IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ibrahim Yazici
January, 2024

Outline

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

Executive Summary

- This report presents a comprehensive analysis of SpaceX Falcon 9 launches with the objective of predicting the outcomes of these missions. Our methodology integrates data collection, wrangling, exploratory and interactive visual analytics, culminating in predictive analysis using sophisticated classification models.
- Data Collection & Wrangling: We leveraged the SpaceX API to extract relevant data on Falcon 9 launches, complemented by additional historical launch records sourced from a dedicated Wikipedia page.
- Exploratory Data Analysis: Our EDA involved a blend of visualization techniques and SQL queries to unveil patterns, anomalies, and relationships within the data. This foundational analysis provided insights into the factors influencing launch success.
- Visual Analytics: Utilizing the Folium library, we conducted interactive visual analytics to spatially examine the launch outcomes. By mapping the launch sites and overlaying key data points, we gained a geographical perspective of the mission successes and failures.
- Predictive Analysis: Our analysis performs the construction and tuning of several classification models to predict launch outcomes. Our iterative process involved building, tuning, and evaluating the models to optimize their performance.
- The findings and predictive models developed through this analysis offer a data-driven approach to enhancing the success rate of Falcon 9 booster landings.

Introduction

- In this capstone, we will develop models to predict if the Falcon 9 first stage will land successfully.
- SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can 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

Executive Summary

- Data collection methodology:
  - Request to the SpaceX API and clean the requested data. Also, perform web scraping to collect Falcon 9 historical launch records from a Wikipedia page.
- Perform data wrangling
  - Convert outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- We collect and make sure the data is in the correct format from the SpaceX API.
- We perform web scraping to collect Falcon 9 historical launch records from https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches.

Data Collection – SpaceX API

- We make a get request to the SpaceX API.
- We define helper functions that will help us use the API to extract information using identification numbers in the launch data.
- GitHub URL of the completed SpaceX API calls notebook: (https://github.com/ibrahimyzc/Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb)

1. Request and parse the SpaceX launch data using the GET request
2. Use json_normalize meethod to convert the json result into a dataframe
3. Take a subset of our dataframe keeping only the features we want
4. Filter the dataframe to only include Falcon 9 launches
5. Deal with Missing Values
6. Export dataset to a CSV

Data Collection - Scraping

- We performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled ”List of Falcon 9 and Falcon Heavy launches”
- GitHub URL of the completed web scraping notebook: (https://github.com/ibrahimyzc/Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb)

1. Import required packages and provide some helper functions to process web scraped HTML table
2. Perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response
3. Create a BeautifulSoup object from the HTML response
4. Extract all column/variable names from the HTML table header
5. Create a dataframe from parsed values
6. Export the dataframe to a CSV

Data Wrangling

- We perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.
- GitHub URL for the data wrangling notebook: (https://github.com/ibrahimyzc/Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb)

1. There are several different cases where the booster did not land successfully. We will convert those outcomes into Training Labels with 1 and 0
2. Calculate the number of launches on each site
3. Calculate the number and occurrence of each orbit
4. Calculate the number and occurence of mission outcome of the orbits
5. Create a landing outcome label from Outcome column

EDA with Data Visualization

- We perform exploratory some Data Analysis and Feature Engineering tasks:
  - ✓ Visualize the relationship between Flight Number and Launch Site
  - ✓ Visualize the relationship between Payload and Launch Site
  - ✓ Visualize the relationship between success rate of each orbit type
  - ✓ Visualize the relationship between FlightNumber and Orbit type
  - ✓ Visualize the relationship between Payload and Orbit type
  - ✓ Visualize the launch success yearly trend
  - ✓ Create dummy variables to categorical columns
  - ✓ Cast all numeric columns to `float64`
- GitHub URL: (https://github.com/ibrahimyzc/Data-Science-Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb)

EDA with SQL

- We execute SQL queries to perform the following tasks:
  - ✓Display the names of the unique launch sites in the space mission
  - ✓Display 5 records where launch sites begin with the string 'CCA'
  - ✓Display the total payload mass carried by boosters launched by NASA (CRS)
  - ✓Display average payload mass carried by booster version F9 v1.1
  - ✓List the date when the first succesful landing outcome in ground pad was acheived.
  - ✓List names of the booster_versions which have carried the maximum payload mass.
- GitHub URL: (https://github.com/ibrahimyzc/Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb)

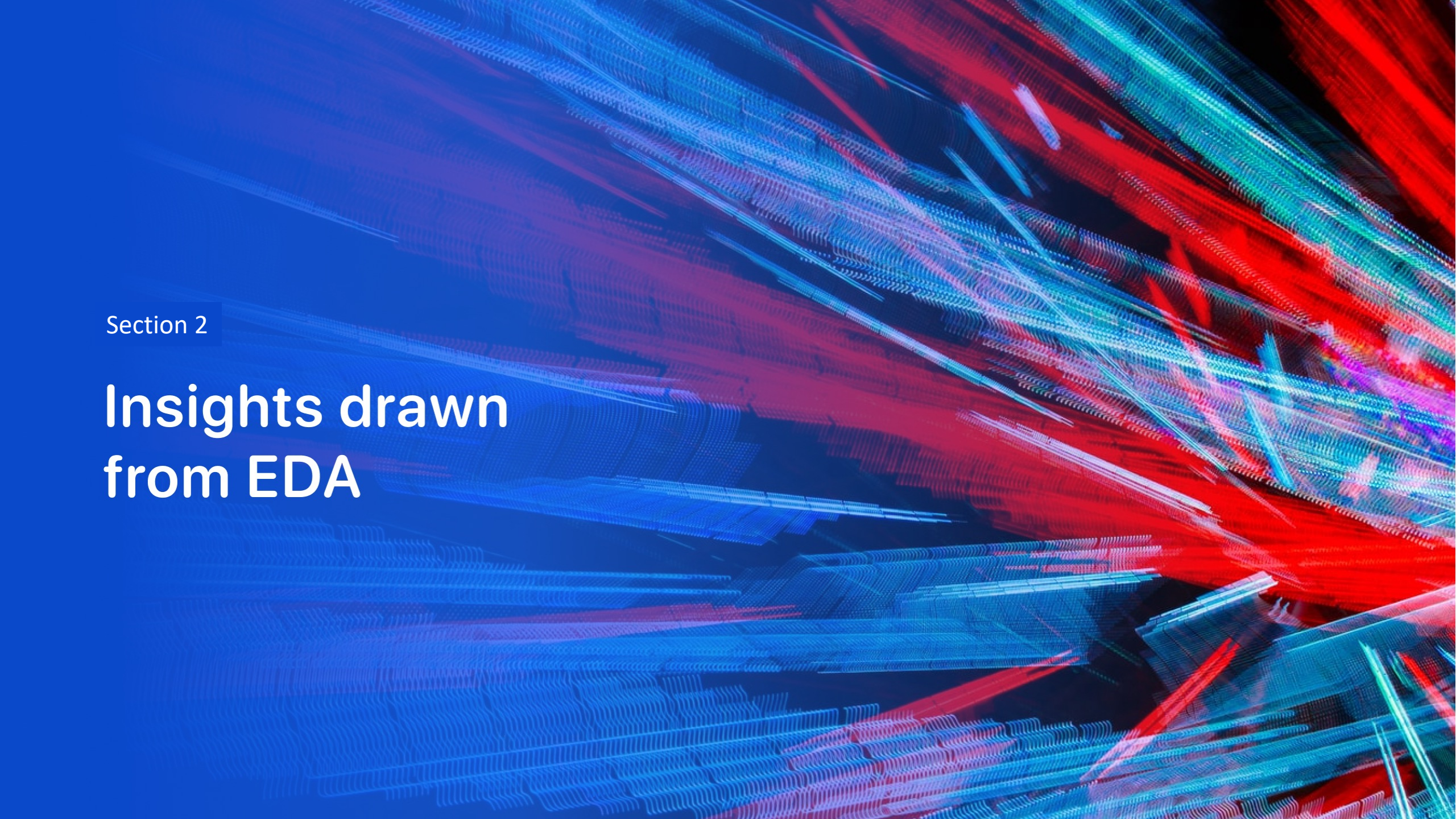
Build an Interactive Map with Folium

- We performed interactive visual analytics using Folium:
  - ✓ Create and add folium.Circle and folium.Marker for each launch site on the site map to explore the map by zoom-in/out the marked areas
  - ✓ Mark the success/failed launches for each site on the map
  - ✓ Calculate the distances between a launch site to its proximities
  - ✓ Add Mouse Position to get the coordinate (Lat, Long) for a mouse over on the map
  - ✓ Create a marker with distance to a closest city, railway, highway, etc.
  - ✓ Draw a line between the marker to the launch site
- GitHub URL: (https://github.com/ibrahimyzc/Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb)

Predictive Analysis (Classification)

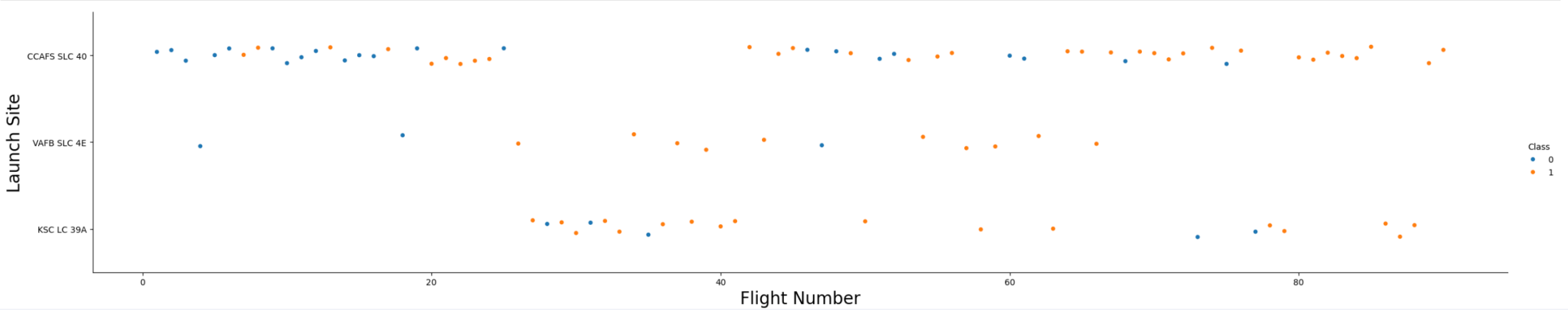
- We performed exploratory Data Analysis and determine Training Labels:
  - Create a column for the class
  - Standardize the data
  - Split into training data and test data
- We determined best Hyperparameter for SVM, Classification Trees and Logistic Regression
- We discovered the method performs best using test data
- GitHub URL: (https://github.com/ibrahimyzc/Data-Science-Capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb)

1. Standardize the data
2. Split the data into training and test data
3. Create a dataframe from parsed values
4. Create a logistic regression object then create a GridSearchCV to find the best parameters
5. Calculate the accuracy on the test data using the method score
6. Repeat the same process for support vector machine, decision tree classifier, and k nearest neighbors.
7. Find the method performs best using the test data


Section 2
Insights drawn from EDA

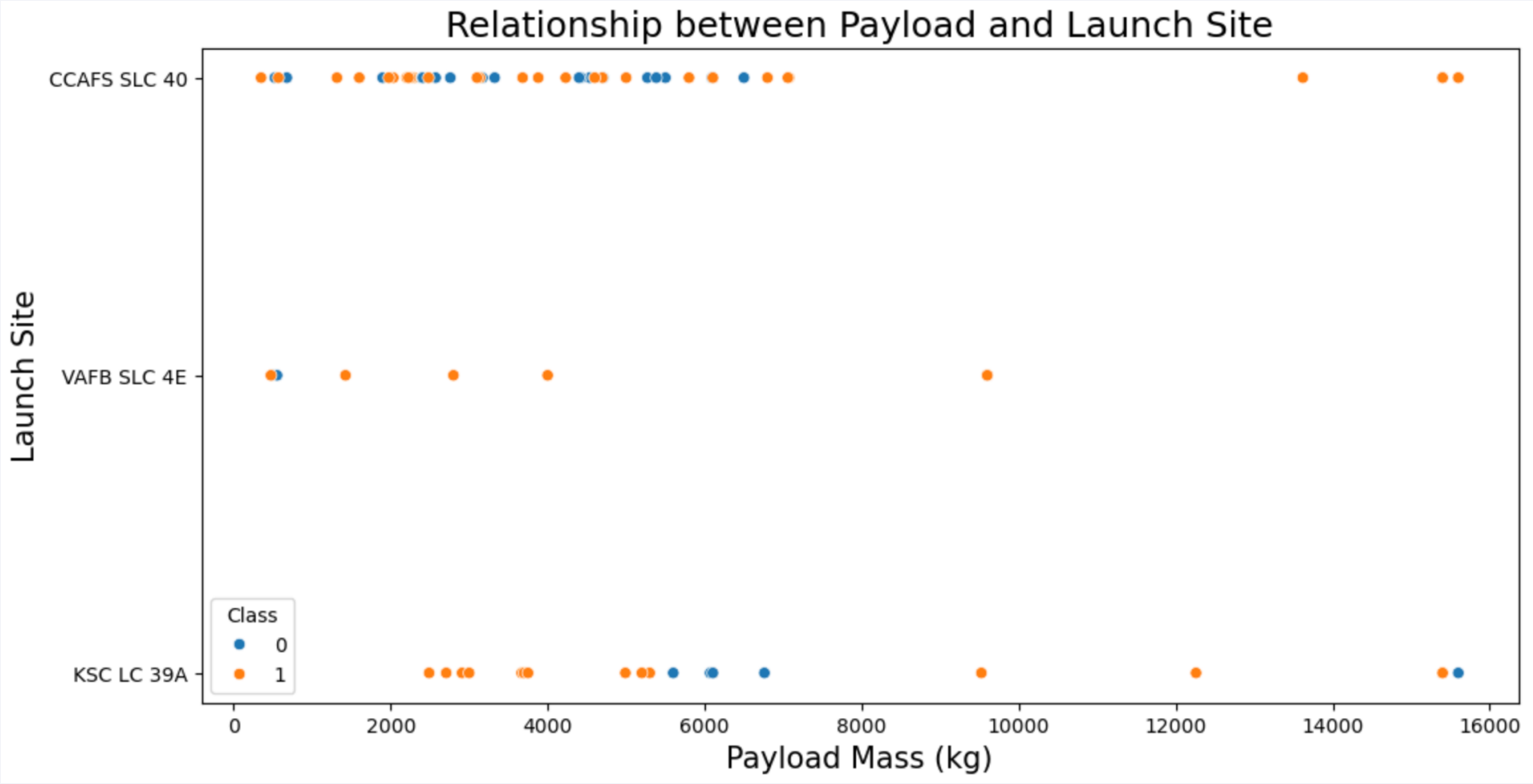
Flight Number vs. Launch Site

Different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%. Below, let's drill down to each site visualize its detailed launch records.



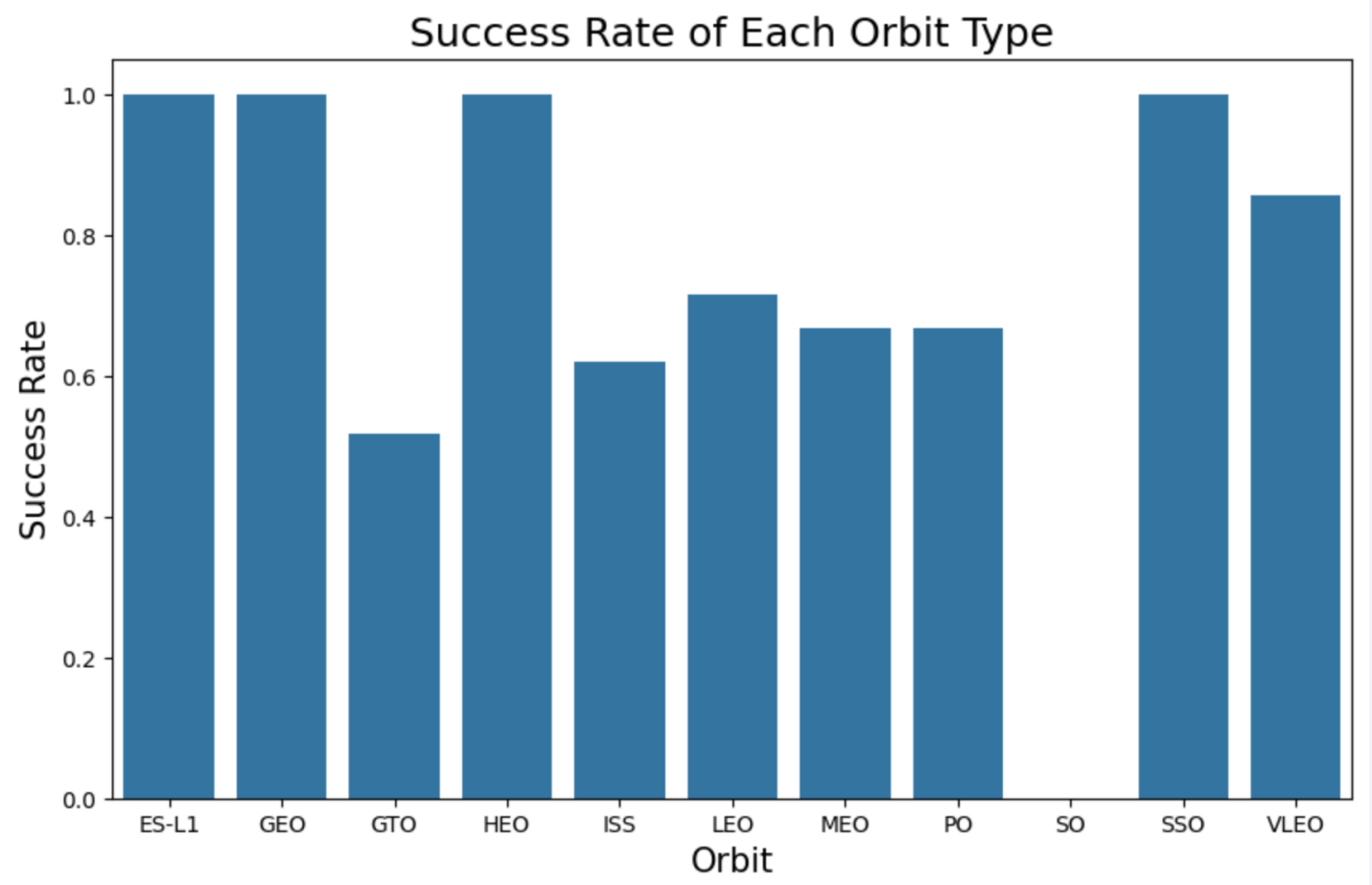
Payload vs. Launch Site

We want to observe if there is any relationship between launch sites and their payload mass. Scatter point chart shows for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).



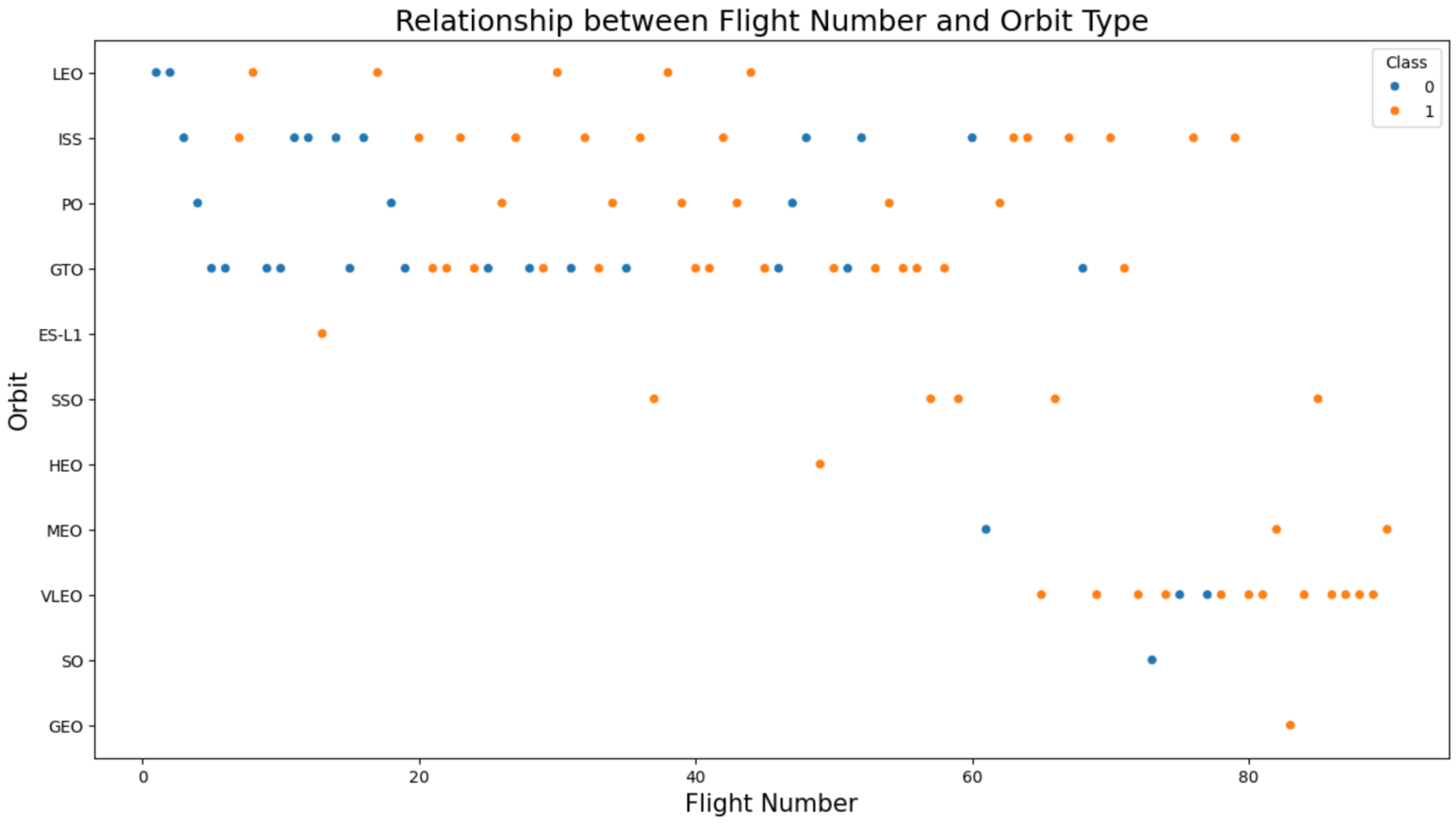
Success Rate vs. Orbit Type

We want to visually check if there are any relationship between success rate and orbit type using a bar chart.



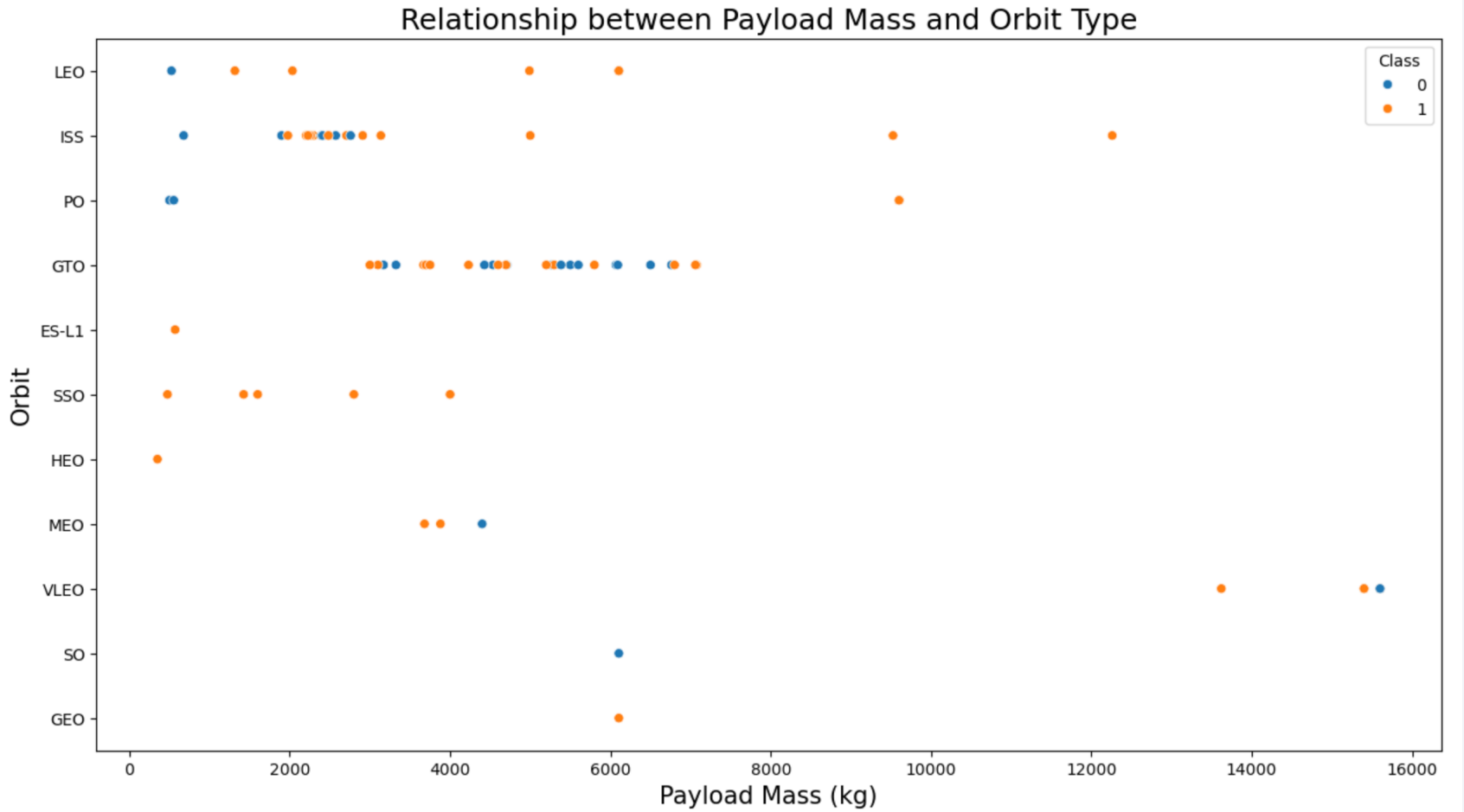
Flight Number vs. Orbit Type

We see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



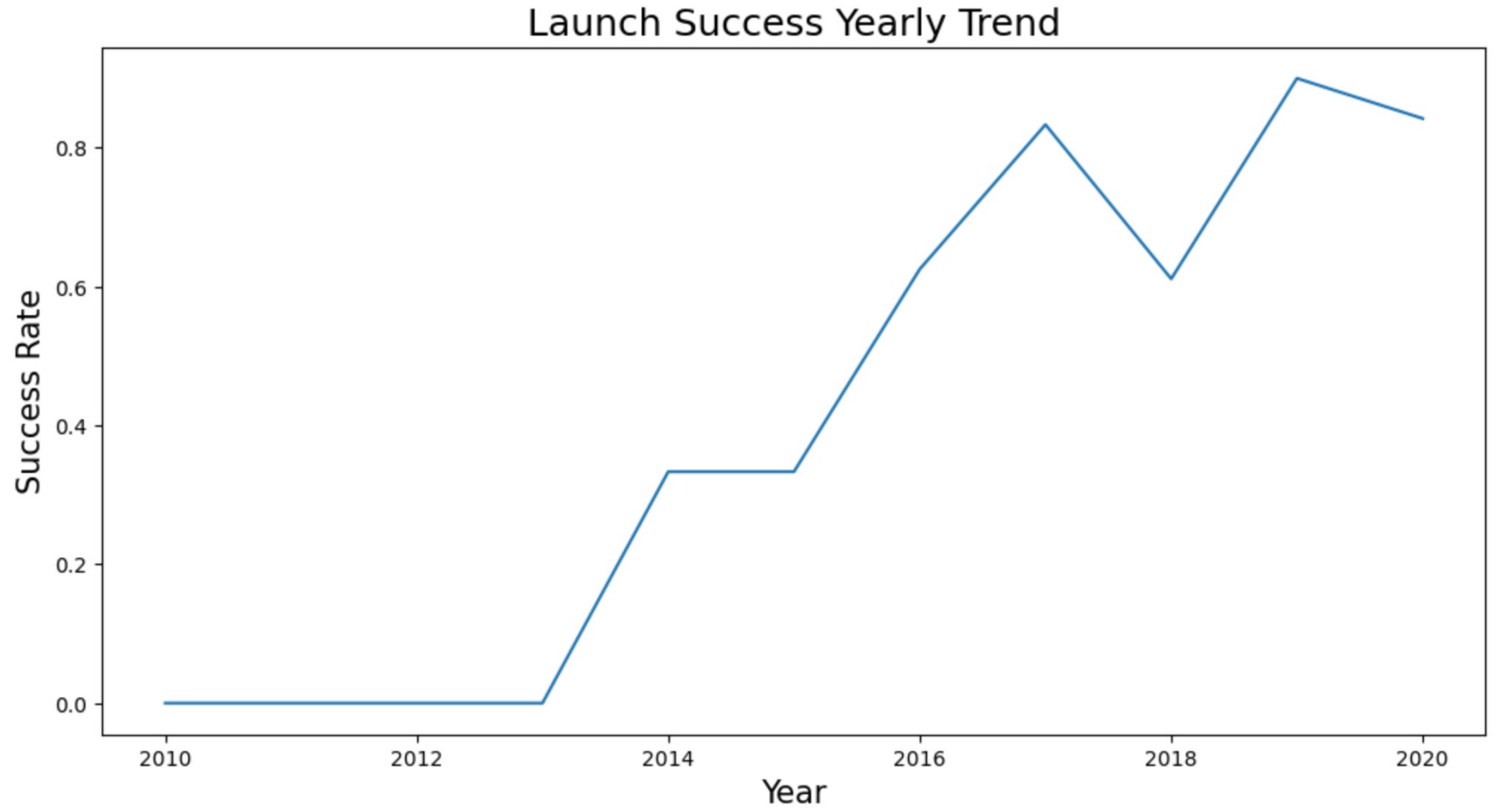
Payload vs. Orbit Type

With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
For GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there.



Launch Success Yearly Trend

We visualize the launch success yearly trend



All Launch Site Names

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

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

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

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

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

```
%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

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

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1'
```

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

AVG("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

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

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

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

MIN("Date")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 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

List the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", 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

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

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

```
 * 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

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

```
%sql SELECT SUBSTR(Date, 6, 2) AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE SUBSTR(Date, 1, 4) = '2015' AND "Landing_Outcom
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
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.

```
%sql SELECT "Landing_Outcome", COUNT(*) AS Outcome_Count FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY Ou
```

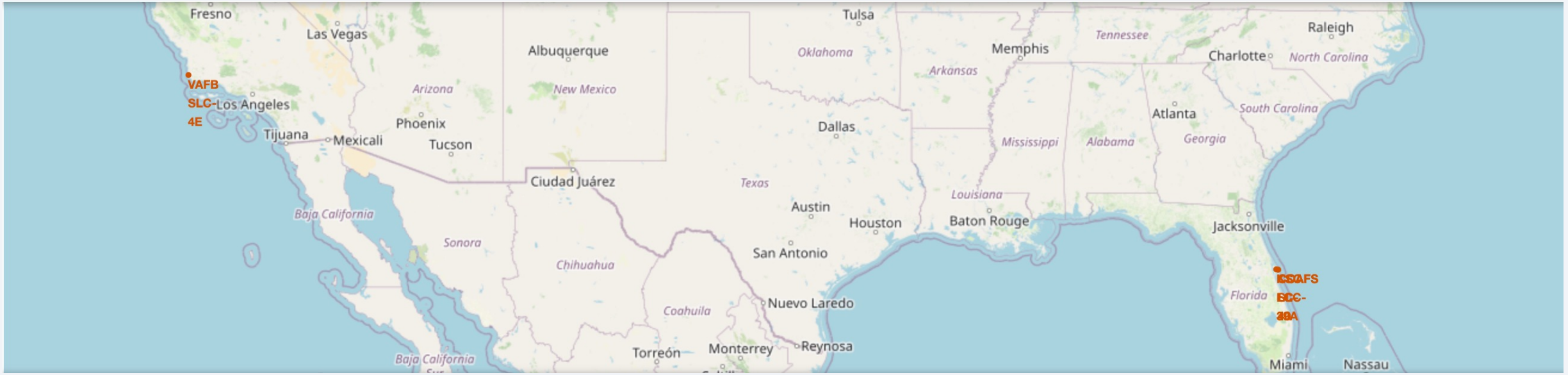
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	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

<Mark all launch sites on a map>

- We add each site's location on a map using site's latitude and longitude coordinates.
- We can explore the map by zoom-in/out the marked areas , and try to answer the following questions:
  - Are all launch sites in proximity to the Equator line?
  - Are all launch sites in very close proximity to the coast?

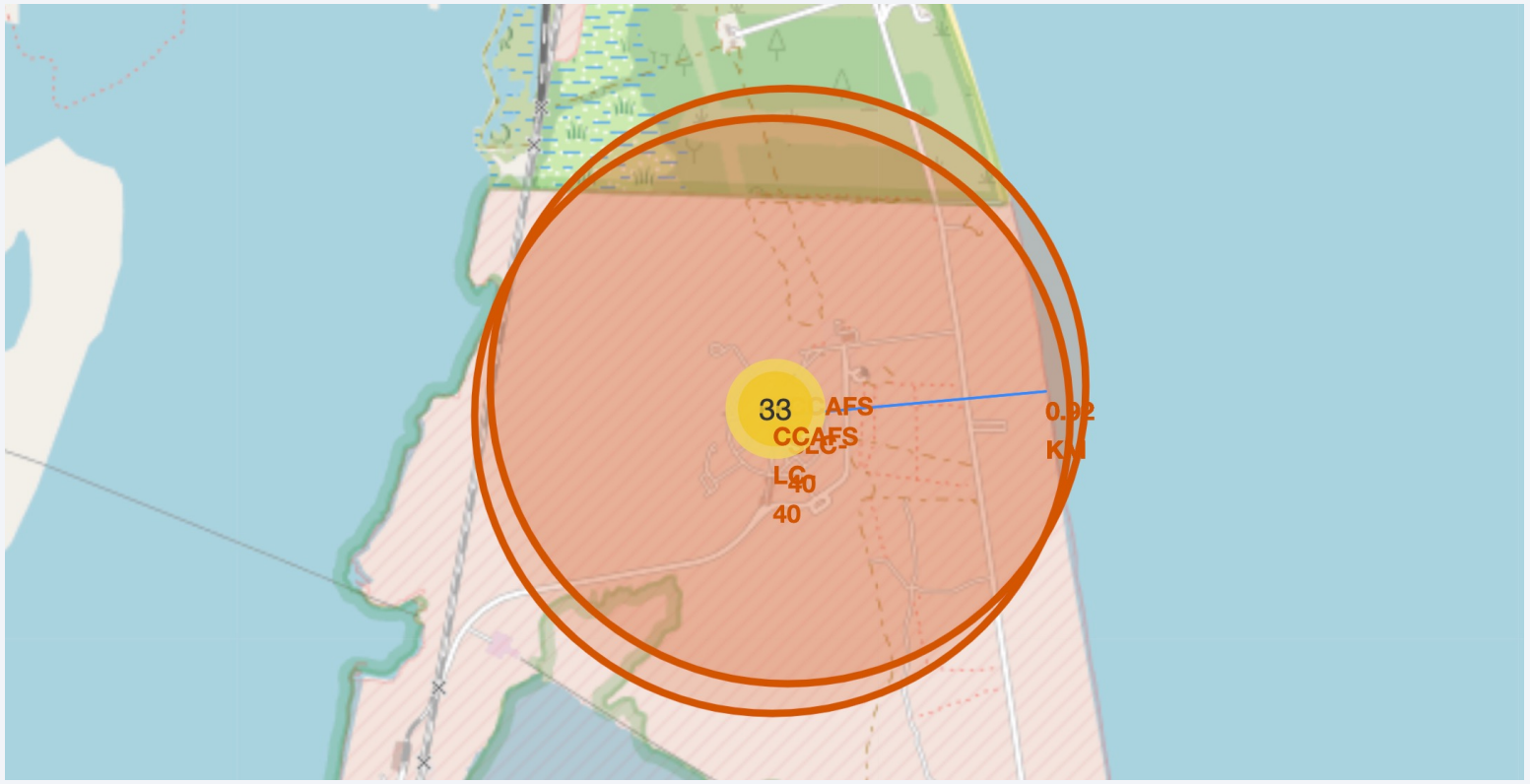


<Mark the success/failed launches for each site on the map>

Enhance the map by adding the launch outcomes for each site, and see which sites have high success rates.

<Calculate the distances between a launch site to its proximities>

- We zoom in to a launch site and explore its proximity to see if you can easily find any railway, highway, coastline, etc.
- We move the mouse to these points and mark down their coordinates in order to the distance to the launch site. Below, the proximity to the coastline is shown on the map as an example.

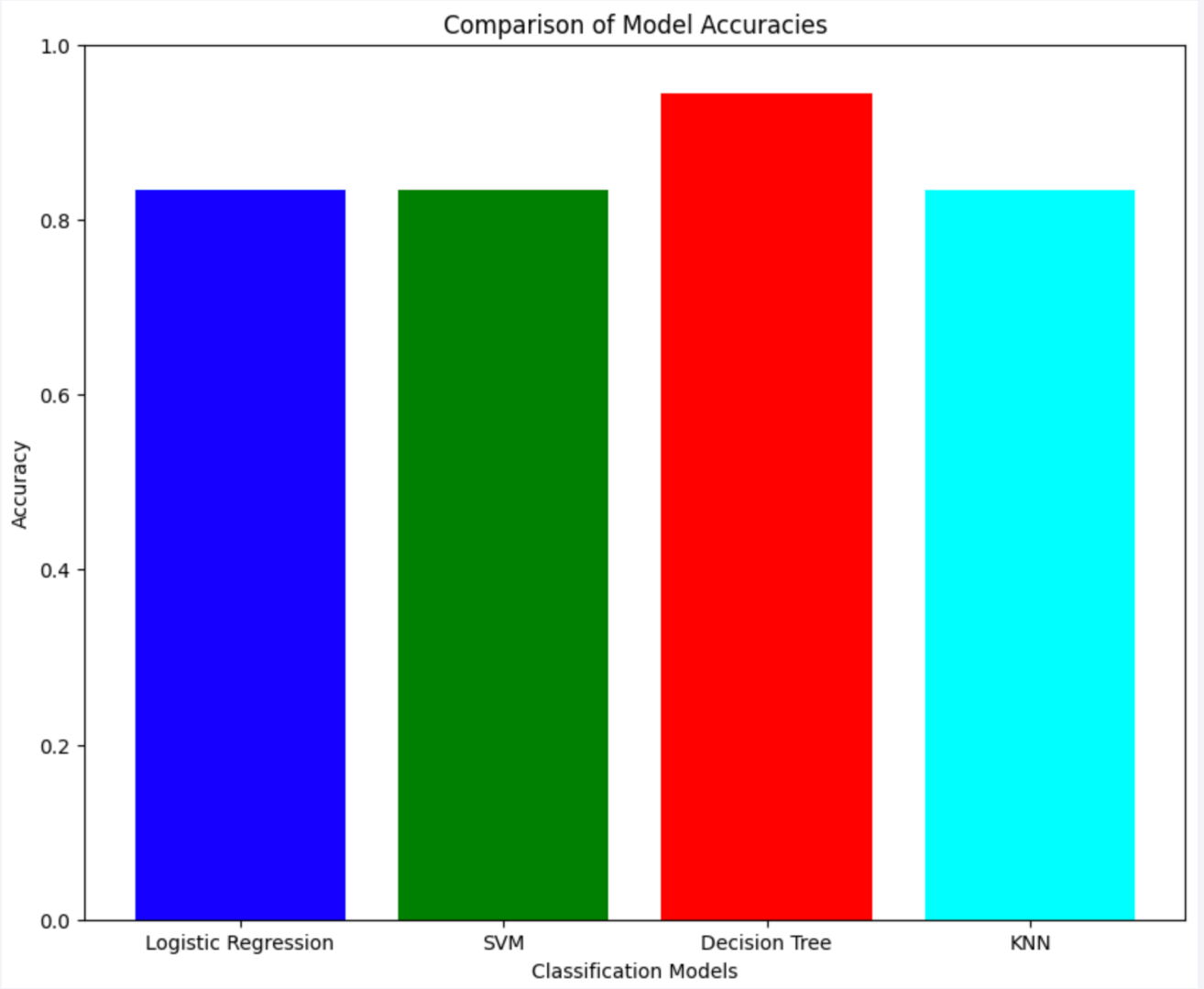


Section 4

Predictive Analysis (Classification)

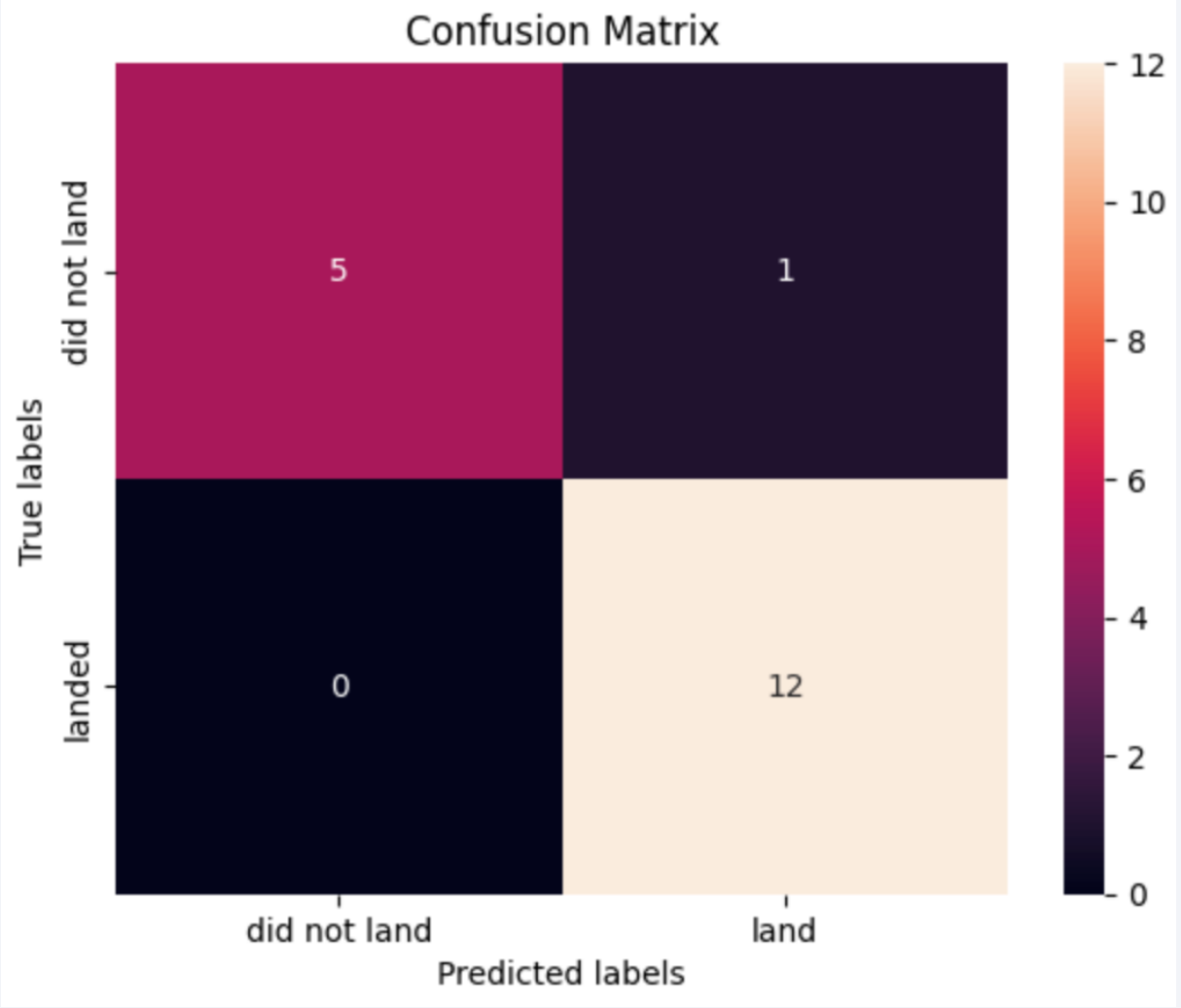
Classification Accuracy

Decision tree model has the highest model accuracy with 0.944 on the test data.



Confusion Matrix for the Decision Tree Model

Examining the confusion matrix, we see that decision tree model can effectively distinguish between the different classes.



Conclusions

- Decision tree is the best performing model among the four models tested.
- Its tuned hpyerparameters are: {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 5, 'splitter': 'random'}. Its accuracy on the train set is 0.87321.
- By the confusion matrix, we can conclude:
  - The model is quite good at predicting when a launch will land successfully, with 12 true positives.
  - It has a small number of false positives, with only 1 instance where it predicted a landing would occur when it did not.
  - There are no false negatives; the model did not miss any landings.
  - The model has 5 true negatives, correctly identifying 5 instances where the launch did not land.
- The remaining 3 models, Logistic Regression, SVM and, KNN displayed similar performances according to the accuracy values.

Appendix

All the files related to this project can be found at:

<https://github.com/ibrahimyzc/Data-Science-Capstone/tree/main>

Thank you!