IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

<Alexander S.>
<2024-12-25>

# Outline

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

# Executive Summary

- Summary of methodologies
  - Data collection with API
  - Data Collection with Web Scraping
  - Data Wrangling
  - EDA with SQL
  - EDA through data visualization
  - Data visualization through Folium
  - Data visualization via Plotly Dash

- Summary of all results
  - Success rate of Stage 1 recovery increased over the years
  - Success rate of Stage 1 recovery is highest at max Payload mass
  - A decision tree classifier provided the most accurate estimation of success on the test set

# Introduction

- Project background and context
  - 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. By determining if the first stage will land, the cost of a launch can be estimated. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- Problems you want to find answers
  - Can one predict the success chance of a launch recovery?

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - Request to the SpaceX API
  - Web scraping from Wikipedia
  - Data was cleaned and categorial variables were one-hot encoded
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Testing of several machine learning algorithms such as SVM, Logistic regression, Tree classifier and KNN, with GridSearchCV

# Data Collection

- SpaceX API:
  - Data was collecting through the SpaceX API via get request.
  - The resulting json file was transformed into a pandas dataframe for further processing.
  - Missing data was removed were necessary.
- Web scraping:
  - Extracted a Falcon 9 launch records HTML table from Wikipedia
  - Parsed the table and converted it into a Pandas data frame

# Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts. See previous slide.
- Add the GitHub URL of the completed SpaceX API calls notebook: https://github.com/AScharf1/Capstone-Project/blob/main/SpaxeX%20API_jupyter-labs-spacex-data-collection-api.ipynb

## Examples of functions:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_s
```

We should see that the request was successfull with the 200 status response code

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

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

```
# Calculate the mean value of PayloadMass column
PayloadMass_mean = df['PayloadMass'].mean()

data_falcon9['PayloadMass'].replace(np.NaN, PayloadMass_mean, inplace = True)
data_falcon9.isnull().sum()
```

# Data Collection – Scraping

- Present your data collection with web scraping using key phrases and flowcharts.
  - BeautifulSoup, HTML, finding the tables and its rows, and columns. Created a dictionary, and filled it with the table information of interest. Transformed dict to pandas DataFrame.
- Add the GitHub URL of the completed Web scraping notebook:
  - Hyperlink

## Examples of functions:

```
# use requests.get() method with the provided static_url
response = requests.get(static_url)
html = response.text
```

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

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

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
soup.find_all('table')
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict['Flight No.'].append(flight_number)
            #print(flight_number)
            datatimelist=date_time(row[0])
```

# Data Wrangling

- Conducted exploratory data analysis to identify patterns and determine training labels.
- Analyzed the number of launches at each site and calculated the frequency of different orbit types.
- Created a landing outcome label from the "Outcome" column and exported the results to a CSV file.
- Github link:
  - https://github.com/AScharf1/Capstone-Project/blob/main/Data%20wrangling_labs-jupyter-spacex-Data%20wrangling.ipynb

# EDA with Data Visualization

- Different plots were used such as categorical scatter plots, line chars and bar charts. To visualize the relationship between e.g.:
  - Flight Number and Launch Site
  - Payload Mass and Launch Site
  - Success Rate and Orbit type
- Additionally, categorial variables were one-hot encoded
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose
- https://github.com/AScharf1/Capstone-Project/blob/main/EDA%20with%20data%20visualization.ipynb

# EDA with SQL 1/2

- Using bullet point format, summarize the SQL queries you performed
- %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
- %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
- %sql Select SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer LIKE 'NASA (CRS)’
- %sql Select AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%%'
- %sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Mission_Outcome LIKE 'Success'
- %sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE Payload_MASS__KG_ > 4000 AND Payload_MASS__KG_ < 6000
- %sql SELECT COUNT(Mission_Outcome) FROM SPACEXTABLE GROUP BY Mission_Outcome

- Continues next page

# EDA with SQL 2/2

- Using bullet point format, summarize the SQL queries you performed

- %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Payload_MASS__KG_ = (SELECT MAX(Payload_MASS__KG_) FROM SPACEXTABLE)

- %sql SELECT substr(Date, 6,2) As 'Month', Landing_Outcome, Booster_Version, Launch_Site  FROM SPACEXTABLE WHERE Landing_Outcome LIKE '%Failure (drone ship)%' AND substr(Date,0,5) = '2015'

- %sql Select COUNT(Landing_Outcomes) FROM SPACEXTABLE WHERE strftime('%Y-%m-%d', Date) > 2010-06-04 AND strftime('%Y-%m-%d', Date) < 2017-03-20 ORDER BY Date DESC

- GitHub URL:
  - https://github.com/AScharf1/Capstone-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

# Build an Interactive Map with Folium

- All launch sites were plotted on a map, with additional map elements like markers, circles, and lines used to represent the success or failure of launches at each site.
- The launch outcomes were categorized as either 0 (failure) or 1 (success) and coloured.
- The color-coded marker clusters, allow to identify Launch Sites with high efficiency.
- Distance to railways and coastline were plottet and calculated.

- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose:
    - https://github.com/AScharf1/Capstone-Project/blob/main/lab_jupyter_launch_site_location.ipynb

# Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard: Pie chart and scatter plot.
- Explain why you added those plots and interactions: To see success rate of different Launch Sites in comparison but also based on Payload mass.
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose
  - https://github.com/AScharf1/Capstone-Project/blob/main/spacex_dash_app.py

# Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- Data was splittet into 80% training set and 20% test set. Testing of several machine learning algorithms such as SVM, Logistic regression, Tree classifier and KNN, with GridSearchCV was done.
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose
  - https://github.com/AScharf1/Capstone-Project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

# Results

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

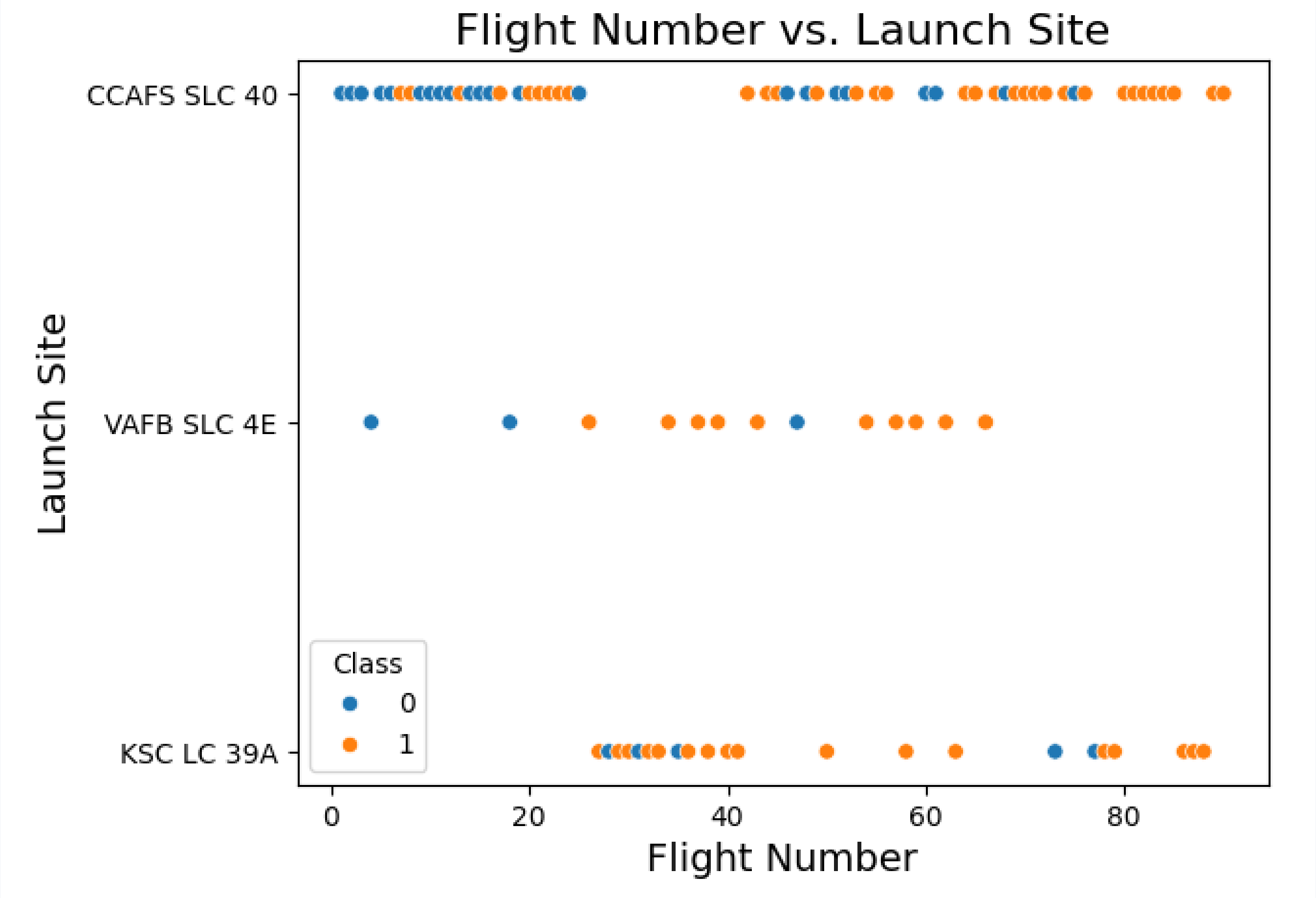
Section 2

# Insights drawn from EDA

# Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site

- Show the screenshot of the scatter plot with explanations
- More success in CCAFS SLC40 and KSC LC 39A with increasing Flight number.

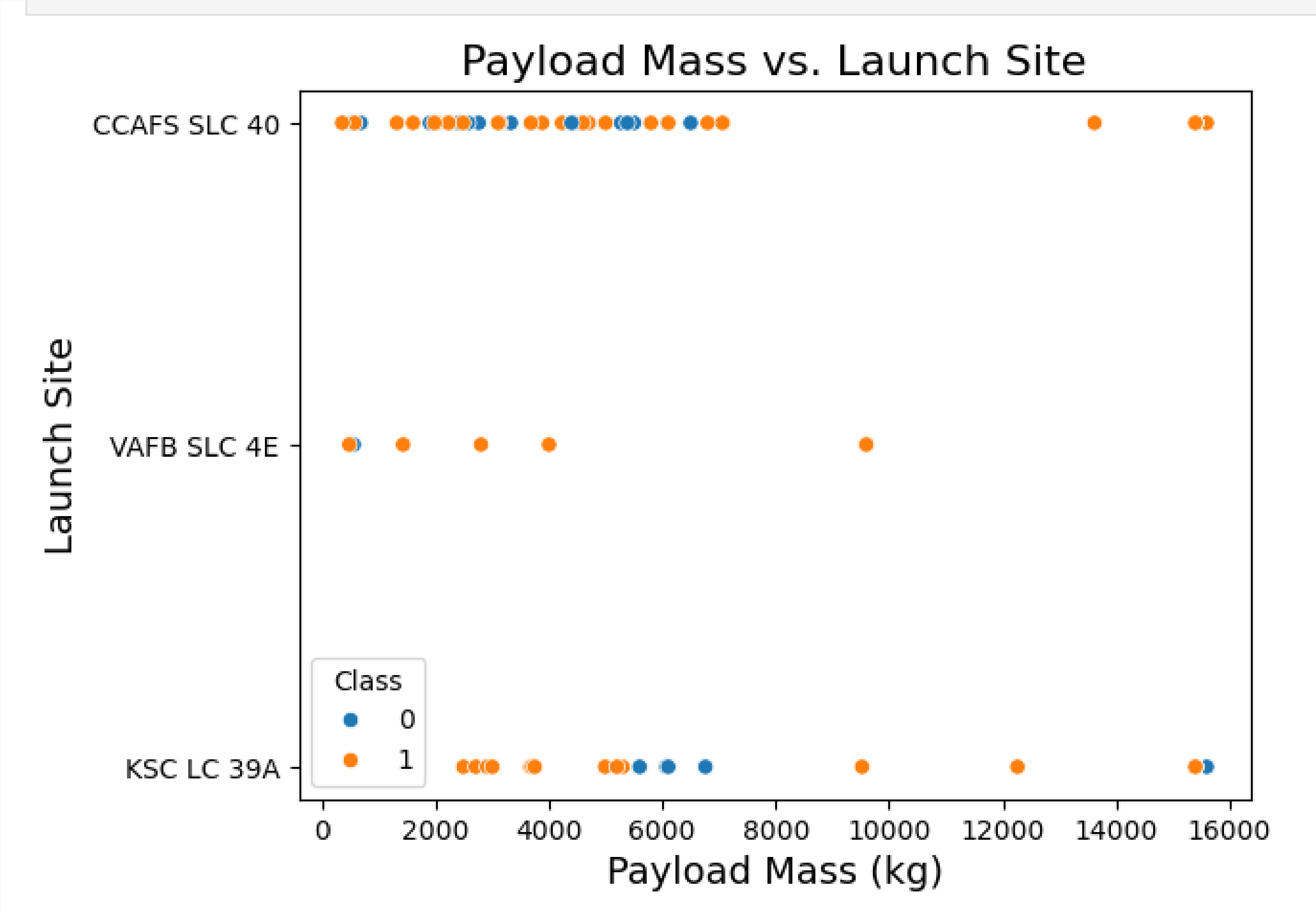


# Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

- Show the screenshot of the scatter plot with explanations:

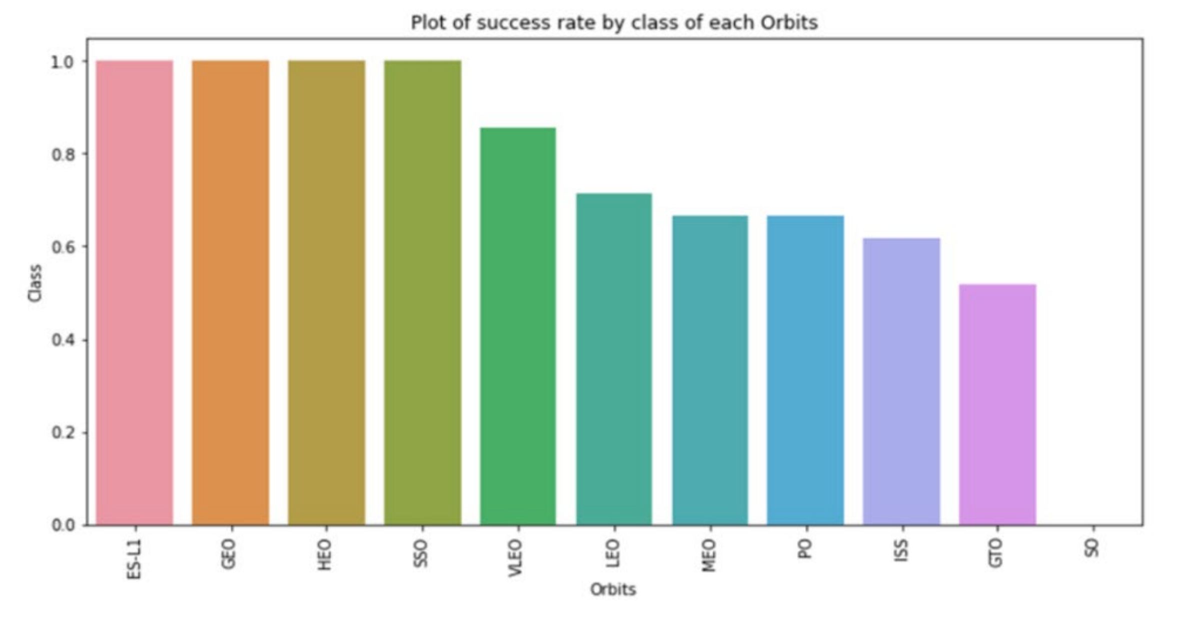
- The VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000)



# Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type

- Show the screenshot of the scatter plot with explanations
- 100% success in the first 4 orbits

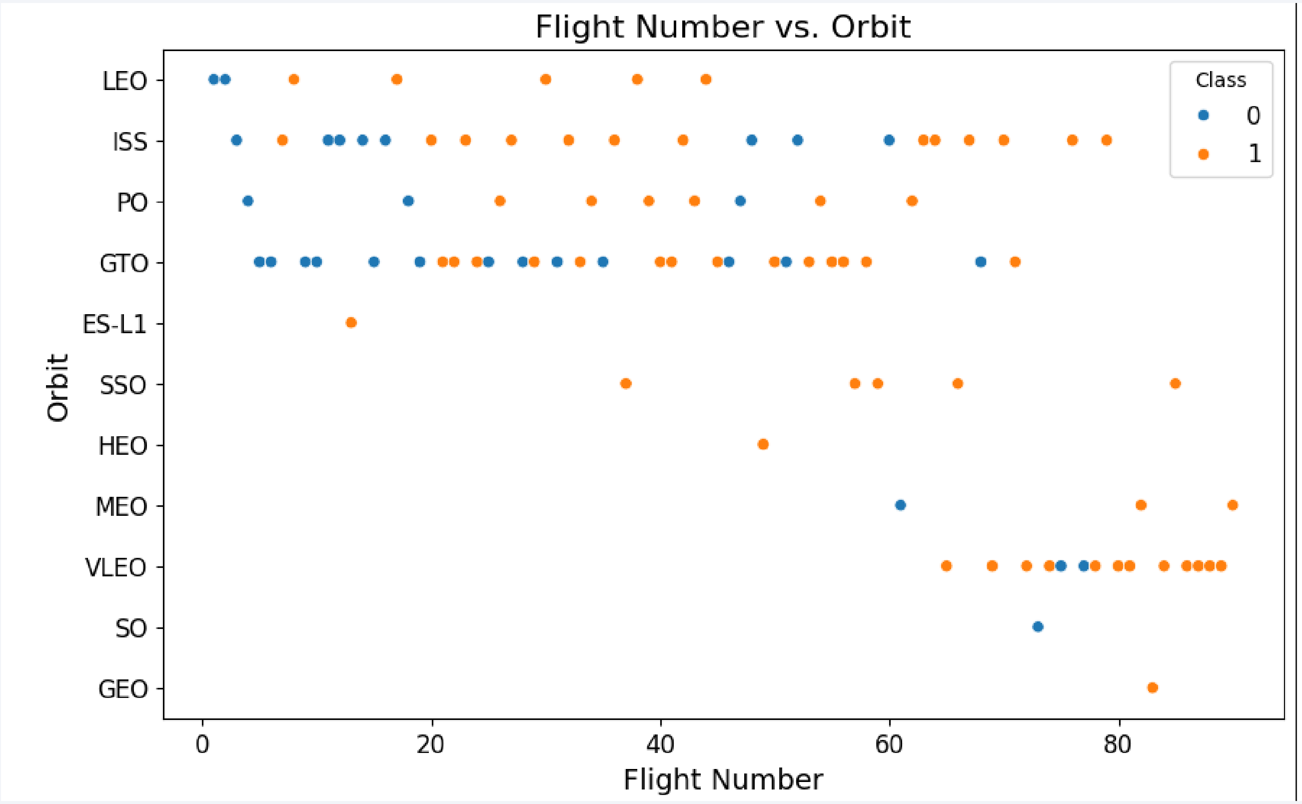


# Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type

- Show the screenshot of the scatter plot with explanations:

- In the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

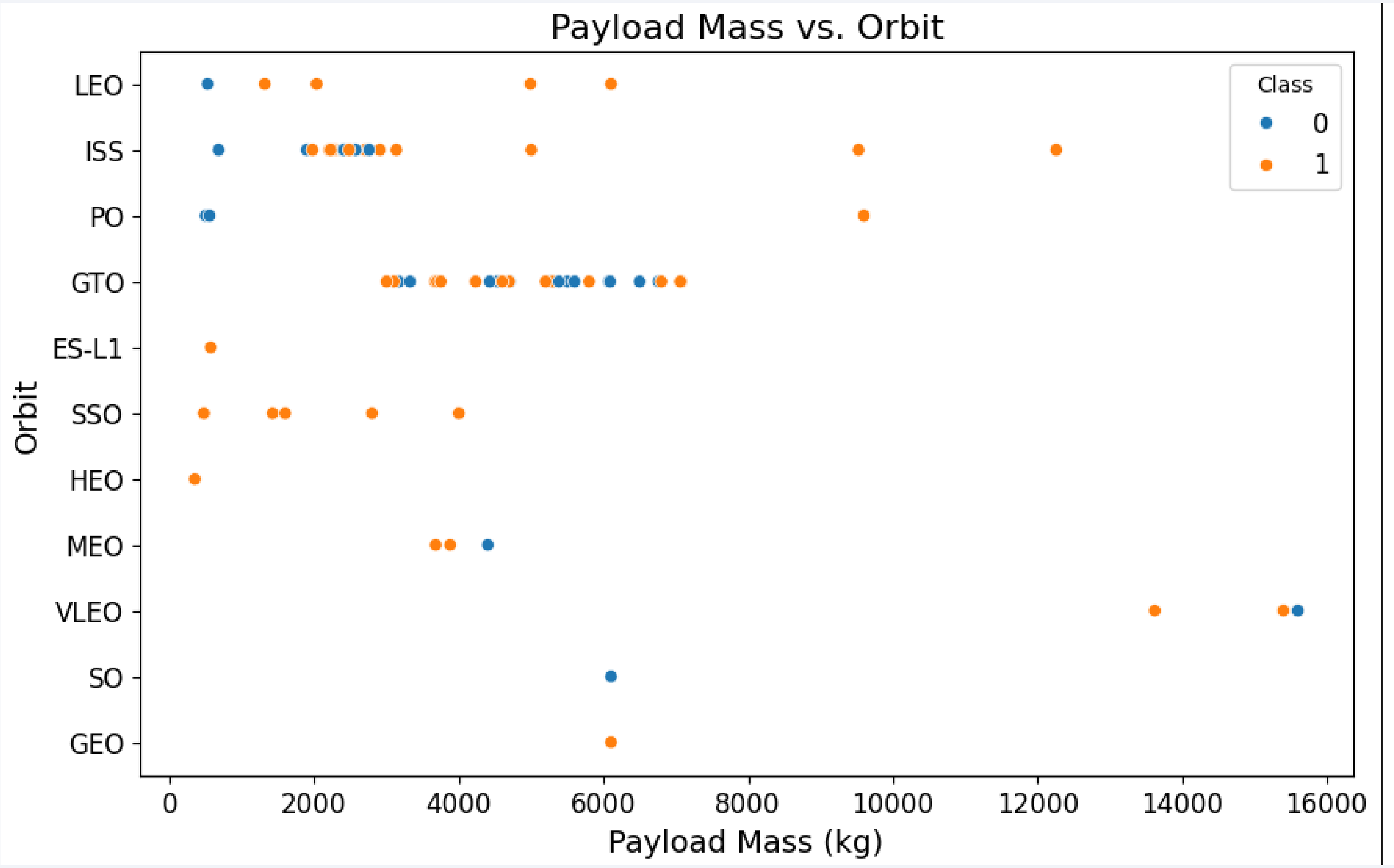


# Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

- Show the screenshot of the scatter plot with explanations

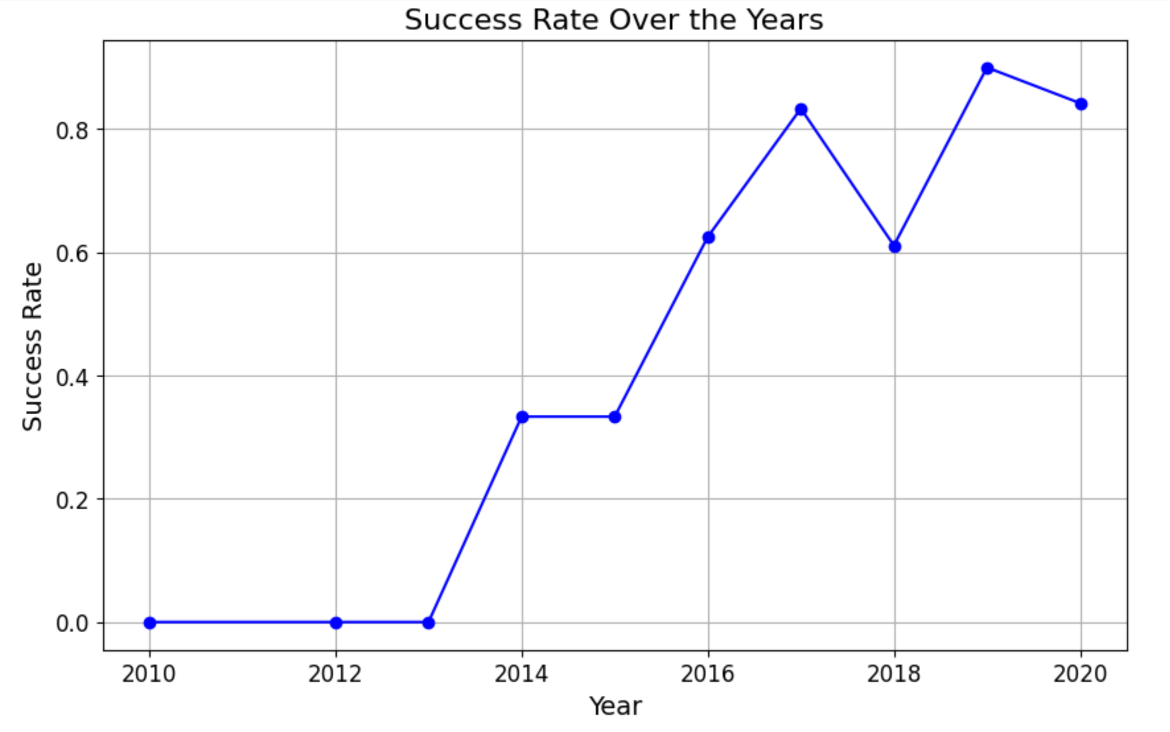
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS. No trend for GTO



# Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations
- Success is getting higher and higher over the years.



# All Launch Site Names

- Find the names of the unique launch sites

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

- Present your query result with a short explanation here:
    - %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
    - “Distinct” takes all unique in the specified column

# Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
- Present your query result with a short explanation here.

| 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

- Calculate the total payload carried by boosters from NASA
- %sql Select SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer LIKE 'NASA (CRS)'
- Present your query result with a short explanation here
- 45596 kg – The total payload.

# Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- %sql Select AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%%'
- Present your query result with a short explanation here
- 2534.67 kg – the average mass

# First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- %sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Mission_Outcome LIKE 'Success'
- Present your query result with a short explanation here
- 2010-06-04 – First time it was a success

# 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 →
- %sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE Payload_MASS__KG_ > 4000 AND Payload_MASS__KG_ < 6000
- Present your query result with a short explanation here. Many booster versions were used in this payload range

| Booster_Version |
|---|
| F9 v1.1 |
| F9 v1.1 B1011 |
| F9 v1.1 B1014 |
| F9 v1.1 B1016 |
| F9 FT B1020 |
| F9 FT B1022 |
| F9 FT B1026 |
| F9 FT B1030 |
| F9 FT B1021.2 |
| F9 FT B1032.1 |
| F9 B4 B1040.1 |
| F9 FT B1031.2 |
| F9 B4 B1043.1 |
| F9 FT B1032.2 |
| F9 B4 B1040.2 |
| F9 B5 B1046.2 |
| F9 B5 B1047.2 |
| F9 B5B1054 |
| F9 B5 B1048.3 |
| F9 B5 B1051.2 |
| F9 B5B1060.1 |
| F9 B5 B1058.2 |
| F9 B5B1062.1 |

# 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

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(Mission_Outcome) FROM SPACEXTABLE GROUP BY Mission_Outcome
```

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

| COUNT(Mission_Outcome) |
|---|
| 1 |
| 98 |
| 1 |
| 1 |

# Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Payload_MASS__KG_ = (SELECT MAX(Payload_MASS__KG_) FRO SPACEXTABLE)
- Present your query result with a short explanation here
- A few booster version have carried the maximum payload.

ut[56]:

| 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 failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- %sql SELECT substr(Date, 6,2) As 'Month', Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE Landing_Outcome LIKE '%Failure (drone ship)%' AND substr(Date,0,5) = '2015'
- Present your query result with a short explanation here. 2 Failures in 2015 for drone ship

| 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

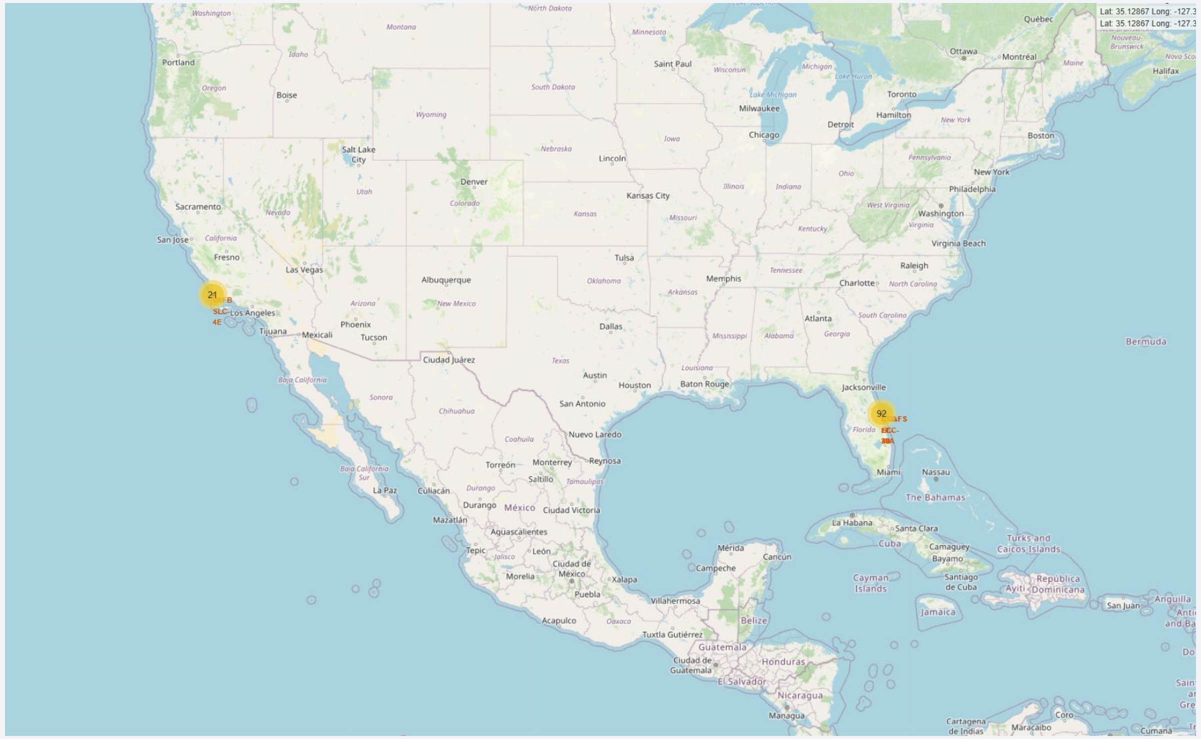
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- %sql Select COUNT(Landing_Outcome) FROM SPACEXTABLE WHERE strftime('%Y-%m-%d', Date) > 2010-06-04 AND strftime('%Y-%m-%d', Date) < 2017-03-20 ORDER BY Date DESC
- Present your query result with a short explanation here

Section 3

# Launch Sites Proximities Analysis

# <Folium Map – Global overview>

- Explain the important elements and findings on the screenshot
- There are Launch Sites on the West and East Coast of the US



# <Folium Map – Launch Site Success>

- Explain the important elements and findings on the screenshot
- VAFB SLC-4E exhibits relative many failures.

# <Folium Map – Proximity to Road>

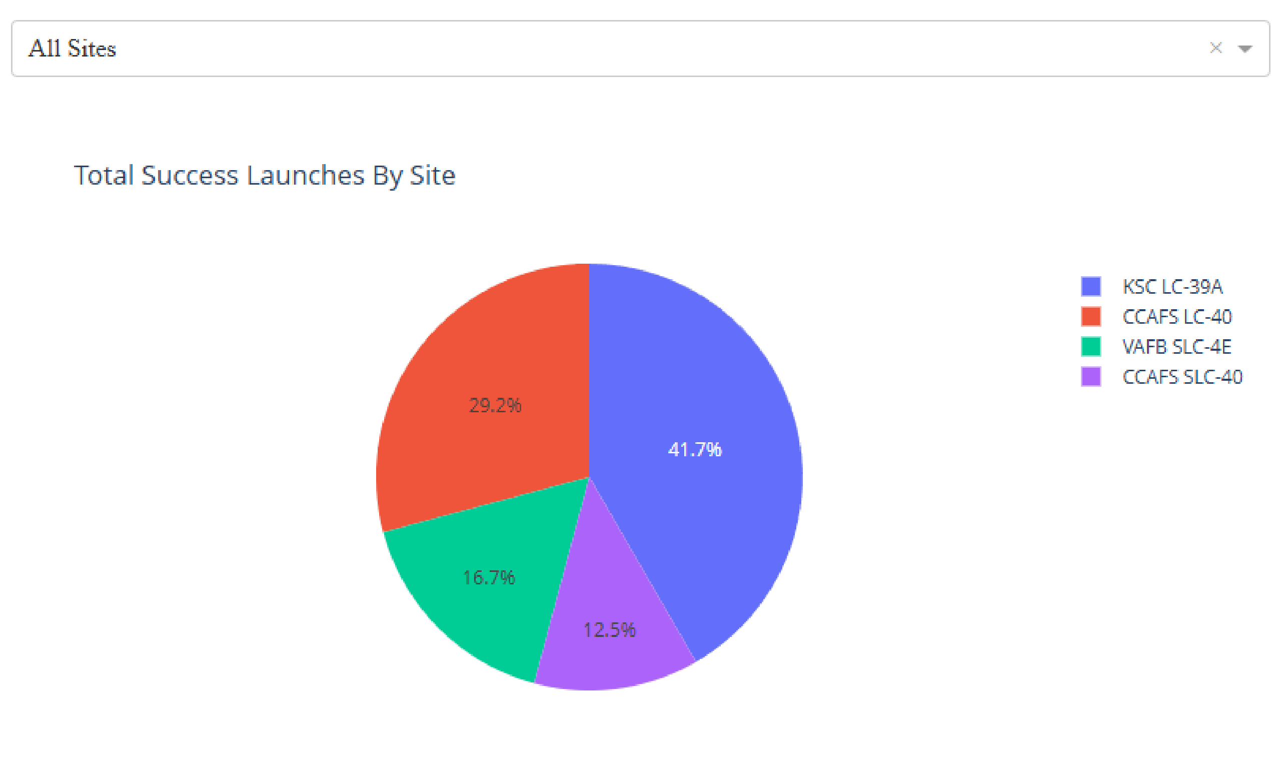
- Explain the important elements and findings on the screenshot
- There is a highway close by. The distance is 0.63 km

Section 4

# Build a Dashboard with Plotly Dash

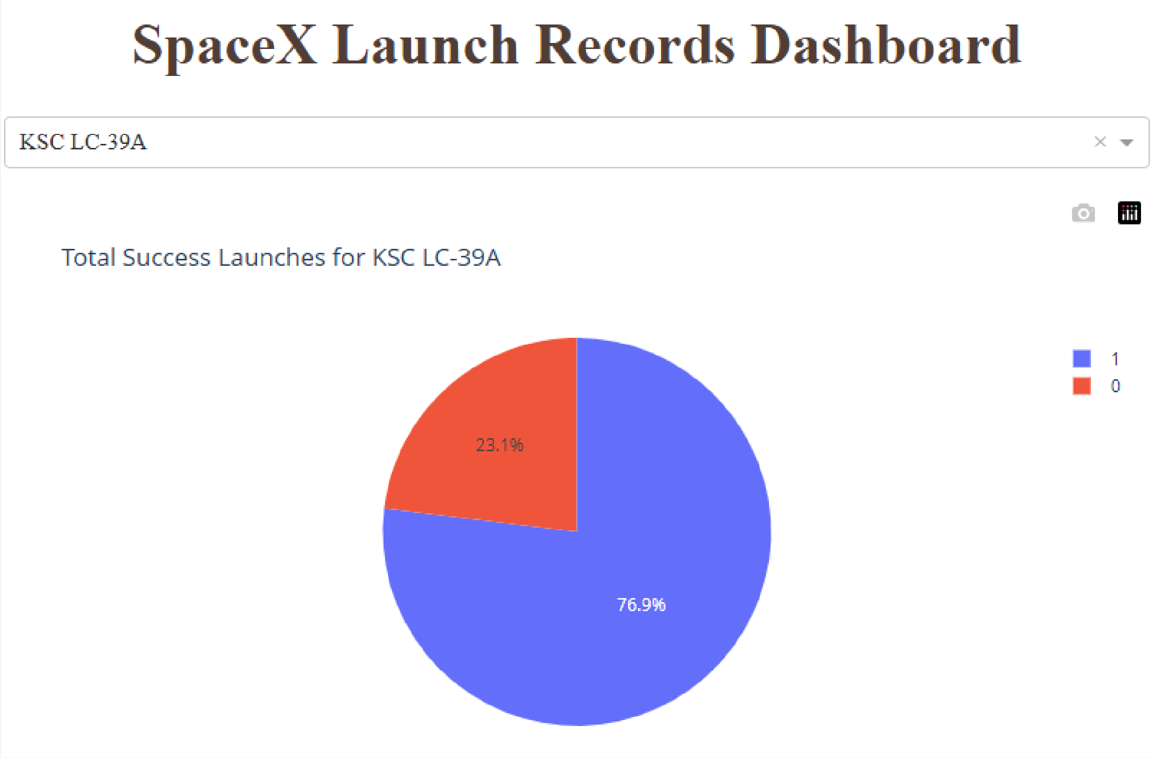
# <Dashboard - Launch success for all sites>

- Explain the important elements and findings on the screenshot
- Launch success is highes at KSC LC-39A



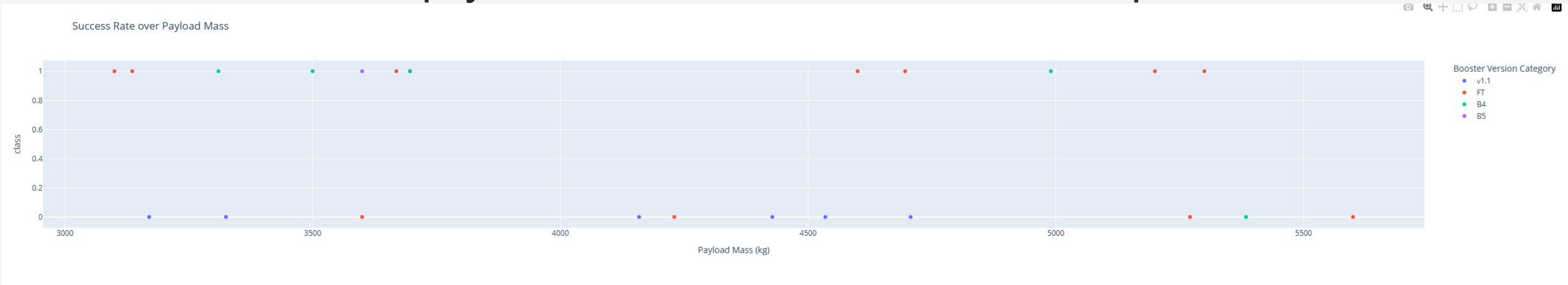
# <Dashboard – Launch Success for KSC LC-39A>

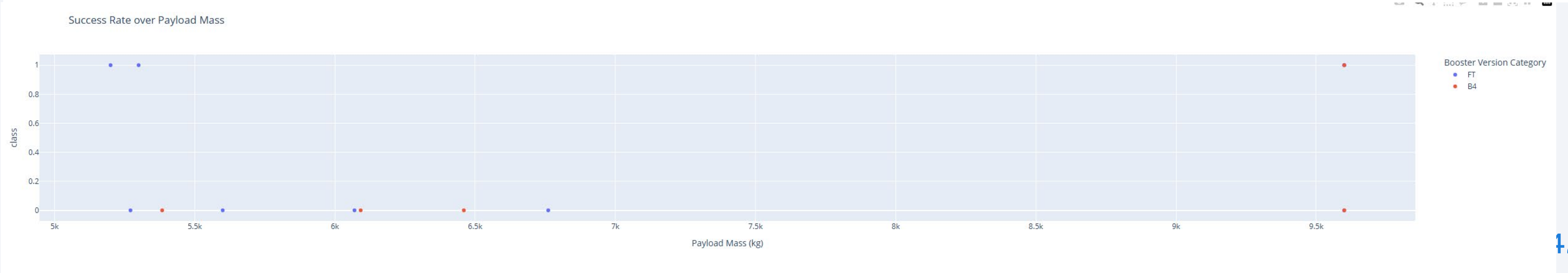
- Explain the important elements and findings on the screenshot → 76.9% Success rate



# <Dashboard  - Payload Mass comparison>

- Replace <Dashboard screenshot 3> title with an appropriate title
- Low success rate for payload mass between 5k and 10k compared to 3k and 6k



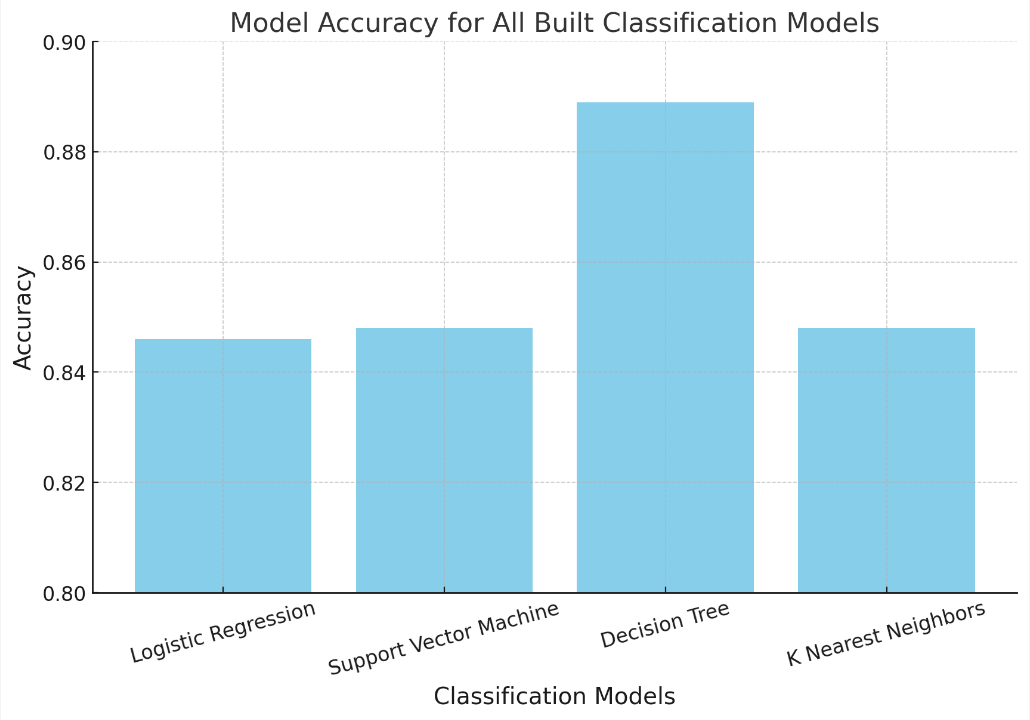



Section 5

# Predictive Analysis (Classification)

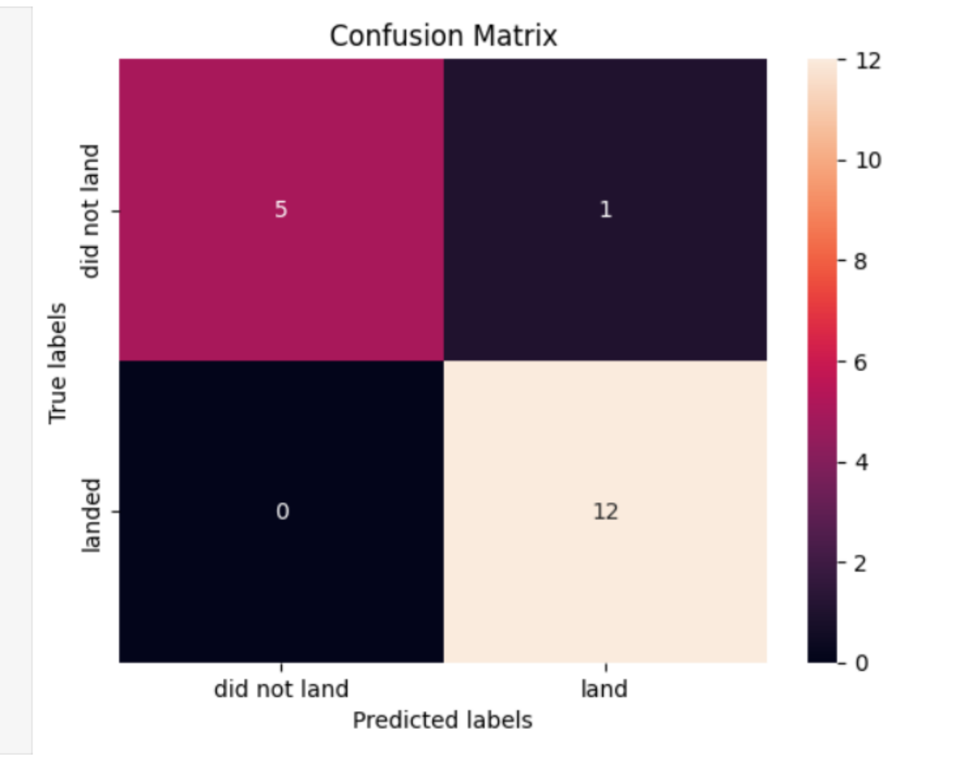
# Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart
- Find which model has the highest classification accuracy: Decision tree



# Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation. From the decision tree classifier:
- 5 True negatives, 1 false negative,
  0 false positives  and 12 true positives
- Very good



# Conclusions

- Success rate of Stage 1 recovery increased over the years
- Success rate of Stage 1 recovery is highest at max Payload mass
- A decision tree classifier provided the most accurate estimation of success on the test set
- Launch Sites close to the equator and in proximity of infrastructure are chosen

# 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!