IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Carlos Bautista>
<10/21/24>

Outline

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

Executive Summary

- The goal of this project is to predict the successful landing of the Falcon 9 rocket’s first stage, a key factor in determining the cost-efficiency of a SpaceX launch. SpaceX advertises a launch cost of $62 million, significantly lower than competitors who charge upwards of $165 million. This price advantage largely stems from SpaceX's ability to reuse the first stage of the Falcon 9 rocket. By accurately predicting whether the first stage will land successfully, we can assess the associated launch costs.
- This information is particularly valuable for potential competitors bidding against SpaceX for rocket launches, as it helps to quantify the cost benefits of reusable rockets. In this project, data is collected via an API, formatted, and analyzed to train a model that predicts landing success based on historical launch data. The project provides insights into the feasibility of reusability and its implications for the broader space industry.

Introduction

- The goal of this project is to predict the successful landing of the Falcon 9 rocket’s first stage, a key factor in determining the cost-efficiency of a SpaceX launch. SpaceX advertises a launch cost of $62 million, significantly lower than competitors who charge upwards of $165 million. This price advantage largely stems from SpaceX's ability to reuse the first stage of the Falcon 9 rocket. By accurately predicting whether the first stage will land successfully, we can assess the associated launch costs.
- This project focuses on predicting the likelihood of a successful landing for the Falcon 9 first stage. By analyzing historical data on previous launches, we aim to build a model that forecasts whether the first stage will land safely. Accurate predictions will not only help SpaceX further optimize its operations but also provide valuable insights for competitors interested in bidding for rocket launch contracts. With access to launch data via an API, this project seeks to understand the key factors influencing landing success and the broader implications for reducing the cost of space travel.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was sourced from IBM Skills Network
- Perform data wrangling
  - Data was processed thru web scraping
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LogReg, Dtree, KNN, SVM

Data Collection

- Data was collected through webscraping

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

Place your flowchart of SpaceX API calls here

Data Collection - Scraping

- IBM-Practice-Labs/Capstone Project 2/module 1 - Intro/web_scraping.py at main · CxLos/IBM-Practice-Labs

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTT

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content, "html.parser")
```

Data Wrangling

- IBM-Practice-Labs/Capstone Project 2/module 1 - Intro/data_wrangling_spacex.py at main · CxLos/IBM-Practice-Labs

EDA with Data Visualization

- IBM-Practice-Labs/Capstone Project 2/module 2 - EDA/labs/jupyter-labs-eda-dataviz-v2.ipynb at main · CxLos/IBM-Practice-Labs
- During the EDA stage of this project, we made some charts using Folium to discover any correlations between the launch site and success rates.

EDA with SQL

- IBM-Practice-Labs/Capstone Project 2/module 2 - EDA/labs/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · CxLos/IBM-Practice-Labs
- During this stage of the project, we used SQL queries to display the launch sites, payload mass, find boosters with specific landing outcomes, and to find the number of successful mission outcomes.

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
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

Build a Dashboard with Plotly Dash

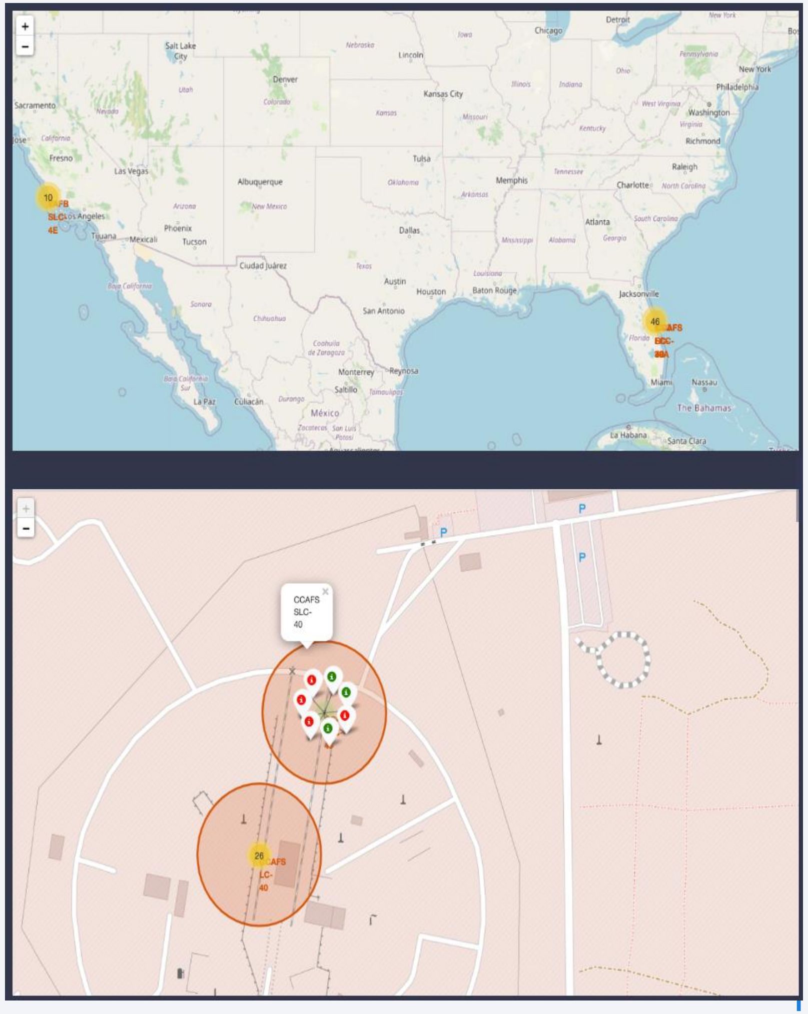
- [IBM-Practice-Labs/Capstone Project 2/module 3 - Interactive Visual Analytics/labs/lab-jupyter-launch-site-location-v2.ipynb at main · CxLos/IBM-Practice-Labs]
- In this dashboard we created a successful landing pie chart by location as well as a scatter plot for payload.

Predictive Analysis (Classification)

- IBM-Practice-Labs/Capstone Project 2/module 4 - Predictive Analysis/labs/SpaceX-Machine-Learning-Prediction-Part-5-v1.ipynb at main · CxLos/IBM-Practice-Labs
- In the Predictive Analysis stage, We utilized Logistic Regression, Decision Tree, SVM and KNN machine learning models and determined the best parameters for each model as well as the best score in order to find out which method worked best.

Results

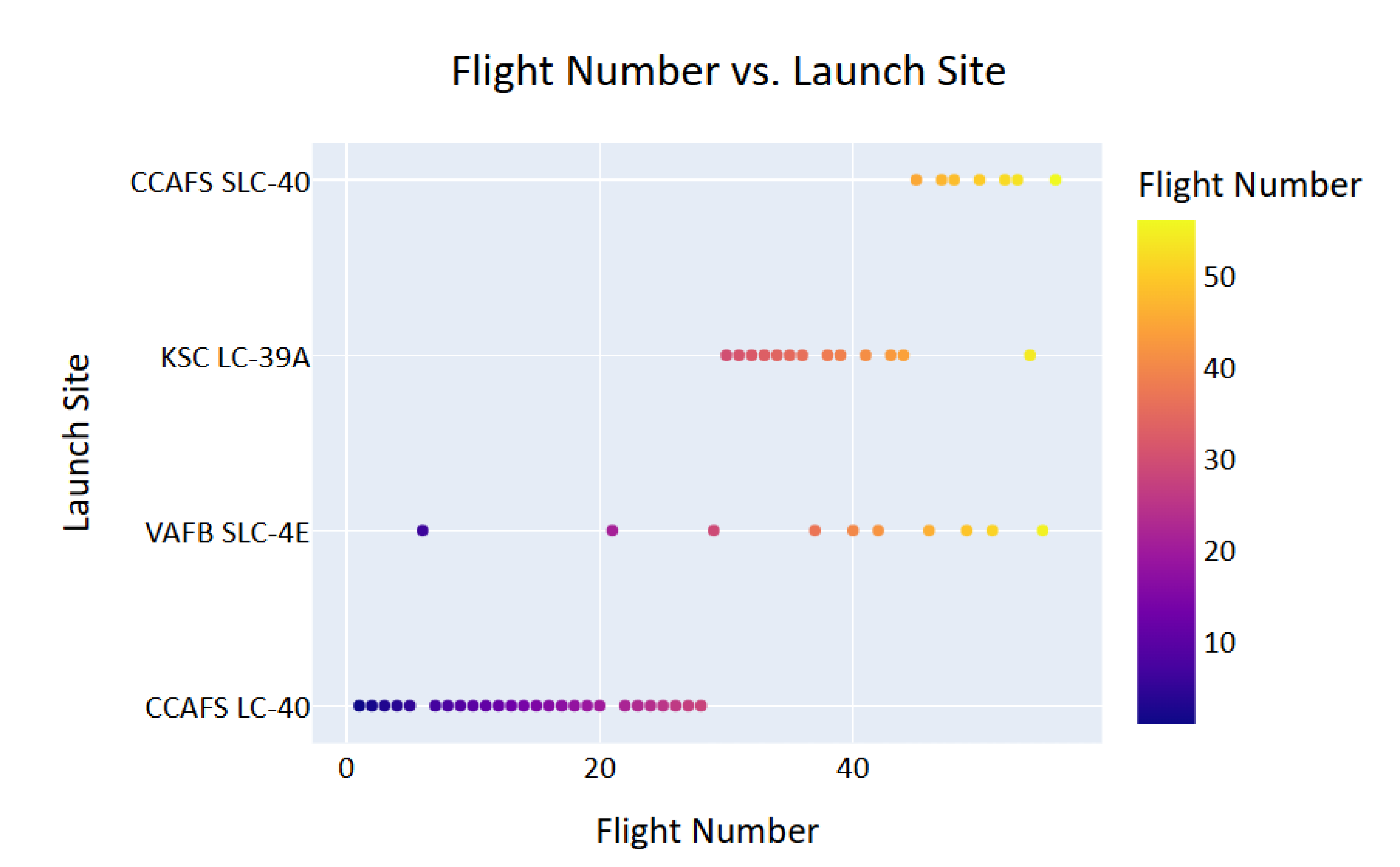
- Exploratory data analysis results
- Predictive analysis results: Decision Tree was the best model for the dataset since it yielded the highest accuracy.



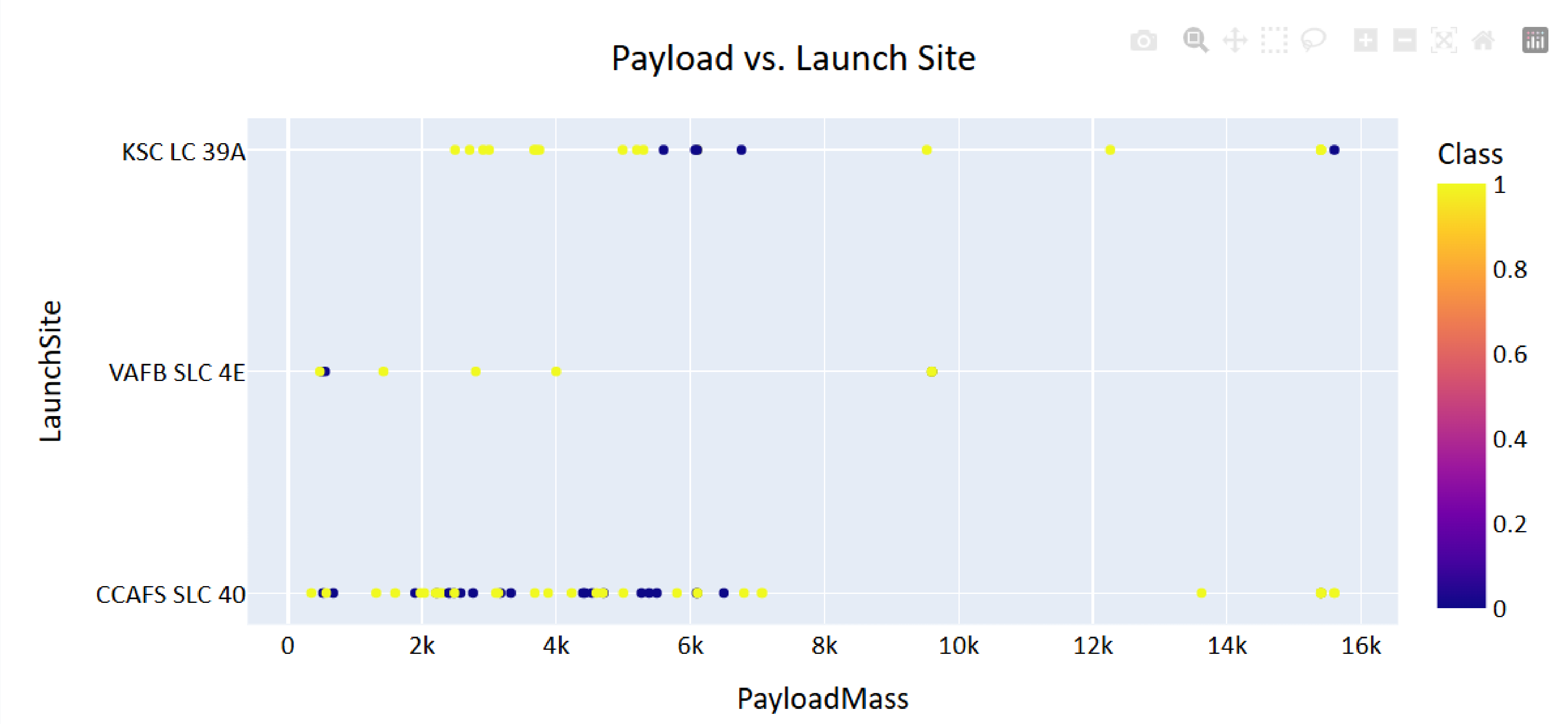
Section 2

Insights drawn from EDA

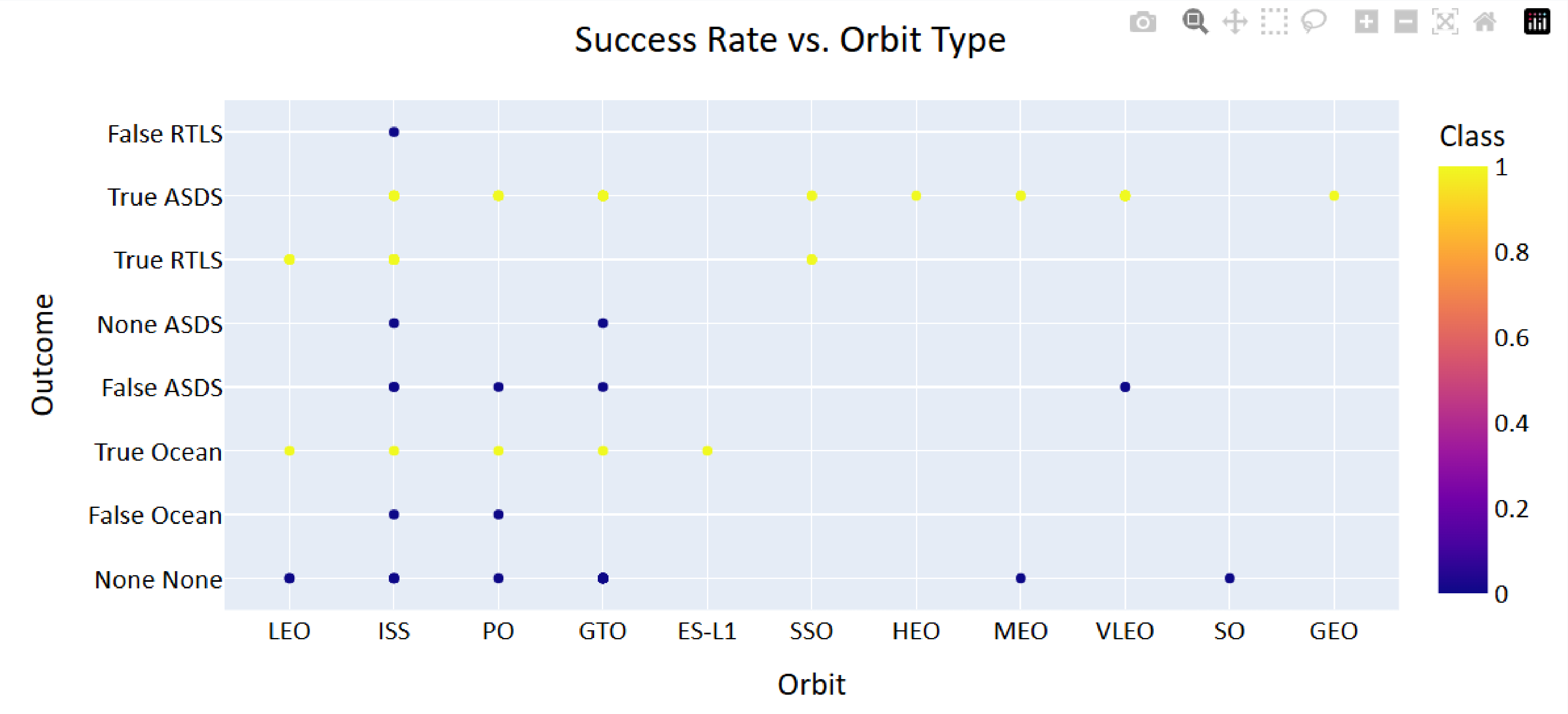
Flight Number vs. Launch Site



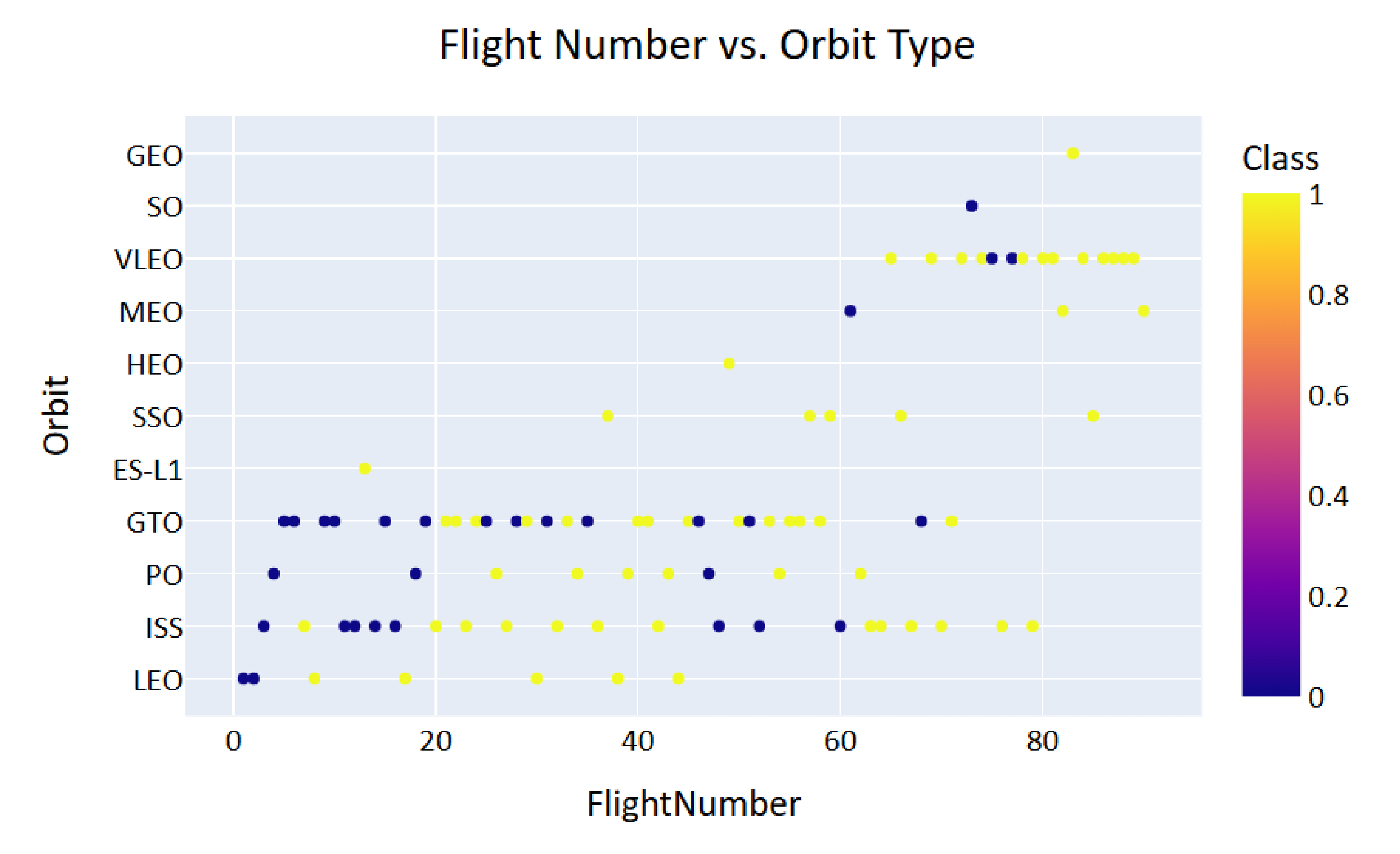
Payload vs. Launch Site



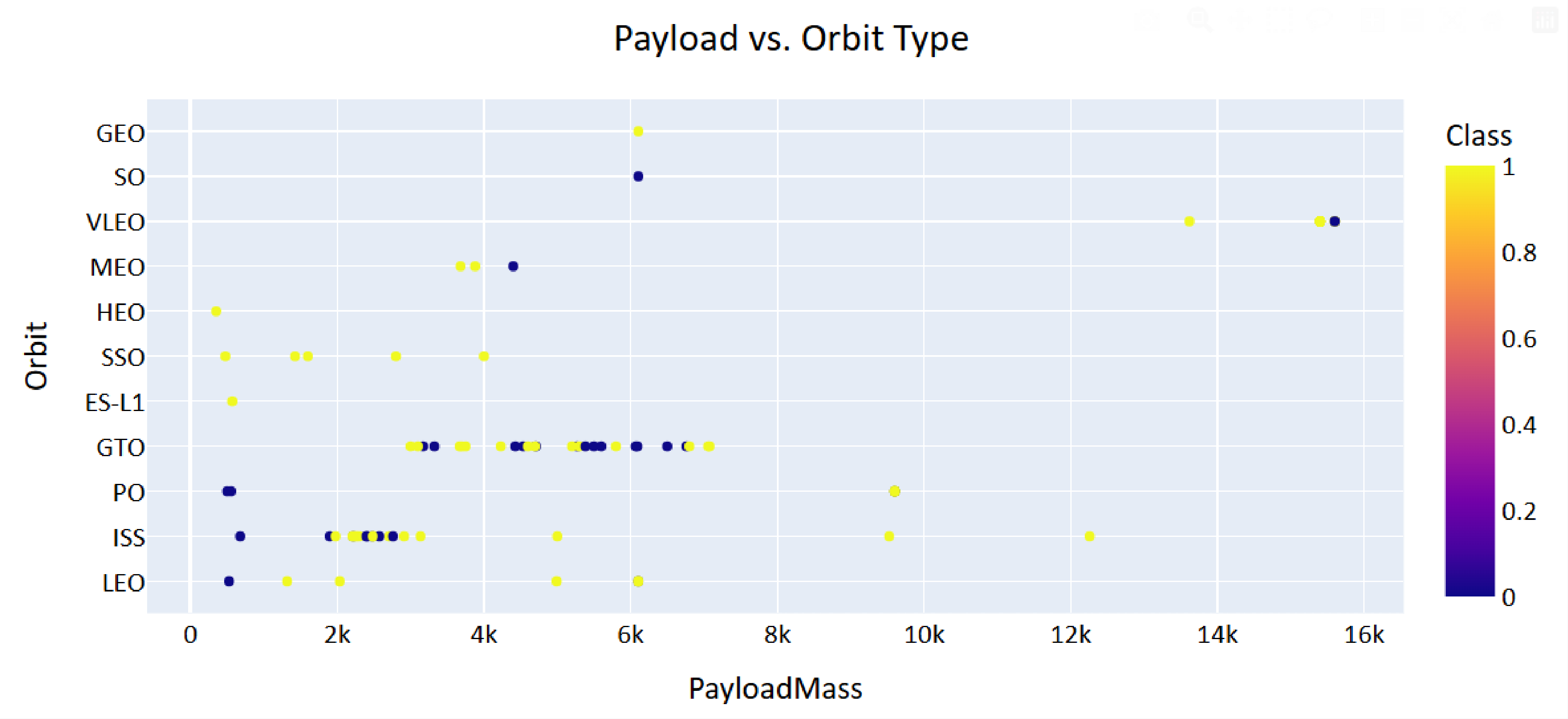
Success Rate vs. Orbit Type



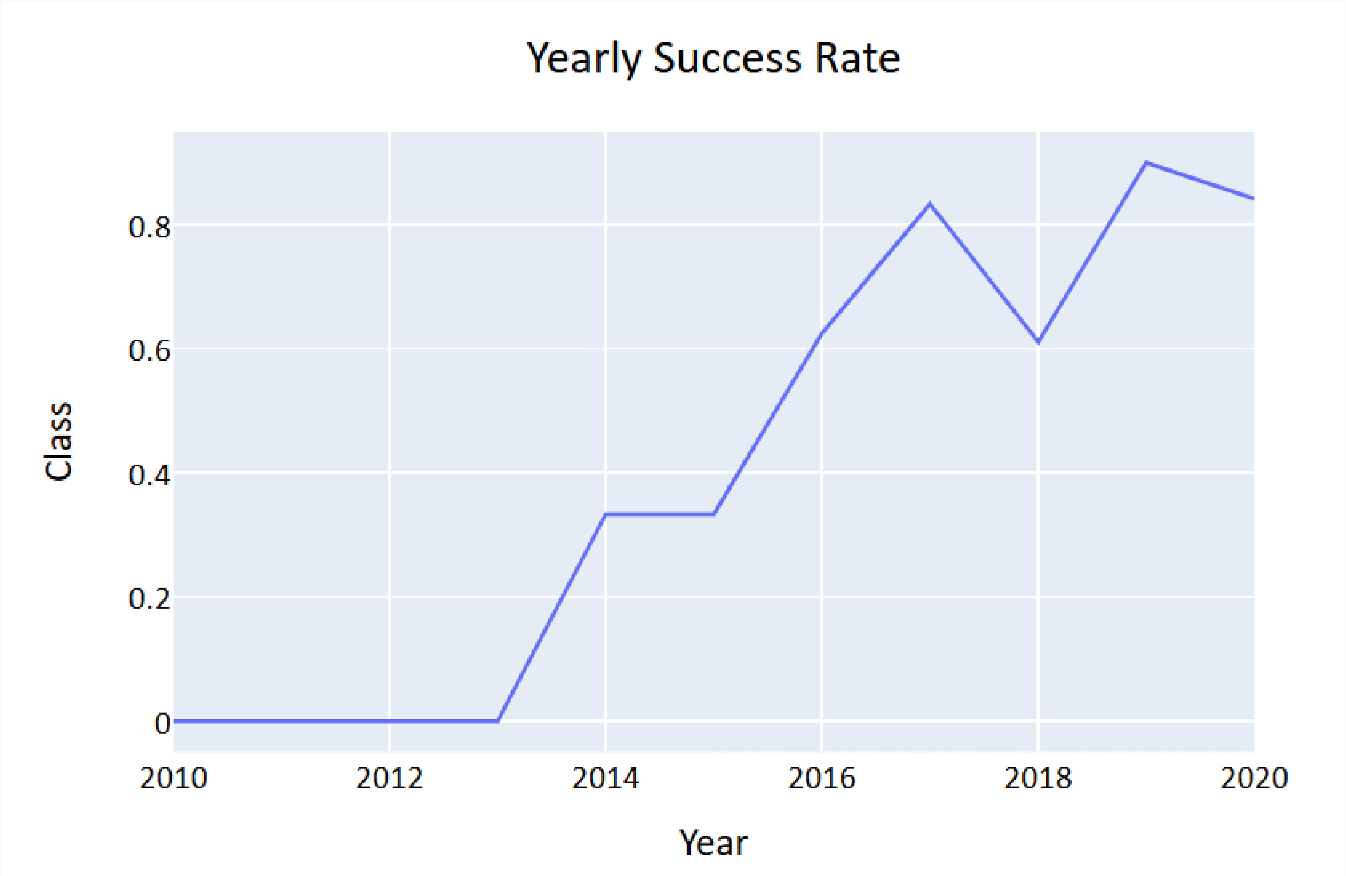
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

- This is an overview of all launch sites in the Spacex database.

```
# 1. All Launch Sites
# print(df['LaunchSite'].value_counts())

# CCAFS SLC 40    55
# KSC LC 39A      22
# VAFB SLC 4E     13
```

Launch Site Names Begin with 'CCA'

- Below are 5 Locations where Launch Site begins with “CCA”

```
# 2. Display 5 records where launch sites begin with the string 'CCA'
# query = pd.read_sql_query("""
#   SELECT Date, "Time (UTC)", Launch_Site
#   FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%'
#   Limit 5
#   """, con)

# print(query)

#          Date Time (UTC)  Launch_Site
# 0  2010-06-04   18:45:00  CCAFS LC-40
# 1  2010-12-08   15:43:00  CCAFS LC-40
# 2  2012-05-22    7:44:00  CCAFS LC-40
# 3  2012-10-08    0:35:00  CCAFS LC-40
# 4  2013-03-01   15:10:00  CCAFS LC-40
```

Total Payload Mass

- Total Payload Mass was 45,596 Kg

```
# 3. Display the total payload mass carried by boosters launched by NASA (CRS)
# query = pd.read_sql_query("""
#   SELECT SUM(PAYLOAD_MASS__KG_)
#   FROM SPACEXTBL
#   WHERE Customer='NASA (CRS)'
#   """, con)

#    SUM(PAYLOAD_MASS__KG_)
# 0                   45596
```

Average Payload Mass by F9 v1.1

- Average Payload Mass was 2,928.4 Kg

```
# 4. Display average payload mass carried by booster version F9 v1.1
# query = pd.read_sql_query("""
#   SELECT AVG(PAYLOAD_MASS__KG_)
#   FROM SPACEXTBL
#   WHERE Booster_Version='F9 v1.1'
#   """, con)

# print(query)

#    AVG(PAYLOAD_MASS__KG_)
# 0                  2928.4
```

First Successful Ground Landing Date

- First successful landing outcome was on December 22nd, 2022

```
# 5. List the date when the first succesful landing outcome in ground pad was acheived.
# query = pd.read_sql_query("""
#   SELECT MIN(Date)
#   FROM SPACEXTBL
#   WHERE Landing_Outcome='Success (ground pad)'
#   """, con)

# print(query)

#     MIN(Date)
# 0  2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- Boosters that have successfully laded on drone ship with a payload mass between 4,000 and 6,000 were:
  - F9 FT B1022
  - F9 FT B1026
  - F9 FT B1021.2
  - F9 FT B1031.2

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

# query = pd.read_sql_query("""
#   SELECT Booster_Version, Landing_Outcome
#   FROM SPACEXTBL
#   WHERE Landing_Outcome='Success (drone ship)'
#   AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
#   """, con)

# print(query)

#   Booster_Version       Landing_Outcome
# 0     F9 FT B1022   Success (drone ship)
# 1     F9 FT B1026   Success (drone ship)
# 2  F9 FT  B1021.2   Success (drone ship)
# 3  F9 FT  B1031.2   Success (drone ship)
```

Total Number of Successful and Failure Mission Outcomes

- Total Successful and Failure mission outcomes was 98

```
# 7. List the total number of successful and failure mission outcomes
# query = pd.read_sql_query("""
#   SELECT Count(Mission_Outcome)
#   FROM SPACEXTBL
#   WHERE Mission_Outcome='Success' OR Mission_Outcome='Failure'
#   """, con)

# print(query)

#    Count(Mission_Outcome)
# 0                      98
```

Boosters Carried Maximum Payload

- The booster that carried the max payload mass was F9 B5 B1048.4

```
# 8. List the names of the booster_versions which have carried the maximum payload mass. Use a
subquery
# query = pd.read_sql_query("""
#   SELECT Booster_Version, MAX(PAYLOAD_MASS__KG_)
#   FROM SPACEXTBL
#   WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
#   """, con)

# print(query)

#   Booster_Version  MAX(PAYLOAD_MASS__KG_)
# 0   F9 B5 B1048.4                   15600
```

2015 Launch Records

- Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015:

```
# 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.
# query = pd.read_sql_query("""
#   SELECT strftime('%B', (Date)) as Month, Landing_Outcome, Booster_Version, Launch_Site
#   FROM SPACEXTBL
#   WHERE Landing_Outcome='Failure (drone ship)'
#   AND strftime('%Y', (Date))='2015'
#   """, con)

# print(query)

#   Month       Landing_Outcome Booster_Version  Launch_Site
# 0   None   Failure (drone ship)    F9 v1.1 B1012  CCAFS LC-40
# 1   None   Failure (drone ship)    F9 v1.1 B1015  CCAFS LC-40
```

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

- Count of landing outcomes between 2010-06-04 and 2017-03-20:

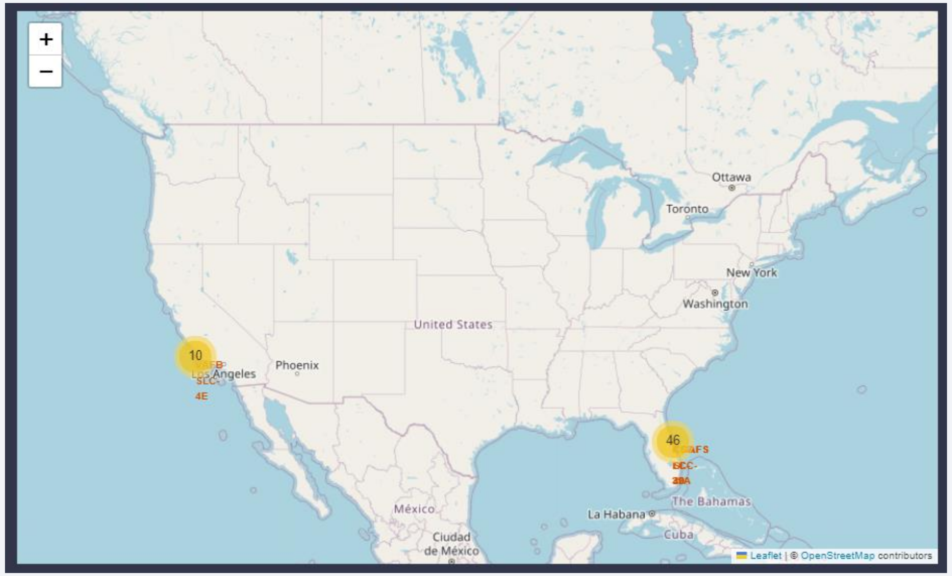
```
# 10. Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))
between the date 2010-06-04 and 2017-03-20, in descending order.
# query = pd.read_sql_query("""
#   SELECT Landing_Outcome, COUNT(Landing_Outcome) as Count
#   FROM SPACEXTBL
#   WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
#   GROUP BY Landing_Outcome
#   ORDER BY Count DESC
#   """, con)

# print(query)

#           Landing_Outcome  Count
# 0              No attempt     10
# 1     Success (drone ship)      5
# 2     Failure (drone ship)      5
# 3    Success (ground pad)      3
# 4      Controlled (ocean)      3
# 5    Uncontrolled (ocean)      2
# 6     Failure (parachute)      2
# 7  Precluded (drone ship)      1       You, 1 second ago • Uncommitted changes
```

Section 3

Launch Sites Proximities Analysis

Launch Site Locations

- In this folium map, you can see the launch site locations shown as marker clusters.



Launch Outcomes

- Folium map showing successful landing outcomes as green and failed in red

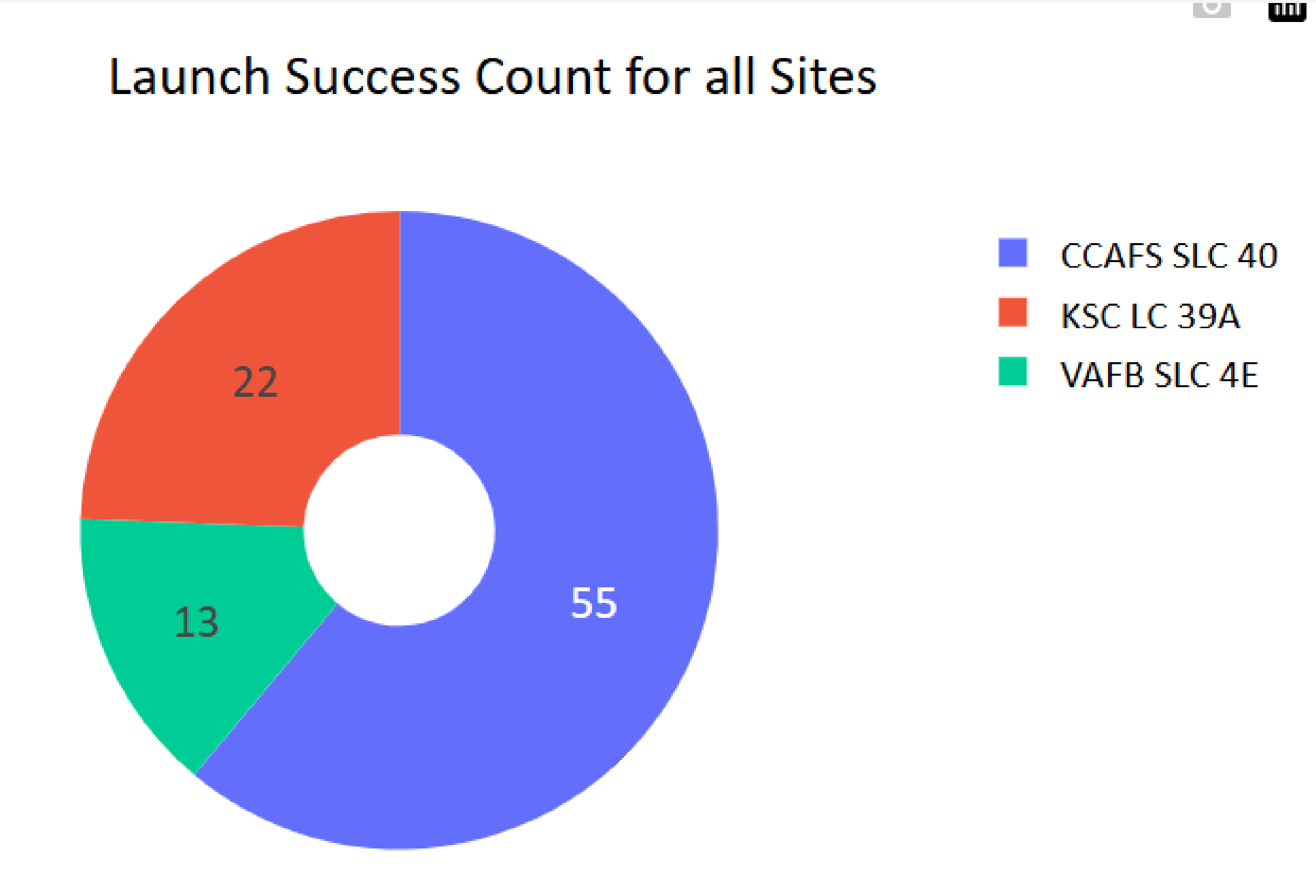
<Folium Map Screenshot 3>

- Folium map showing selected launch site to its proximities such as railway, highway, coastline, with calculated distance.

Section 4

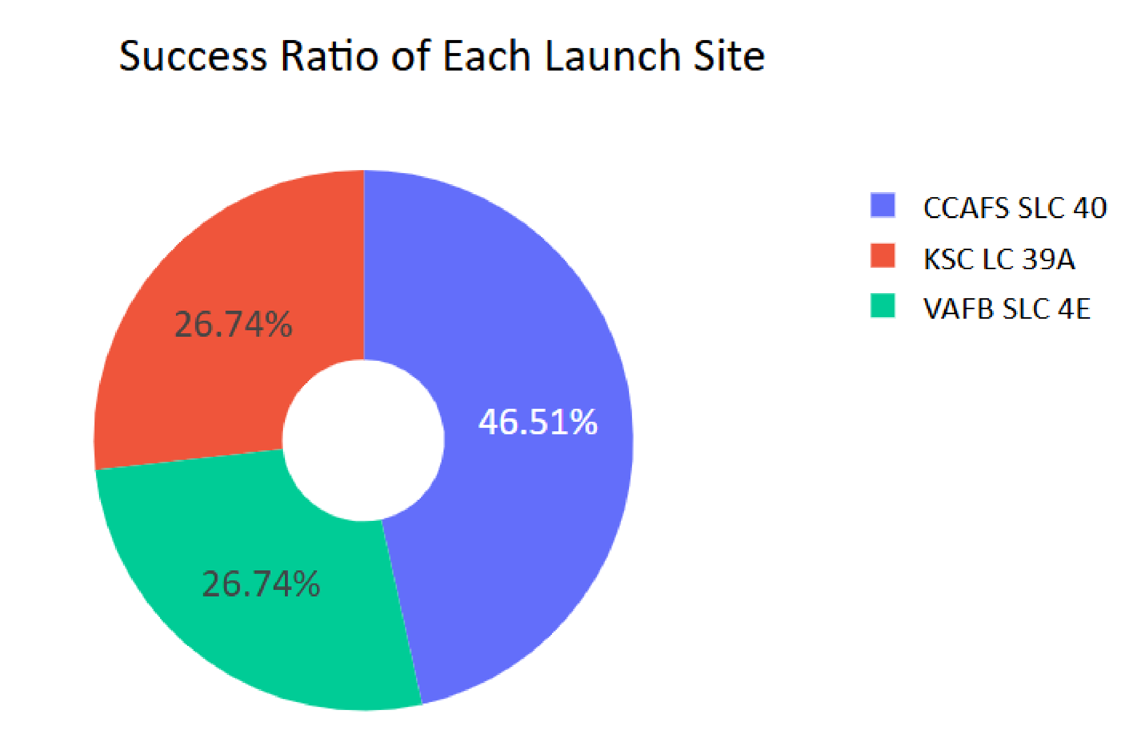
Build a Dashboard with Plotly Dash

Successful Launch Outcomes by Site



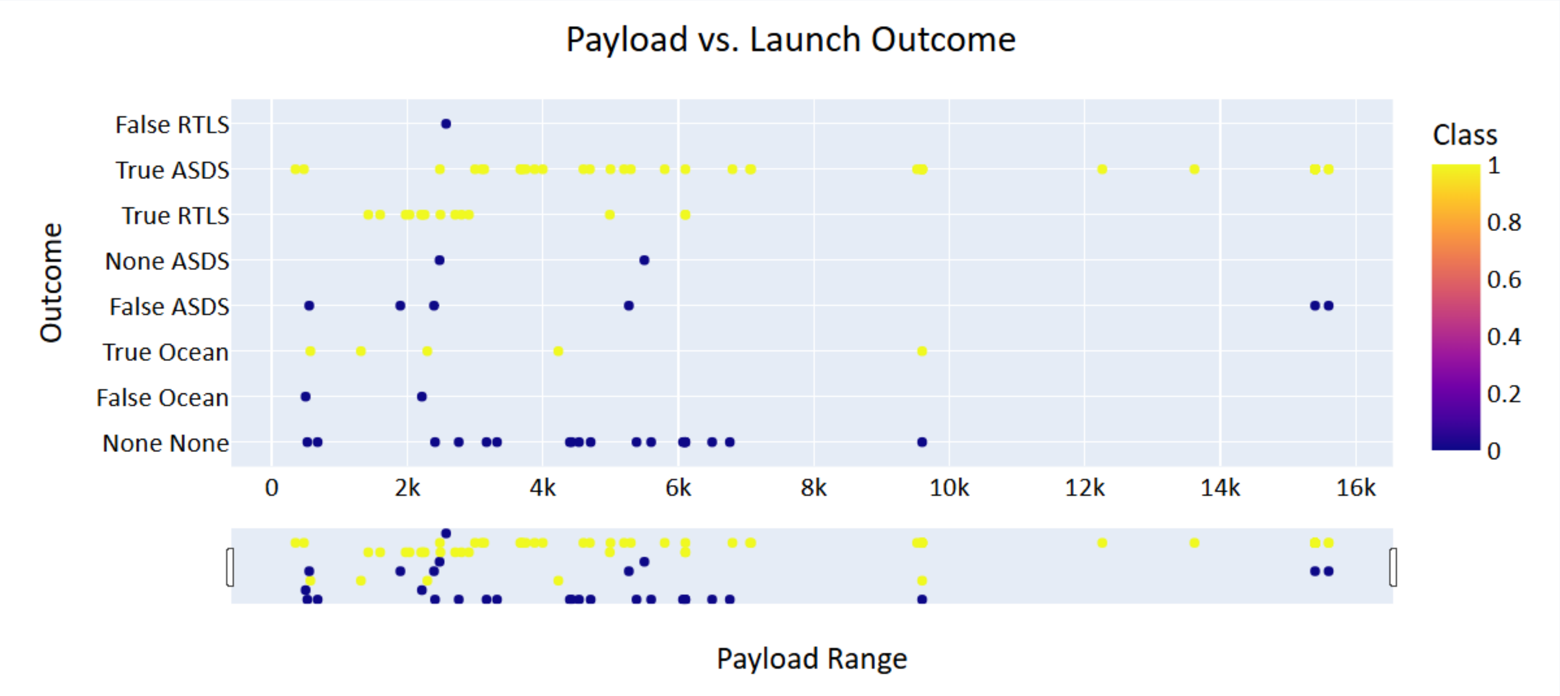
Success Rate by Launch Site

- Piechart for the launch site with highest launch success ratio



Payload vs. Launch Outcome

- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

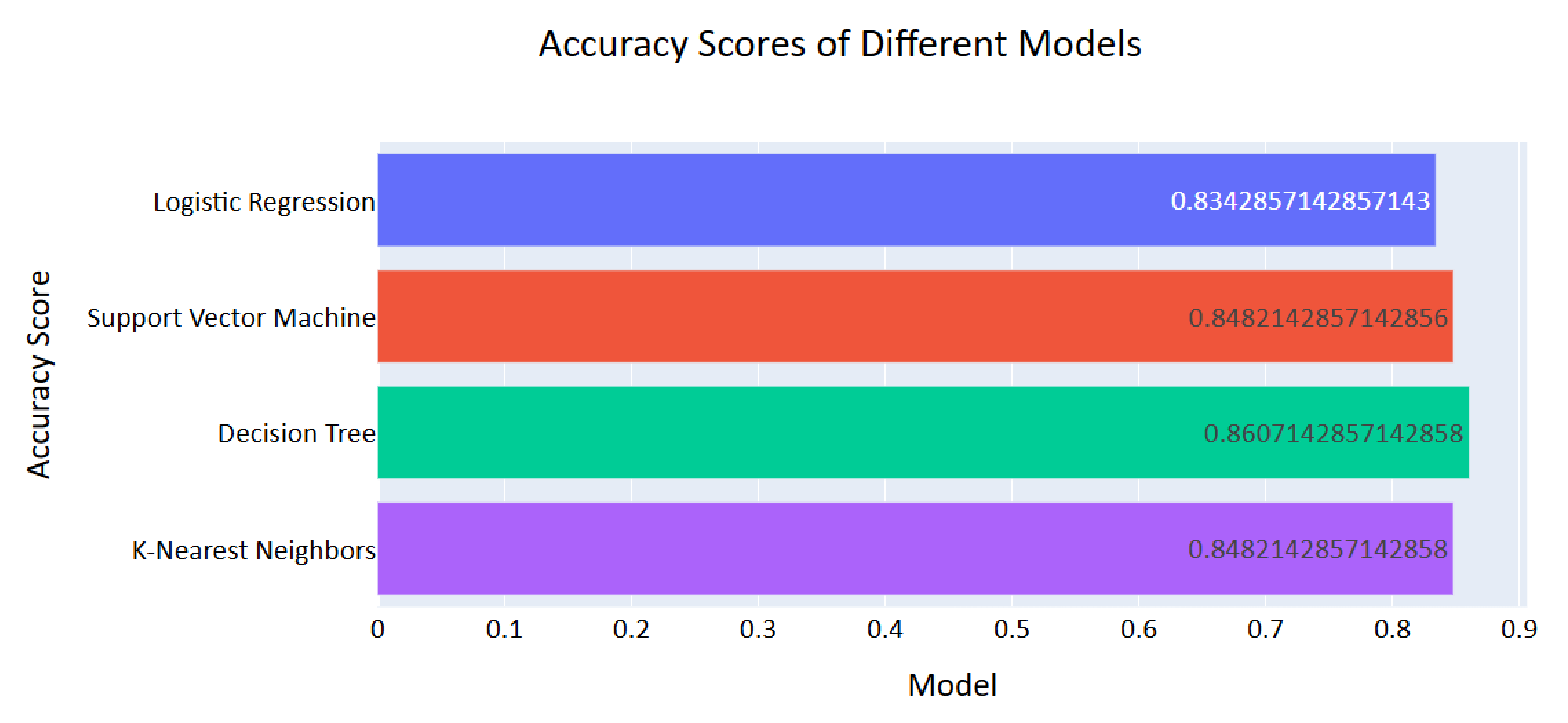


Section 5

Predictive Analysis (Classification)

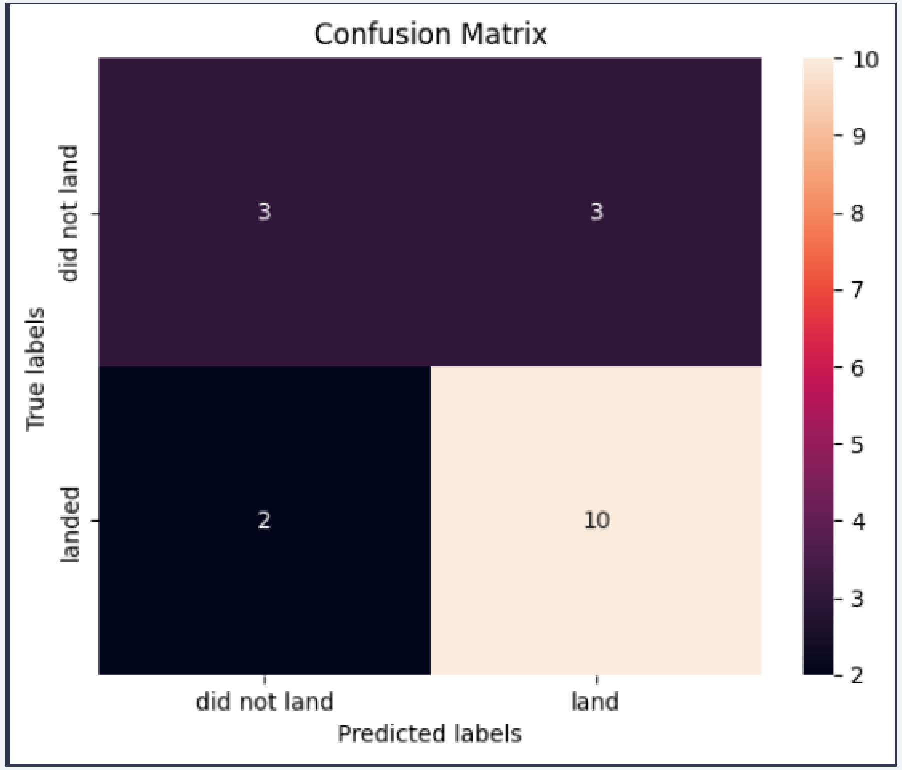
Classification Accuracy

- Highest accuracy model is Decision Tree



Confusion Matrix

- Confusion Matrix for Decision Tree



Conclusions

- Point 1
- Point 2
- Point 3
- Point 4
- …

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

Thank you!