IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

SC LEE
2025/1/24

# Outline

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

# Executive Summary

- Summary of methodologies:
  - We collect the required information and perform necessary preprocessing on them.
  - We performed some exploration of the data, including using data visualization and data search (SQL) to help us understand the pattern of the data we collected.
  - We train four different prediction models and calculate their accuracy.
- Summary of all results:
  - We created four scatter charts to confirm the relationship between the two variables; a bar chart showing the success rate of different orbit types, and a line chart showing the yearly trend of Launch success.
  - We present the SQL search results for the conditions we are interested in.
  - We created a folium map of launch sites.
  - We created a dashboard of mission success rates and payloads at launch sites.
  - We calculated the accuracy and confusion matrix of the four prediction models.

# Introduction

- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch.
- Our main goal is to build a model to predict whether the first stage of SpaceX Falcon 9 will land successfully.

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - Retrieving information form the json files of SpaceX API.
  - Webscraping the information form the HTML file of Wikipedia.
- Perform data wrangling
  - We deal missing values and label whether the land of first stage is successful or not.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We use four different models: logistic regression, SVM, decision tree, and KNN.

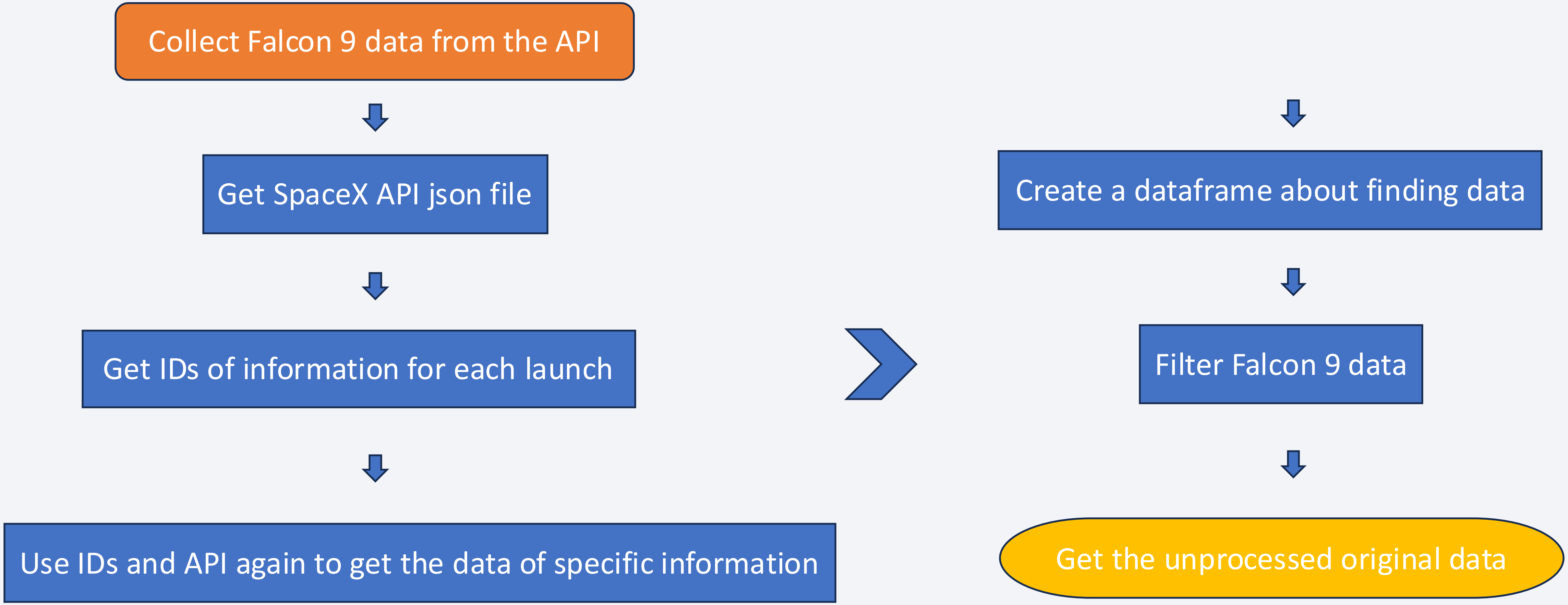
# Data Collection

- We collect information from two places: one is SpaceX API, the other is Wikipedia.
- The information which we get from SpaceX API is the json file. During the data collection process, we will request data from the SpaceX API many times. Please see the Data Collection – SpaceX API slide for details.
- In Wikipedia, there is a table with SpaceX past launch records. We use web scraping to find the table in HTML file. Specifically, we parse HTML file using package BeautifulSoup and find the data we need. Please see the Data Collection – Scraping slide for details.

# Data Collection – SpaceX API

- For SpaceX API, we get the information we want from the json files as follows.
  1. Get the IDs of the information of each launch from the json file of the API "/launches/past".
  2. Use the API again to get information about the launches using the IDs given for each launch. For example, the columns 'rocket', 'payloads', 'launchpad', and 'cores'. In this part, we have some simple functions to facilitate to use IDs to retrieve data.
  3. Add all the collected data into a dictionary and convert it into a dataframe.
  4. Filter the dataframe to only include 'Falcon 9' launches.
- Click here for GitHub URL of the SpaceX API notebook.

# Data Collection – SpaceX API Flowchart



Collect Falcon 9 data from the API

Get SpaceX API json file

Get IDs of information for each launch

Use IDs and API again to get the data of specific information

Create a dataframe about finding data

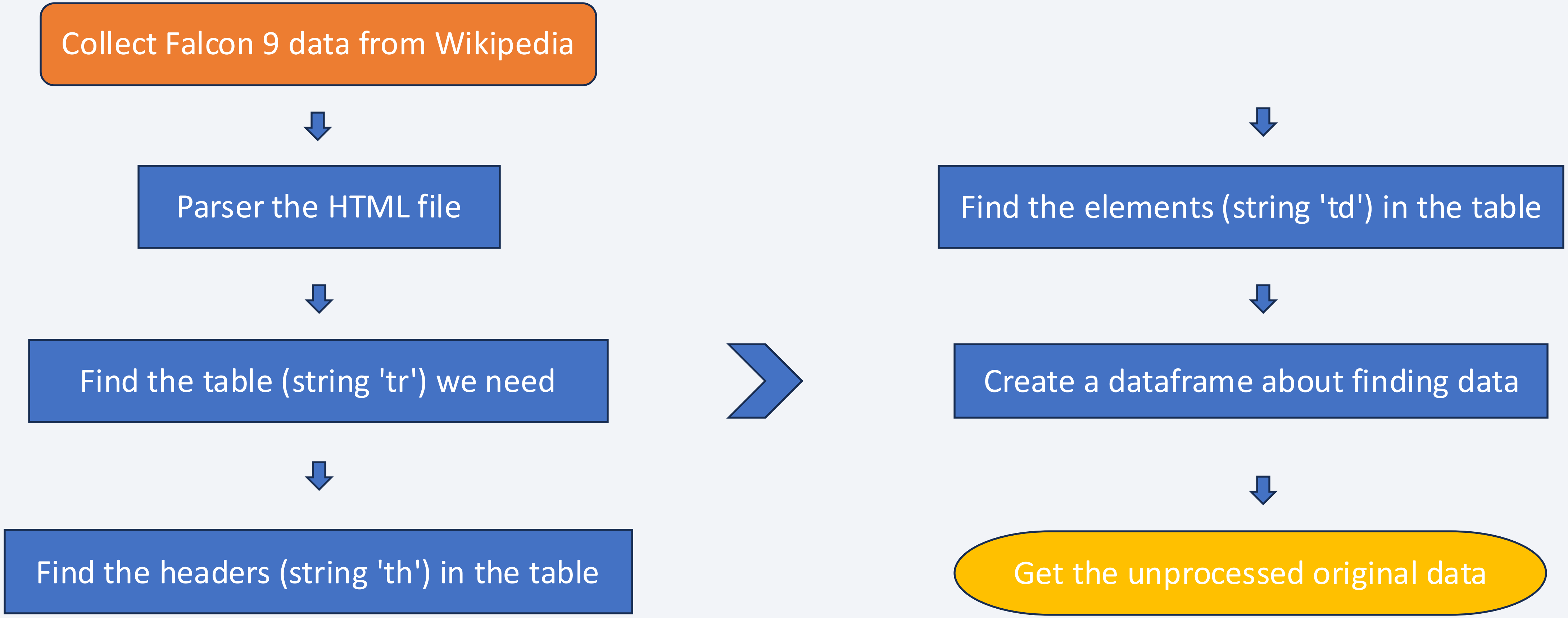
Filter Falcon 9 data

Get the unprocessed original data

# Data Collection – Scraping

- For Wikipedia, we get the information we want from the HTML file as follows.
  1. Use package "BeautifulSoup" to parser the HTML file.
  2. Use command "find_all" to find the specific string in HTML file. In this case, we find 'tr' which means table in HTML. Note that we also need to check the table number we need, for example, the third table in this project.
  3. Find all strings 'th' which means the headers of the table we have found in step 2.
  4. Find all strings 'td' which means the elements of the table we have found in step 2.
  5. Add all the collected data into a dictionary iteratively and convert it into a dataframe.
- Click here for GitHub URL of the Scraping notebook.

# Data Collection – Scraping Flowchart


Collect Falcon 9 data from Wikipedia
Parser the HTML file
Find the table (string 'tr') we need
Find the headers (string 'th') in the table
Find the elements (string 'td') in the table
Create a dataframe about finding data
Get the unprocessed original data

# Data Wrangling

- First, we preprocess our data:
  - Check if there are some missing values. In this project, column 'PayloadMass' have 5 missing values. We use the mean to process missing values.
  - Identify which columns are numerical and categorical.
- Exploratory Data Analysis:
  - Calculate the number of launches on each site.
  - Calculate the number and occurrence of each orbit.
  - Calculate the number and occurrence of mission outcome of the orbits.
- Create a landing outcome label from Outcome column. We named this column 'Class'. If the value is zero, the first stage did not land successfully; one means the first stage landed successfully.
- Click here for GitHub URL of the Data Wrangling notebook.

# EDA with Data Visualization

- We plot three different types of charts, namely scatter chart, bar chart, and line chart.
- For scatter chart, we use it to compare the relationship between two variables, such as 'FlightNumber' vs 'PayloadMass', 'FlightNumber' vs 'LaunchSite', and so on.
- For bar chart, we want to visually check if there are any relationship between success rate and orbit type.
- For line chart, we visualize the launch success yearly trend.
- Click here for GitHub URL of the EDA with Data Visualization notebook.

# EDA with SQL (part 1)

- SQL queries summary:
  - The names of the unique launch sites
  - 5 records where launch sites begin with the string 'CCA'
  - The total payload mass carried by boosters launched by NASA (CRS)
  - Average payload mass carried by booster version F9 v1.1
  - The date when the first successful landing outcome in ground pad was achieved
  - The names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

# EDA with SQL (part 2)

- SQL queries summary:
  - The total number of successful and failure mission outcomes
  - The names of the booster versions which have carried the maximum payload mass
  - The records which will display the month names, failure landing outcomes in drone ship, booster versions, launch site for the months in year 2015
  - The count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order
- Click here for GitHub URL of the EDA with SQL notebook.

# Build an Interactive Map with Folium

- First, we add four launch sites to a folium map. Each coordinate have a highlighted circle area with a text label. In this way, we can intuitively present the latitude and longitude information on the map.
- Second, we mark the success/failed launches for each site on the map. We create "MarkerCluster" object to help identify overlapping markers at the same location. Also, we add colors to distinguish between success (green) and failure (red).
- Finally, we mark the closest coastline and calculate the distance between the coastline point and the launch site. And add the distance line in the map for better visibility. We also do the same with highways.
- Click here for GitHub URL of the Interactive Map with Folium notebook.

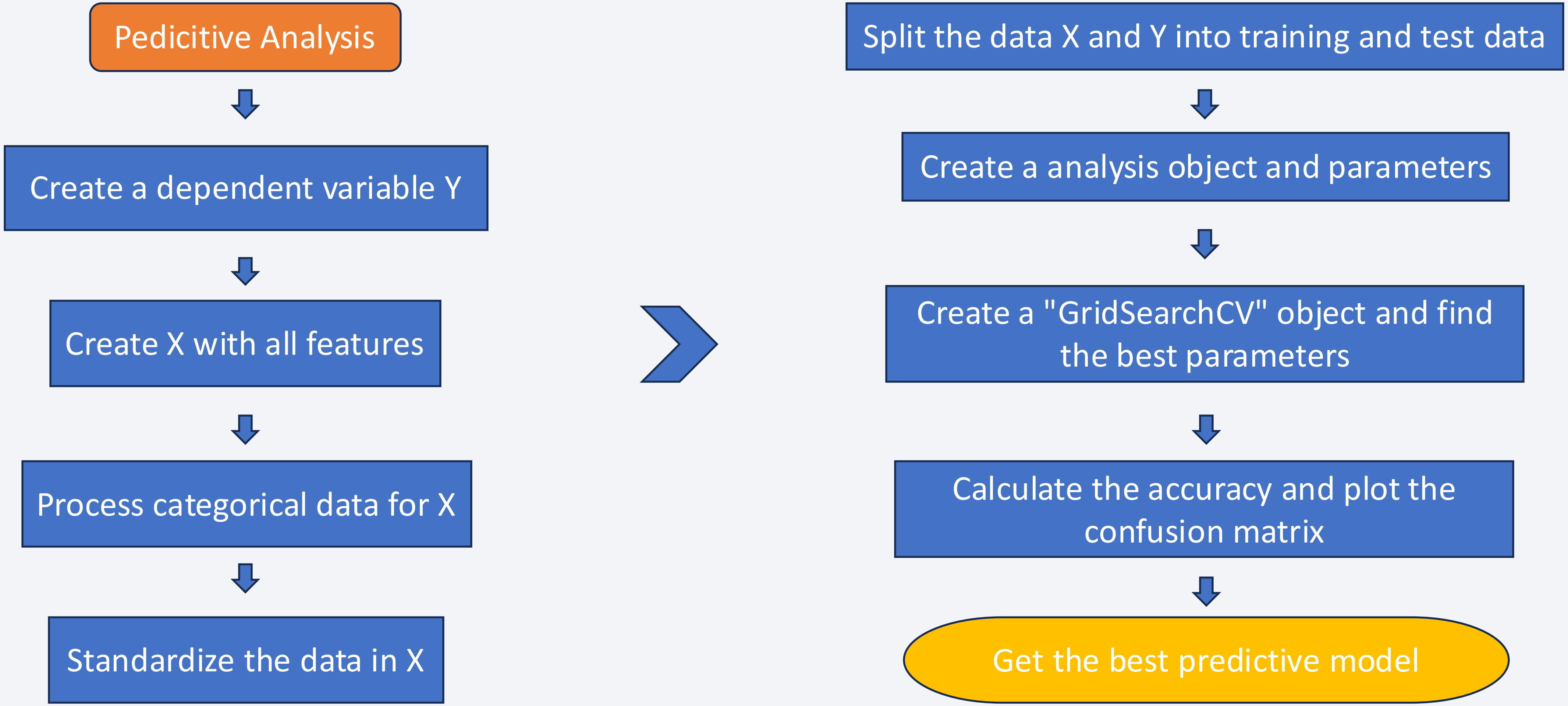
# Build a Dashboard with Ploty Dash

- We add a dropdown list to enable launch site selection. There are 5 options, four different locations and one all-inclusive option.
- We add a pie chart to show the total successful launches count for all sites. If a specific launch site was selected, we show the success vs failed counts for the site.
- We add a scatter chart to show the correlation between payload and launch success.
- We add a slider to select payload range, so that we can control the range of the scatter chart.
- Click here for GitHub URL of the Dashboard with Ploty Dash notebook.

# Predictive Analysis (Classification)

- The predictive analysis process is as follows.
  1. Create a variable Y that contains only the column 'Class' in our data. It is the dependent variable.
  2. Create a variable X that contains all the features in our data. They are the independent variables.
  3. Use dummy variables to process categorical data for X.
  4. Standardize the data in X.
  5. Split the data X and Y into training and test data. We set the test set to 20% of the total and random_state to 2.
  6. Create a analysis object and corresponding appropriate parameters. In this project, we use four different analysis models, namely logistic regression, support vector machine, decision tree, and k nearest neighbors.
  7. Create a "GridSearchCV" object with cv = 10 which will find the best parameters from parameters of the model we chose.
  8. Calculate the accuracy of each model and plot the confusion matrix. Compare them to find the best model.
- Click here for GitHub URL of the Predictive Analysis notebook.

# Predictive Analysis (Classification) Flowchart



Pedicitive Analysis

Create a dependent variable Y

Create X with all features

Process categorical data for X

Standardize the data in X

Split the data X and Y into training and test data

Create a analysis object and parameters

Create a "GridSearchCV" object and find the best parameters

Calculate the accuracy and plot the confusion matrix

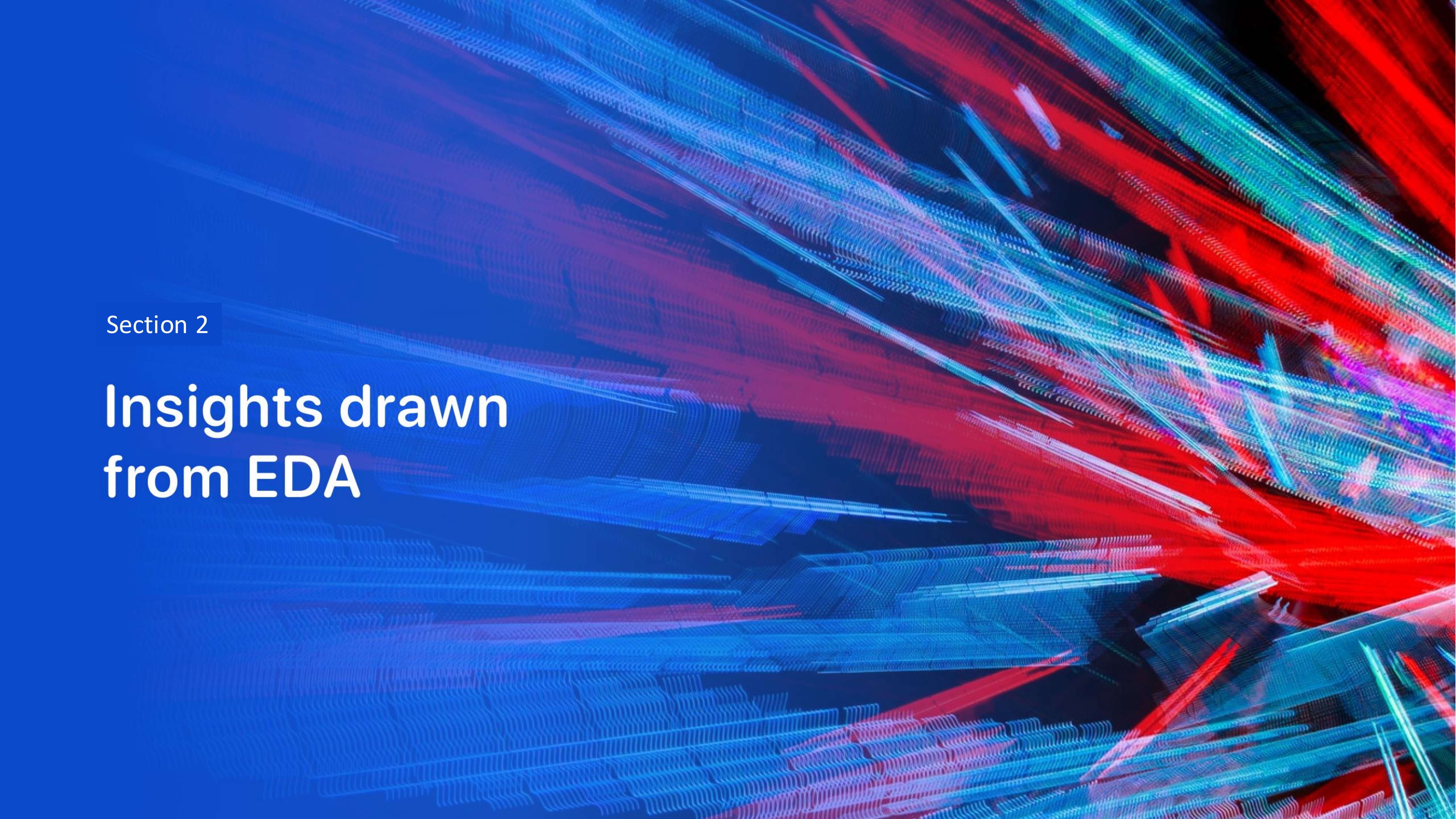
Get the best predictive model

# Results

- Exploratory data analysis results will be shown in section 2 (includes Data visualization and SQL).
- Interactive analytics demo in screenshots will be shown in section 3 and 4.
  - Section 3 is for interactive map with folium.
  - Section 4 is for the Dashboard with Plotly Dash.
- Predictive analysis results will be shown in section 5.

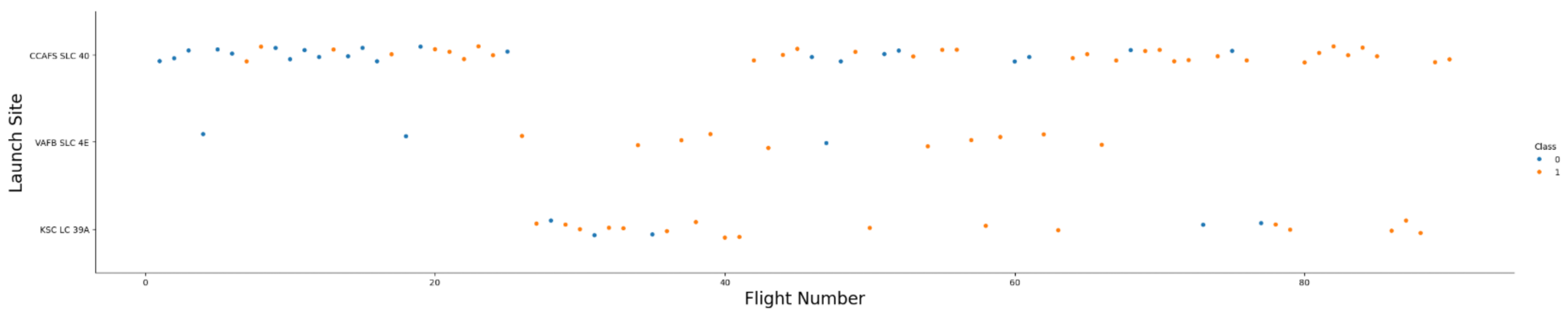
Section 2

# Insights drawn from EDA



# Flight Number vs. Launch Site

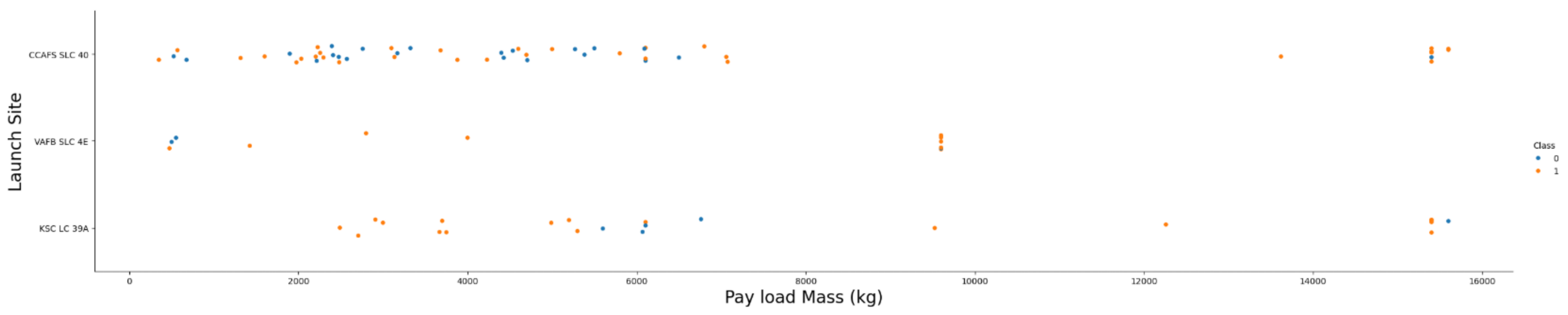
- A scatter plot of Flight Number vs. Launch Site



- We can find that at low flight numbers, launch sites CCAFS-SLC 40 and VAFB-SLC 4E have lower success rates; conversely, at high flight numbers, all launch sites have higher success rates.

# Payload vs. Launch Site

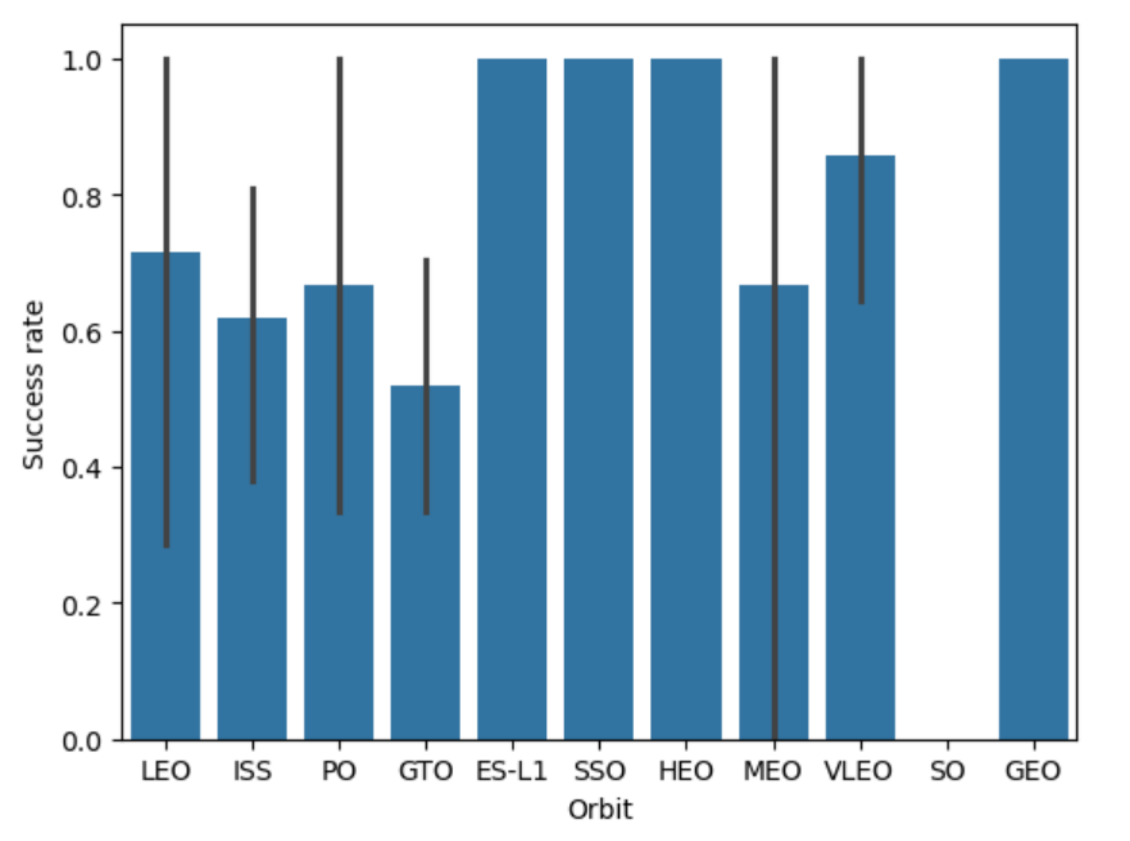
- A scatter plot of Payload vs. Launch Site



- We can find that there are no rockets launched for heavy payload mass (greater than 10000 kg) at the VAFB-SLC 4E launch site. In addition, for launch sites CCAFS-SLC 40, the payload does not seem to have a significant relationship with its success rate.

# Success Rate vs. Orbit Type

- A bar chart for the success rate of each orbit type

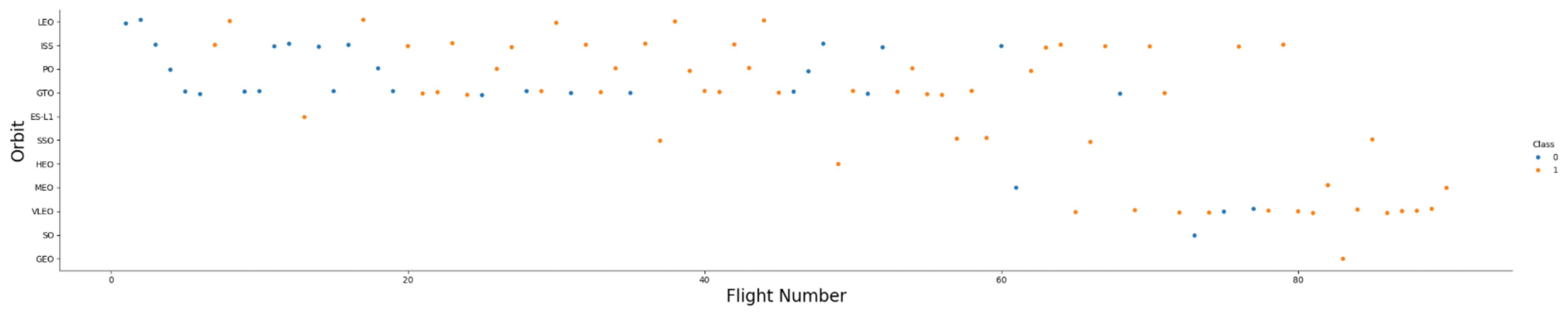


- We can find that orbits ES-L1, SSO, HEO and GEO have the highest success rate, while orbit SO has the lowest success rate.

- The other orbits have a success rate of about 60%.

# Flight Number vs. Orbit Type

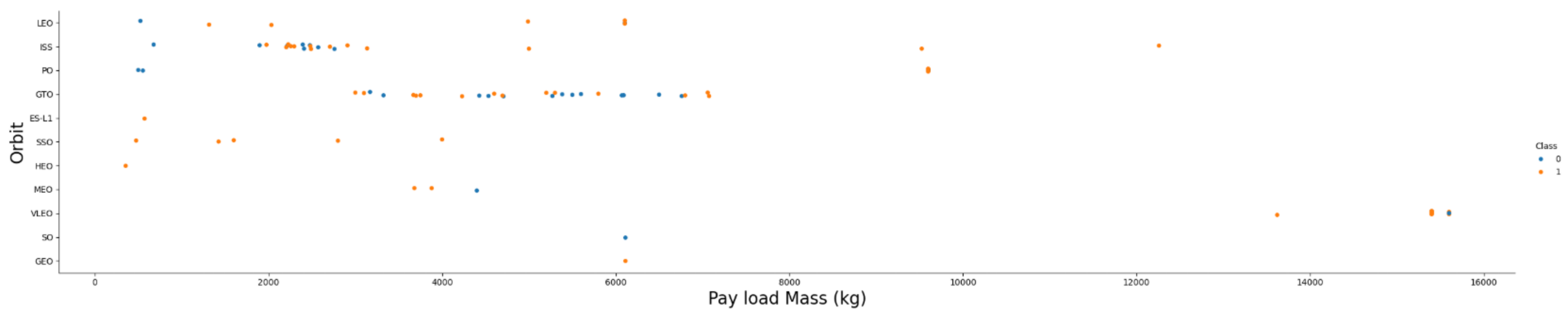
- A scatter point of Flight number vs. Orbit type



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

# Payload vs. Orbit Type

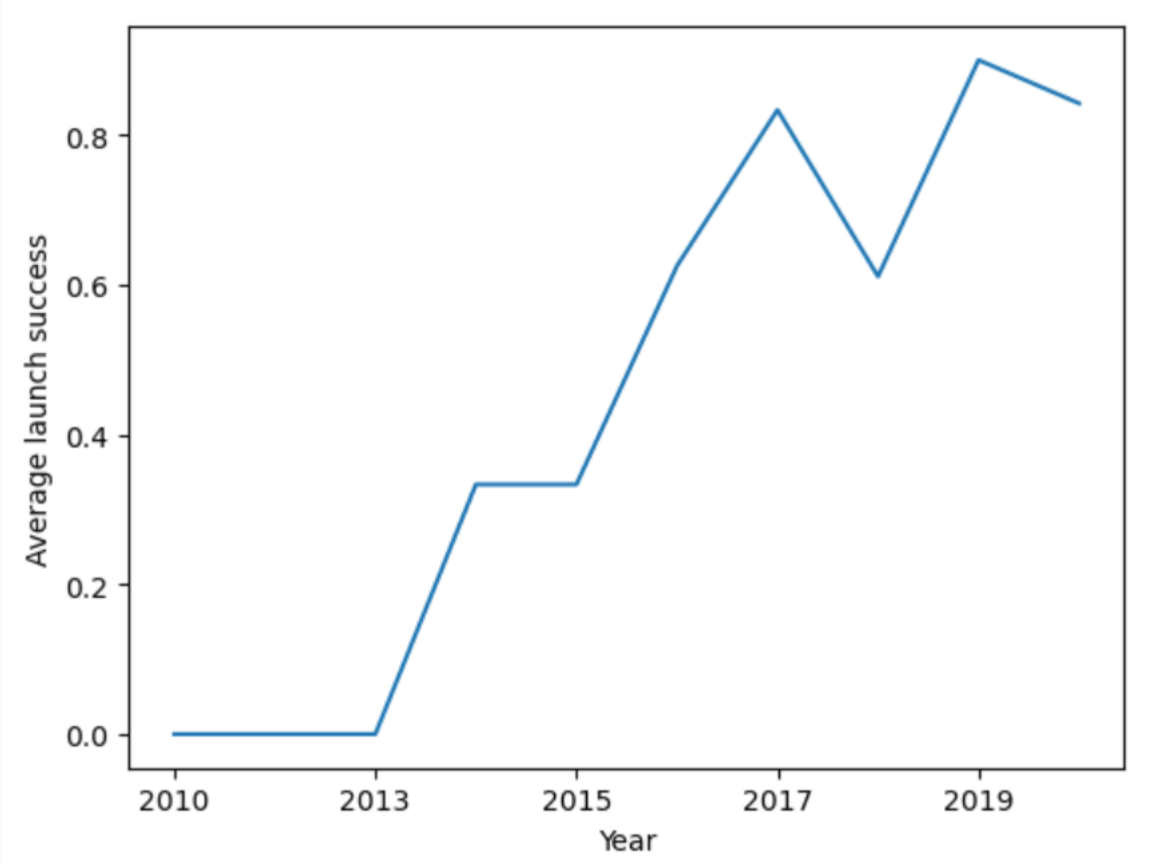
- A scatter point of payload vs. orbit type



- We can find that the successful landing or positive landing rate are more for Polar, LEO and ISS with heavy payloads. However for GTO we cannot distinguish this well as both positive landing rate and negative landing (unsuccessful mission) are both there here.

# Launch Success Yearly Trend

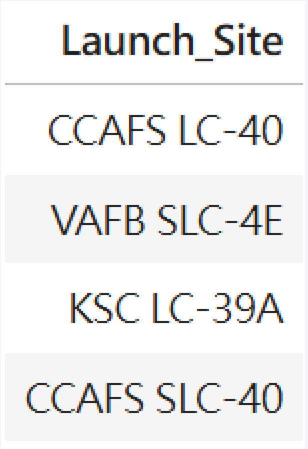
- Show a line chart of yearly average success rate



- We can find that the success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.

# All Launch Site Names

- Find the names of the unique launch sites

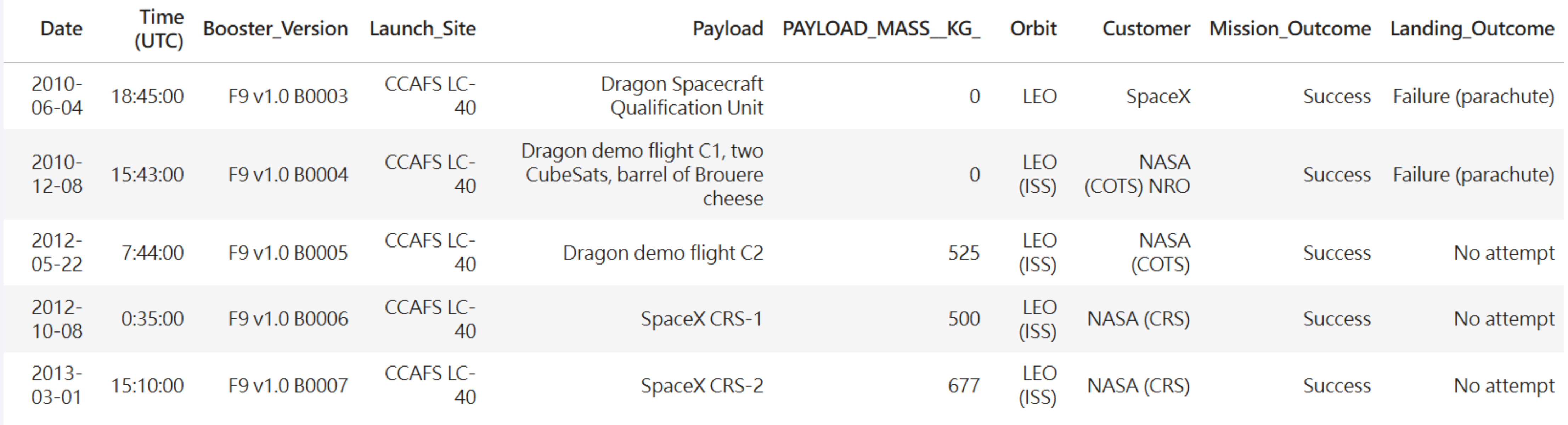


| Launch_Site |
|---|
| CCAFS LC-40 |
| VAFB SLC-4E |
| KSC LC-39A |
| CCAFS SLC-40 |

- There are four different launch sites, namely CCAFS LC-40, VAFB SLC-4E, KSC LC-39A and CCAFS SLC-40.

# Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

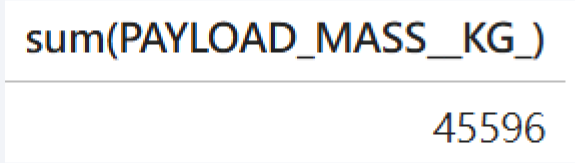


| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing_Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 2010-06-04 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 2010-12-08 | 15:43:00 | F9 v1.0 B0004 | CCAFS LC-40 | Dragon demo flight C1, two CubeSats, barrel of Brouere cheese | 0 | LEO (ISS) | NASA (COTS) NRO | Success | Failure (parachute) |
| 2012-05-22 | 7:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 2012-10-08 | 0:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 2013-03-01 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

- These 5 records are sorted from earliest to latest.

# Total Payload Mass

- Calculate the total payload carried by boosters from NASA

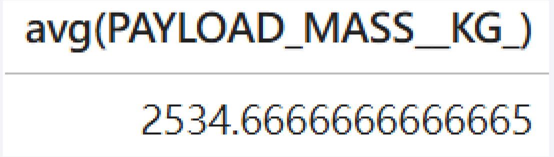


| sum(PAYLOAD_MASS__KG_) |
|---:|
| 45596 |

- The total payload carried by boosters from NASA is 45596 kg.

# Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

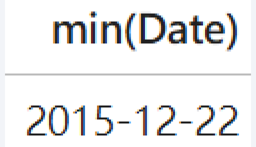


| avg(PAYLOAD_MASS__KG_) |
|---:|
| 2534.6666666666665 |

- The average payload mass carried by booster version F9 v1.1 is 2534.67 kg.

# First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

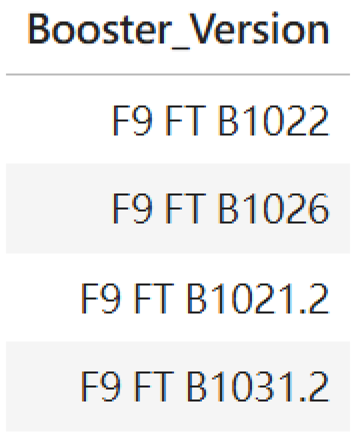


| min(Date) |
|---|
| 2015-12-22 |

- The dates of the first successful landing outcome on ground pad is 2015-15-22.

# 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

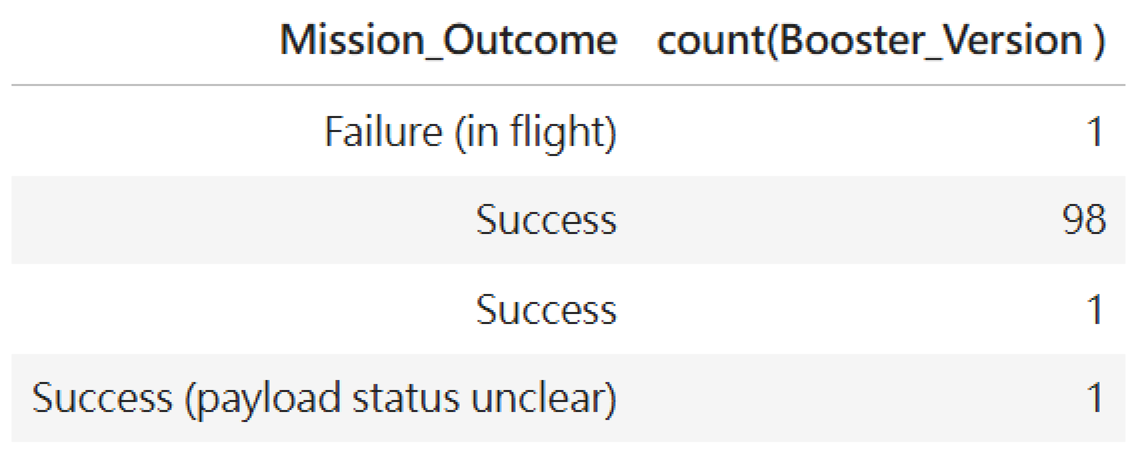


| Booster_Version |
|---:|
| F9 FT B1022 |
| F9 FT B1026 |
| F9 FT B1021.2 |
| F9 FT B1031.2 |

- There are four boosters that satisfy the above conditions.

# Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

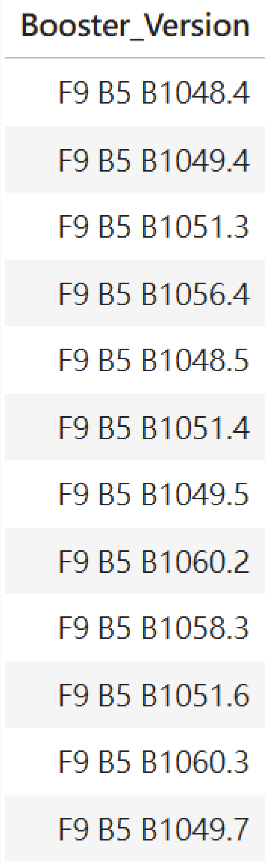


| Mission_Outcome | count(Booster_Version ) |
|---:|---:|
| Failure (in flight) | 1 |
| Success | 98 |
| Success | 1 |
| Success (payload status unclear) | 1 |

- Thera is only one failure (in flight) mission outcomes.

# Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

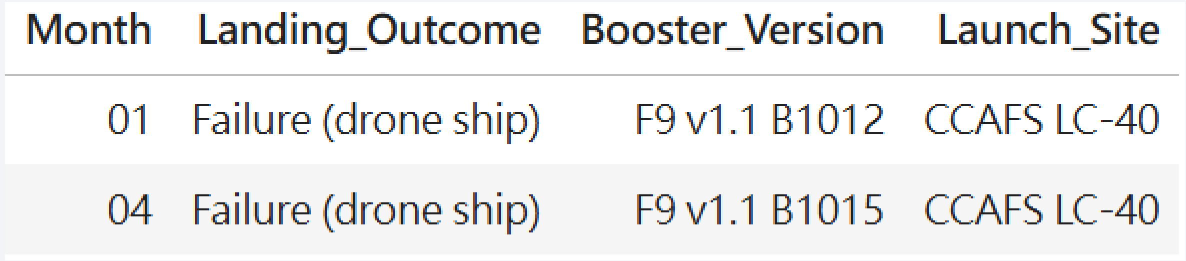


| Booster_Version |
|---|
| F9 B5 B1048.4 |
| F9 B5 B1049.4 |
| F9 B5 B1051.3 |
| F9 B5 B1056.4 |
| F9 B5 B1048.5 |
| F9 B5 B1051.4 |
| F9 B5 B1049.5 |
| F9 B5 B1060.2 |
| F9 B5 B1058.3 |
| F9 B5 B1051.6 |
| F9 B5 B1060.3 |
| F9 B5 B1049.7 |

- There are 12 boosters that carried the maximum payload mass.

# 2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for the months in year 2015

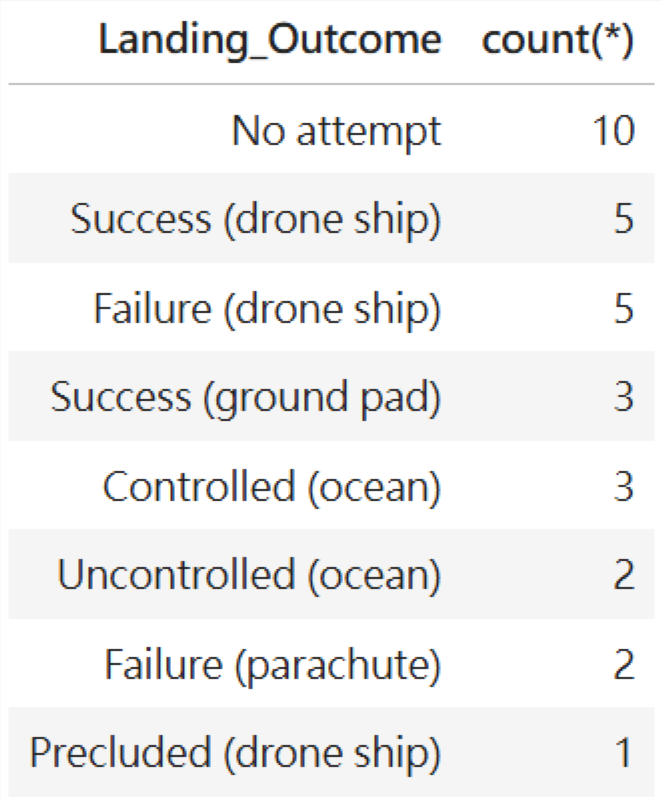


| Month | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| 01 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

- There were two failed landing outcomes in drone ship in 2015, in January and April.

# 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



| Landing_Outcome | count(*) |
|---:|---:|
| No attempt | 10 |
| Success (drone ship) | 5 |
| Failure (drone ship) | 5 |
| Success (ground pad) | 3 |
| Controlled (ocean) | 3 |
| Uncontrolled (ocean) | 2 |
| Failure (parachute) | 2 |
| Precluded (drone ship) | 1 |

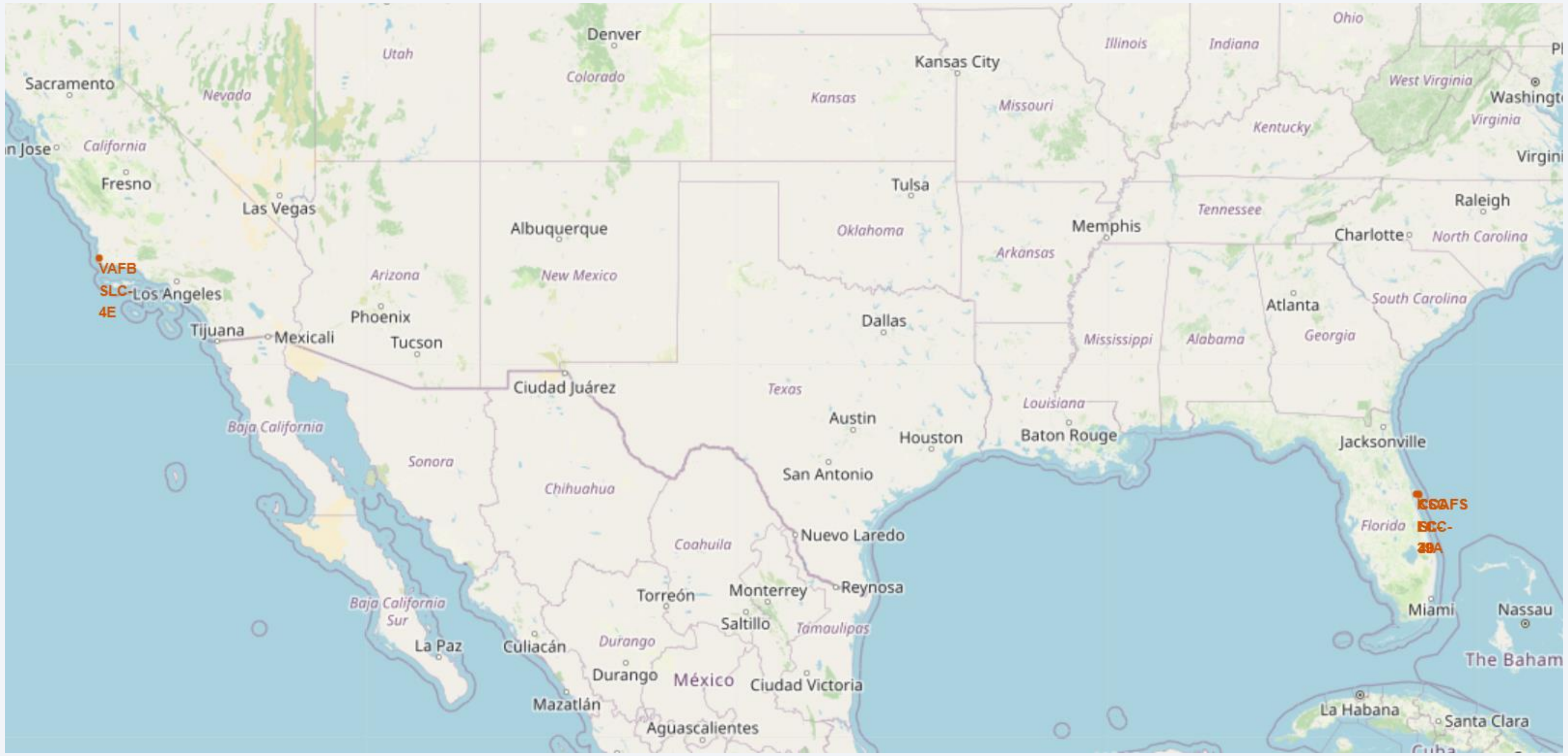
- During this period, no landing attempt had the most results, with 10, followed by success (drone ship) and failure (drone ship), with 5 each.

Section 3

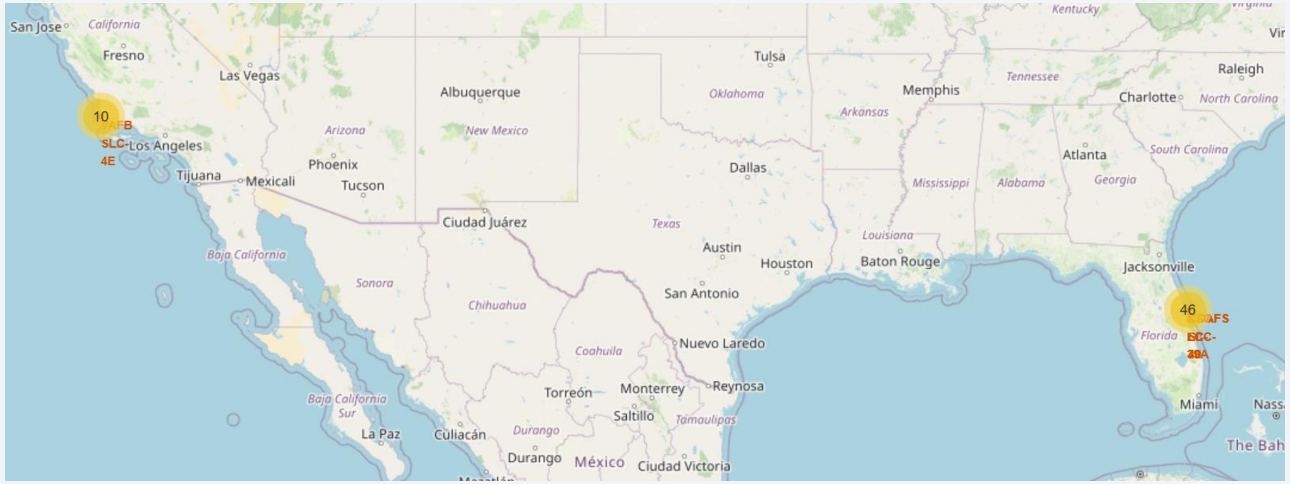
# Launch Sites Proximities Analysis

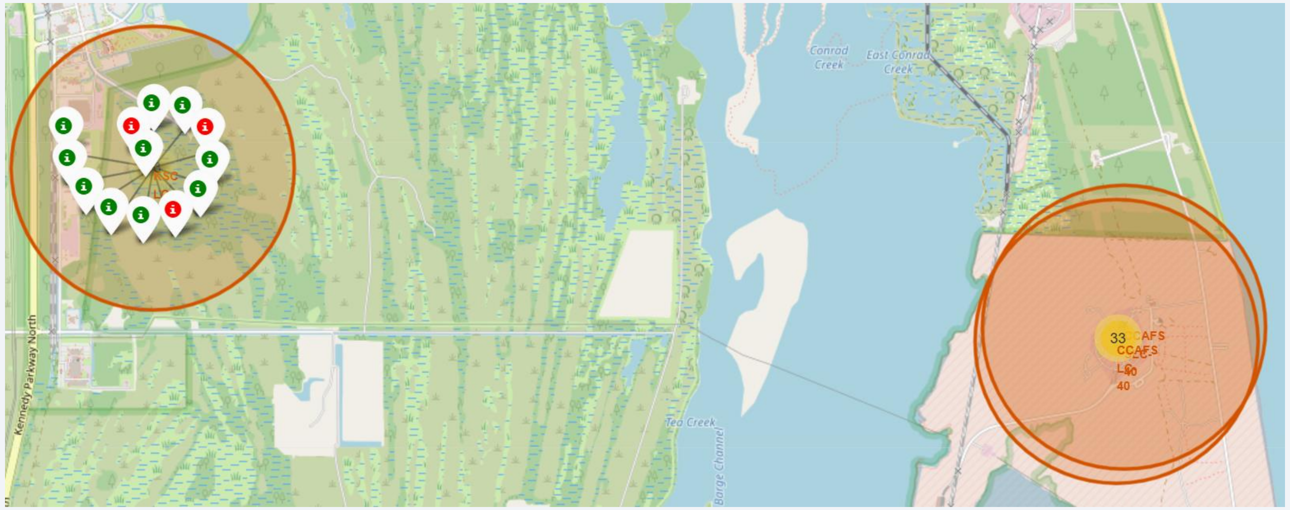
# Mark all launch sites on a Folium map

- We mark all launch sites on map as below. We can find that all launch sites are in very close proximity to the coast.



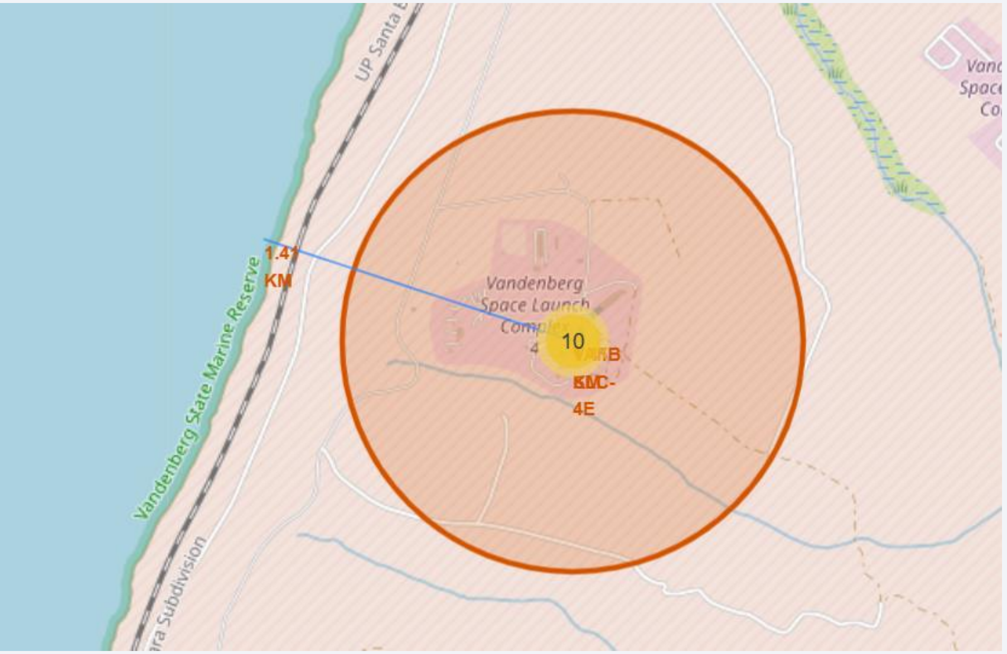
# Mark the success/failed launches for each site on the map

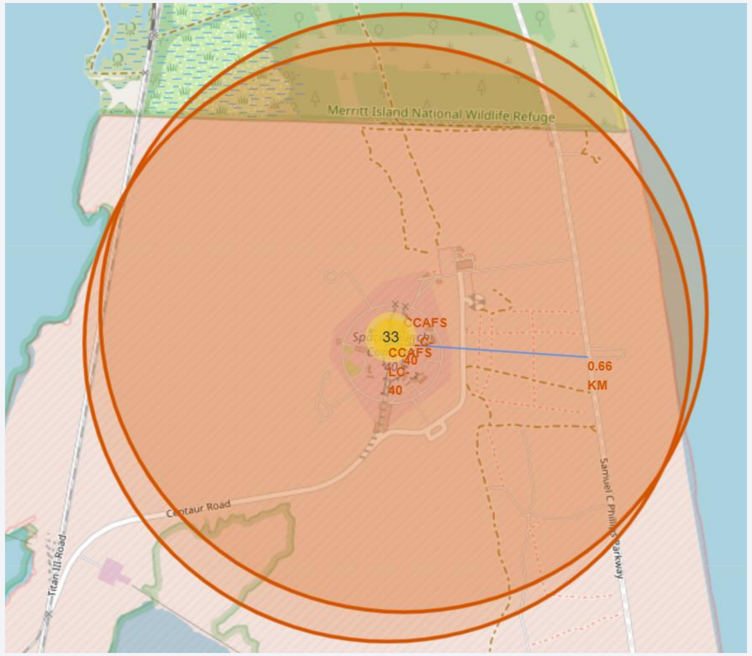




- We use "MarkerCluster" object to help identify overlapping markers at the same location.
- We can find that there are 10 missions on the west coast and 46 missions on the east coast.
- We add colors to distinguish between success (green) and failure (red).

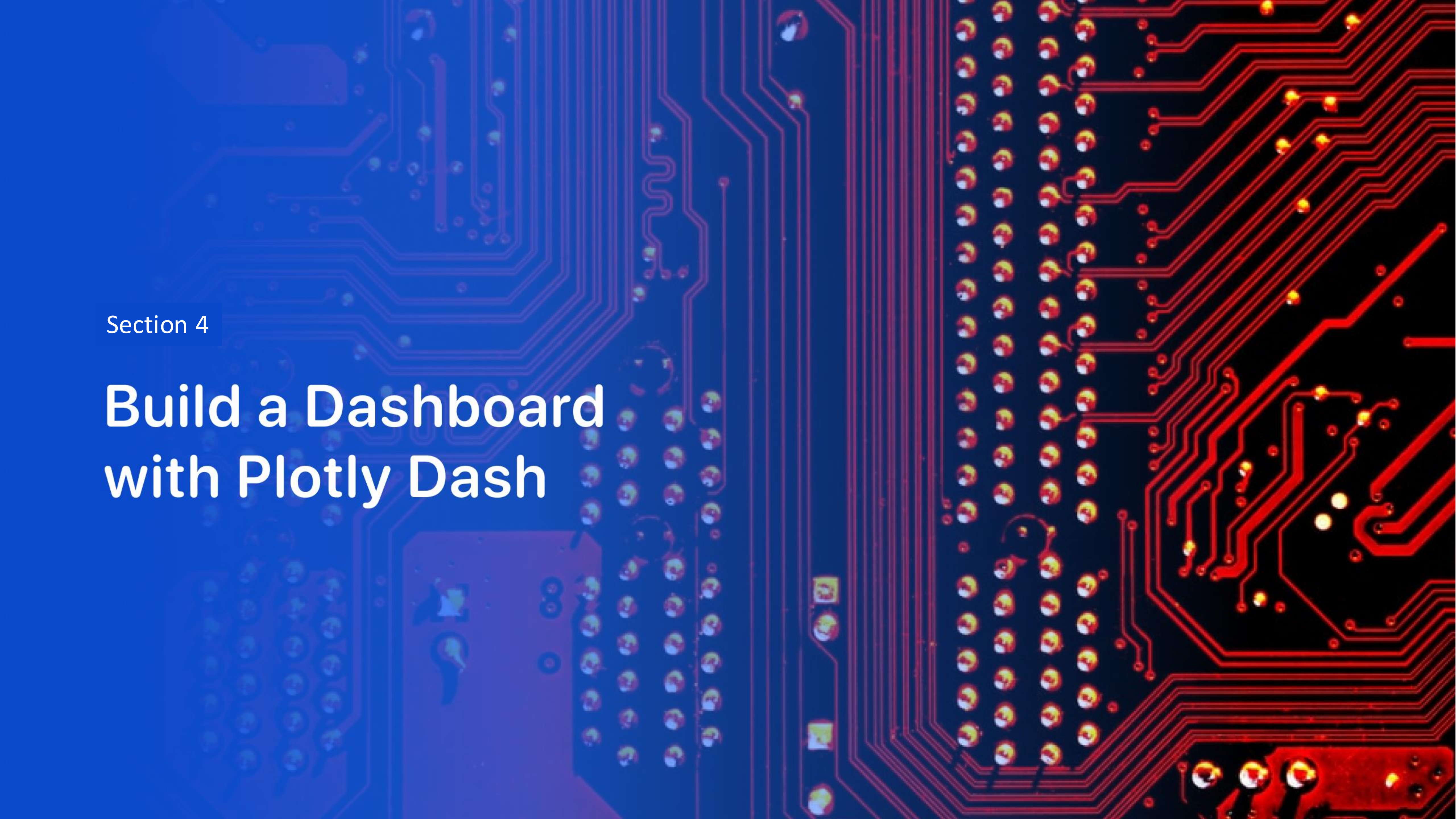
# Calculate the distances between a launch site to its proximities


UP Santa
Vandenberg State Marine Reserve
1.41 KM
Vandenberg Space Launch Complex 4
10
VAFB SLC-4E
ra Subdivision


Merritt Island National Wildlife Refuge
33
CCAFS
CCAFS LC-40
0.66 KM
Centaur Road
Samuel C Phillips Parkway
Titan III Road

- Launch site VAFB SLC-4E is less than 2 km from the nearest coast and railway.

- Launch sites CCAFS SLC-40 and CCAFS LC-40 are less than 1 km from the nearset highway and coast.

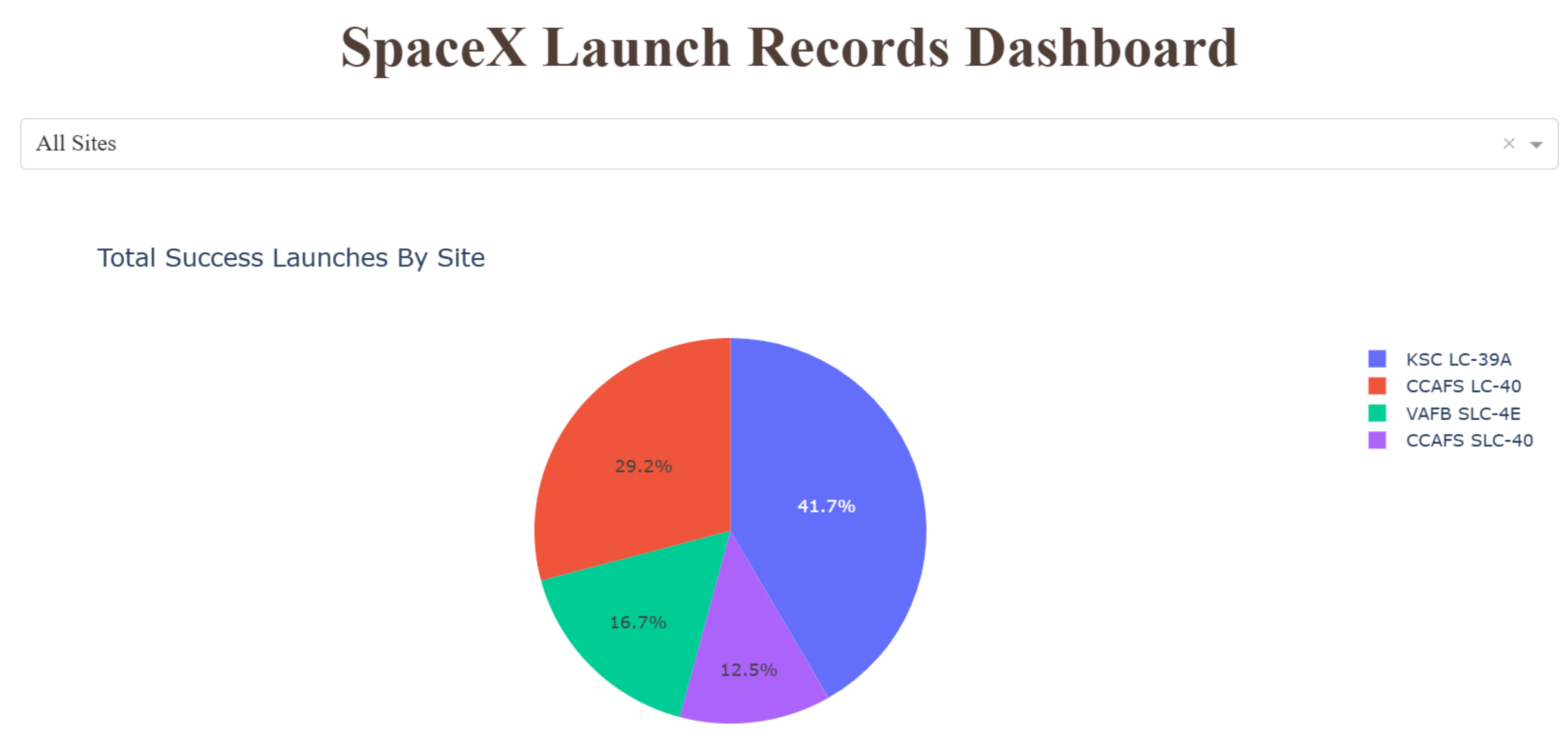


Section 4

# Build a Dashboard with Plotly Dash

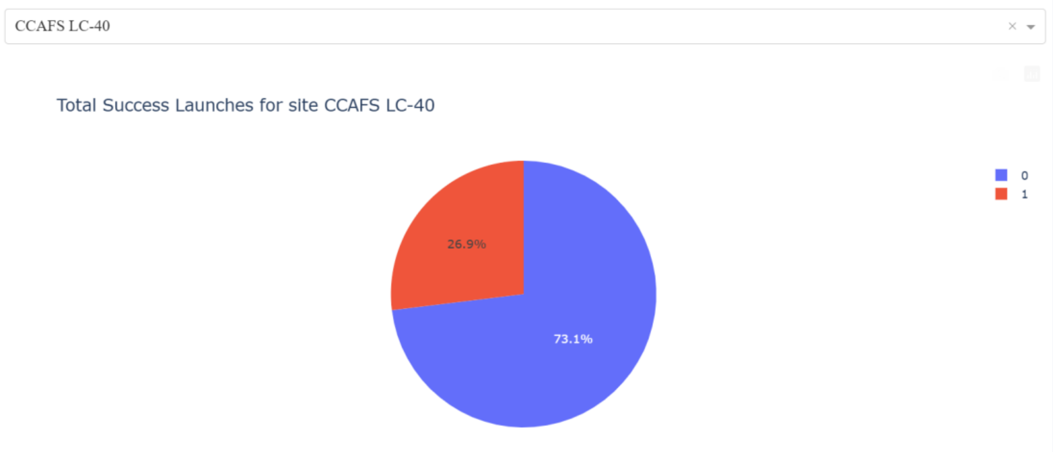
# The pie chart of total successful launches count for all sites (Dashboard)

- We can find that among all the successful launch sites, the site KSC LC-39A has the highest percentage, and the site CCAFS SLC-40 has the lowest.



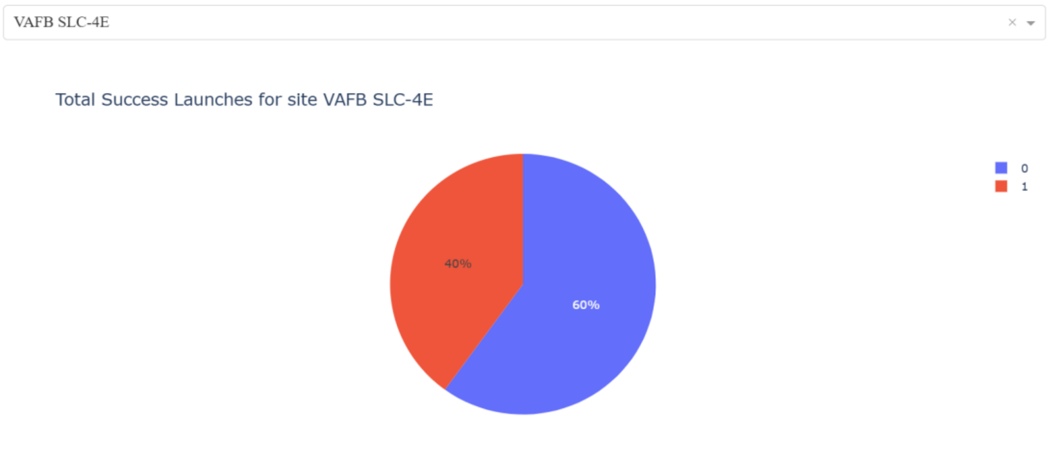
# The pie charts of success rate for specific launch site (Dashboard)

- We can find that the site KSC LC-39A has the highest success rate, while the site CCAFS LC-40 has the lowest.



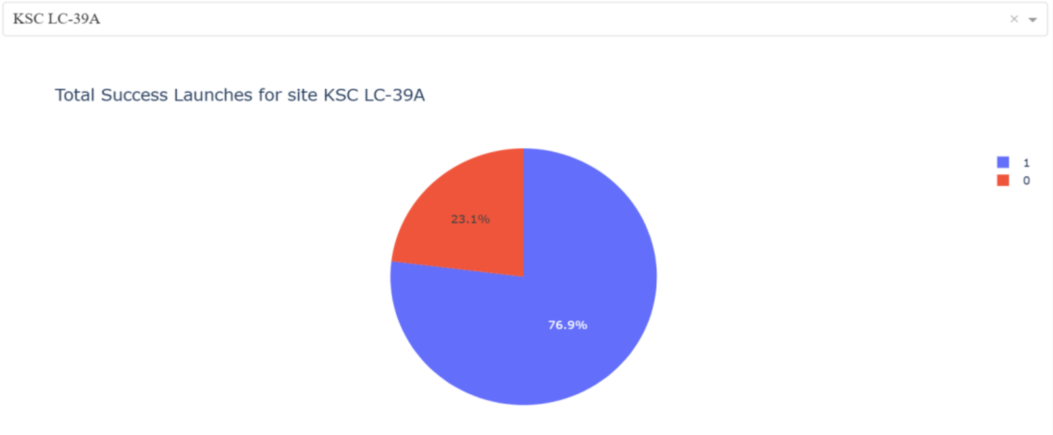
CCAFS LC-40

Total Success Launches for site CCAFS LC-40



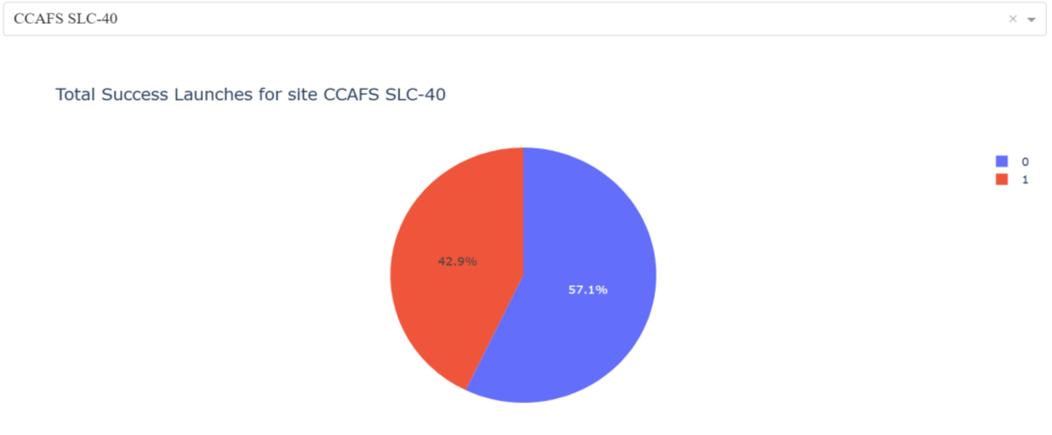
VAFB SLC-4E

Total Success Launches for site VAFB SLC-4E



KSC LC-39A

Total Success Launches for site KSC LC-39A

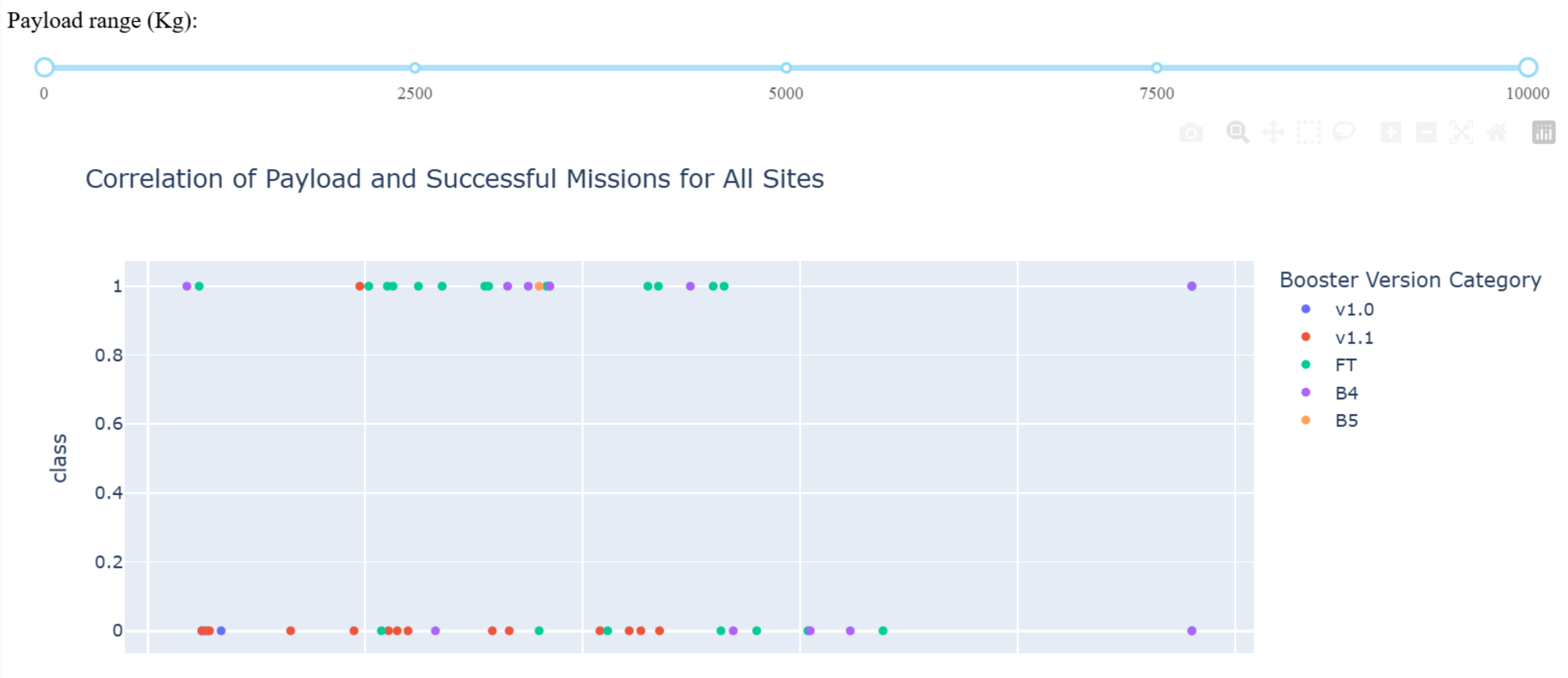


CCAFS SLC-40

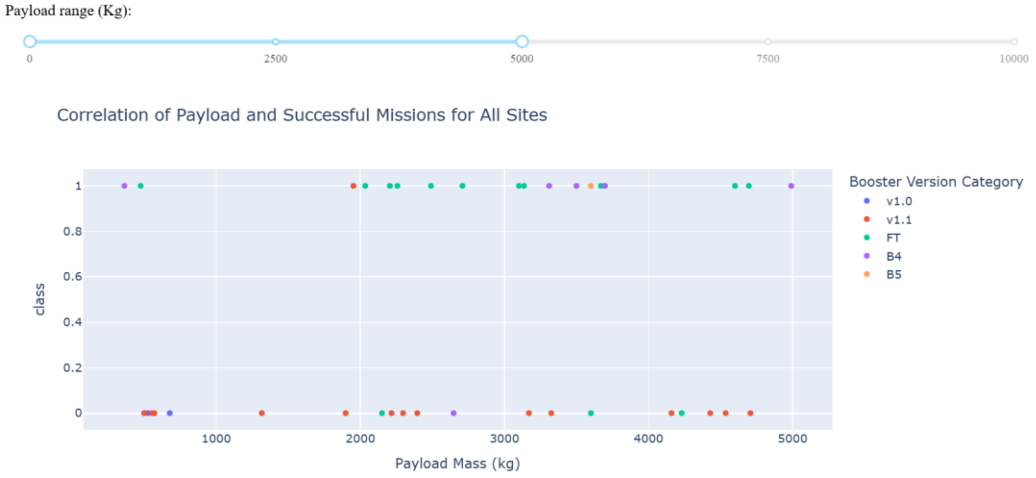
Total Success Launches for site CCAFS SLC-40

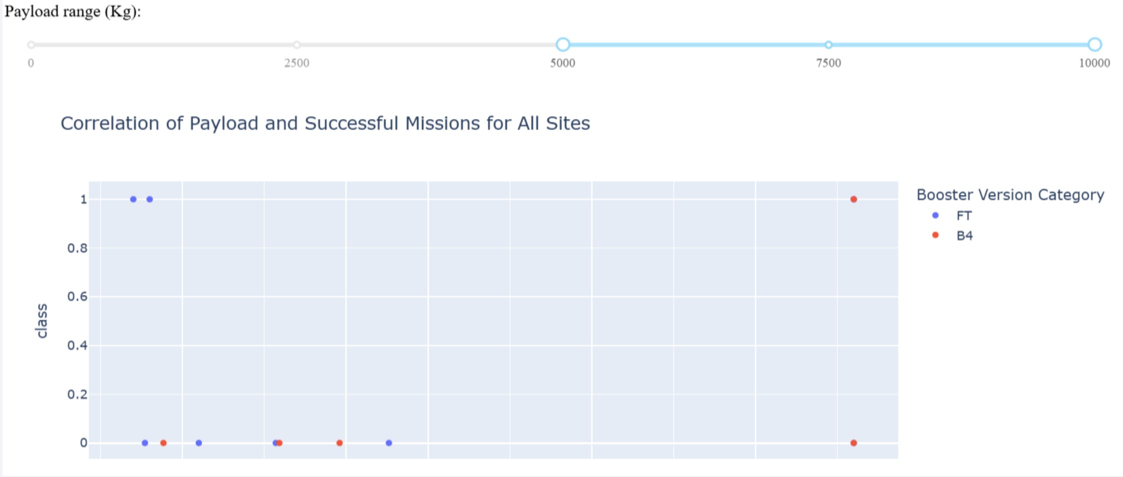
# Correlation of Payload and Successful Missions for All Sites (Dashboard) 1

- We show the all range of payload (0-10000 kg) for booster as below. We can find that the booster version FT generally has a higher success rate, while the booster versions v1.0 and v1.1 have a lower success rate.



# Correlation of Payload and Successful Missions for All Sites (Dashboard) 2





- At low payload (0-5000 kg), we can more clearly see the previous discussion. The booster version FT and even B4 generally have a higher success rate, while the booster versions v1.0 and v1.1 have a lower success rate.

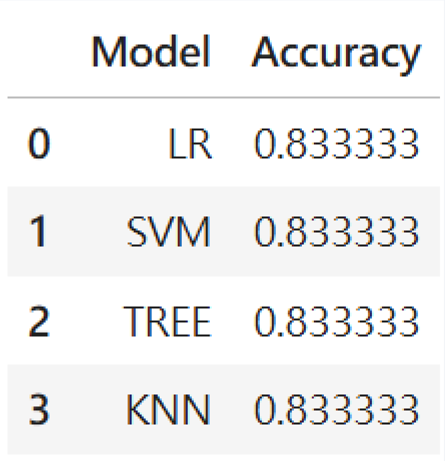
- At high payload (5000-10000 kg), we can find that only the booster version FT and B4 have records and their success rates are not high.

Section 5

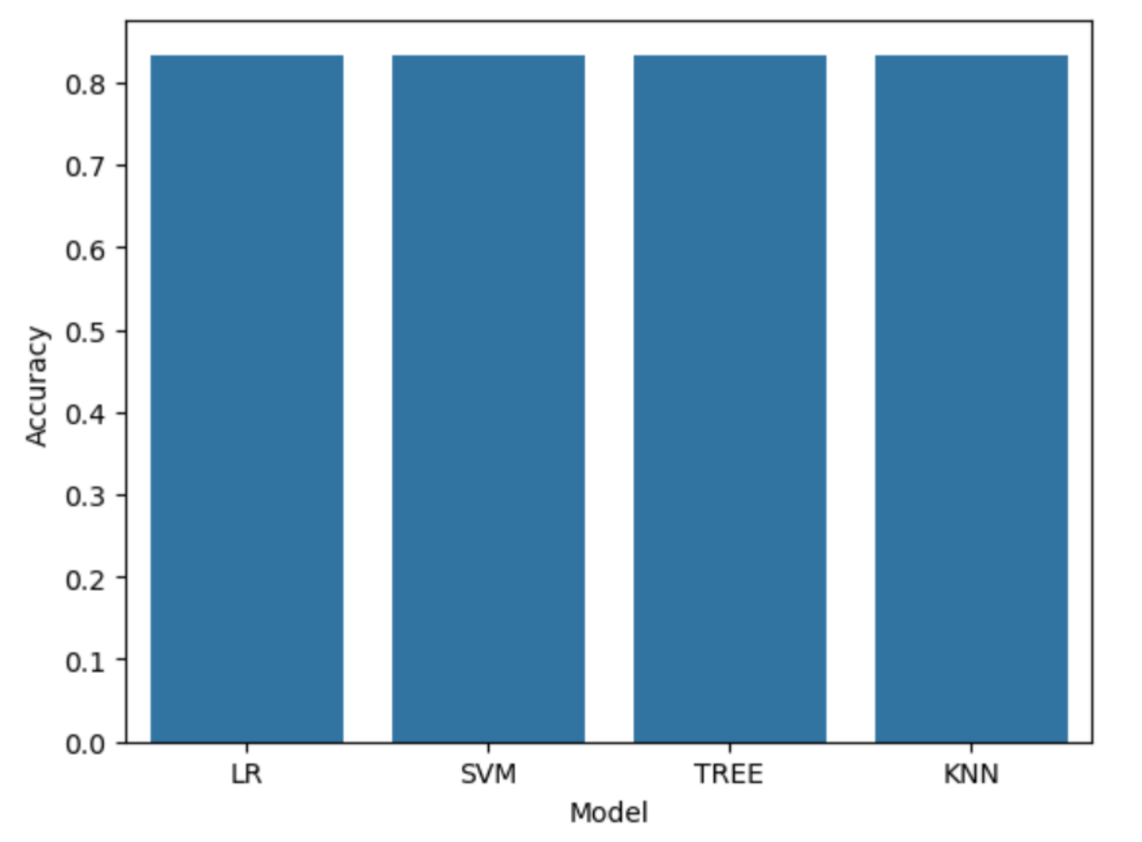
# Predictive Analysis (Classification)

# Classification Accuracy

- We show a table and a bar chart about the accuracy for all built classification models.
- Obviously, we can find that all models have the same accuracy of 0.833, which means that the four models we selected in this project have the same predictive ability.

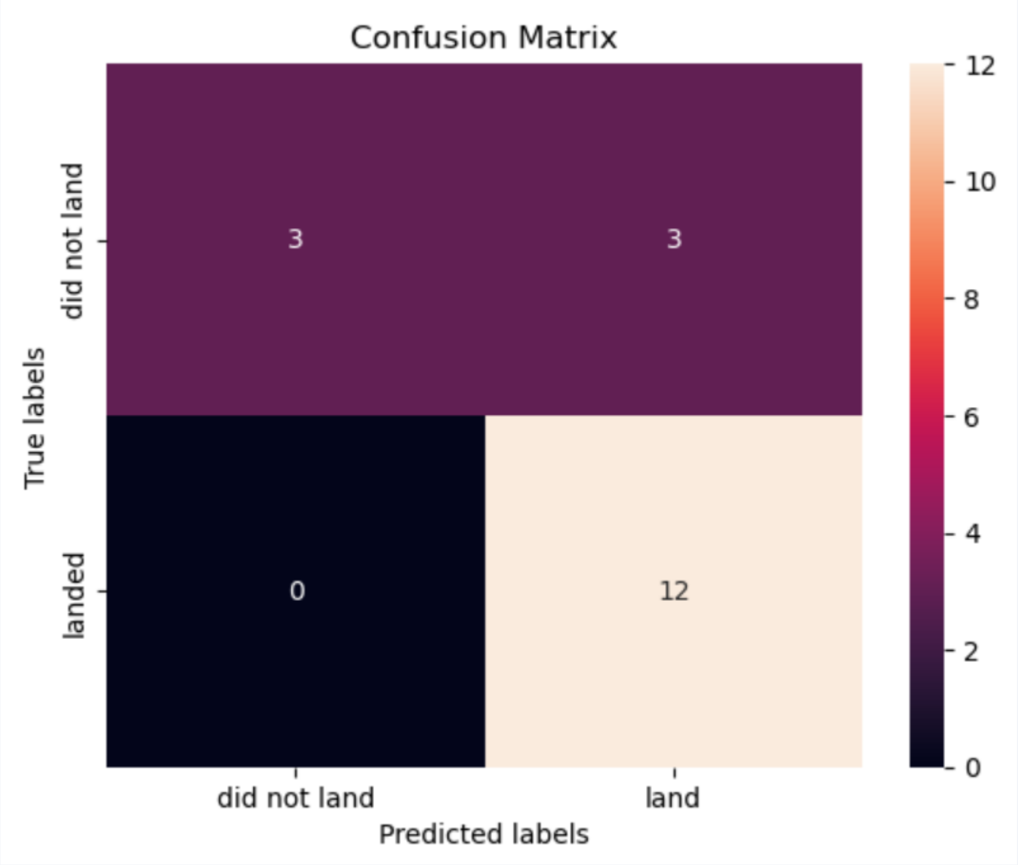


|   | Model | Accuracy |
|---|---|---|
| 0 | LR | 0.833333 |
| 1 | SVM | 0.833333 |
| 2 | TREE | 0.833333 |
| 3 | KNN | 0.833333 |



# Confusion Matrix

- Since all models have the same accuracy and same confusion matrix, we show the confusion matrix of one of them as below.



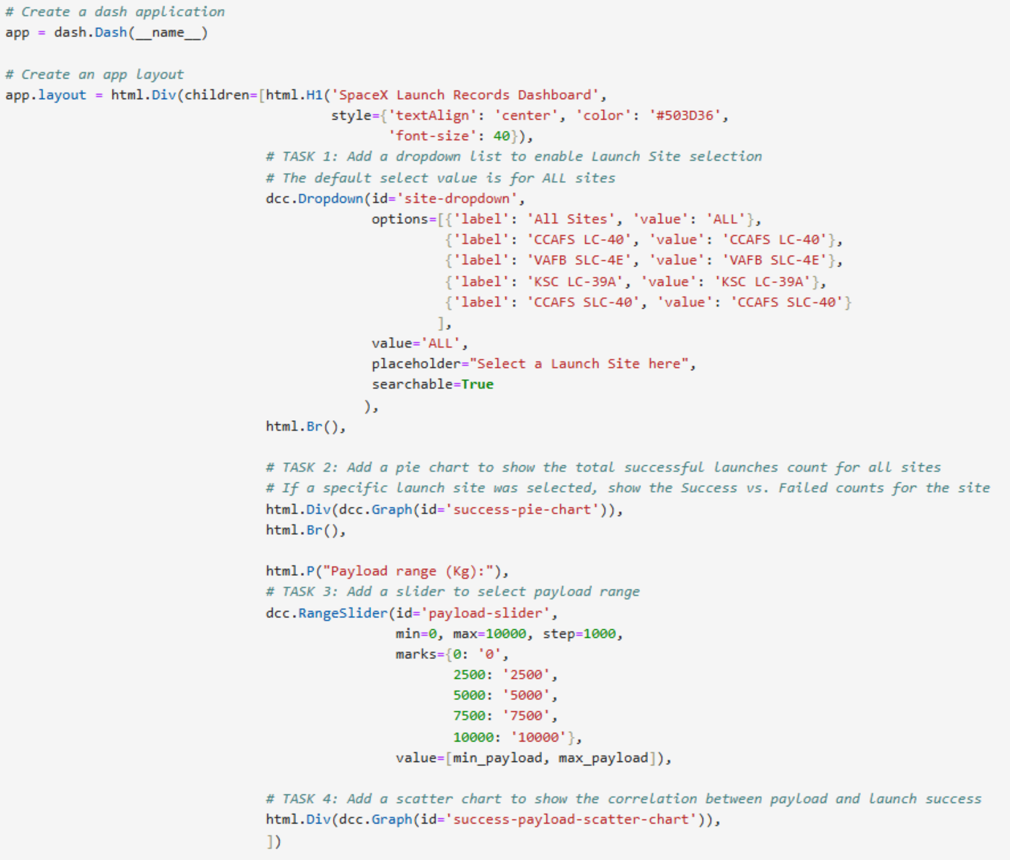
- Examining the confusion matrix, we see that all models can distinguish between the different classes. We see that the problem is false positives.
    - True Postive : 12 (True label is landed, Predicted label is also landed)
    - False Postive : 3 (True label is not landed, Predicted label is landed)

# Conclusions

- For our main purpose, predicting the SpaceX Falcon 9 First Stage Landing. The four models we built are logistic regression (LR), support vector machine (SVM), decision tree (TREE) and k nearest neighbors (KNN). Their training results are as follows.
- All models have the same accuracy of 0.833. We can choose the model we prefer to use.
- All models have the false positives problem. This can be used as the main direction for improvement in future research.

# Appendix

- For Dashborad with Ploty Dash, we provide the skeleton of the program we used, along with screenshots of the code, for reference.
- <u>The skeleton</u>



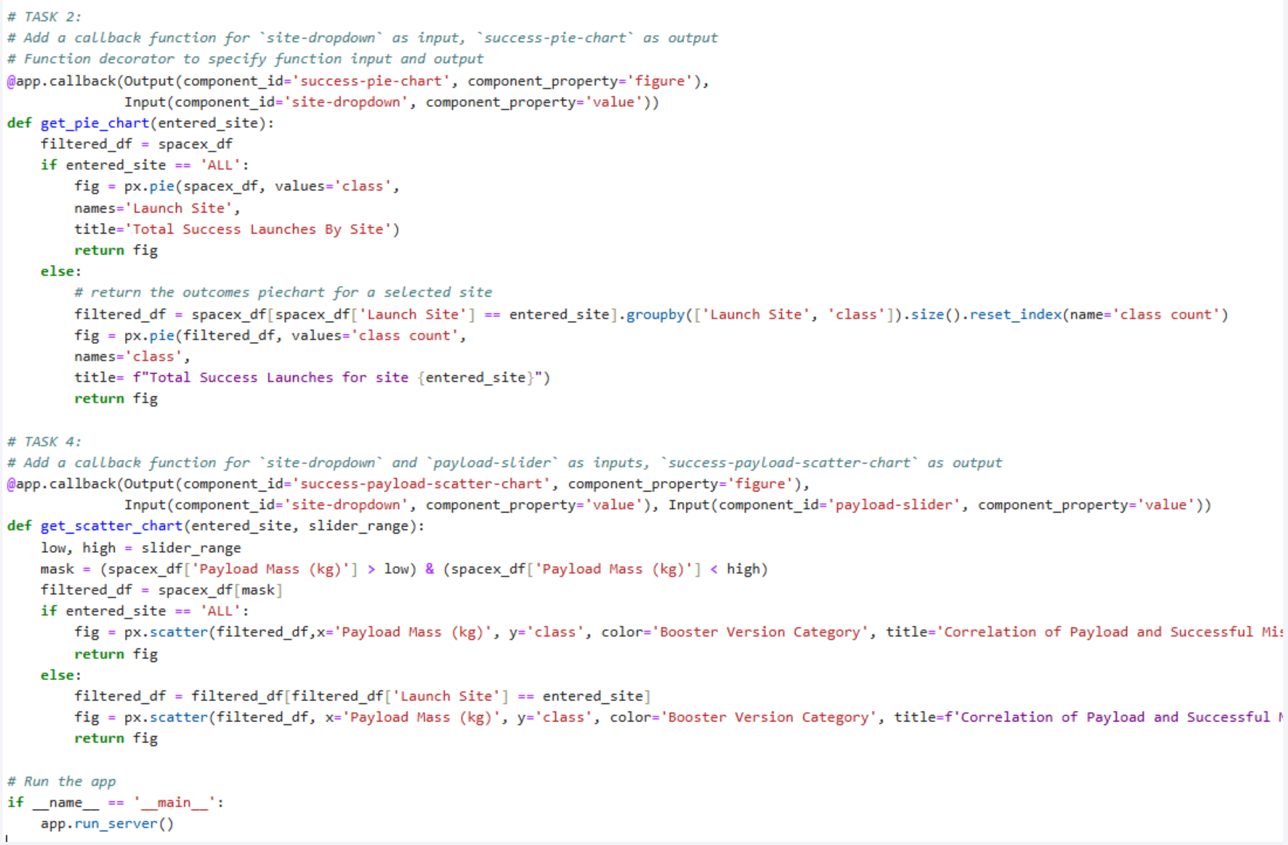
```
# Create a dash application
app = dash.Dash(__name__)

# Create an app layout
app.layout = html.Div(children=[html.H1('SpaceX Launch Records Dashboard',
                                        style={'textAlign': 'center', 'color': '#503D36',
                                               'font-size': 40}),
                                # TASK 1: Add a dropdown list to enable Launch Site selection
                                # The default select value is for ALL sites
                                dcc.Dropdown(id='site-dropdown',
                                             options=[{'label': 'All Sites', 'value': 'ALL'},
                                                      {'label': 'CCAFS LC-40', 'value': 'CCAFS LC-40'},
                                                      {'label': 'VAFB SLC-4E', 'value': 'VAFB SLC-4E'},
                                                      {'label': 'KSC LC-39A', 'value': 'KSC LC-39A'},
                                                      {'label': 'CCAFS SLC-40', 'value': 'CCAFS SLC-40'}
                                                     ],
                                             value='ALL',
                                             placeholder="Select a Launch Site here",
                                             searchable=True
                                            ),
                                html.Br(),

                                # TASK 2: Add a pie chart to show the total successful launches count for all sites
                                # If a specific launch site was selected, show the Success vs. Failed counts for the site
                                html.Div(dcc.Graph(id='success-pie-chart')),
                                html.Br(),

                                html.P("Payload range (Kg):"),
                                # TASK 3: Add a slider to select payload range
                                dcc.RangeSlider(id='payload-slider',
                                                min=0, max=10000, step=1000,
                                                marks={0: '0',
                                                       2500: '2500',
                                                       5000: '5000',
                                                       7500: '7500',
                                                       10000: '10000'},
                                                value=[min_payload, max_payload]),

                                # TASK 4: Add a scatter chart to show the correlation between payload and launch success
                                html.Div(dcc.Graph(id='success-payload-scatter-chart')),
                                ])
```



```
# TASK 2:
# Add a callback function for `site-dropdown` as input, `success-pie-chart` as output
# Function decorator to specify function input and output
@app.callback(Output(component_id='success-pie-chart', component_property='figure'),
              Input(component_id='site-dropdown', component_property='value'))
def get_pie_chart(entered_site):
    filtered_df = spacex_df
    if entered_site == 'ALL':
        fig = px.pie(spacex_df, values='class',
        names='Launch Site',
        title='Total Success Launches By Site')
        return fig
    else:
        # return the outcomes piechart for a selected site
        filtered_df = spacex_df[spacex_df['Launch Site'] == entered_site].groupby(['Launch Site', 'class']).size().reset_index(name='class count')
        fig = px.pie(filtered_df, values='class count',
        names='class',
        title= f"Total Success Launches for site {entered_site}")
        return fig

# TASK 4:
# Add a callback function for `site-dropdown` and `payload-slider` as inputs, `success-payload-scatter-chart` as output
@app.callback(Output(component_id='success-payload-scatter-chart', component_property='figure'),
              Input(component_id='site-dropdown', component_property='value'), Input(component_id='payload-slider', component_property='value'))
def get_scatter_chart(entered_site, slider_range):
    low, high = slider_range
    mask = (spacex_df['Payload Mass (kg)'] > low) & (spacex_df['Payload Mass (kg)'] < high)
    filtered_df = spacex_df[mask]
    if entered_site == 'ALL':
        fig = px.scatter(filtered_df,x='Payload Mass (kg)', y='class', color='Booster Version Category', title='Correlation of Payload and Successful Mis
        return fig
    else:
        filtered_df = filtered_df[filtered_df['Launch Site'] == entered_site]
        fig = px.scatter(filtered_df, x='Payload Mass (kg)', y='class', color='Booster Version Category', title=f'Correlation of Payload and Successful M
        return fig

# Run the app
if __name__ == '__main__':
    app.run_server()
```

# Thank you!