IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

James Estell
October 16, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection w/ API and Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis
  - Visual Analytics and Dashboard Build
  - Predictive Analytics
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive Analytics screenshots
  - Predictive Analysis results

Introduction

- SpaceX is a spacecraft manufacturer and launch service provider. SpaceX is much cheaper than other launch service providers because it can often reuse of the first stage of its rockets if it successfully lands. So if we can predict whether or not a rocket's first stage will land we can more accurately predict the cost of a launch.
- So we want to look gather data from SpaceX's previous launches. Analyze the data to determine trends correlated to whether or not the first stage of a rocket will successfully land. Finally we want to build a predictive model that we can use to more accurately determine the likelihood of a successful landing of the first stage of a SpaceX rocket launch

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was collected through accessing SpaceX's API and webscrapping from SpaceX's Wikipedia page
- Perform data wrangling
  - One-hot encoding used to create a binary classification for landing outcomes
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build, tune, and evaluate classification models

Data Collection

- SpaceX API
  - Use the Requests library to call the SpaceX API
  - From the API calls create a Pandas dataframe with data regarding SpaceX Falcon 9 launches
  - Clean up missing values
- Wikipedia Web Scraping
  - BeautifulSoup library used to scrap the list of historical Falcon 9 SpaceX launches including their booster landing outcomes
  - Create Pandas dataframe from scrapped list

Data Collection – SpaceX API

- Use a get request to retrieve the launch data from SpaceX REST API
- Then the json containing the launch data convert to a dataframe
- Finally, do some basic data wrangling to clean up missing values and filter for Falcon 9 launches

Github link: https://github.com/JamesEstell/IBM-Data-Science-Capstone/blob/main/Data%20Collection%20API.ipynb

1. Use a get request to retrieve launch data from the SpaceX API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

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

2. Use the json_normalize method to convert the json results from the get request into a dataframe

```
# Use json_normalize meethod to convert the json result into a dataframe

response2 = requests.get(static_json_url)
results = response2.json()
data = pd.json_normalize(results)
```

3. Filter for only Falcon 9 rockets and clean up missing values in the dataframe

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df_launch[df_launch['BoosterVersion']!='Falcon 1']
data_falcon9
```

```
# Calculate the mean value of PayloadMass column
PayloadMassMean = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, PayloadMassMean, inplace=True)
data_falcon9
```

Data Collection - Scraping

- Start by using a get request and beautifulsoup to create an html parser of the Falcon launch data from wikipedia
- Then create a dictionary and parse through the data in the html table map to the dictionary
- Finally convert the dictionary to a dataframe

Github Link: https://github.com/JamesEstell/IBM-Data-Science-Capstone/blob/main/Web%20scraping.ipynb

1. Use a get request to retrieve Falcon 9 and Falcon Heavy launch data from wikipedia and beautifulsoup to create a html parser for the results.

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy
```

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

response = requests.get(static_url).text
```

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

2. Extract the column names from the table

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

headers = first_launch_table.find_all('th')
extract_column_from_header(headers[0])

for col in first_launch_table.find_all('th'):
    col_name=extract_column_from_header(col)

    if col_name is not None and len(col_name) > 0:
        column_names.append(col_name)
```

3. Create a dictionary object to hold the launch data and use a for loop to parse through the html table and add the data from the respective column to our dictionary.

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

4. Finally, convert your dictionary into a dataframe.

```
df=pd.DataFrame(launch_dict)
```

Data Wrangling

- We did some basic exploratory data analysis to better understand and see trends in the launch sites, orbits, and landing outcomes.
- Then we created a binary classification for landing outcomes to determine success of each landing. This classification will be used for further exploratory data analysis and predictive analysis of factors leading to a successful SpaceX launch and landing.

Github Link: https://github.com/JamesEstell/IBM-Data-Science-Capstone/blob/main/Data%20wrangling.ipynb

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

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

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
```

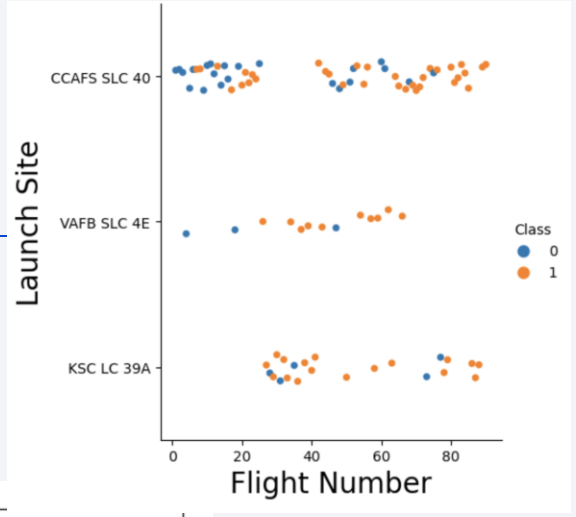
```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

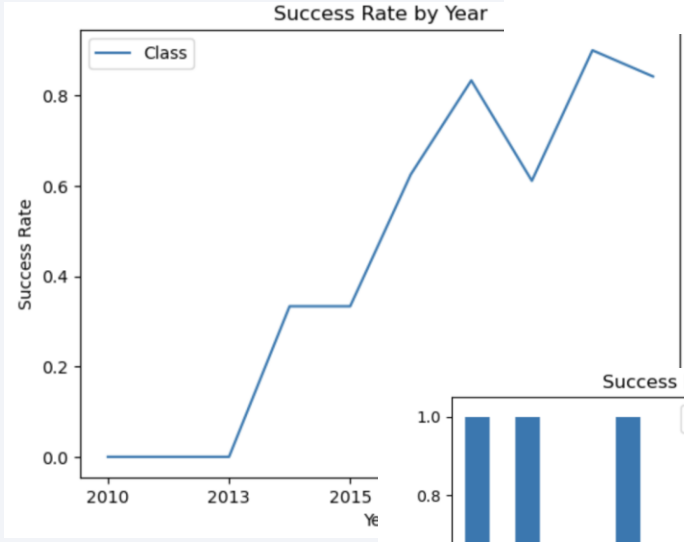
```
True ASDS       41
None None       19
True RTLS       14
False ASDS       6
True Ocean       5
False Ocean      2
None ASDS        2
False RTLS       1
```

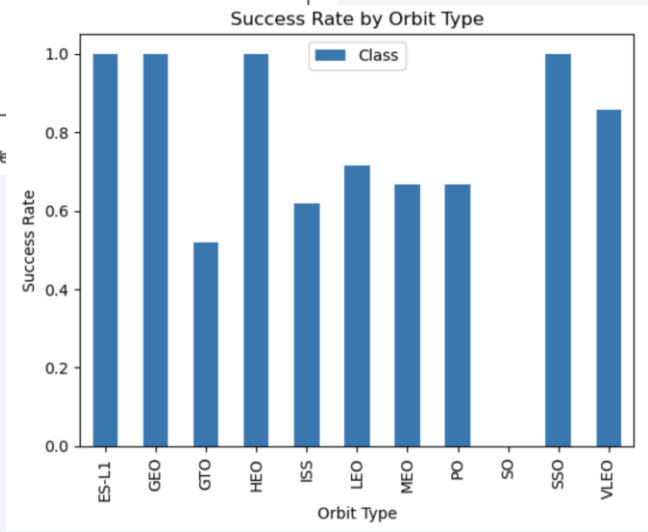
EDA with Data Visualization

- We continued our exploratory data analysis by using data visualizations to determine the relationship between certain variable and landing outcomes
- Some of these included the looking at the impacts of launch site, payload mass, orbit, and year of launch on the landing outcomes

Ref: https://github.com/JamesEstell/IBM-Data-Science-Capstone/blob/main/EDA%20Dataviz.ipynb







EDA with SQL

- Next we used SQL to continue doing additional exploratory data analysis. Below are some of the queries we reviewed:
  - All the unique launch sites
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The date of the first successful landing on a ground pad
  - The boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - The total number of missions by landing outcome
  - The names of the booster versions which have carried the maximum payload mass

Github Link: https://github.com/JamesEstell/IBM-Data-Science-Capstone/blob/main/EDA%20SQL.ipynb

Build an Interactive Map with Folium

- Next we used folium to create interactive maps of the launch data
- First we created circles on the map to identify the different launch sites
- Then we created green and red markers at each location corresponding to successful and failed launches that turned into to clusters as you zoomed out to display the total number of launches at a location
- Finally, we added lines that calculated the distance between the launch sites and key proximate features such as railroads, highways, and coastlines

Github Link: https://github.com/JamesEstell/IBM-Data-Science-Capstone/blob/main/Launch%20Sites%20Locations%20Analysis%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Next we used plotly to create an interactive dashboard to further visualize our data
- We created a pie chart to display the total number of launches by launch site
- We also created a scatter plot to display the impact that payload mass has on the landing outcome of different booster versions
- Additionally, we added a dropdown to analyze the data by site or adjust the range of the payload mass displayed

Github link: https://github.com/JamesEstell/IBM-Data-Science-Capstone/blob/main/spacex%20dashboard%20app.py

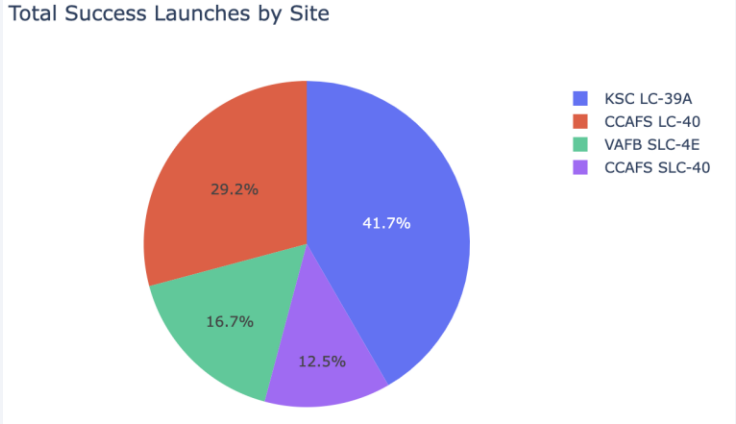
Predictive Analysis (Classification)

- Finally, we began to train, test, evaluate, improve, and compare different predictive models for determining landing outcome success
- We started by splitting our dataset into test and train sets
- Because we are trying to predict a binary classification we decided to train and evalute the following models: ogistic regression, support vector machine, decision tree classifier, k-nearest neighbors
- We create each model using the training set to train them and gridsearch to determine the best hyperparameters for each
- Then using the test set we tested each model and plotted the results on a confusion matrix to evaluate
- Lastly, compared the best scores from each model to determine the preferred predictive model for determining landing outcomes for SpaceX rocket launches

Github Link: https://github.com/JamesEstell/IBM-Data-Science-Capstone/blob/main/Machine%20Learning%20Prediction.ipynb

Results

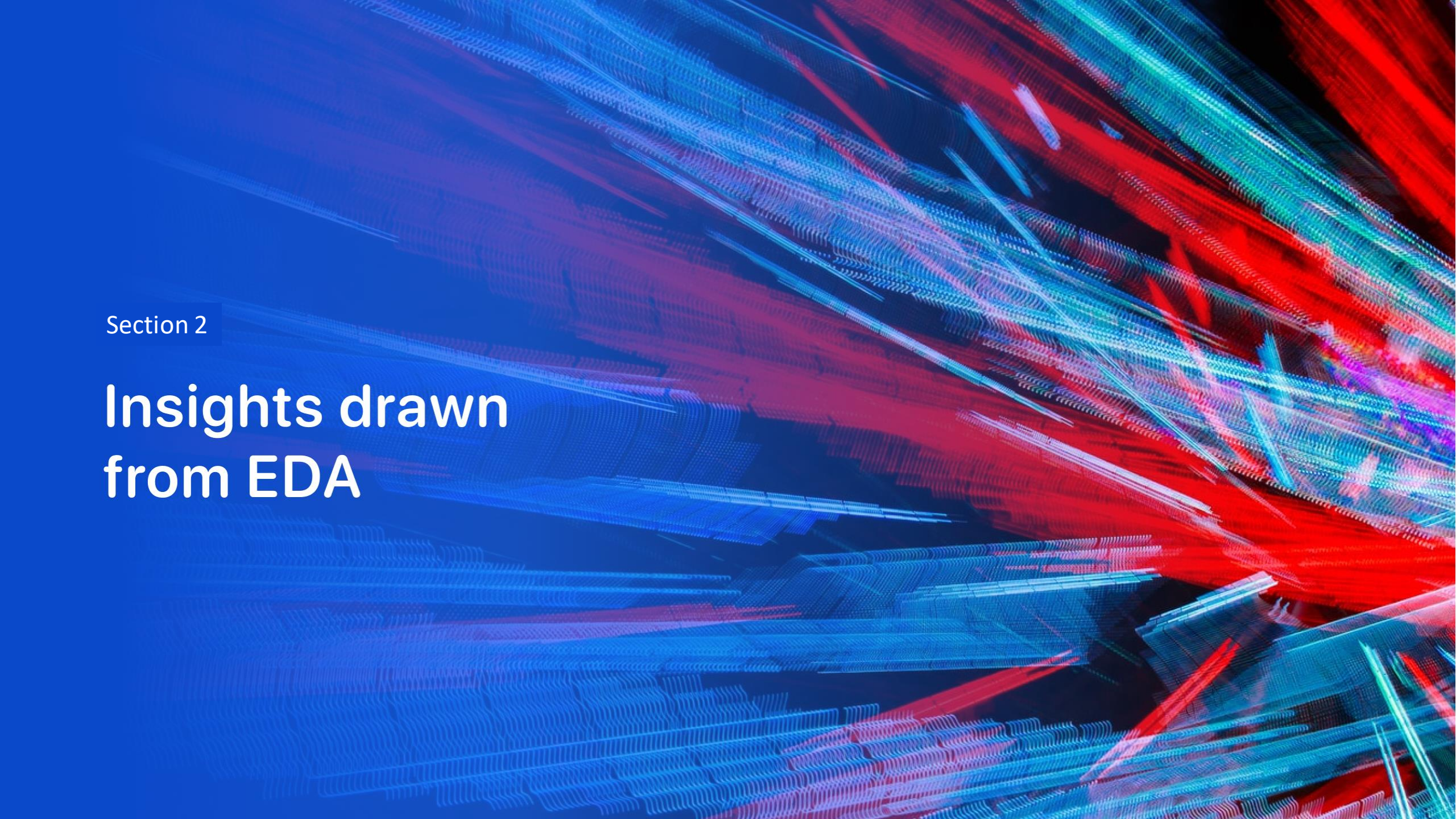
- Our initial exploratory data analysis determined that there positive correlation between landing outcome and orbit (specifically ES-L1, GEO, SSO, and HEO) and that more recent launches have a much higher landing success rate
- Interactive analytics demo in screenshot:



- Our predictive analysis results determined that our logistic regression, support vector model, and k-nearest neighbors models all were tied for our best predictive model for determining the success or failure of the landing of a SpaceX launch

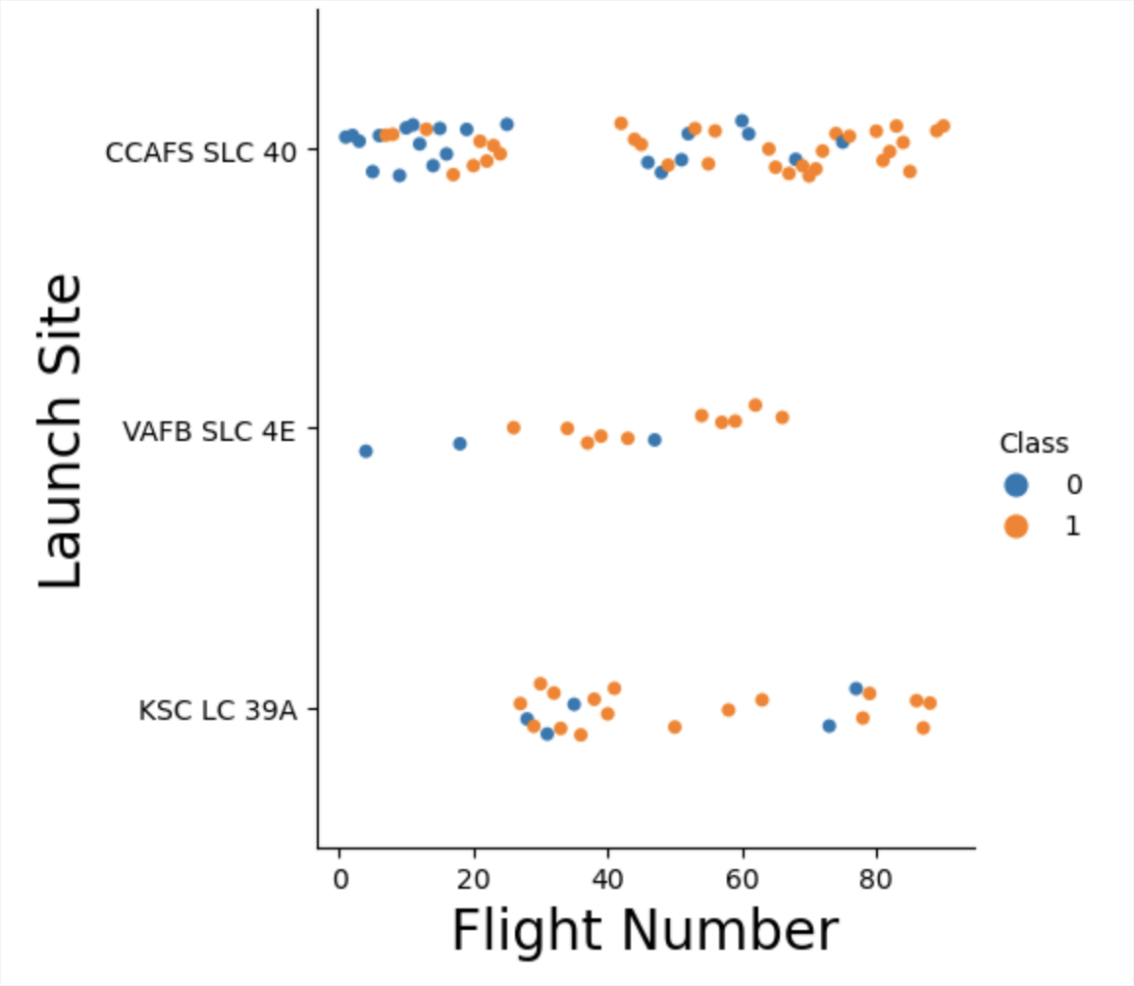
Section 2

Insights drawn from EDA



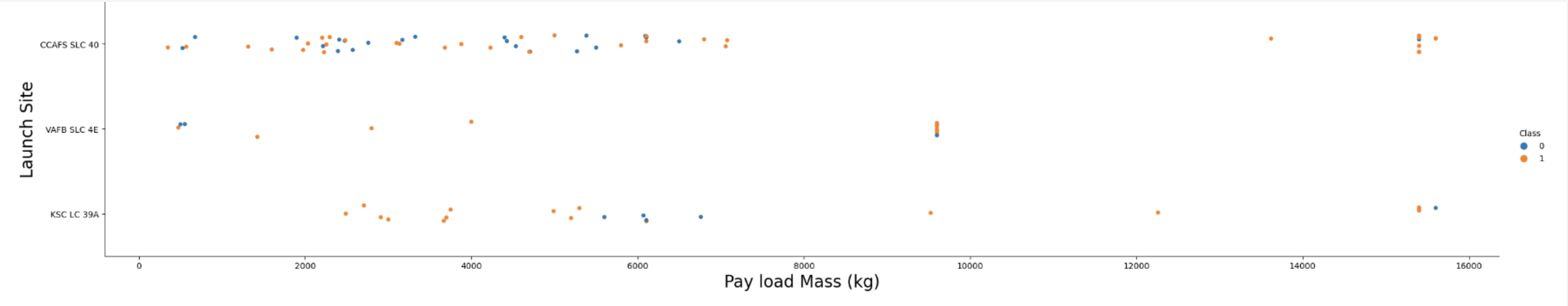
Flight Number vs. Launch Site

- Scatter plot of Flight Number vs Launch Site with the color of the points based on the success of the landing outcome



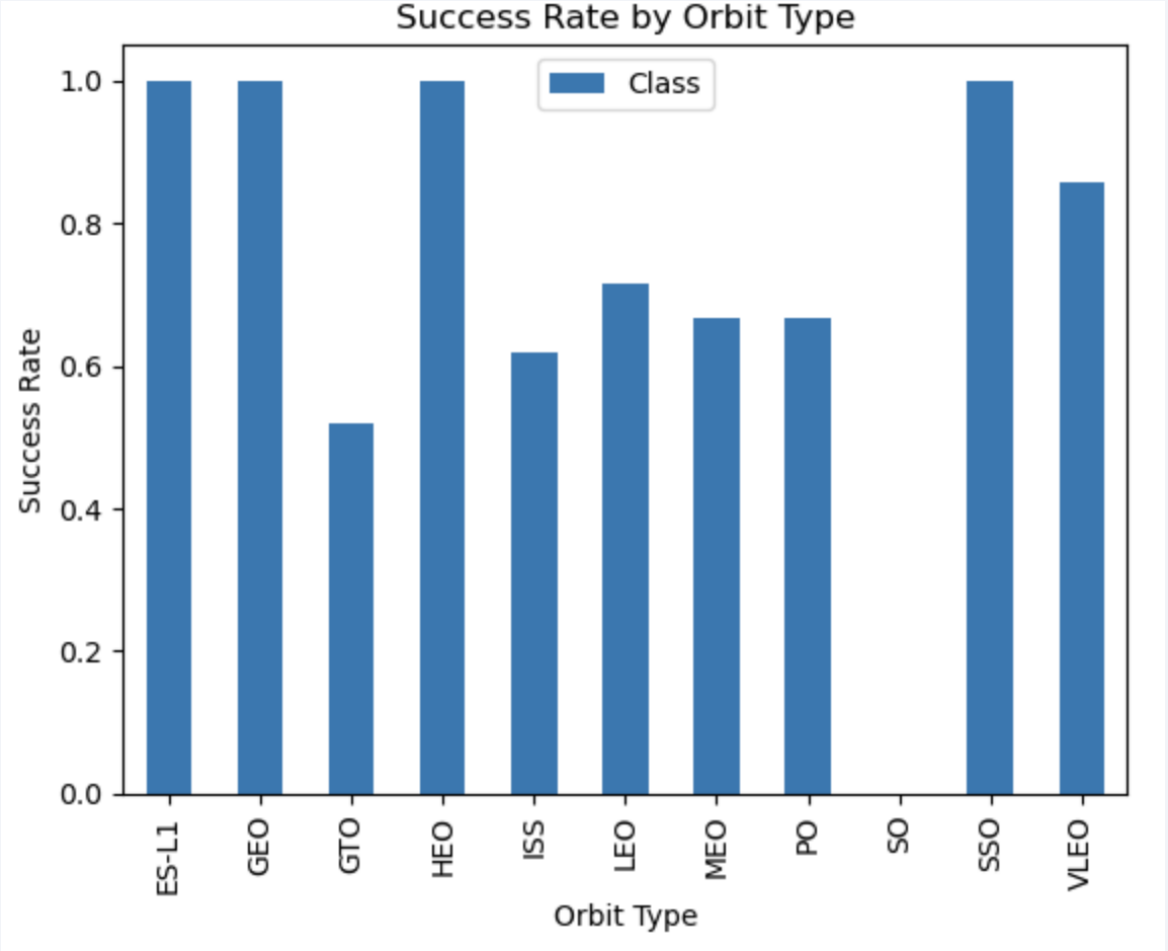
Payload vs. Launch Site

- Scatter plot of Payload Mass (kg) vs. Launch Site with the color of the points based on the success of the landing outcome



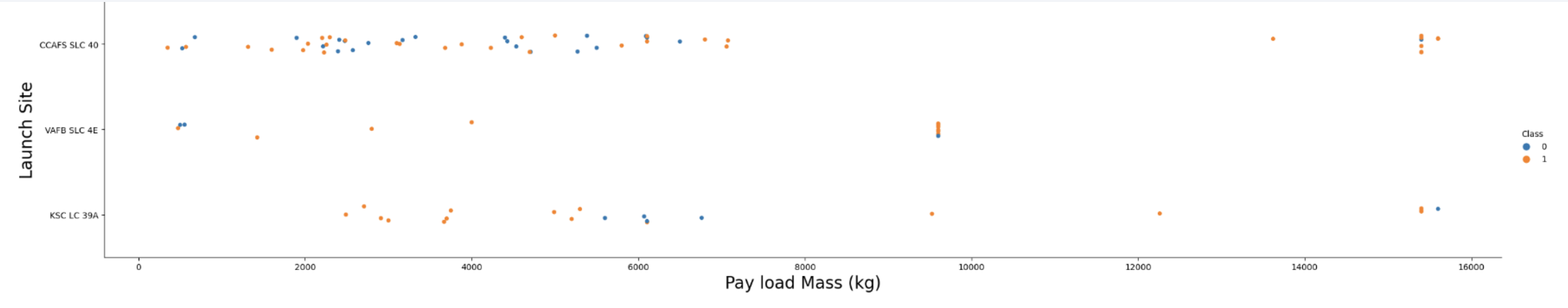
Success Rate vs. Orbit Type

- Bar chart of the success rate of each Orbit Type



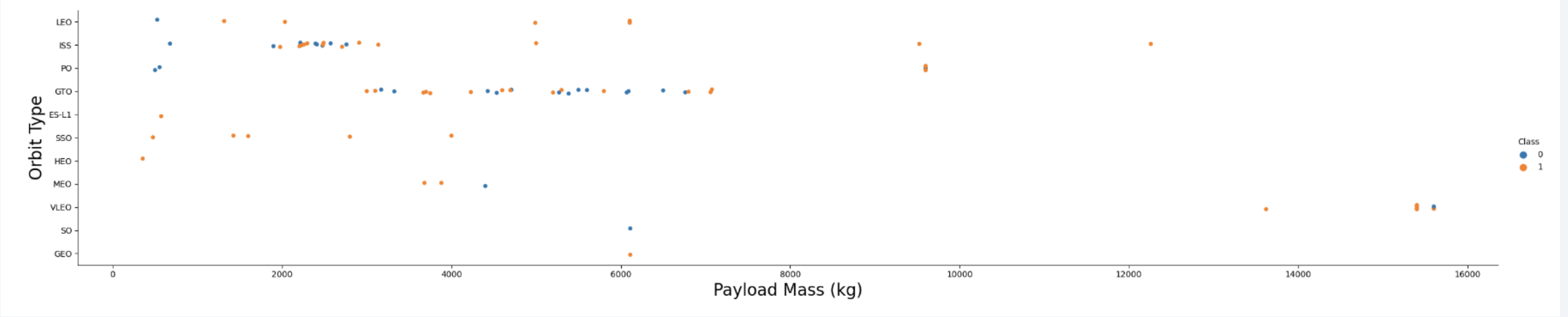
Flight Number vs. Orbit Type

- Scatter point of Flight number vs. Orbit type with the color of the point based on the success of the landing



Payload vs. Orbit Type

- Scatter point of Payload Mass (kg) vs. Orbit Type with the color of the point based on the success of the landing



Launch Success Yearly Trend

- Trend line showing the showing the landing outcome success rate year over year

All Launch Site Names

- We queried the dataset to list the unique names of all the launch sites

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

Launch Site Names Begin with 'CCA'

- To better understand the dataset we queried 5 records where launch sites begin with `CCA`

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (p
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 (p
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

Total Payload Mass

- Next we queried the dataset to calculate the total payload carried by boosters from NASA

NASA Total Payload Mass (kg)
107010

Average Payload Mass by F9 v1.1

- Then we queried the dataset to find the average payload mass (kg) carried by booster version F9 v1.1

F9 v1.1 Avg Total Payload Mass (kg)
2534.6666666666665

First Successful Ground Landing Date

- Next, we found the date of the first successful landing on ground pad

First Successful Landing Outcome in Ground Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Then we found the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Boosters w/ Payload Mass 4000-6000kg and Successful Drone Ship Landing
F9 FT B1021.2
F9 FT B1031.2
F9 FT B1022
F9 FT B1026

Total Number of Successful and Failure Mission Outcomes

- Then, we queried the data set for the total number missions for each landing outcome

Landing_Outcome	qty
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1
No attempt	1

Boosters Carried Maximum Payload

- Then we queried the data set for the names of the booster which have carried the maximum payload mass

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

2015 Launch Records

- Then we queried the data set for the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

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

- Finally we queried the data set for the landing outcomes between the date 2010-06-04 and 2017-03-20, sorted in descending order by total amount

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


Section 3
Launch Sites
Proximities Analysis

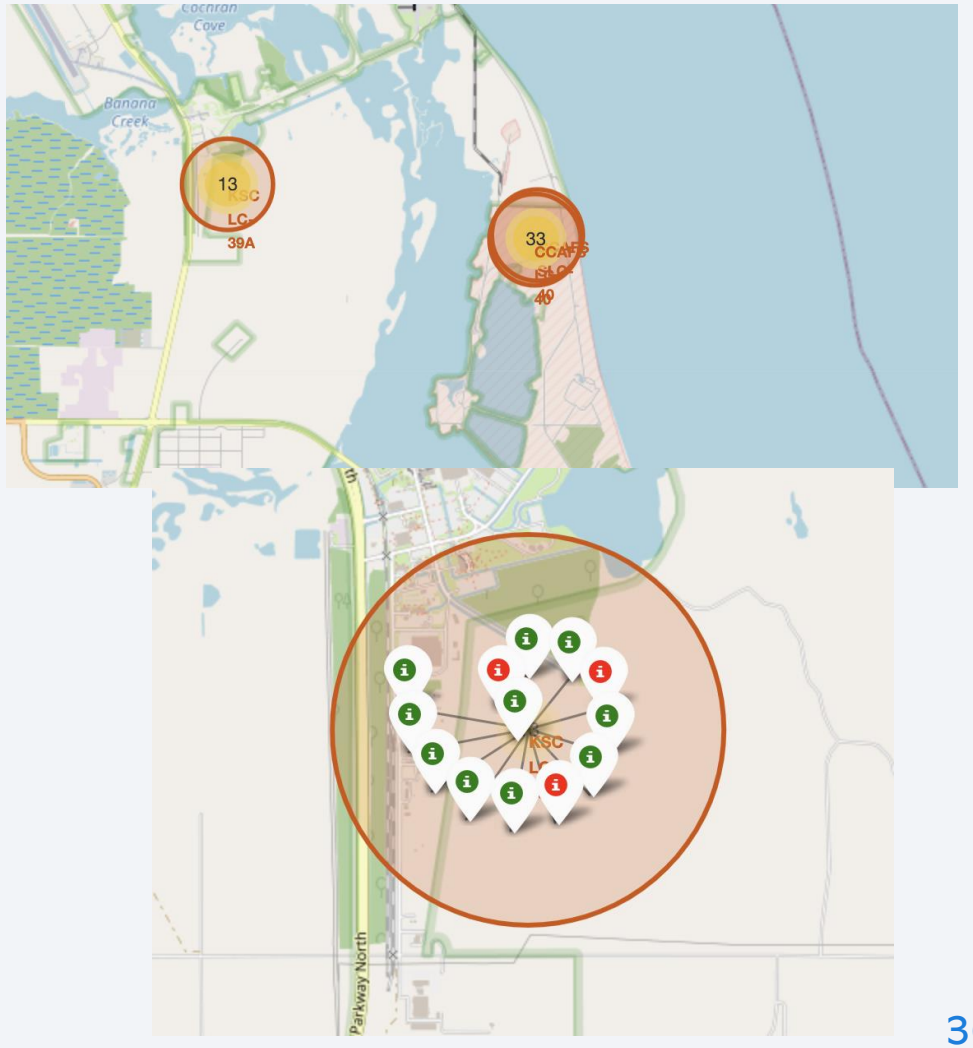
Map of Launch Sites

- Interactive map using folium displaying all 4 launch site locations using a small orange circle marker and the launch site name

Map of Launch Sites and Landing Outcomes

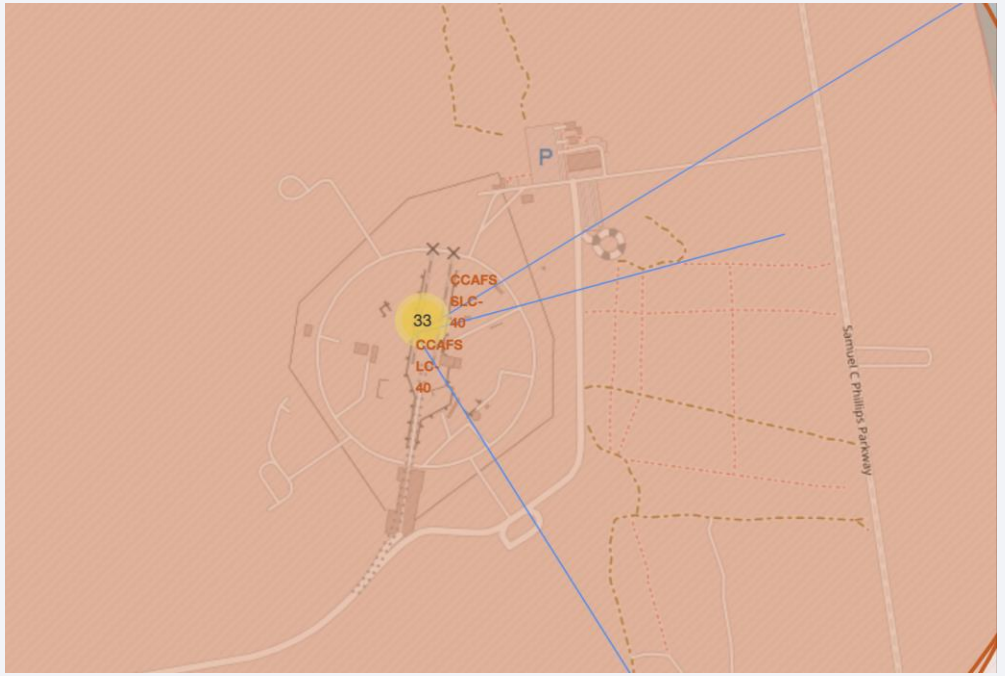
- An interactive map created with folium that displays the launch site locations and the number of launches from each site

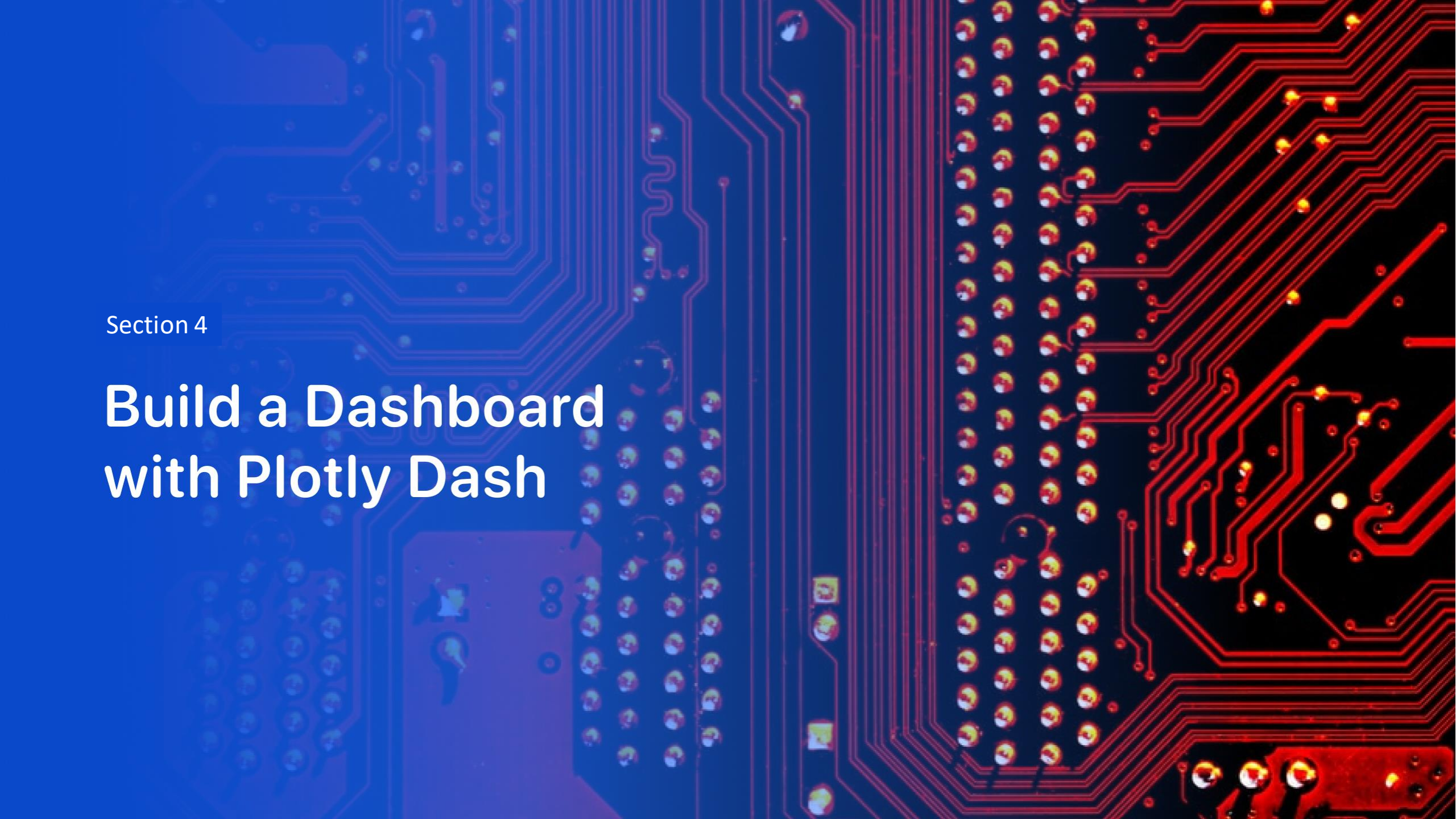
- When a site is clicked on it will display indicator markers for each launch that are either green or red based on whether to landing outcome was successful or unsuccessful respectively



Map of Launch Sites and Proximate Features

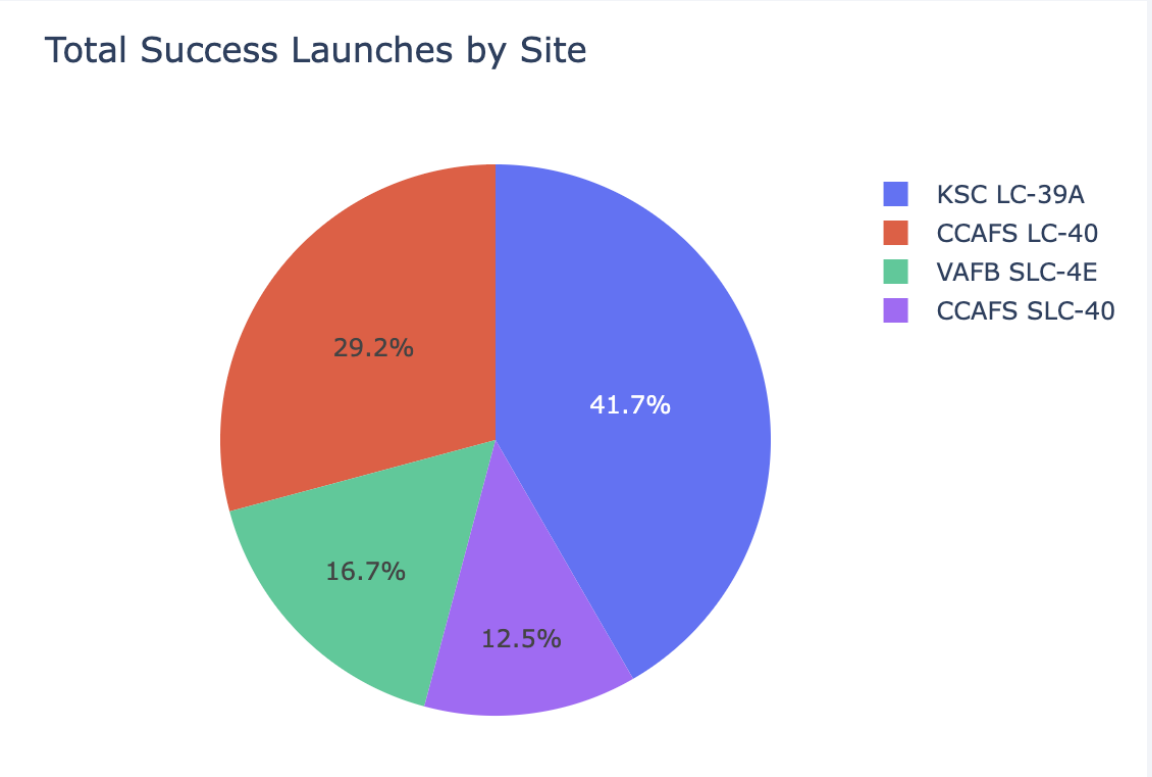
- An interactive folium map similar to the previous two, that additional shows lines to connecting between the launch site and nearby features such as railways, roads, and coastlines




Section 4
Build a Dashboard with Plotly Dash

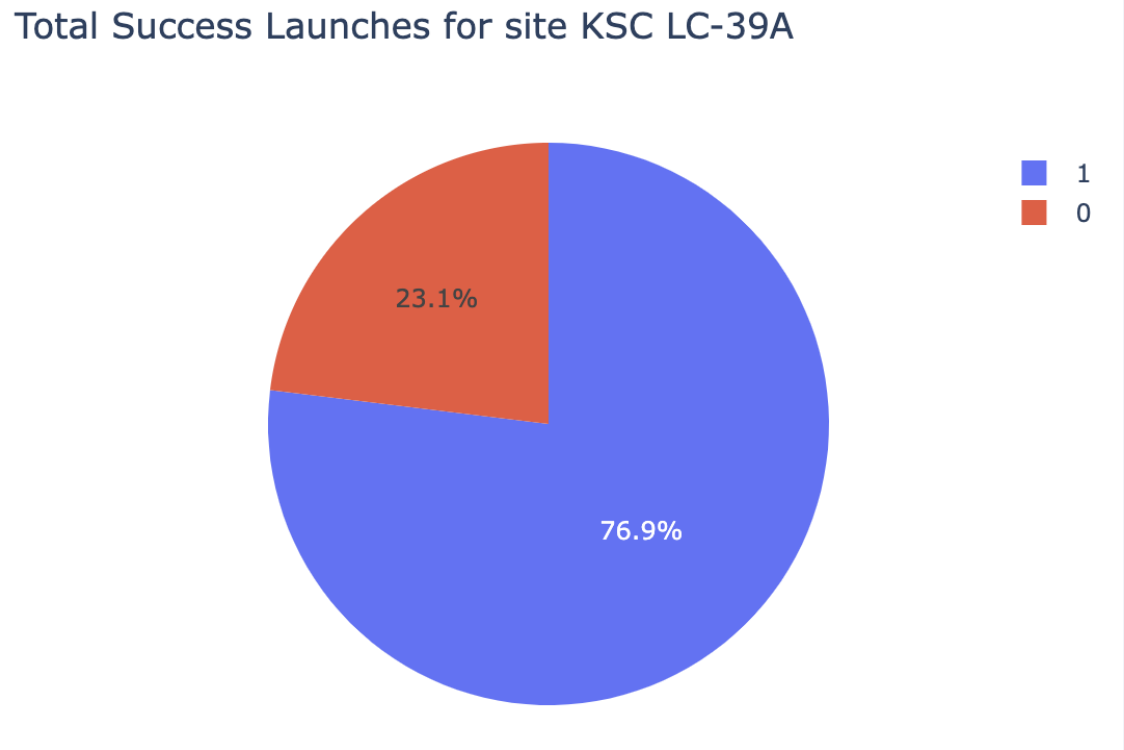
Total Successful Launches by Site

- An interactive piechart of the proportion of successful launches by site
- Hovering over a slice of the pie will show the number of successful launches for each site



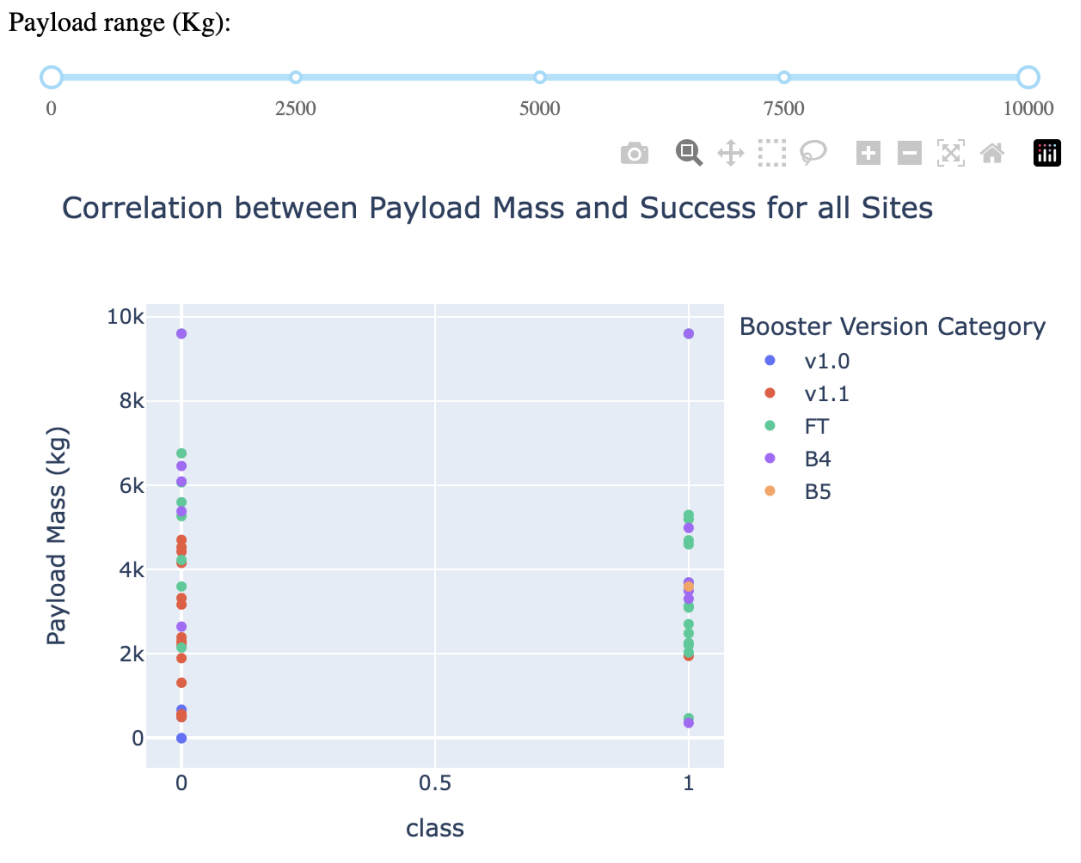
Total Launches for KSC LC-39A launch site

- An interactive piechart of the proportion of successful vs unsuccessful launches at KSC LC-39A, the site with the most total successful launches
- The red slice shows failed landings while the blue slice shows the successful landing outcomes
- Hovering over a slice of the pie will show the number of either successful or unsuccessful landings



Correlation between Payload Mass and Success

- An interactive scatter plot of the correlation between payload mass (kg) and launch outcome, with sliders for selecting the payload mass range
- The plot uses different colored points to identify the booster version used in the launch
- Hoovering over a point will show the booster version, payload mass, and landing outcome for that launch

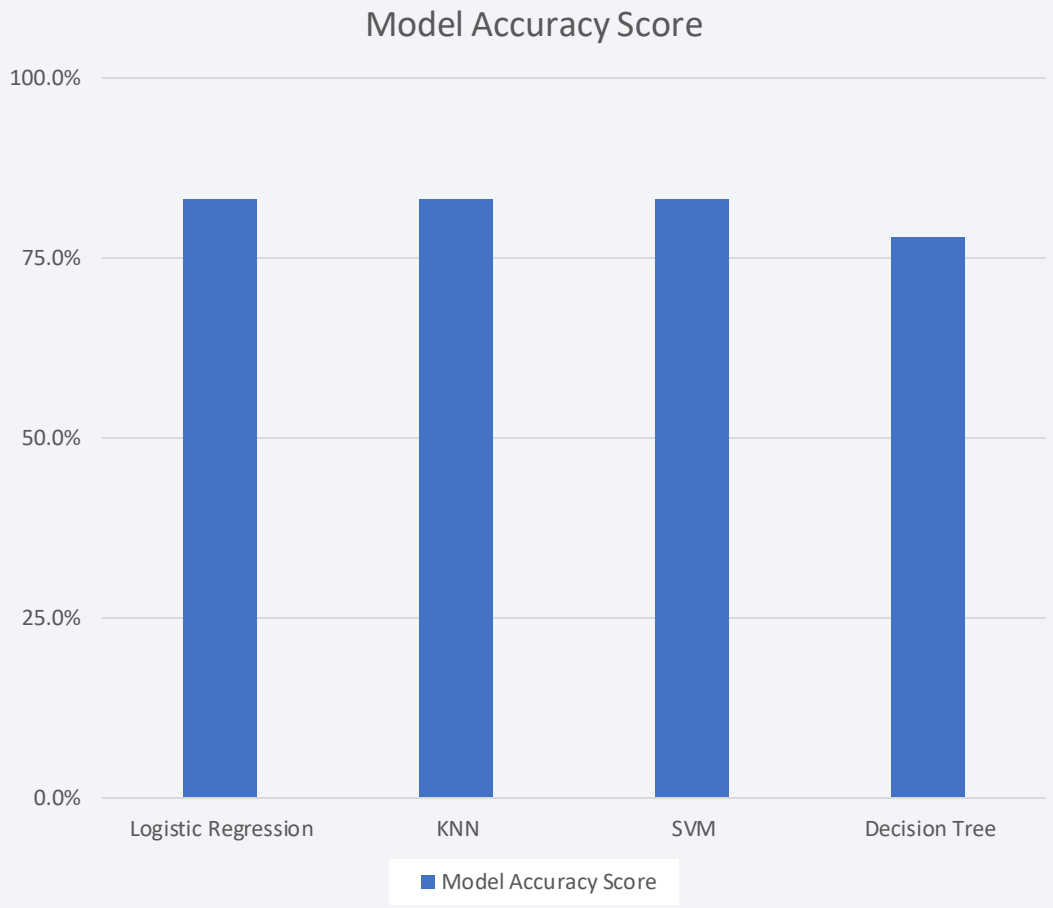


Section 5

Predictive Analysis (Classification)

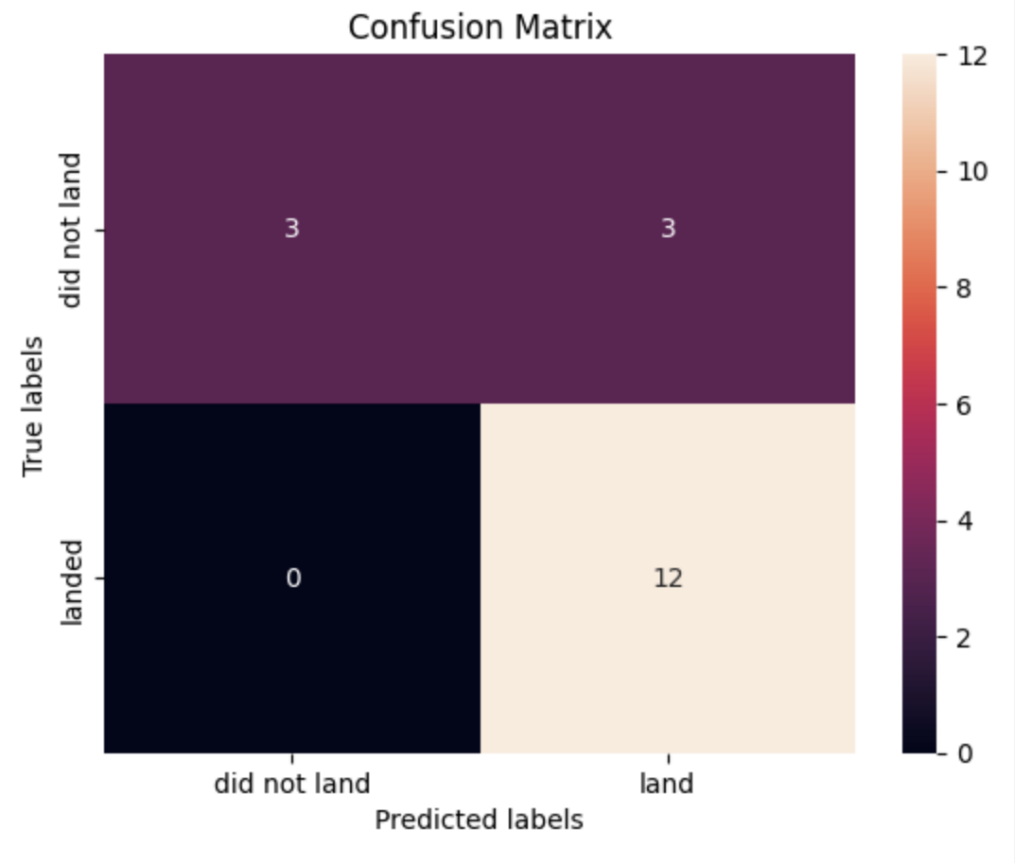
Classification Accuracy

- The resulting accuracy scores for the models returned that the Logistic Regression, K-Nearest Neighbor, and Support Vector Model all had the same classification accuracy scores, matching for the highest score



Confusion Matrix

- Here is the confusion matrix that resulted from the Logistic Regression, K-Nearest Neighbors, and Support Vector Model
- It shows the correct predictions in the upper-left corner and bottom-right corner, along with false positives in the upper-right, and the false negatives in the lower-left



Conclusions

- There was a positive correlation between landing outcome and orbit (specifically ES-L1, GEO, SSO, and HEO)
- There was a also a positive correlation based on recent launches having a much higher landing success rate
- The KSC LC-39A launch site also has the most successful landing outcomes compared to the other launch sites
- Our logistic regression, support vector model, and k-nearest neighbors models all were tied for our best predictive model for determining the success or failure of the landing of a SpaceX launch

Thank you!