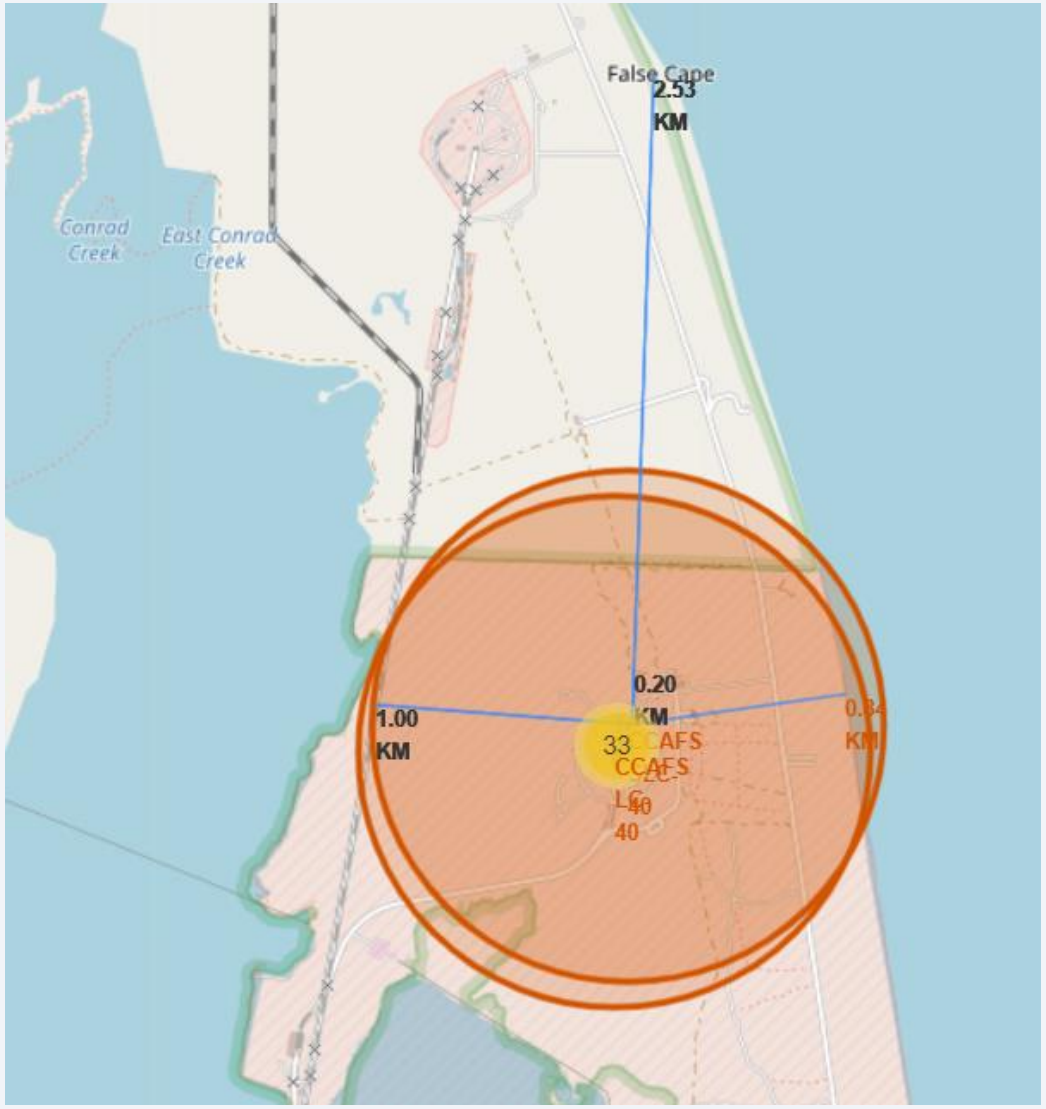
Map of Launch Outcomes

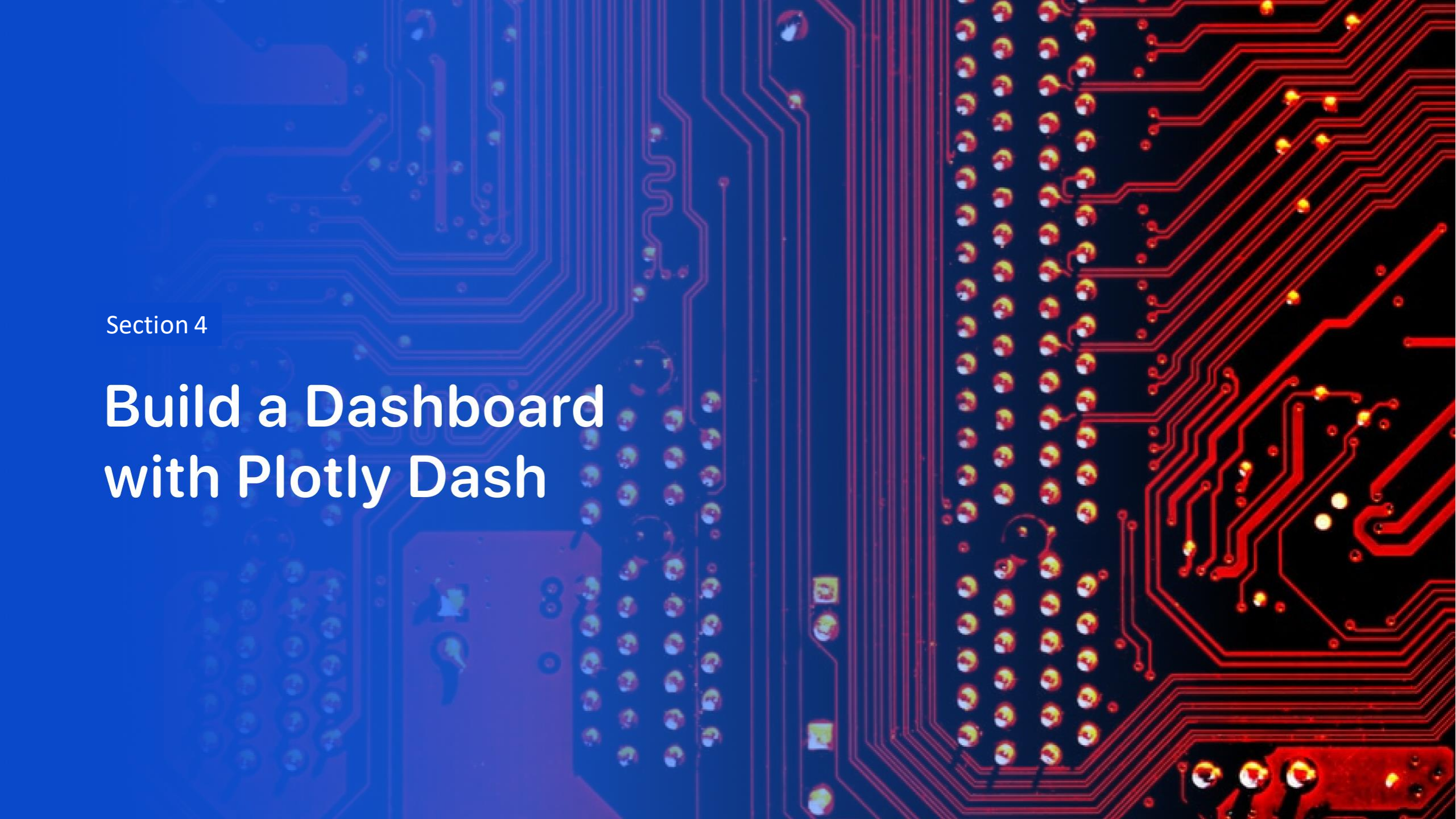
- Screenshot the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot
  - Green pins = successful launch, red = failure
  - This is done for all sites to see # of each outcome



Map of Locations Nearby

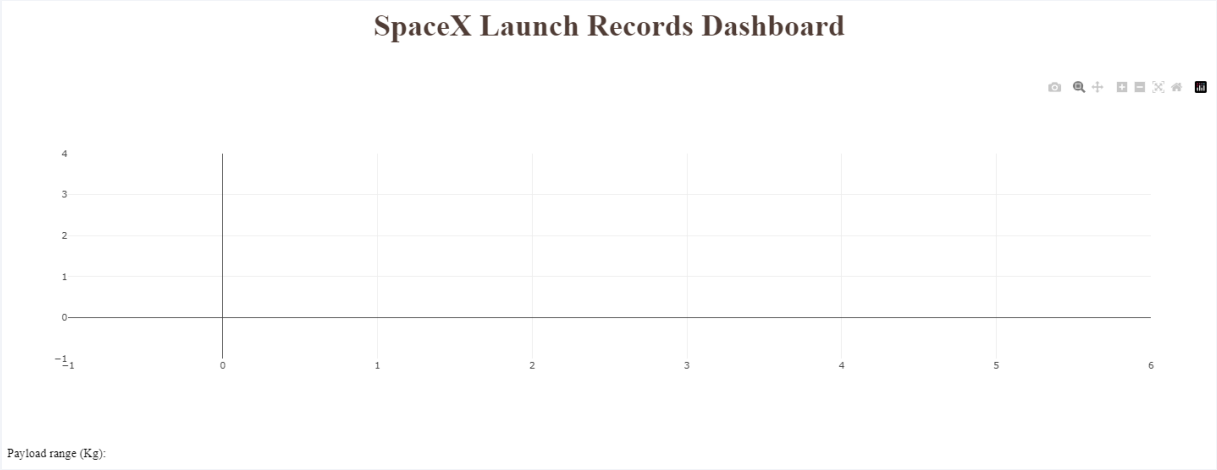
- Screenshot launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- Explain the important elements and findings on the screenshot
    - This map shows how far the launch site is to each of the specified locations and plots a line connecting the points

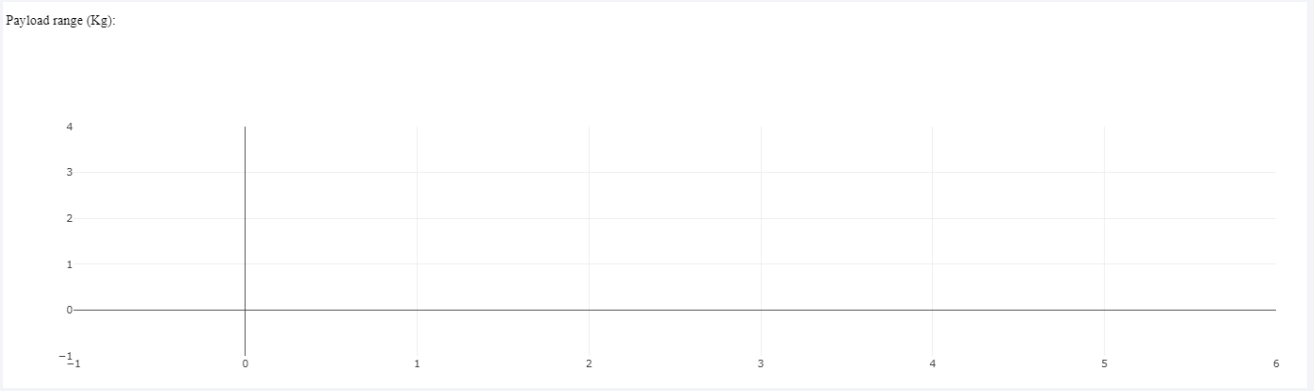



Section 4
Build a Dashboard with Plotly Dash

Dashboard Overview

- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot





<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

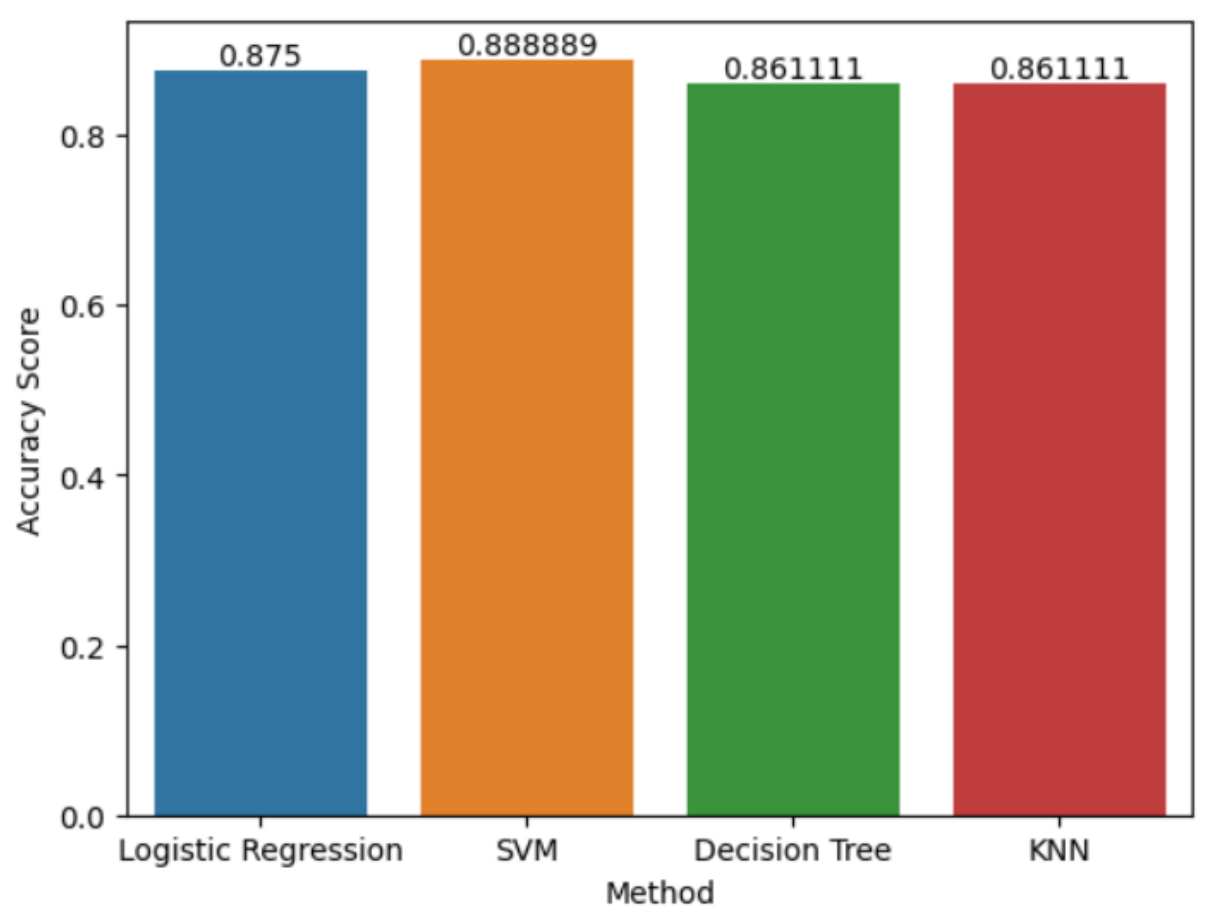
- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

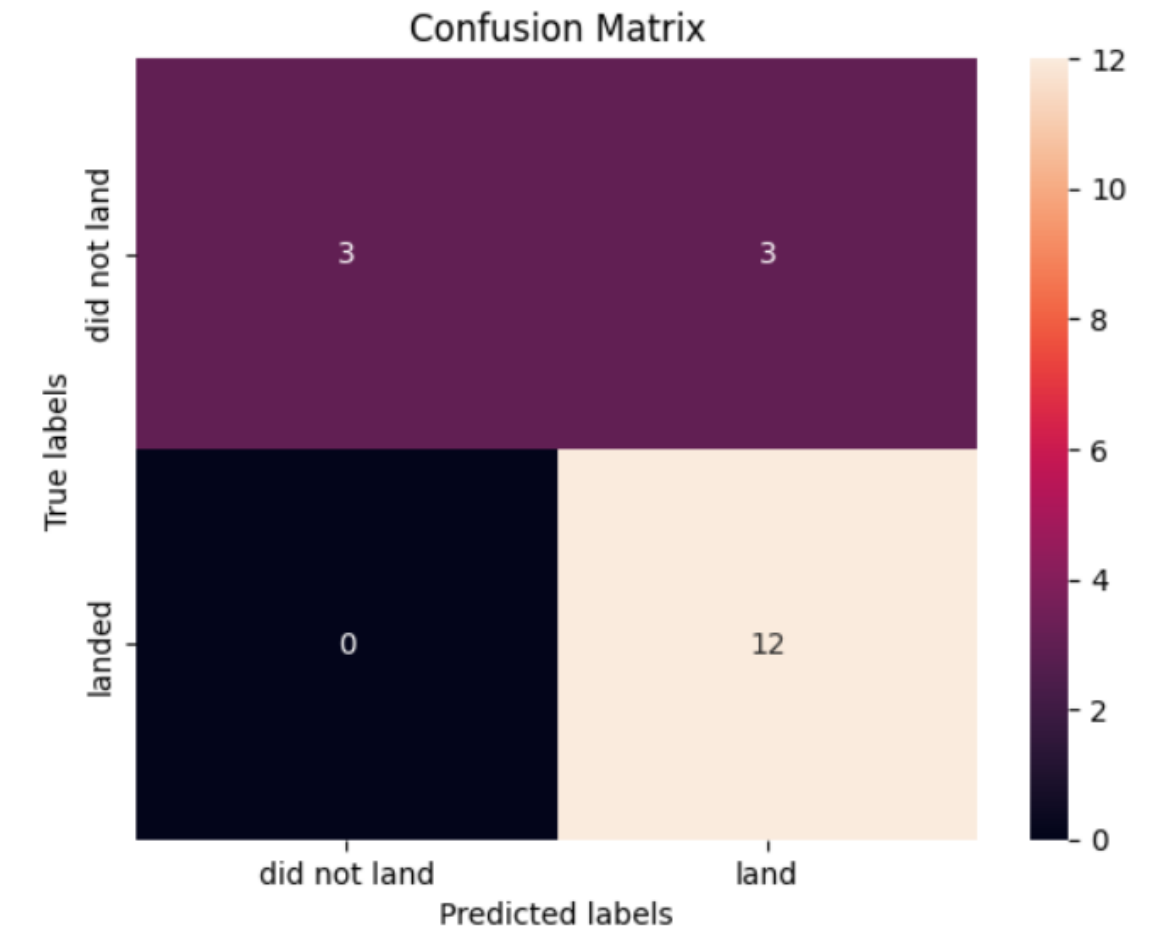
- Visualize the built model accuracy for all built classification models, in a bar chart



- Model with the highest classification accuracy:
  - SVM

Confusion Matrix

- Confusion matrix of the best performing model



- Explanation
  - 3 False positives (top right of matrix)

Conclusions

- SVM has proven to be the most effective model
- The confusion matrix has relatively high accuracy as only 3 out of 18 labels were incorrectly identified
- Decision Tree and KNN had the lowest accuracy out of all the models

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

- For additional information please refer to my GitHub files

Thank you!