IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Danielle LaVigne
July 22, 2025

Outline

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

Executive Summary

- Summary of methodologies
  - The following methodologies were applied to SpaceX Launch Data:
    - Data Collection via SpaceX API and Web Scraping via Wikipedia HTML Table
    - Data Preparation using Cleansing and Transformation techniques
    - Exploratory Data Analysis using Data Wrangling, SQL, and Visualization
    - Interactive Map and Dashboard Creation for Visual Insights
    - Predictive Analysis and Data Modelling to forecast Falcon 9 Landing Success

- Summary of all results
  - Exploratory Data Analysis:
    - Landing success of Falcon 9 improved progressively over time with an overall success rate of 67%
    - Orbit types ES-L1, GEO, HEO, and SSO have the highest success rates at 100%.
    - Launch site KSC LC 39A had the highest launch success rate at 77%.
  - Data Visualization and Analysis:
    - Launch sites are near coastlines and not near cities presumably to decrease the chance of destruction is launch fails.
    - Launch sire are close to the equator presumably to take advantage of the Earth's rotation speed which lessens the fuel consumption at take-off.
    - Launch sites are close to railways and highways presumably to be close to a mode of transportation for equipment and supply delivery.
  - Predictive Analysis
    - All models performed nearly the same, with the Decision Tree model just slightly performing better.

Introduction

- **Project Background and Context**

  SpaceX has gained worldwide attention for a series of historic milestones. It is the only private company ever to return a spacecraft from low-earth orbit, which it first accomplished in December 2010. Spaces X's Falcon 9 launches like regular rockets. SpaceX advertises these rocket launches on its website at a cost of $62 million whereas other providers cost upwards of $165 million per launch. Most of the SpaceX savings can be attributed the ability to reuse the first stage. The goal is to predict if the first stage will land successfully.

- **Problems you want to find answers**
  - What is the overall Falcon 9 landing success rate over the course of its launch history?
  - Do the number of flights, payload mass, launch site locations, or orbit type contribute to predicting if Falcon 9's first stage will land successfully?
  - What is the best model for predicting the success of Falcon 9 landings?

Section 1

Methodology

Methodology

- Data collection methodology:
  - Data was collected via SpaceX API and web scraping via Wikipedia HTML table.
- Perform data wrangling:
  - Data was processed by filtering for Falcon 9 data, replacing null values in the Falcon 9 pay load mass with the calculated mean, and using one hot encoding to convert the Falcon 9 categorical variables (orbit type, launch site, landing pad, and serial identifier) to numbers.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models:
  - The model was built using a machine learning pipeline. The data was preprocessed, standardized, and split into testing and training sets. Grid searches were used to tune the model by finding the best parameters for each model. Logistic regression, decision tree, support vector machines, and K nearest neighbors models were evaluated using a confusion matrix.

Data Collection – SpaceX API

SpaceX API Launch Data

- Defined a series of functions so data from the API call could be appended to lists.
- Requested and parsed data from the API using a get request.
- Normalized the JSON response to create a Pandas dataframe.
- Retrieved additional data using the launch identifiers to call the API.
- Stored the data in global variable lists.
- Constructed the data set by combining the global variables into a dictionary.
- Created a Pandas dataframe from the dictionary.
- Filtered the data, creating a new dataframe containing Falcon 9 data only.
- Below is the GitHub URL of the completed SpaceX API calls notebook: https://github.com/ddlavigne/SpaceX-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Request rocket launch data from SpaceX API

spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)

↓

Request and parse SpaceX data using GET request

static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
response=requests.get(static_json_url)

↓

Decode response as a JSON and normalize creating a dataframe

data = pd.json_normalize(response.json())

↓

Get data about launches using launch IDs

data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

↓

Combine columns into a dictionary and create a dataframe

launch_dict = {'FlightNumber': list(data['flight_number']),…
df = pd.DataFrame(launch_dict)

↓

Filter dataframe for Falcon 9 launches and save to new dataframe

data_falcon9 = df[df['BoosterVersion']!='Falcon 1']

Data Collection - Scraping

SpaceX Wikipedia Data Web Scraping

- Defined a series of functions to process web scraped HTML table.
- Scraped the data from a snapshot of the List of Falcon 9 and Falcon Heavy launches Wiki page using an HTML get request.
- Assigned the HTML response to a Beautiful Soup object using the HTML parser..
- Extracted the variable names from the HTML table header.
- Created dictionary using the keys from the extracted column names.
- Filled the dictionary launch records extracted from the table rows
- Created a data frame launch dictionary.
- Below is the GitHub URL of the completed SpaceX web scraping notebook: https://github.com/ddlavigne/SpaceX-Capstone/blob/main/jupyter-labs-webscraping.ipynb

Scrape data from Falcon 9 and Falcon Heavy launches Wikipage

```
static_url = "https://en.wikipedia.org/w/index.php?title=
List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
response = requests.get(static_url)
```

Create BeautifulSoup object from HTML response and extract column headers

```
soup = BeautifulSoup(html_content, 'html.parser')
html_tables = soup.find_all('table')
first_launch_table = html_tables[2]
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
```

Create dictionary Pandas Dataframe by parsing HTML tables and fill with data

```
launch_dict= dict.fromkeys(column_names)
for table_number,table in enumerate(soup.find_all('table',"wikitable
plainrowheaders collapsible")):
    for rows in table.find_all("tr"):
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        row=rows.find_all('td')...
```

Create a dataframe from dictionary

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items()})
```

Data Wrangling

- After data was filtering for Falcon 9 data missing values were identified for each variable.
- Null values in the Falcon 9 pay load mass variable were replaced with the calculated mean.
- Exploratory data analysis was conducted to get an idea of the data content, including number of launches per launch site, number and occurrence of each orbit type, and landing outcomes.
- Landing outcomes were enumerated, a landing outcome label was created, and the landing success rate was calculated.
- One hot encoding was used to convert the Falcon 9 categorical variables (orbit type, launch site, landing pad, and serial identifier) to numbers.
- Casted the entire dataframe changing data types from an integer to a float64.
- Below are the GitHub URL of the completed data wrangling notebooks:
  https://github.com/ddlavigne/SpaceX-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

  https://github.com/ddlavigne/SpaceX-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

  https://github.com/ddlavigne/SpaceX-Capstone/blob/main/edadataviz.ipynb

Filter dataframe for Falcon 9 launches and save to new dataframe

```
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

Identify missing values in each attribute

```
data_falcon9.isnull().sum()
```

Calculate numbers for key variables

```
counts1 = df['LaunchSite'].value_counts()
counts2 = df['Orbit'].value_counts()
landing_outcomes = df['Outcome'].value_counts()
```

Enumerate landing outcomes, create a label, and calculate success rate

```
for i,outcome in enumerate(landing_outcomes.keys())
landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df['Outcome']]
df["Class"].mean()
```

Calculate mean for PayloadMass and replace null values with mean value

```
mean_PayloadMass = data_falcon9['PayloadMass'].mean(axis=0)
data_falcon9['PayloadMass'].replace(np.nan, mean_PayloadMass, inplace=True)
```

Create dummy variables, use OneHotEncoder to assign variable values, add to df

```
features_one_hot=pd.get_dummies(features, columns=['Orbit', 'LaunchSite', 'LandingPad', 'Serial'], dtype=int)
df =pd.concat([df,features_one_hot],axis =1)
```

Cast all numeric columns to float variables

```
features_one_hot = features_one_hot.astype(np.float64)
```

EDA with Data Visualization

The charts plotted on the SpaceX launch data include:

- Flight Number vs. Payload Mass – Scatter plot used to visualize if Flight Number and Payload variables would affect the launch outcome
- Flight Number vs. Launch Site – Scatter plot used to visualize if Flight Number and Launch Site variables would affect the launch outcome
- Payload Mass vs. Launch Site – Scatter plot used to visualize if Payload Mass and Launch Site variables would affect the launch outcome
- Flight Number vs. Orbit Type – Scatter plot used to visualize if Flight Number and Orbit Type variables would affect the launch outcome
- Payload Mass vs. Orbit Type – Scatter plot used to visualize if Payload Mass and Orbit Type variables would affect the launch outcome
- Orbit Type Launch Success Rate – Bar chart used to visualize the relationship between launch success rate and Orbit Type
- Launch Success Yearly Trend – Line plot used to visualize the annual launch success trend
- Below is the GitHub URL of the completed EDA with data visualization notebook: https://github.com/ddlavigne/SpaceX-Capstone/blob/main/edadataviz.ipynb

EDA with SQL

Summary of SQL queries performed on the SpaceX launch data:

- Retrieved a list of unique launch sites
- Retrieved a list of 5 records where the launch site begins with "CCA"
- Retrieved the total payload mass carried by boosters launched by "NASA (CRS)"
- Retrieved the average payload mass carried by booster version "F9 v1.1"
- Retrieved the date when the first successful landing outcome in ground pad was achieved
- Retrieved a list of names of the boosters that have success in drone ship with a payload mass between 4000 and 6000
- Retrieved the total number of successful and failure mission outcomes
- Retrieved a list all booster versions that have carried the maximum payload mass, using a subquery with an aggregate function
- Retrieved a list of records for the months in year 2015 with a failure landing in drone ship outcome that includes the month names, booster versions, and launch site
- Retrieved the count of the landing outcomes between the date 2010-06-04 and 2017-03-20, in descending rank order
- Below is the GitHub URL of the completed EDA with SQL notebook:
  https://github.com/ddlavigne/SpaceX-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Summary of map objects created and added to a folium map representing SpaceX launch sites:

- A circle was added on the NASA Johnson Space Center coordinates, a pop-up displaying its name, and a marker icon showing the text name. The purpose was to easily visualize the proximity of the NASA Johnson Space Center to launch sites.
- A circle was added on each launch site (CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E) coordinates, a pop-up displaying the names, and a marker for each text name. The purpose was to easily visualize the proximity of each launch site to the nearest railroad, highway, and coastline.
- A marker cluster was added on each launch site with a white marker and an icon for each successful launch (green) and failed launch (red). The purpose was to easily visualize the launch successes and failures at launch site.
- A marker to show the coordinates of an area based on the mouse position at the moment. The purpose was to easily identify the coordinates of the railroad, highway, and coastline nearest each launch site.
- A marker to indicate the distance a railroad, highway, and coastline are from the VAFB SLC-4E launch site. The purpose was to easily visualize the distance between the VAFB SLC-4E launch site and the nearest railroad, highway, and coastline.
- A line between each of the railroad, highway, and coastline and the VAFB SLC-4E launch site distance marker. The purpose was to easily visualize the VAFB SLC-4E launch site with the nearest railroad, highway, and coastline.
- Below is the GitHub URL of the completed interactive map with Folium map:
  https://github.com/ddlavigne/SpaceX-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Summary of plots, graphs and interactions on the dashboard:

- Launch Site Drop-down Input Component – this was added so the user can select one of the launch sites or all of the launch sites to see changes to the pie and scatter plots in each scenario.
- Range Slider input component – this was added so the user can change the Payload Mass range to see the changes in launch successes vs. failures in the scatter plot based on varying payload masses.
- Pie Chart output component – this was added to visualize the launch success rates for each launch site or a combination of all launch sites, depending on the Launch Site selected in the drop-down by the user.
- Scatter Chart output component – this was added to visualize the varying launch successes vs. failures for each Booster version or all Booster versions, based on the Payload Mass range chosen by the user.
- Below is the GitHub URL of the completed Plotly Dash lab: https://github.com/ddlavigne/SpaceX-Capstone/blob/main/spacex-dash-app.py

Predictive Analysis (Classification)

- After a function was built to plot the confusion matrix, the SpaceX data was loaded.
- An array was created and assigned to the y variable representing the class (launch success vs. failure).
- All other columns in the dataframe were standardized and assigned to the X variable.
- The data was split into testing and training sets using the train_test_split function, where the testing set comprised of 20% of the data.
- Model objects were created, grid searches were applied to each, and model accuracy was calculated. Models included Logistic Regression, Decision Tree, Support Vector Machine, and K Nearest Neighbors.
- A confusion matrix was produced from each model to represent the difference between predicted vs. actual launch success.
- Test data accuracy scores were calculated for each model.
- A bar chart was created representing the accuracy rate of each model and best performing model was identified.
- Below is the GitHub URL of the completed predictive analysis related notebook: https://github.com/ddlavigne/SpaceX-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

<u>Numpy array created for Class column and assigned to variable y</u>

```
y = data['Class'].to_numpy()
```

↓

<u>All other data standardized, transformed, and assigned to variable X</u>

```
transform = preprocessing.StandardScaler()
X = transform.fit_transform(X)
```

↓

<u>Data split into training and testing data sets with test set being 20% of the data</u>

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,random_state=2)
```

↓

<u>Grid Search applied to Logistic Regression to get best parameters and model accuracy calculated</u>

```
logreg_cv = GridSearchCV(lr, parameters, cv = 10)
logreg_cv.fit(X_train, y_train)
test_score_lr = logreg_cv.score(X_test, y_test)
```

↓

<u>Logistic Regression model Confusion Matrix plotted</u>

```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(y_test,yhat)
```

↓

<u>Grid Search and Confusion Matrix applied to all other models and each model's accuracy calculated</u>

Additional Models include: Decision Tree, Support Vector Machine, and K Nearest Neighbors

↓

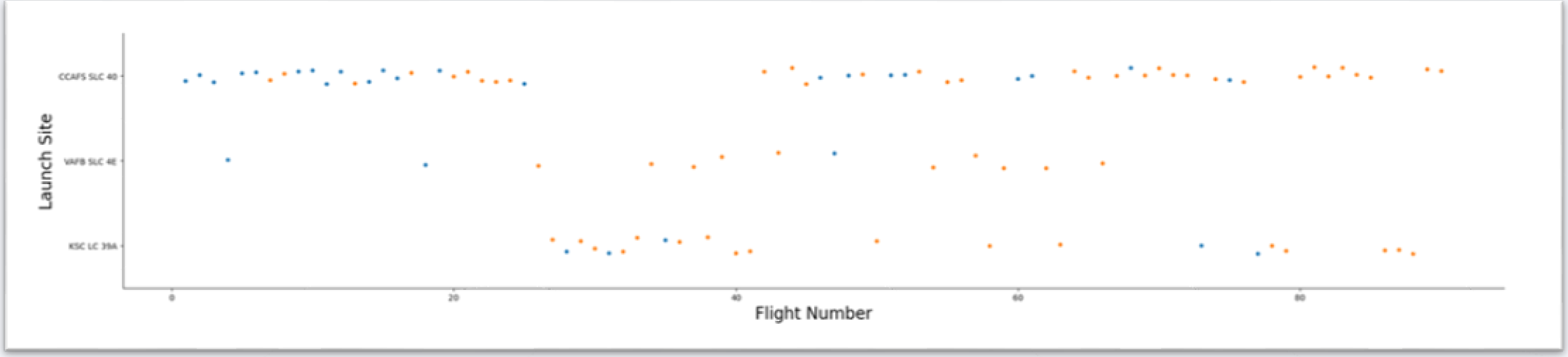
<u>Bar chart created to show model accuracy</u>

```
plt.figure(figsize=(8, 5))
sns.barplot(x='Model', y='Accuracy', data=df_accuracy, palette='Blues')
plt.xlabel('Classification Model', fontsize=12)
plt.ylabel('Accuracy', fontsize=12)
plt.title('Accuracy of Classification Models', fontsize=14)
plt.show()
```

Section 2

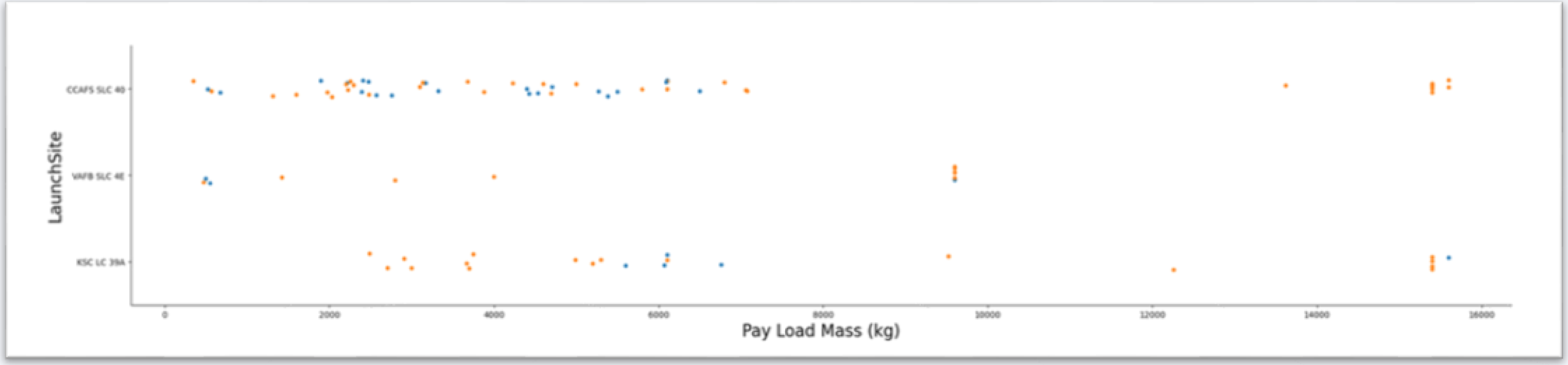
Insights drawn from EDA

Flight Number vs. Launch Site



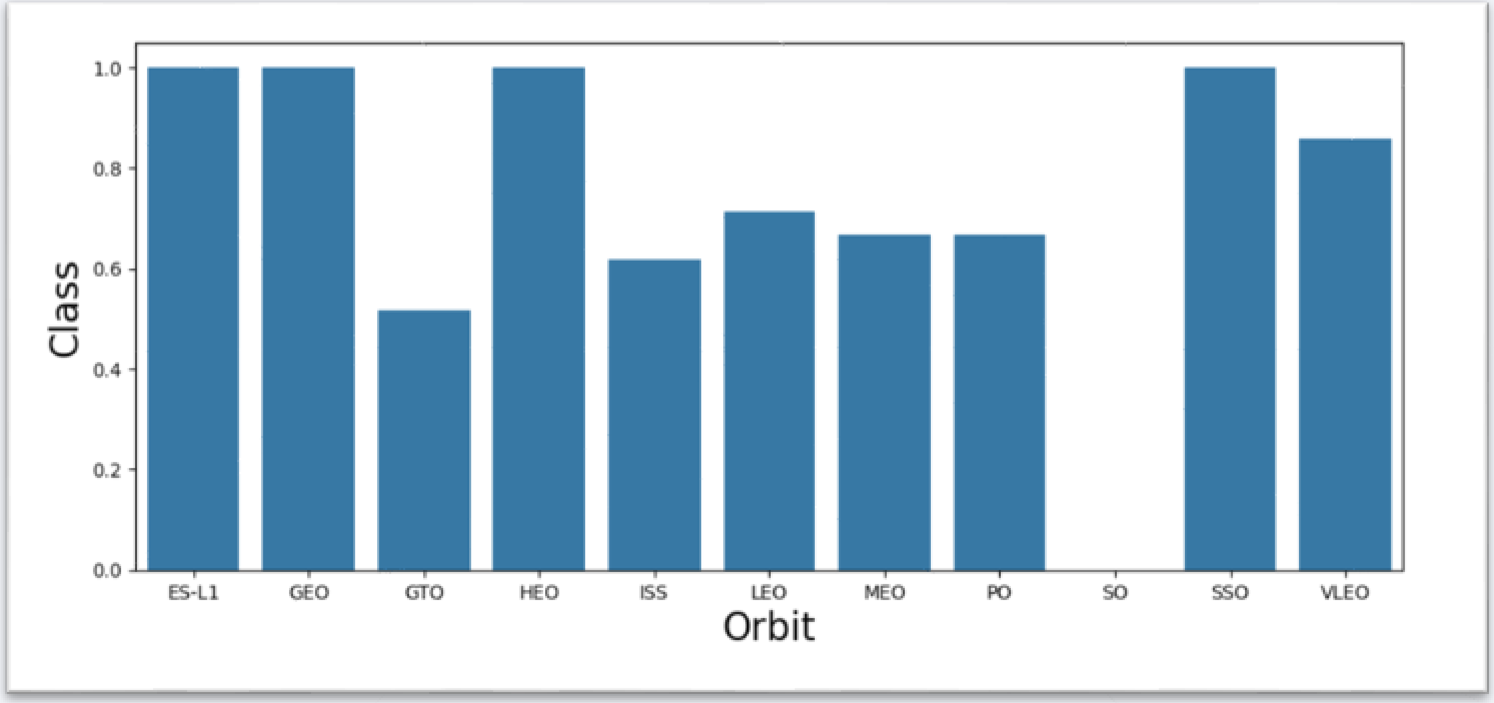
- Orange dot means the launch was successful
- Blue dot means the launch was unsuccessful
- Conclusion: The more flights each launch site conducted the greater the success rate

Payload vs. Launch Site



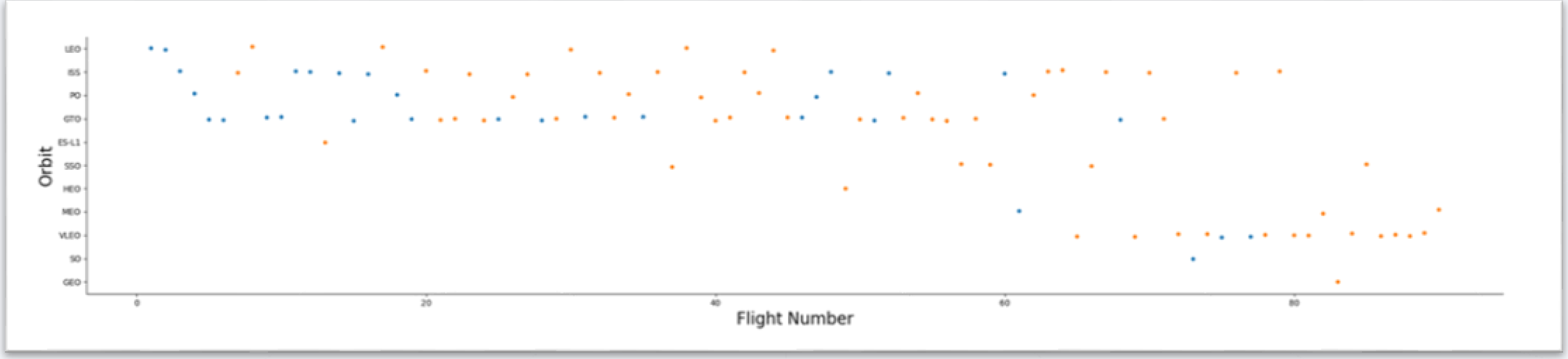
- Orange dot means the launch was successful
- Blue dot means the launch was unsuccessful
- Conclusion: Payload doesn't appear to be a factor in predicting launch success regardless of launch site

Success Rate vs. Orbit Type



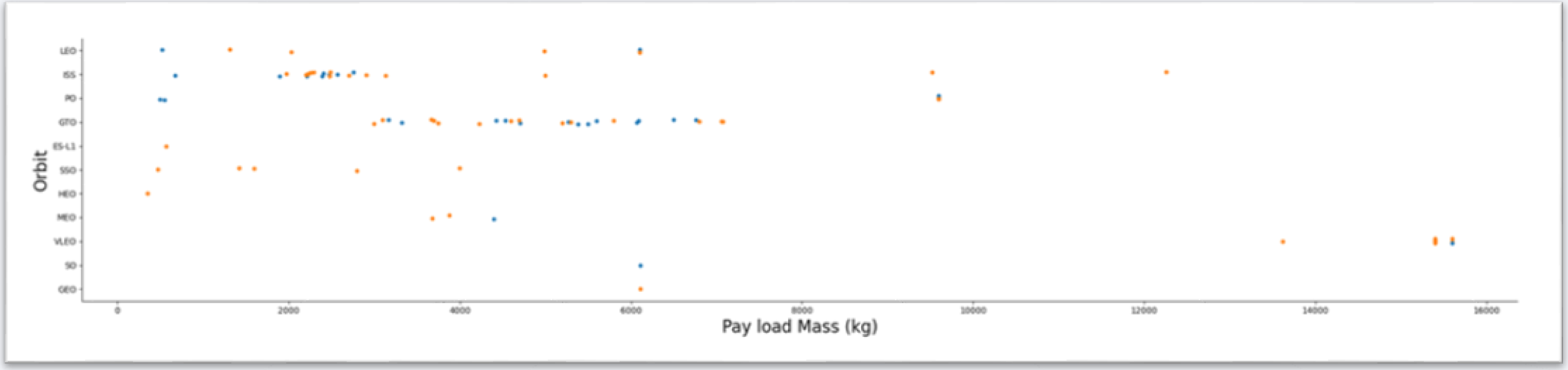
- Class 1.0 means the average launch was successful
- Class 0.0 means the average launch was unsuccessful
- Conclusion: Orbit types ES-L1, GEO, HEO, and SSO have the highest launch success rates at 100%

Flight Number vs. Orbit Type



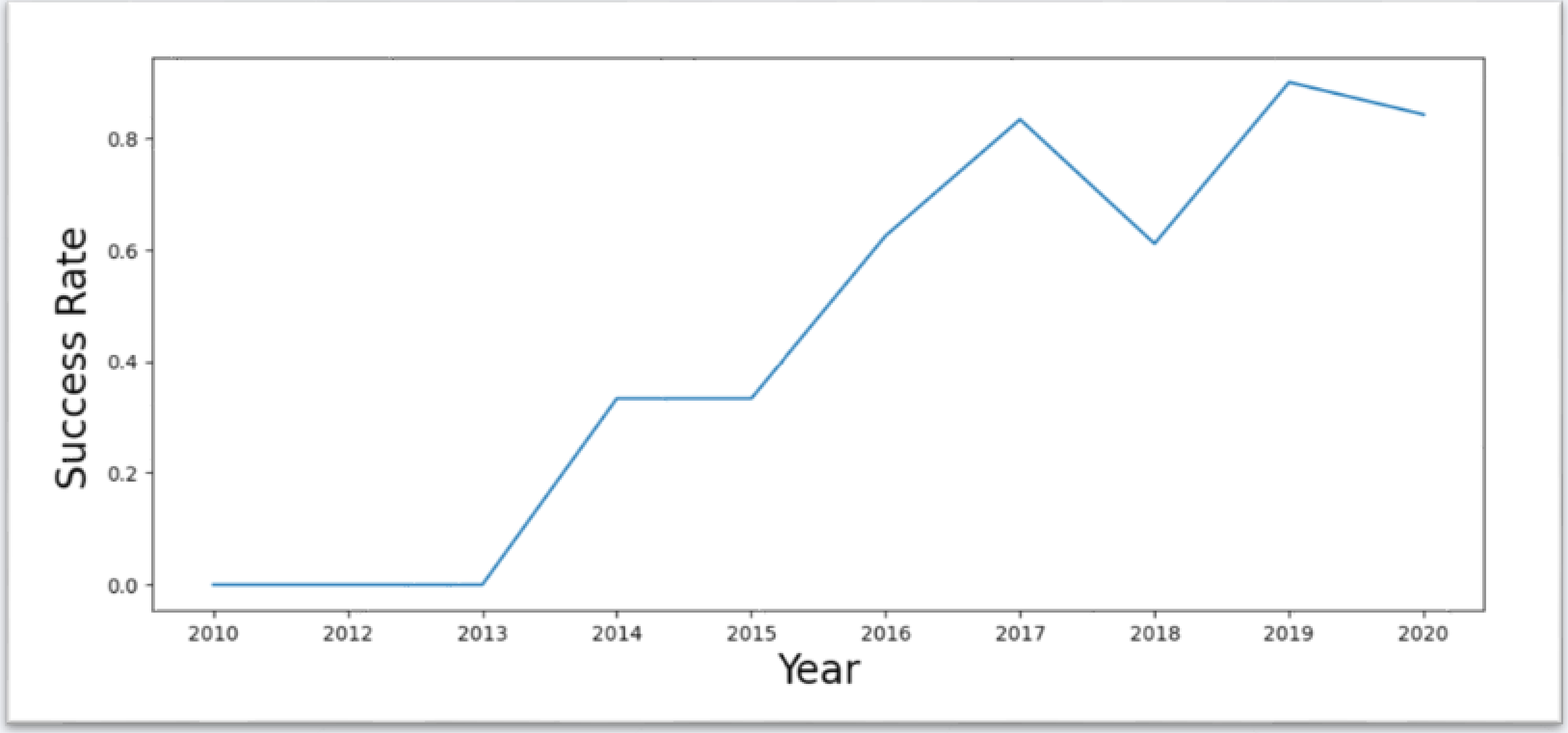
- Orange dot means the launch was successful
- Blue dot means the launch was unsuccessful
- Conclusion: Orbit type may have some influence of launch success, but more analysis would be required to find out

Payload vs. Orbit Type



- Orange dot means the launch was successful
- Blue dot means the launch was unsuccessful
- Conclusion: Payload Mass may or may not have anything to do with success rate of certain Orbit types, but more analysis would be needed to determine this

Launch Success Yearly Trend



- Success Rate above 0.6 means the average launch was for the year successful
- Success Rate 0.0 Blue means the average launch for the year was unsuccessful
- Conclusion: The more launches performed the better the success rate becomes

All Launch Site Names

Launch Site Names:

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

Query to find the unique launch sites in the SpaceX table:

```
%sql Select distinct "Launch_Site" from SPACEXTABLE;
```

Launch Site Names Begin with 'CCA'

Five records where launch sites begin with "CCA":

Date	Time (UTC)	BoosterVersion	LaunchSite	Payload	Payload Mass	Orbit	Customer	Mission Outcome	LandingOutcome
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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

Query to find 5 records where launch sites begin with "CCA":

```
%sql Select * from SPACEXTABLE where "Launch_Site" like 'CCA%' Limit 5;
```

Total Payload Mass

Total payload mass carried by boosters launched by NASA:

sum(PAYLOAD_MASS__KG_) = 45596

Query to find the total payload mass carried by boosters launched by NASA:

```
%%sql  Select sum(PAYLOAD_MASS__KG_)
       from SPACEXTABLE
       where "Customer" = 'NASA (CRS)';
```

Average Payload Mass by F9 v1.1

Average payload mass carried by booster version F9 v1.1:

avg(PAYLOAD_MASS__KG_) = 2928.4

Query to find the average payload mass carried by booster version F9 v1.1:

```
%%sql  Select avg(PAYLOAD_MASS__KG_)
       from SPACEXTABLE
       where "Booster_Version" = 'F9 v1.1';
```

First Successful Ground Landing Date

Date of the first successful landing outcome on ground pad:

min("Date") = 2015-12-22

Query to find the date of the first successful landing outcome on ground pad:

```
%%sql  Select min("Date")
       from SPACEXTABLE
       where "Landing_Outcome" = 'Success (ground pad)';
```

Successful Drone Ship Landing with Payload between 4000 and 6000

The names of boosters that successfully landed on drone ship with a payload mass between 4000 and 6000:

- F9 FT B1022
- F9 FT B1026
- F9 FT B1021.2
- F9 FT B1031.2

Query to find the names of boosters that successfully landed on drone ship with a payload mass between 4000 and 6000:

```
%%sql  Select distinct "Booster_Version"
       from SPACEXTABLE
       where "Landing_Outcome" =  'Success (drone ship)'
       and "PAYLOAD_MASS__KG_" > 4000 and "PAYLOAD_MASS__KG_" < 6000;
```

Total Number of Successful and Failure Mission Outcomes

Total number of successful and failed mission outcomes:

- Successful = 100
- Failed = 1

Query to find the number of successful and failed mission outcomes:

```
%sql Select count(*) from SPACEXTABLE where "Mission_Outcome" like 'Success%';
%sql Select count(*) from SPACEXTABLE where "Mission_Outcome" like 'Failure%';
```

Boosters Carried Maximum Payload

Names of the boosters which have carried the maximum payload mass:

- 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

Query to find the names of the boosters which have carried the maximum payload mass:

```
%%sql   Select distinct "Booster_Version"
        from SPACEXTABLE
        where "PAYLOAD_MASS__KG_" =
          (Select max("PAYLOAD_MASS__KG_")
          from SPACEXTABLE)
        order by "Booster_Version";
```

2015 Launch Records

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

Month Name	Landing Outcome	Booster Version	Launch Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Query to find the failed landing outcomes in drone ship, their booster versions, and launch sitenames for the year 2015:

```
%%sql   Select Case substr(Date, 6,2) when '01' then 'January'
                 when '04' then 'April' End as Month_Name,
                 Landing_Outcome, Booster_Version, Launch_Site
        from SPACEXTABLE
        where "Landing_Outcome" =  'Failure (drone ship)'
        and substr(Date,0,5)='2015';
```

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

Count of landing outcomes between the 6/4/2010 and 3/20/2017, with count ranked in descending order:

Landing Outcome	Count Outcome
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

Query to find the count of landing outcomes between the 6/4/2010 and 3/20/2017, ranked in descending order:

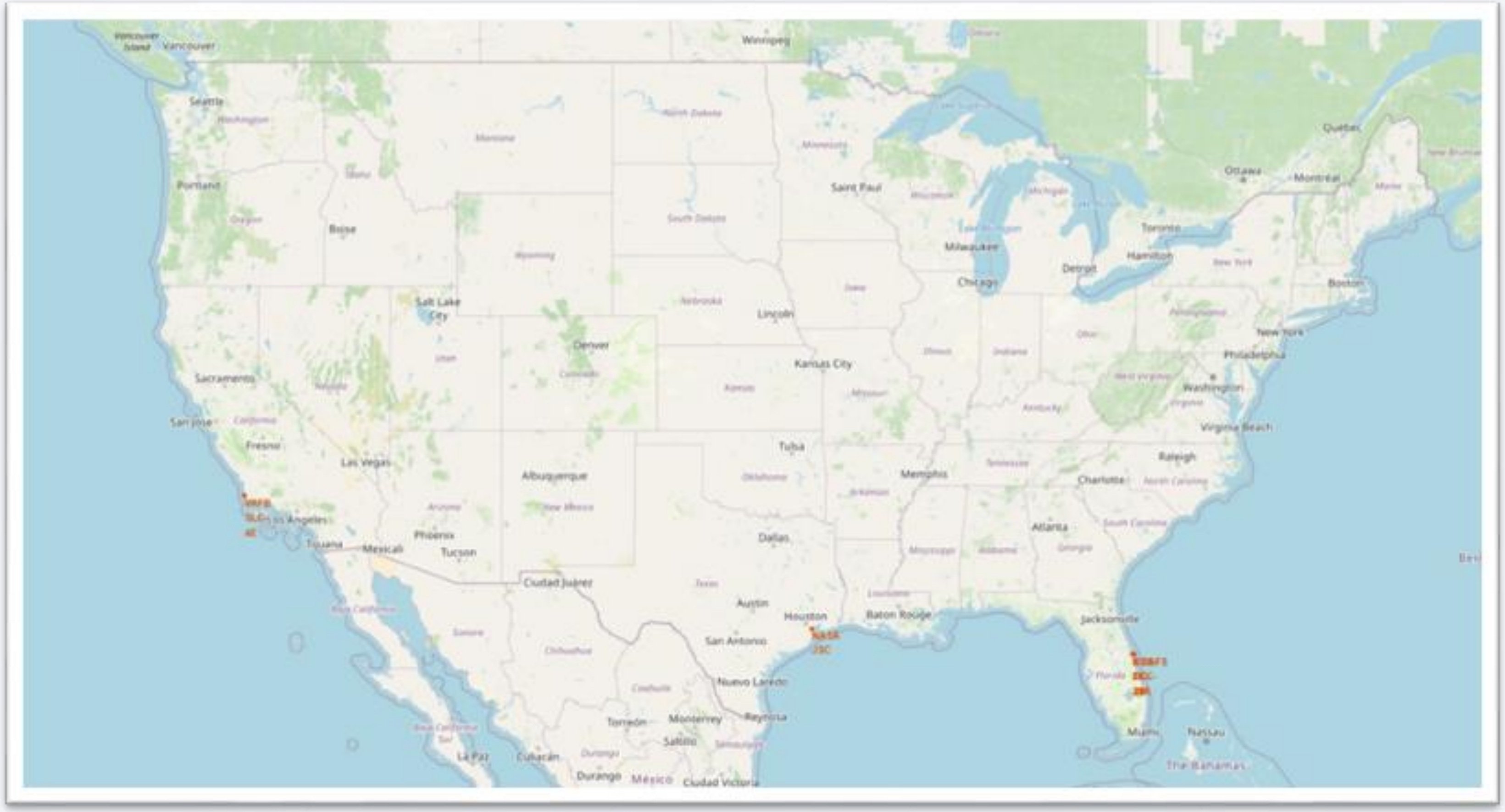
```
%%sql  Select Landing_Outcome, count(*) as Count_Outcome
       from SPACEXTABLE
       where Date between '2010-06-04' and '2017-03-20'
       and substr(Date,0,5)='2015'
       group by Landing_Outcome
       Order by Count_Outcome desc;
```

Section 3

Launch Sites Proximities Analysis

Folium Map Launch Sites and Space Center

- The red markers are indicative of 4 launch sites on US coasts and the NASA Johnson Space Center in Houston.
- The Launch Sites include three within a short distance from each other on the east coast in Florida (CCAFS LC-40, CCAFS SLC-40, and KSC LC-39A) and the other one by itself on the west coast in California (VAFB SLC-4E) .

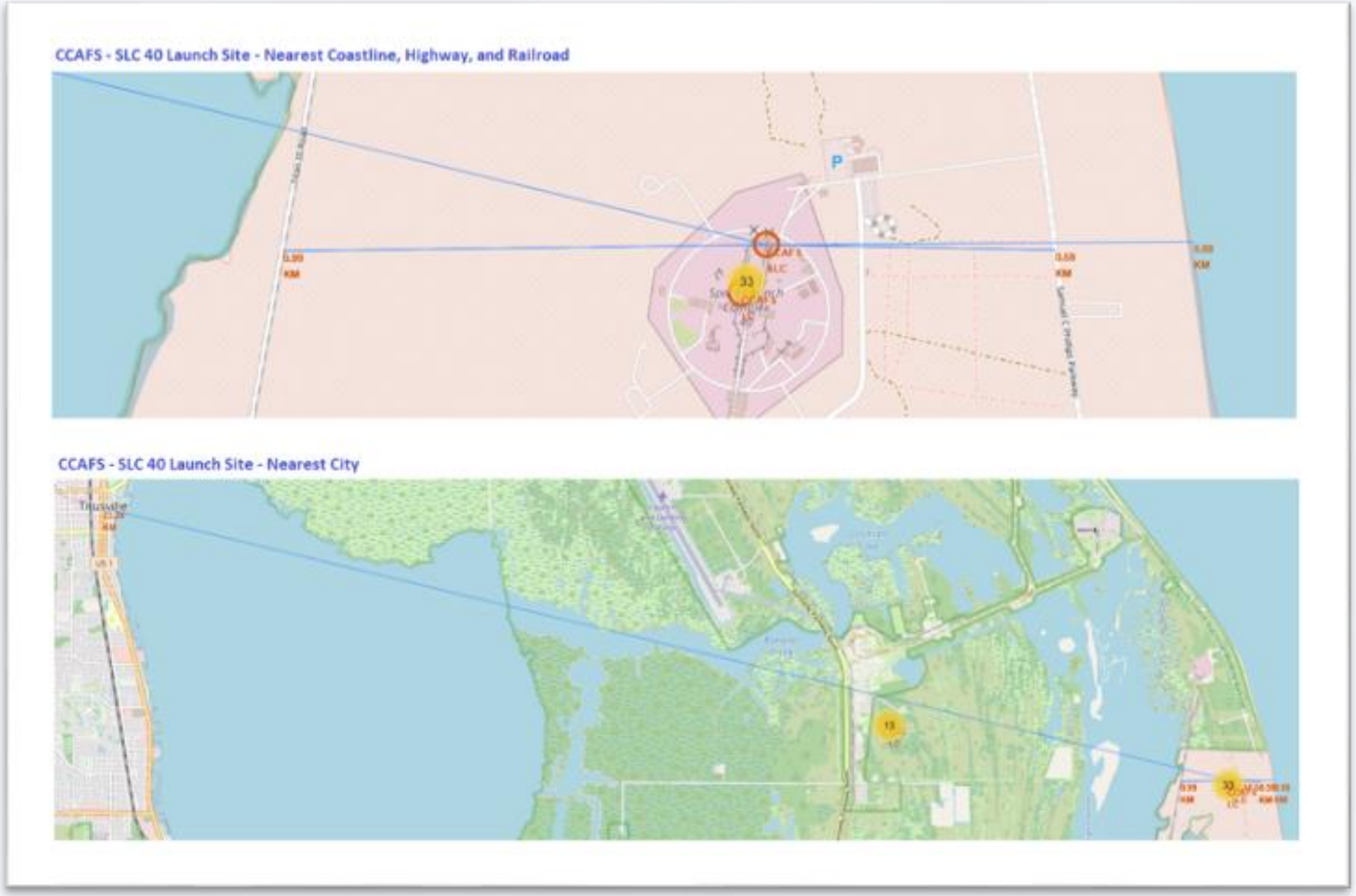


Folium Map Launch Site Outcomes

- Each Launch Site is shown here with the launch outcomes indicated by a marker in red, for failed, or green, for succeeded.
- This provides a quick visual representation of how the launch sites stack up against each other with regard to successful launches.

Folium Map CCAFS – SLC 40 Proximities

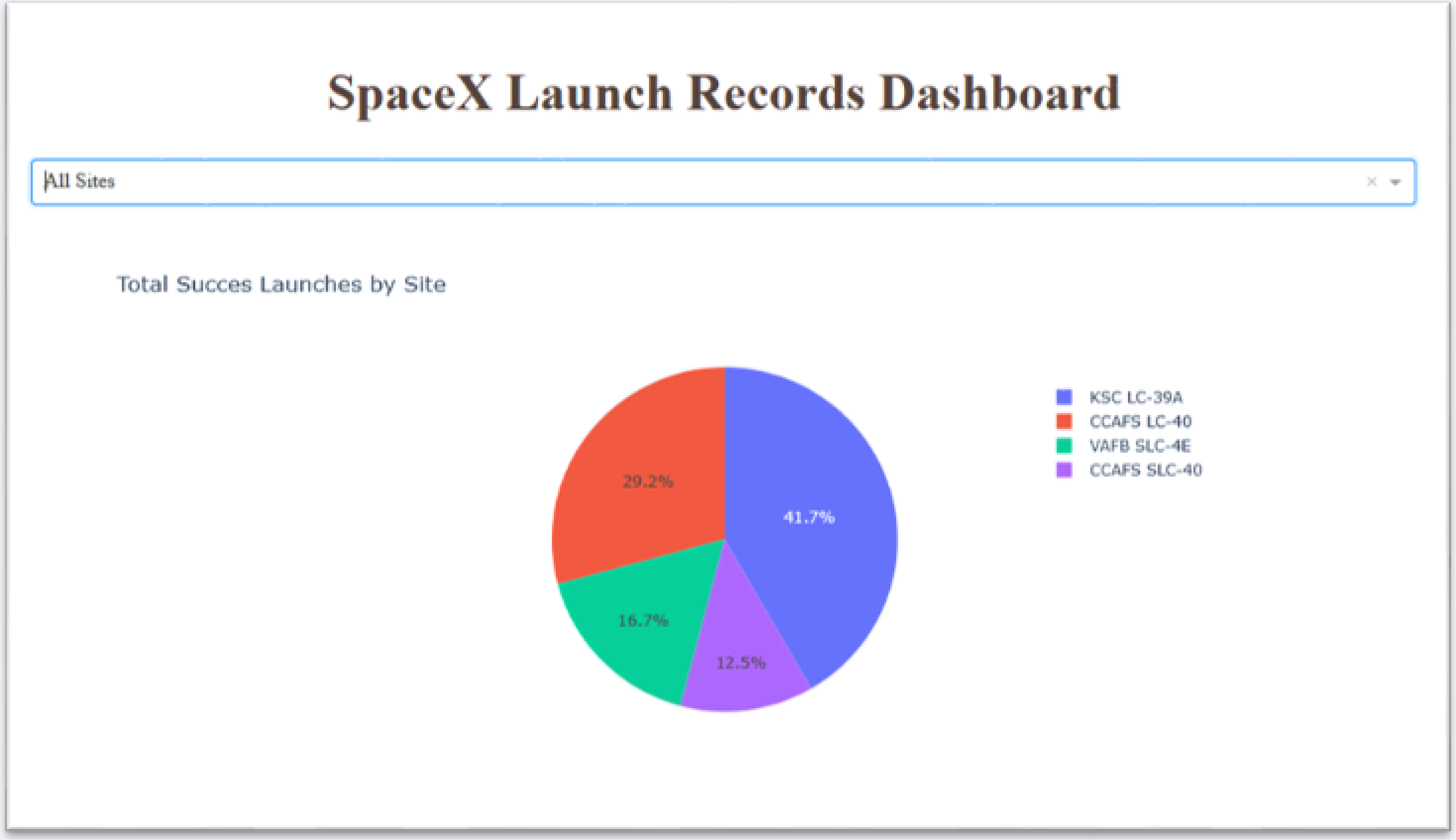
- Launch Site CCAFS – SLC 40 is shown here with its nearest coastline, highway, and railroad each being less than a mile away, while its nearest city is over 20 miles away.
- This suggests it is advantageous for launch sites to be located:
    - Close to railways and highways for easy access and deliveries
    - Close to coastlines for the purpose of aborting the launch in an area that isn't close to people or properties
    - Farther away from nearest cities to avoid damage to people and properties in the case of launch or landing crash



Section 4

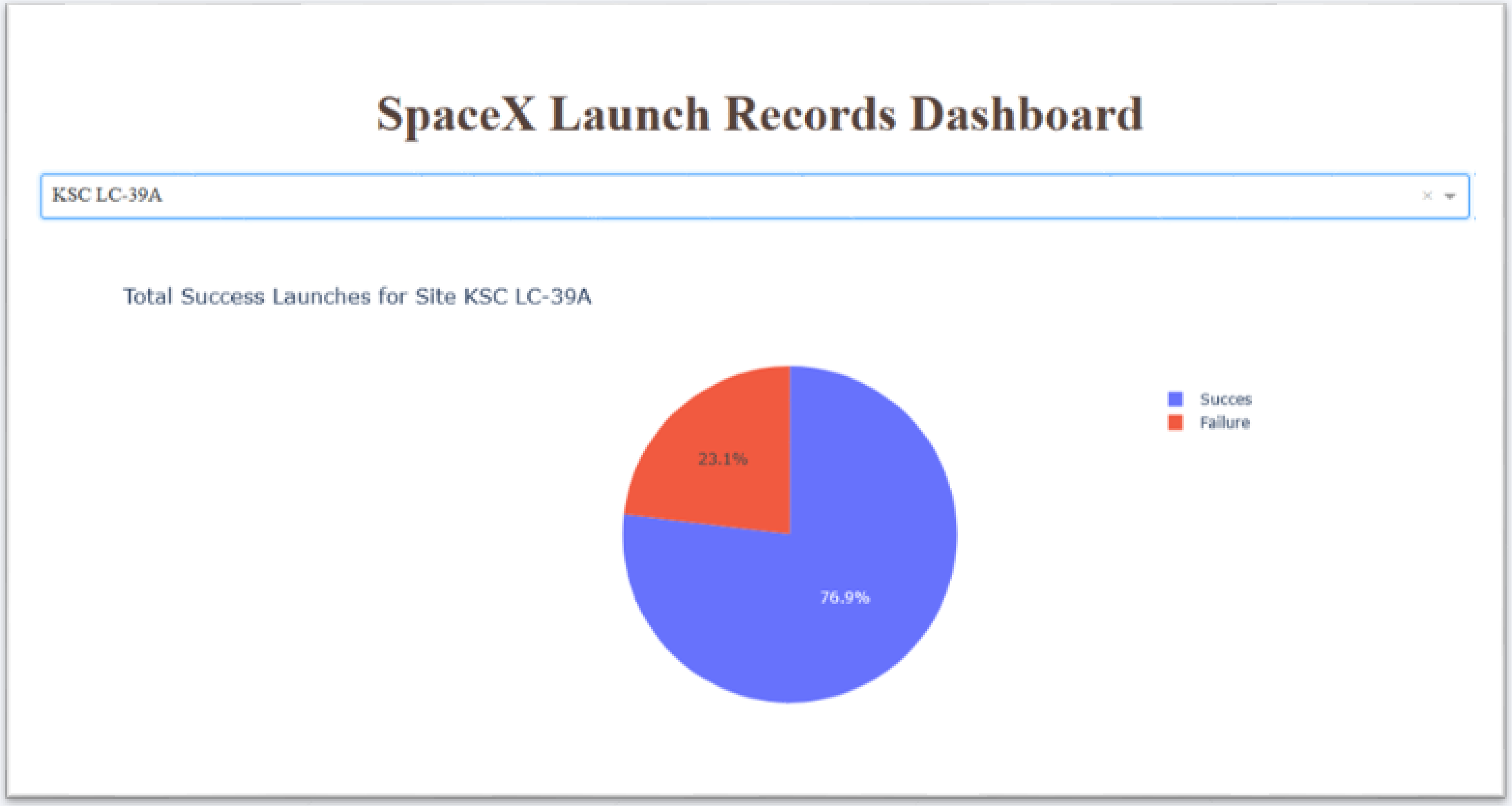
Build a Dashboard with Plotly Dash

Overall Launch Site Success Rates



- Launch site KSC LC-39A has the most successful launches with 41.7% of the successful launches overall
- Launch site CCAFS SLC-40 has the fewest successful launches with only 12.5% of the successful launches overall

Launch Site KSC LC-39A



Launch site KSC LC-39A has the most successful launch rate out of all 4 launch sites at 76.9%

SpaceX Payload Mass vs. Launch Success for All Sites

Payload Mass 0 kg – 5000 kg:

- All 5 Booster Versions (v1.0, v1.1, FT, B4, and B5) have carried payload masses below 5000kg.
- The FT Booster Version has had the most launch successes in this payload range with 11 total.
- For this payload range, the B5 Booster Version has the highest launch success rate at 100%, whereas the v1.0 Booster Version has the lowest launch success rate at 0%.

Payload Mass 5000 kg – 10000 kg:

- Of all the Booster Versions regardless of payload mass, only FT and B4 have carried payload masses in this range.
- The FT Booster Version has had the most launch successes in this payload range with 2 total.
- For this payload range, the FT Booster Version has the highest success rate, although it is low at 33%, whereas the B4 Booster Version has the lowest launch success rate at 20%.

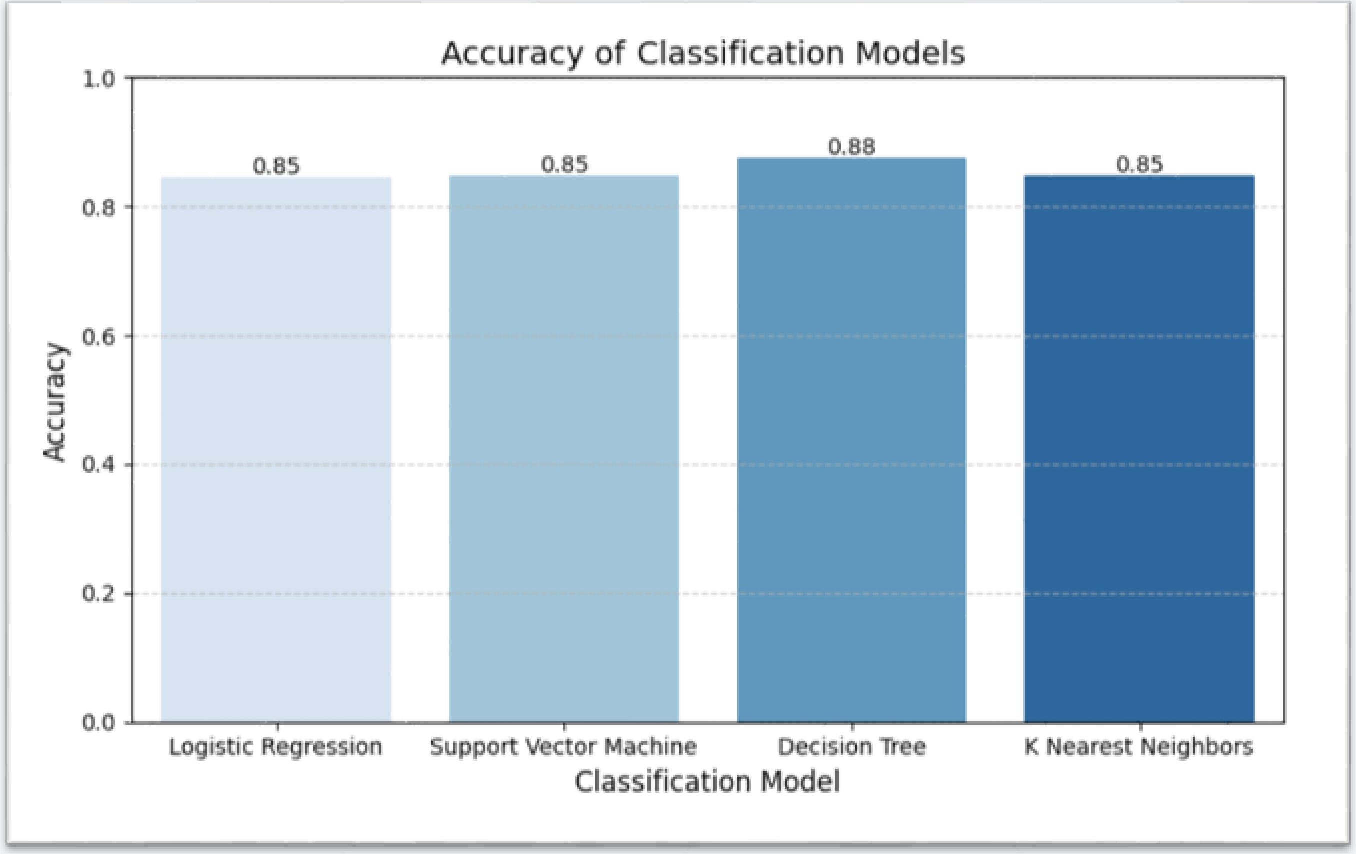
Conclusions:

- The FT Booster Version has had the most launch successes overall with 13 total.
- The B5 Booster Version has the highest launch success rate overall at 100% for payload masses between 0 kg and 10000 kg.
- The v1.0 Booster Version has the lowest launch success rate overall at 0% for payload masses between 0 kg and 10000 kg.
- Booster versions with payload mass ranges between 0 kg and 5000 kg have the highest launch success rate at 46% whereas payload mass ranges between 5000 kg and 10000 kg only have a launch success rate of 27%.

Section 5

Predictive Analysis (Classification)

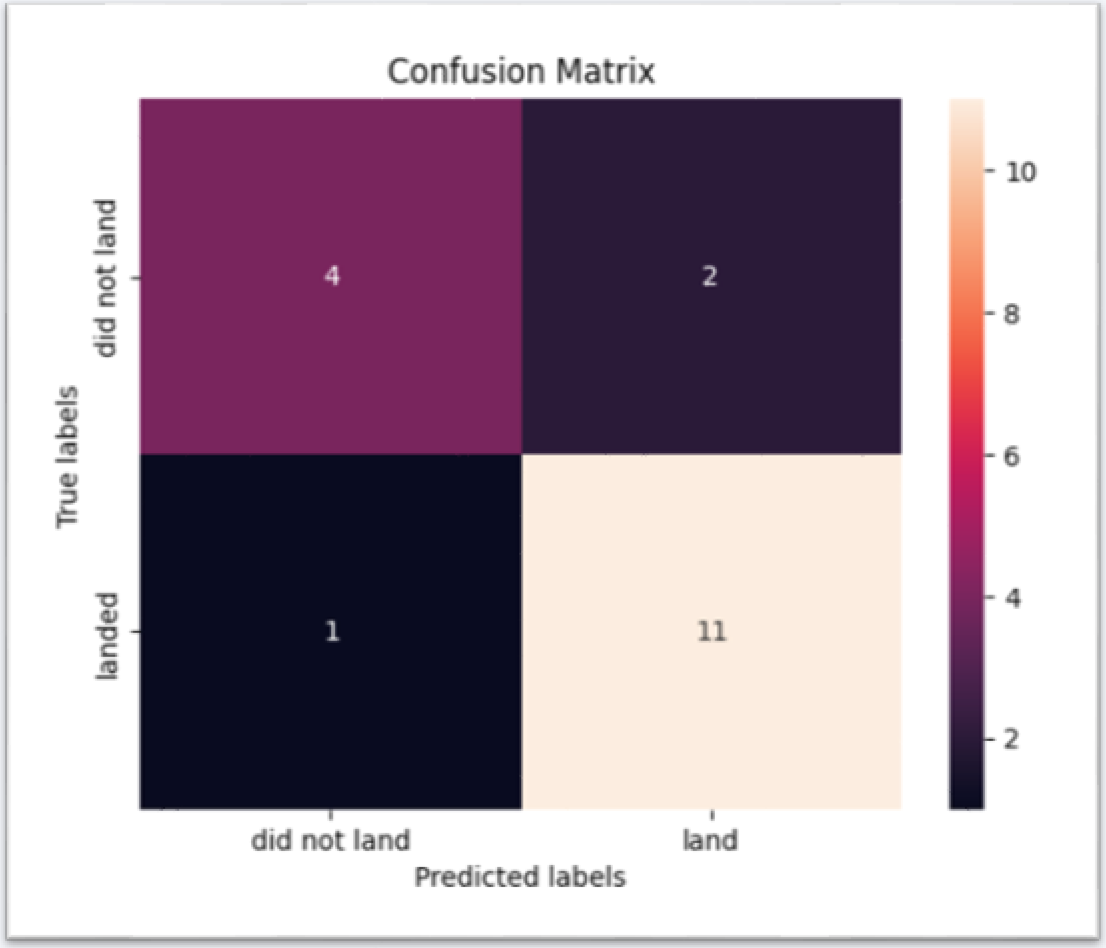
Classification Accuracy



The Decision Tree model has the highest classification accuracy at 88%

Confusion Matrix

- The Decision Tree classification model performed the best, although only slightly better than the other models
- The Confusion Matrix shown here is that of the Decision Tree
- Based on the Confusion Matrix, the Decision Tree model performed as follows:
    - False Positive - 2 launches would have a failed landing even though they were predicted to succeed
    - True Positive - 11 launches would have a successful landing and also were predicted to succeed
    - False Negative - 4 launches would have failed landing and also were predicted to fail
    - True Negative - 1 launch would have a successful landing even though it was predicted to fail
- The Decision Tree model showed 1 fewer false positive than the other models, indicating it performed better. In this type of scenario it costly and dangerous to predict something will succeed to then have it fail



Conclusions

- The overall launch success rate is 67%, showing the more flights performed at launch site, the greater the success rate is at that launch site, suggesting that there would be l higher likelihood of success for each addition launch.
- Of all the launch sites, KSC LC 39A had the highest launch success rate at 77% and captured 42% of the total success across all sites, although launch site doesn't seem to be an predictor of success.
- Launch sites are close to coastlines, highways, and railways, but farther away from cities, speaking to the need for easy equipment deliver and being more isolated to ensure an aborted launch does not harm people or properties.
- Orbit types ES-L1, GEO, HEO, and SSO have the highest launch success rates at 100%.
- The Decision Tree model performed better than Logistic Regression, Support Vector Machine, and K Nearest Neighbors models, but only slightly, which means further modeling would be advisable.

Appendix

Data Sets Used

- Data Collection - API
  - https://api.spacexdata.com/v4
  - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json
- Data Collection – Web Scaping
  - https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Data Wrangling
  - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv
- Exploratory Data Analysis with SQL
  - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv
- Exploratory Data Analysis with Visualization and Machine Learning Prediction
  - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv
- Interactive Visual Analytics and Dashboards
  - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_geo.csv
- Build an Interactive Dashboard with Plotly Dash
  - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_dash.csv
  - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/t4-Vy4iOU19i8y6E3Px_ww/spacex-dash-app.py
- Machine Learning Prediction
  - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_3.csv

Data Sets Created

- Data Collection API
  - dataset_part_1.csv - https://github.com/ddlavigne/SpaceX-Capstone/blob/main/dataset_part_1.csv
- Data Collection with Web Scraping
  - spacex_web_scraped.csv - https://github.com/ddlavigne/SpaceX-Capstone/blob/main/spacex_web_scraped.csv
- Data Wrangling
  - dataset_part_2.csv - https://github.com/ddlavigne/SpaceX-Capstone/blob/main/dataset_part_2.csv
- Exploratory Data Analysis with Visualization
  - dataset_part_3.csv - https://github.com/ddlavigne/SpaceX-Capstone/blob/main/dataset_part_3.csv

Thank you!