IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Chunjie Zhang
July 27, 2022

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data collection and wrangling
  - Exploratory analysis of data
  - Interactive visual analysis of data
  - Machine learning models and prediction

- **Summary of all results**
  - Parameters that are critical for predicting launch results include Flight Number, Payload Mass, Flights, Block, Reused Count, Orbit, Launch Site, Landing Pad, and Serial. These parameters were selected based on various means of exploratory analysis of data.
  - Machine learning models were constructed using Logic Regression, SVC, Decision Tree, and KNN. KNN and SVC showed highest accuracy of ca. 0.83.

Introduction

- **Project background and context**

SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars while other providers cost upward of 165 million dollars each. Much of the savings is because SpaceX can reuse the first stage.

If we can determine if the first stage will land successfully, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- **Problems you want to find answers**

In this project, we first understood the launch results and factors that affected the launches by analyzing and visualizing historical data. Based on the findings, we will build machine learning models and use them to predict if the Falcon 9 first stage will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Launch data collected from SpaceX REST API
  - Web Scraping from Wikipedia page
- Perform data wrangling
  - Converted data into data frame
  - Converted the launch outcome into digits
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Plotted launch results again various parameters such as Flight Number, Launch Site, Payload Mass, Orbit, etc.
  - Used SQL to perform operations on data and extract insight
- Perform interactive visual analytics using Folium and Plotly Dash
  - Marked locations and measured distances from proximities
  - Created interactive dashboard
- Perform predictive analysis using classification models
  - Prepared data and build machine learning models
  - Used multiple models to predict launch result

Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

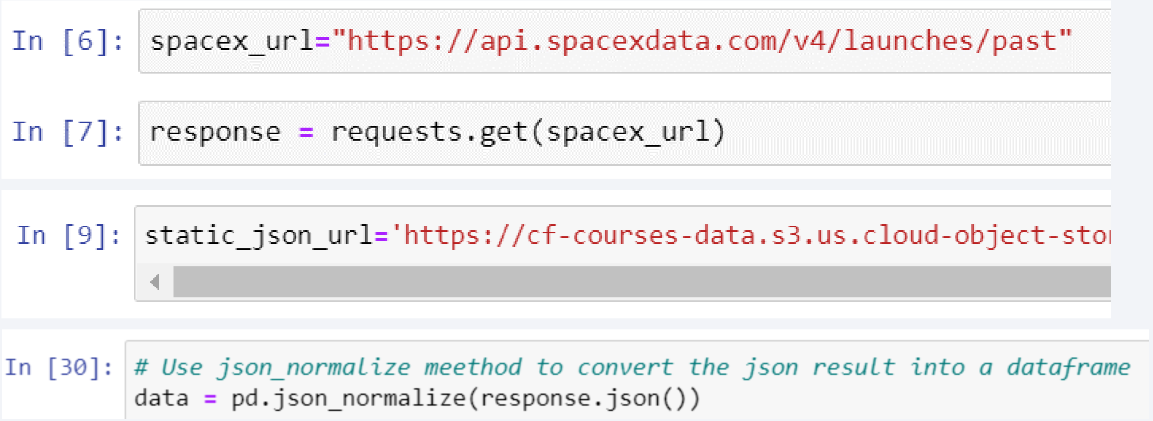
- Present your data collection with SpaceX REST calls using key phrases and flowcharts

  1. Define helper functions to help API extract information using identification numbers in the launch data.
  2. Request and parse launch data using the GET request
  3. Filter the dataframe to only include Falcon 9 launches
  4. Replace missing values with means

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

  - https://github.com/czhang234/Coursera_Capstone_Project/blob/9032a4560b85bfc6162b863663b10daf65e59488/Capstone_Week%201_spacex-data-collection-api.ipynb

Place your flowchart of SpaceX API calls here:



```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)

In [9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-stoı

In [30]: # Use json_normalize meethod to convert the json result into a dataframe
         data = pd.json_normalize(response.json())
```

Data Collection - Scraping

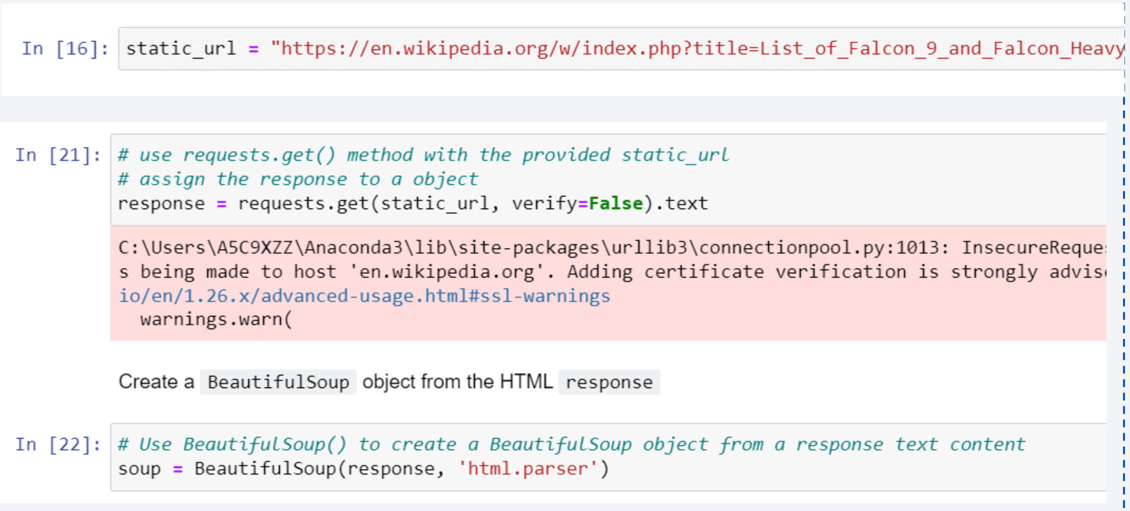
- Present your web scraping process using key phrases and flowcharts

    1. Define helper functions for you to process web scraped HTML table
    2. Request the Falcon9 Launch Wiki page from its URL
    3. Extract all column/variable names from the HTML table header
    4. Create a data frame by parsing the launch HTML tables

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

    - https://github.com/czhang234/Coursera_Capstone_Project/blob/9032a4560b85bfc6162b863663b10daf65e59488/Capstone_Week%201_webscraping.ipynb

Place your flowchart of web scraping here



```
In [16]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy
```

```
In [21]: # use requests.get() method with the provided static_url
         # assign the response to a object
         response = requests.get(static_url, verify=False).text
```

```
C:\Users\A5C9XZZ\Anaconda3\lib\site-packages\urllib3\connectionpool.py:1013: InsecureReque
s being made to host 'en.wikipedia.org'. Adding certificate verification is strongly advis
io/en/1.26.x/advanced-usage.html#ssl-warnings
  warnings.warn(
```

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

```
In [22]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
         soup = BeautifulSoup(response, 'html.parser')
```

Data Wrangling

- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts
  - Identify and calculate the percentage of the missing values in each attribute
  - Identify which columns are numerical and categorical
  - Calculate the number of launches for each site.
  - Calculate the number and occurrence of each orbit
  - Calculate the number and occurrence of mission outcome per orbit type
  - Create a landing outcome label from Outcome column

- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose
  - https://github.com/czhang234/Coursera_Capstone_Project/blob/9032a4560b85bfc6162b863663b10daf65e59488/Capstone_Week%201_spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
  - Scatter plot: FlightNumber vs. PayloadMassand
  - Scatter plot: FlightNumber vs LaunchSite
  - Scatter plot: Payload and Launch Site
  - Bar chart: success rate and orbit type
  - Scatter plot: FlightNumber and Orbit type
  - Scatter plot: Payload and Orbit type
  - Line plot: launch success yearly trend

These charts were used to understand which parameters affect launch results and in what ways.

- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose
  - https://github.com/czhang234/Coursera_Capstone_Project/blob/9032a4560b85bfc6162b863663b10daf65e59488/Capstone_Week%202_EDA%20with%20Visualization%20Lab.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - %sql select distinct(Launch_Site) from SPACEXTBL;
  - %sql select * from SPACEXTBL where Launch_site like 'CCA%' limit 5;
  - %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = 'NASA (CRS)';
  - %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version = 'F9 v1.1’;
  - %sql select min(date) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)’;
  - %sql select Booster_Version from SPACEXTBL where (Landing_Outcome = 'Success (drone ship)') and (PAYLOAD_MASS__KG_ between 4000 and 6000);
  - %sql select count(Mission_Outcome) from SPACEXTBL where Mission_Outcome like '%Success%’;
  - %sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
  - %%sql select substr(Date, 4, 2), Booster_Version, Launch_Site from SPACEXTBL where Landing _Outcome like '%Failure%' and substr(Date,7,4)='2015’;
  - %%sql select * from SPACEXTBL where Landing _Outcome like '%Success%' and Date between '04-06-2010' and '20-03-2017' order by Date Desc;

- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose
  - https://github.com/czhang234/Coursera_Capstone_Project/blob/9032a4560b85bfc6162b863663b10daf65e59488/Capstone_Week%202_Complete%20the%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
  - Mark all launch sites on a map
  - Mark the success/failed launches for each site on the map
  - Calculate the distances between a launch site to its proximities, e.g. highway, railroad, city
- Explain why you added those objects
  - Those objects are factors to build a launch site due to ease of transportation and safety reasons of launching rockets.
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose
  - https://github.com/czhang234/Coursera_Capstone_Project/blob/9032a4560b85bfc6162b863663b10daf65e59488/Capstone_Week_3_Launch_Sites_Locations_Analysis_with_Folium.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
  - Dropdown list to enable Launch Site selection + callback function for `site-dropdown` as input, `success-pie-chart` as output
  - Pie chart to show the total successful launches count for all sites
  - Slider to select payload range + callback function for `site-dropdown` and `payload-slider` as inputs, `success-payload-scatter-chart` as output
  - Scatter chart to show the correlation between payload and launch success

- Explain why you added those plots and interactions
  - The plots and interactions allow us to visualize launch results of each site.

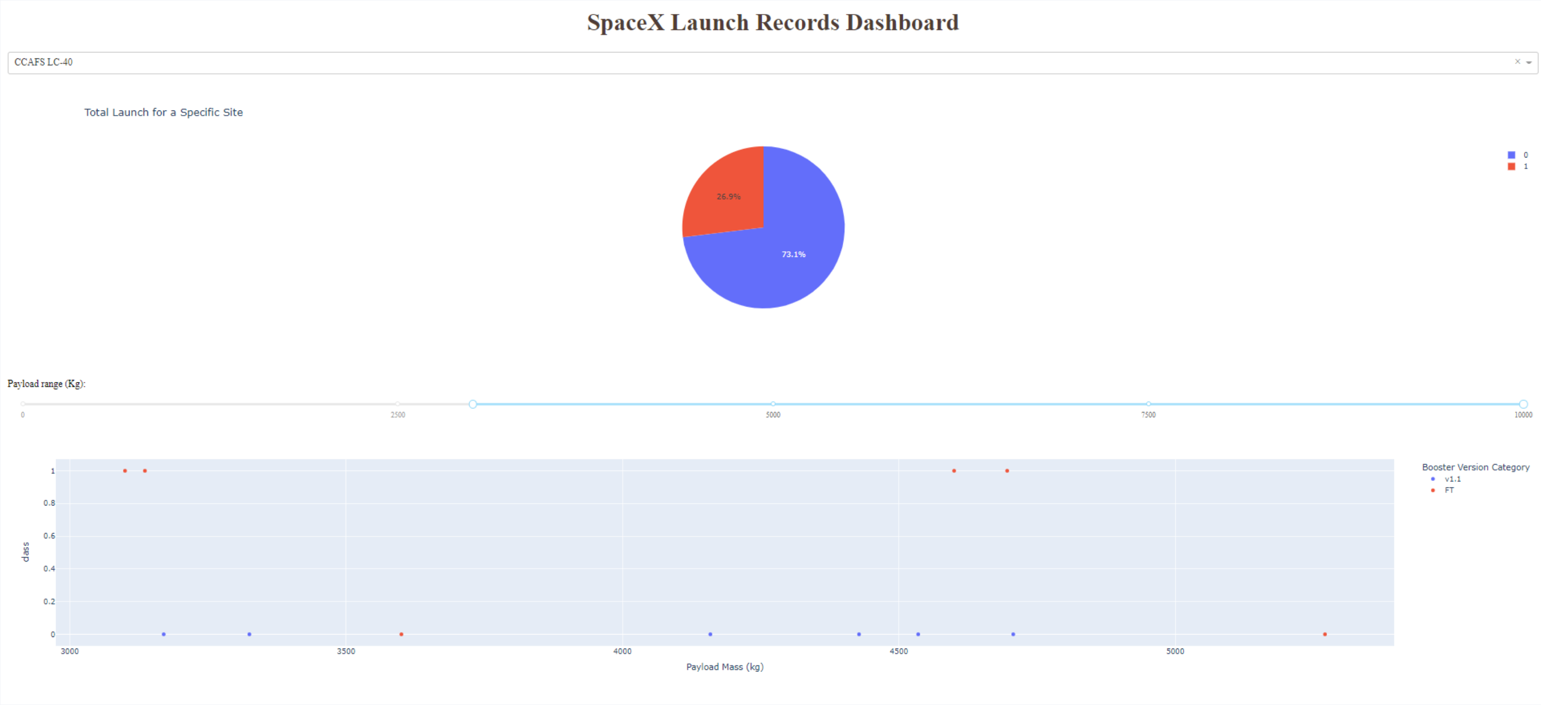
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose
  - https://github.com/czhang234/Coursera_Capstone_Project/blob/9032a4560b85bfc6162b863663b10daf65e59488/Capstone_Week%203_Dashboard.ipynb

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
  - Select relevant parameters and do feature engineering:
    - Create dummy variables to categorical columns
    - Cast all numeric columns to float64
  - Build machine learning models:
    - Select X and Y. Standardize X.
    - Perform train_test_split on X and Y
    - Select models for classification: logic regression, SVC, Decision Tree, KNN
  - Use GridSearchCV to optimize hyperparameters
  - Calculate the accuracy on the test data using the method score
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose
  - https://github.com/czhang234/Coursera_Capstone_Project/blob/9032a4560b85bfc6162b863663b10daf65e59488/Capstone_Week%205_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- Exploratory data analysis results
  - Launch results are affected by parameters including Flight Number, Launch Site, Payload Mass, Orbit, etc.
- Interactive analytics demo in screenshots



SpaceX Launch Records Dashboard

CCAFS LC-40

Total Launch for a Specific Site

Payload range (Kg):

- Predictive analysis results
  - KNN and SVC showed highest accuracy of ca. 0.83.

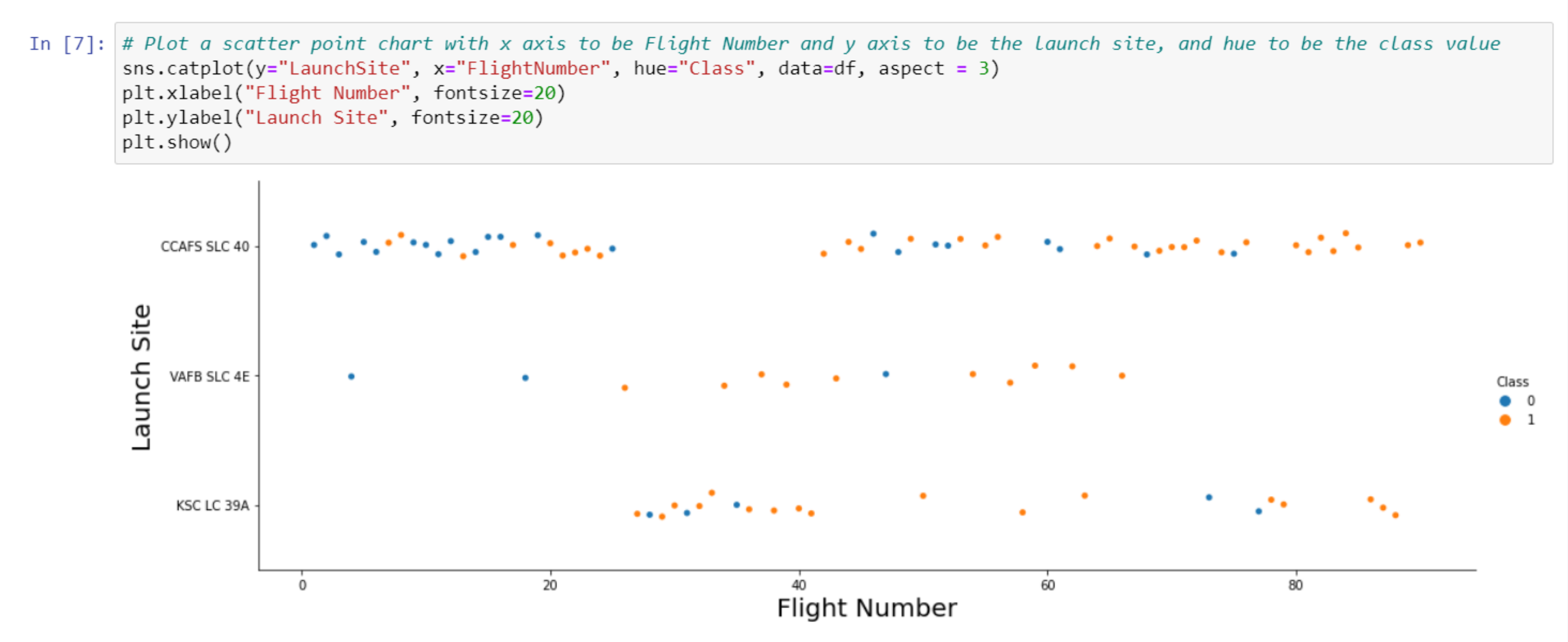
Section 2

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
  - VAFB took on less launch missions after 60 launches by SpaceX. CCAFS took on most launch missions continuously.

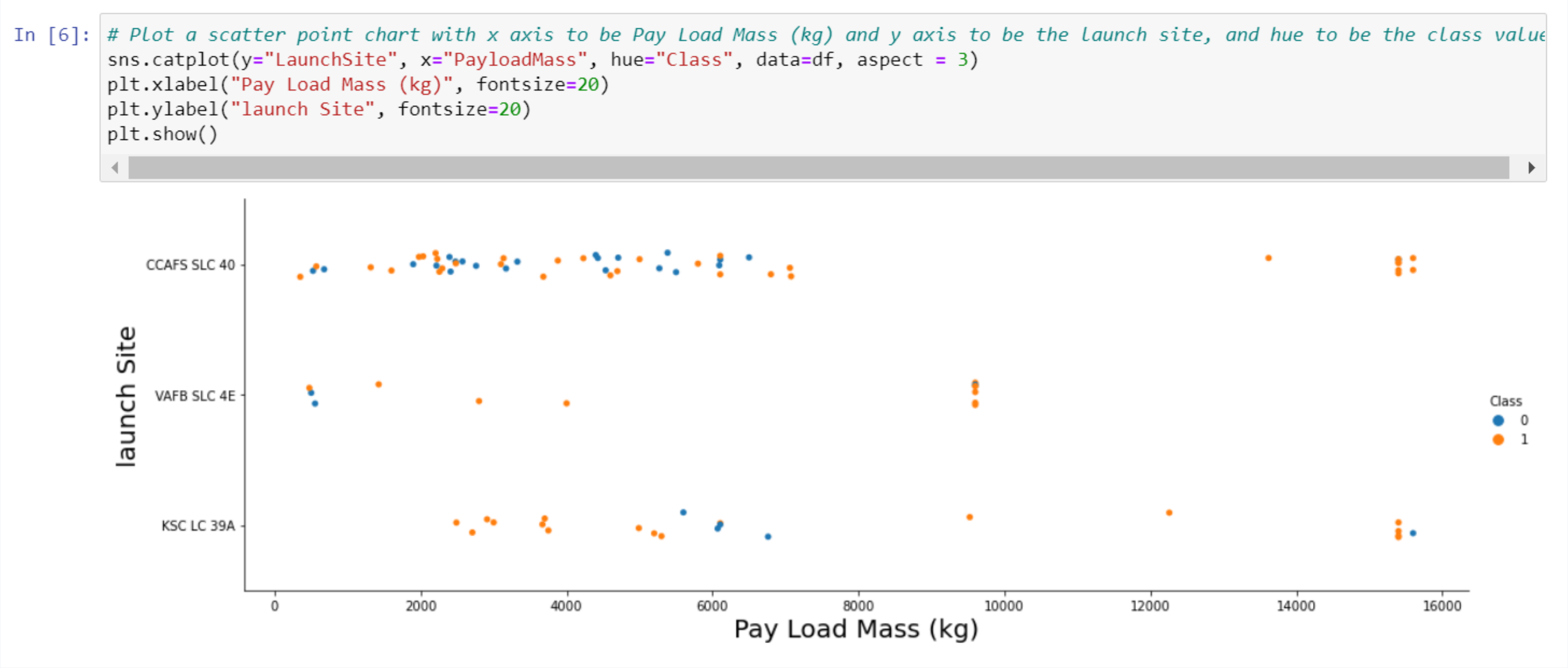


```
In [7]: # Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
        sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 3)
        plt.xlabel("Flight Number", fontsize=20)
        plt.ylabel("Launch Site", fontsize=20)
        plt.show()
```

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

- Show the screenshot of the scatter plot with explanations
  - Most launches had payload less than 10000 kg. CCAFS and KSC took both heavy and light payloads. VAFB only launch lightweights.

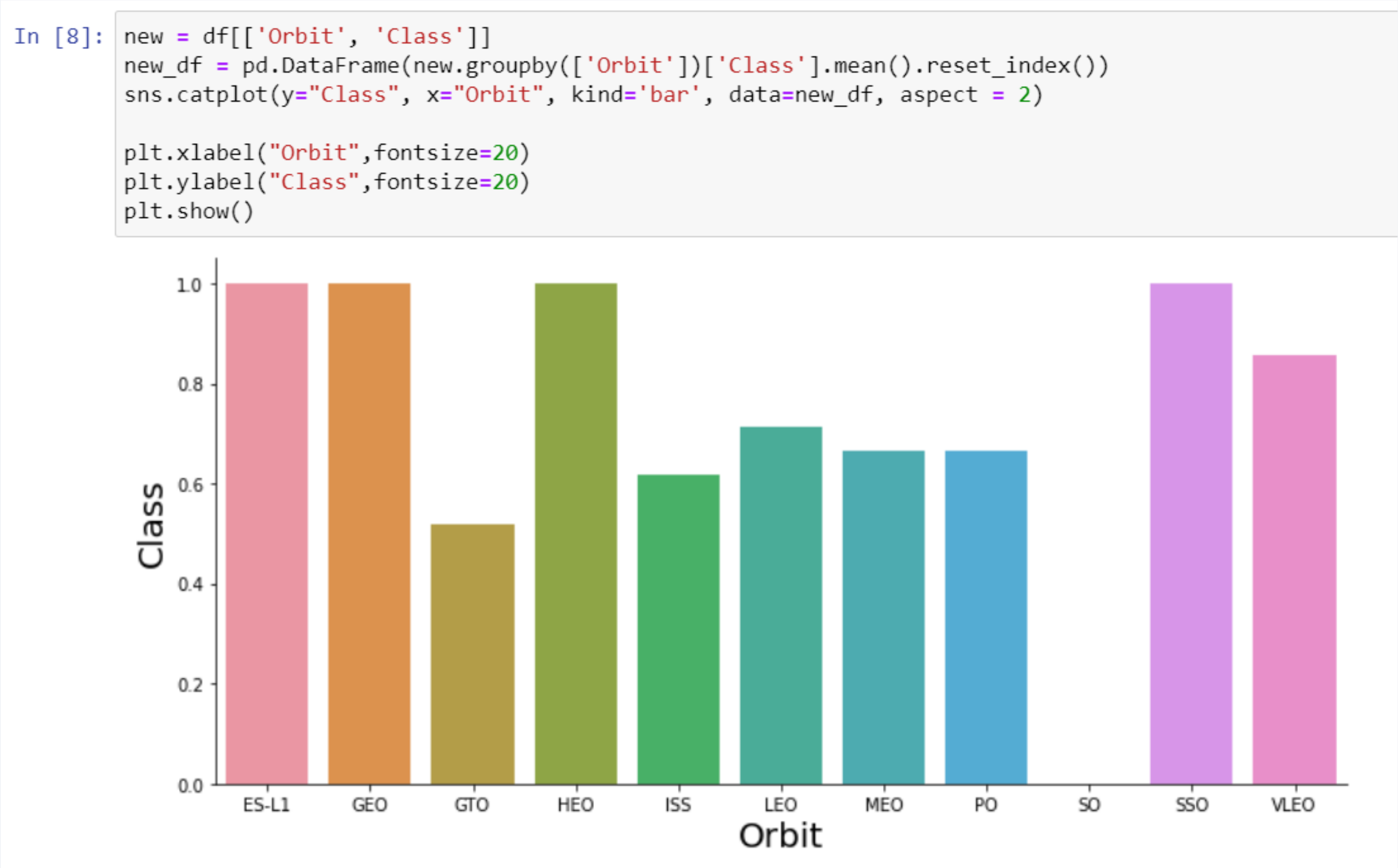


In [6]:

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 3)
plt.xlabel("Pay Load Mass (kg)", fontsize=20)
plt.ylabel("launch Site", fontsize=20)
plt.show()
```

Success Rate vs. Orbit Type

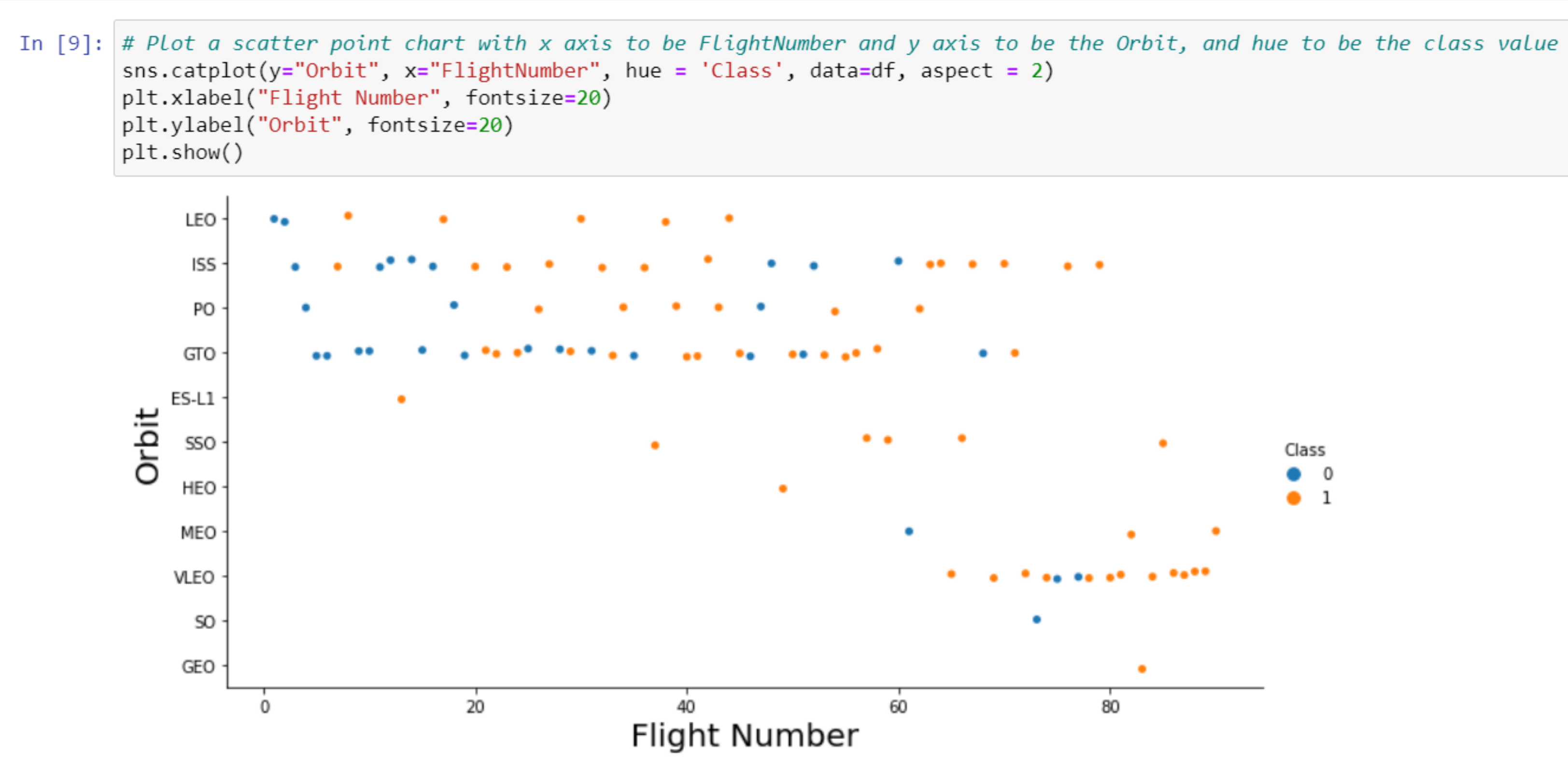
- Show a bar chart for the success rate of each orbit type
- Show the screenshot of the scatter plot with explanations
    - Success rate highly depends on orbit type. SO had the lowest success rate of 0. GTO, ISS, LEO, MEO and PO had success rate of about 0.6.



```
In [8]: new = df[['Orbit', 'Class']]
        new_df = pd.DataFrame(new.groupby(['Orbit'])['Class'].mean().reset_index())
        sns.catplot(y="Class", x="Orbit", kind='bar', data=new_df, aspect = 2)

        plt.xlabel("Orbit",fontsize=20)
        plt.ylabel("Class",fontsize=20)
        plt.show()
```

Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot with explanations
    - Early flights were concentrated on LEO, ISS, PO and GTO. Later, flights after #60 shifted to SSO, HEO, MEO and VLEO.

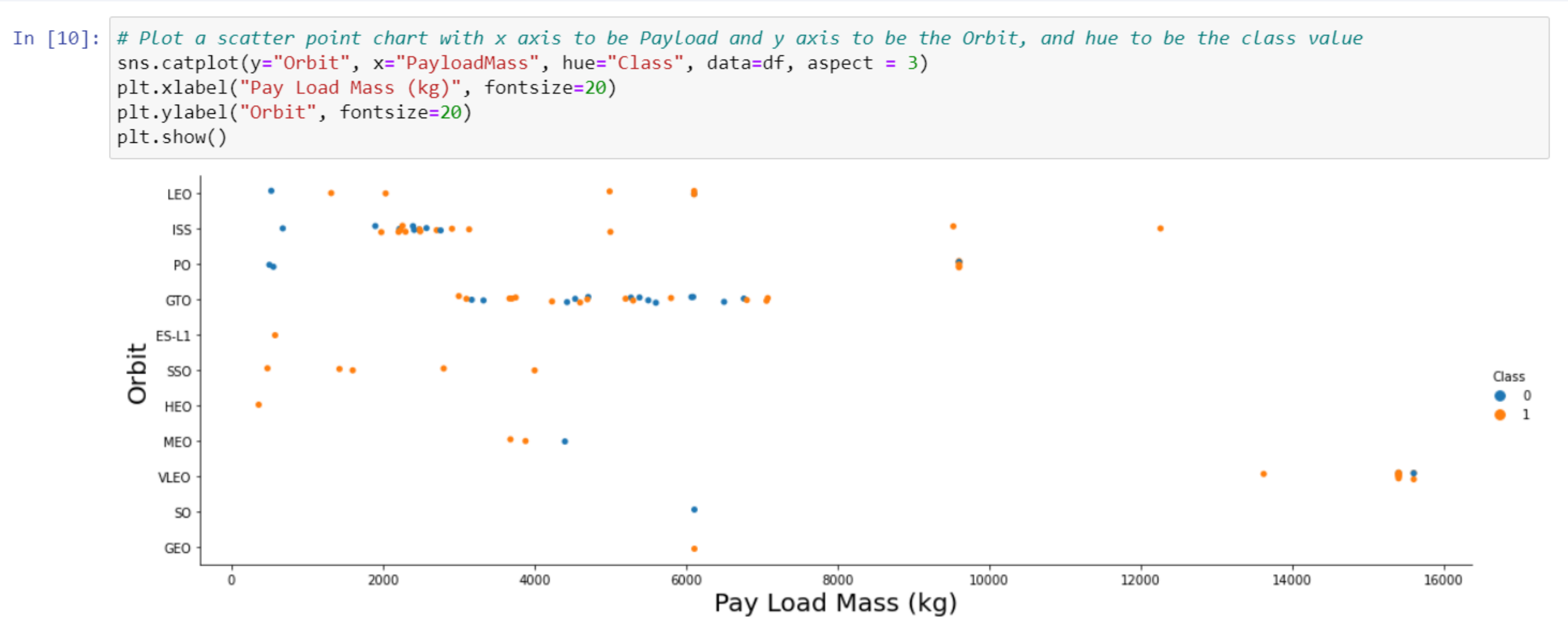


```
In [9]: # Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
        sns.catplot(y="Orbit", x="FlightNumber", hue = 'Class', data=df, aspect = 2)
        plt.xlabel("Flight Number", fontsize=20)
        plt.ylabel("Orbit", fontsize=20)
        plt.show()
```

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

- Show the screenshot of the scatter plot with explanations
  - Payload less than 8000 kg was launched to LEO, ISS, PO, GTO, SSO, and HEO orbits. Heavy payloads above 10000 kg were launched to PO and VLEO.

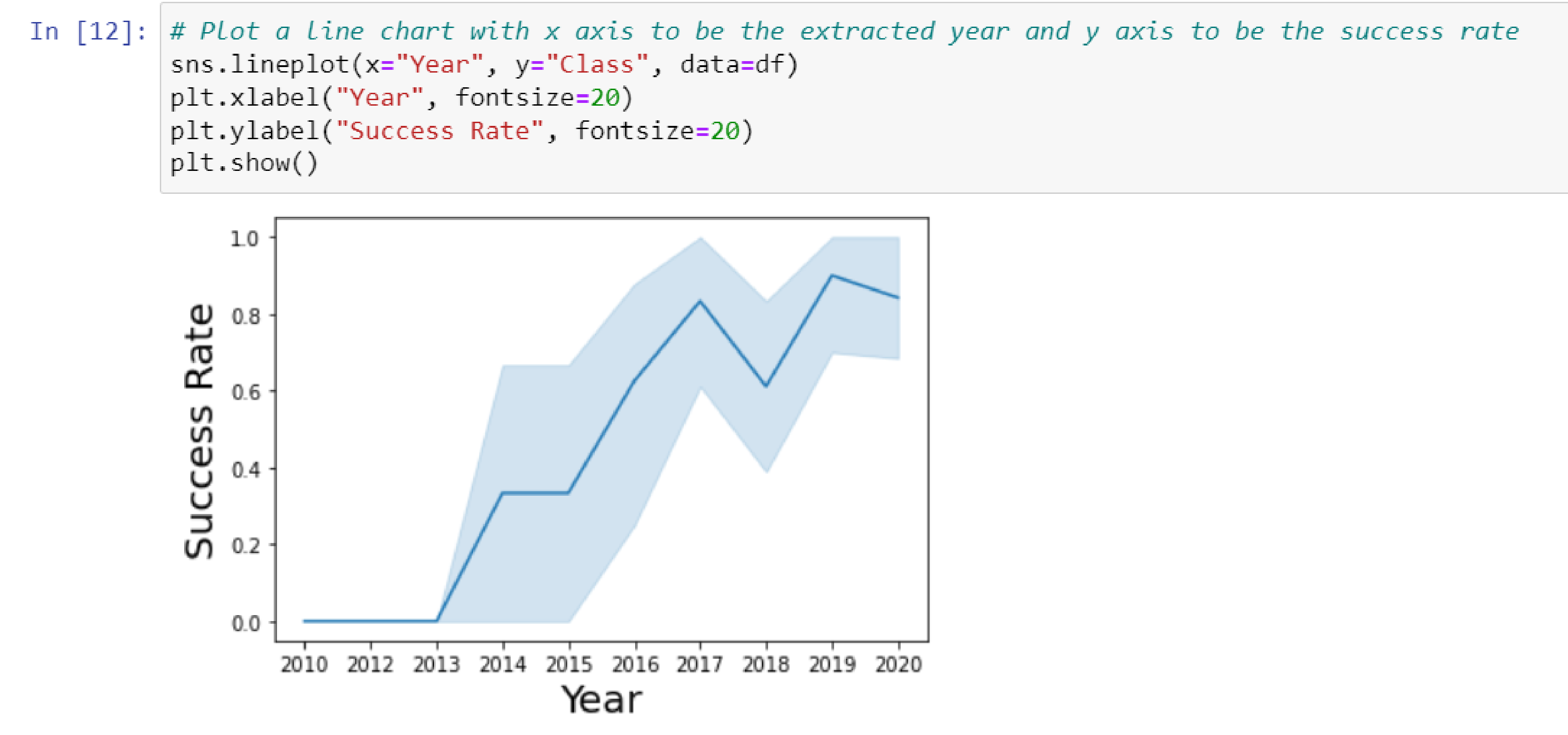


```
In [10]: # Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
         sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 3)
         plt.xlabel("Pay Load Mass (kg)", fontsize=20)
         plt.ylabel("Orbit", fontsize=20)
         plt.show()
```

Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations
    - Launch success rate increases significantly from 2013.

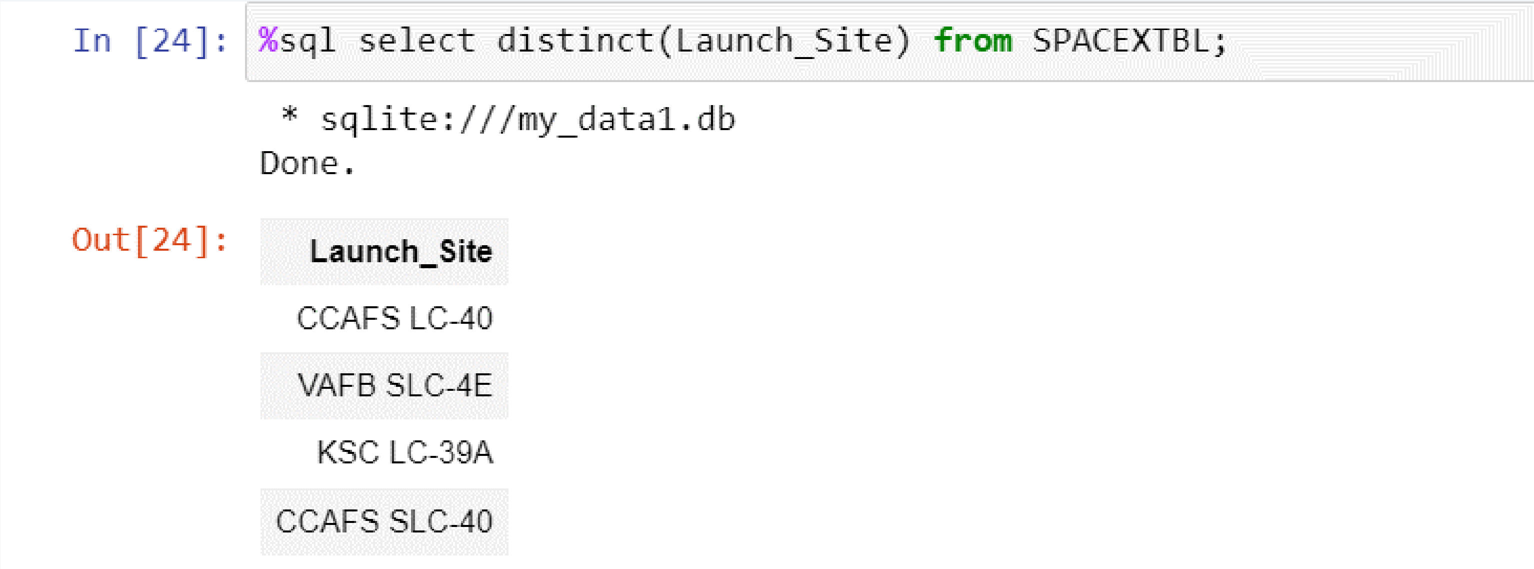


In [12]:

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
sns.lineplot(x="Year", y="Class", data=df)
plt.xlabel("Year", fontsize=20)
plt.ylabel("Success Rate", fontsize=20)
plt.show()
```

All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here

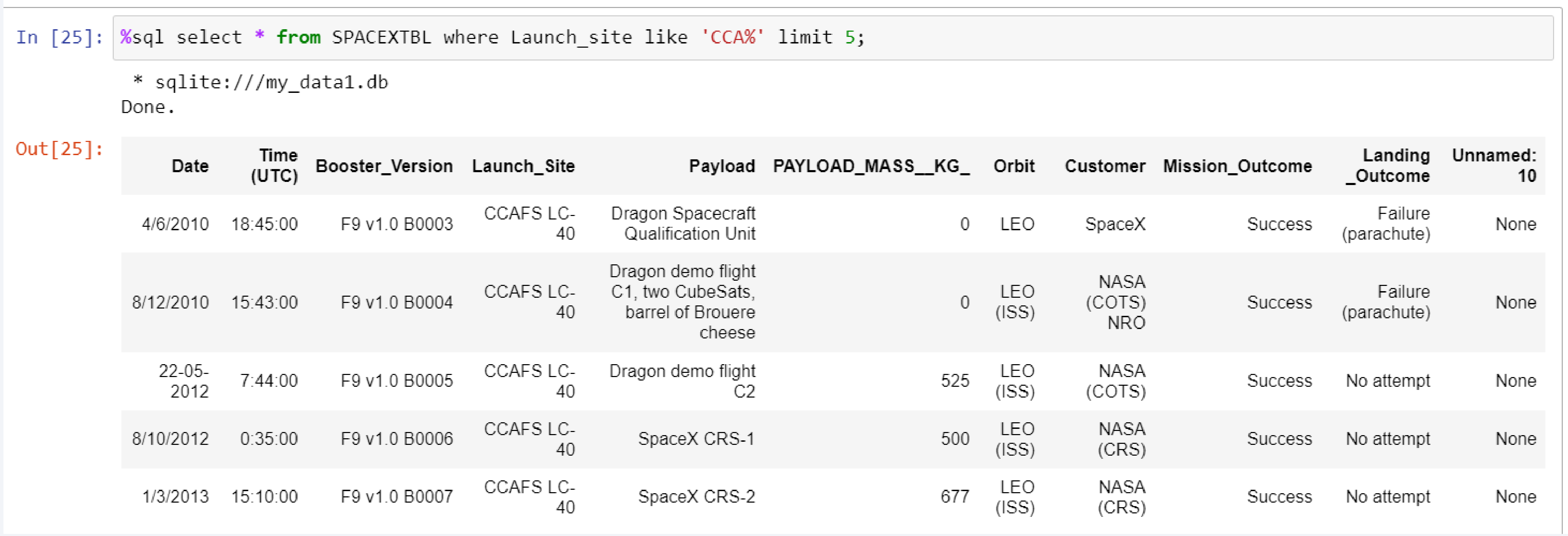


```
In [24]: %sql select distinct(Launch_Site) from SPACEXTBL;

 * sqlite:///my_data1.db
Done.

Out[24]:
```

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here



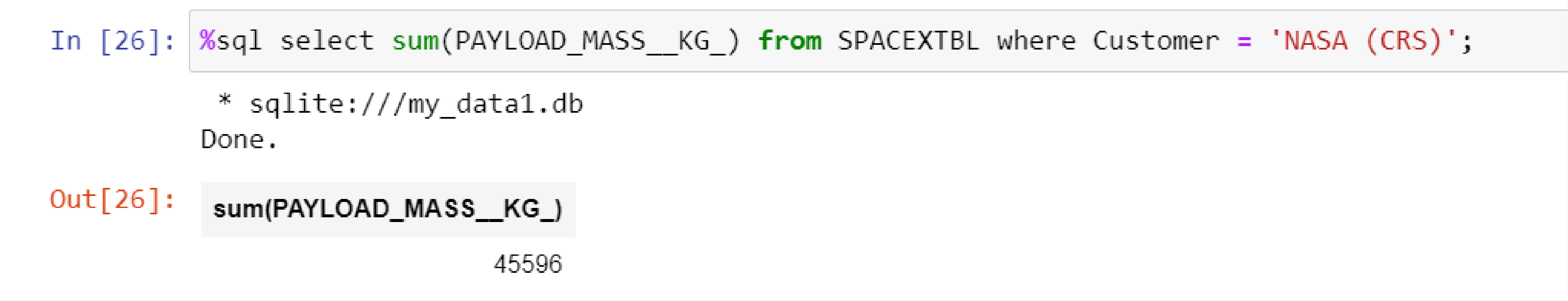
```
In [25]: %sql select * from SPACEXTBL where Launch_site like 'CCA%' limit 5;

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

Out[25]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome	Unnamed: 10
4/6/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)	None
8/12/2010	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)	None
22-05-2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt	None
8/10/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt	None
1/3/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt	None

Total Payload Mass

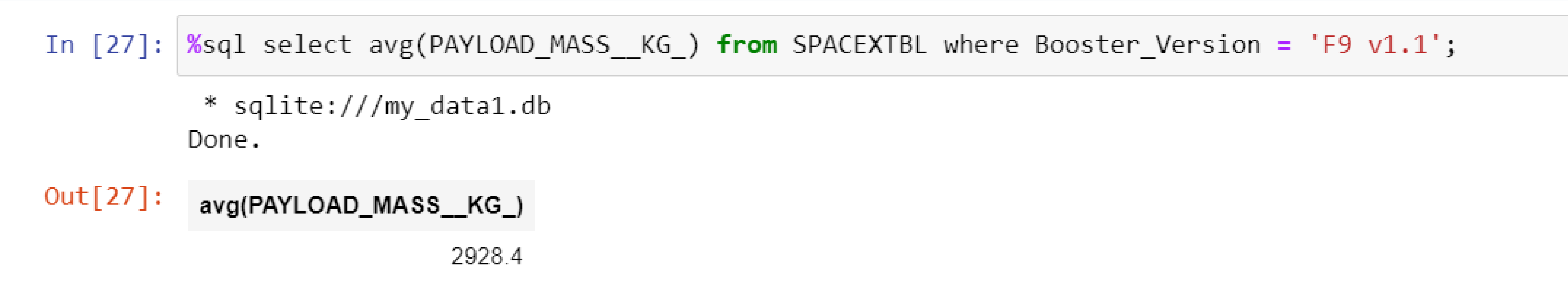
- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here



```
In [26]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = 'NASA (CRS)';

          * sqlite:///my_data1.db
         Done.

Out[26]: sum(PAYLOAD_MASS__KG_)
                          45596
```

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here



```
In [27]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version = 'F9 v1.1';

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

Out[27]:

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

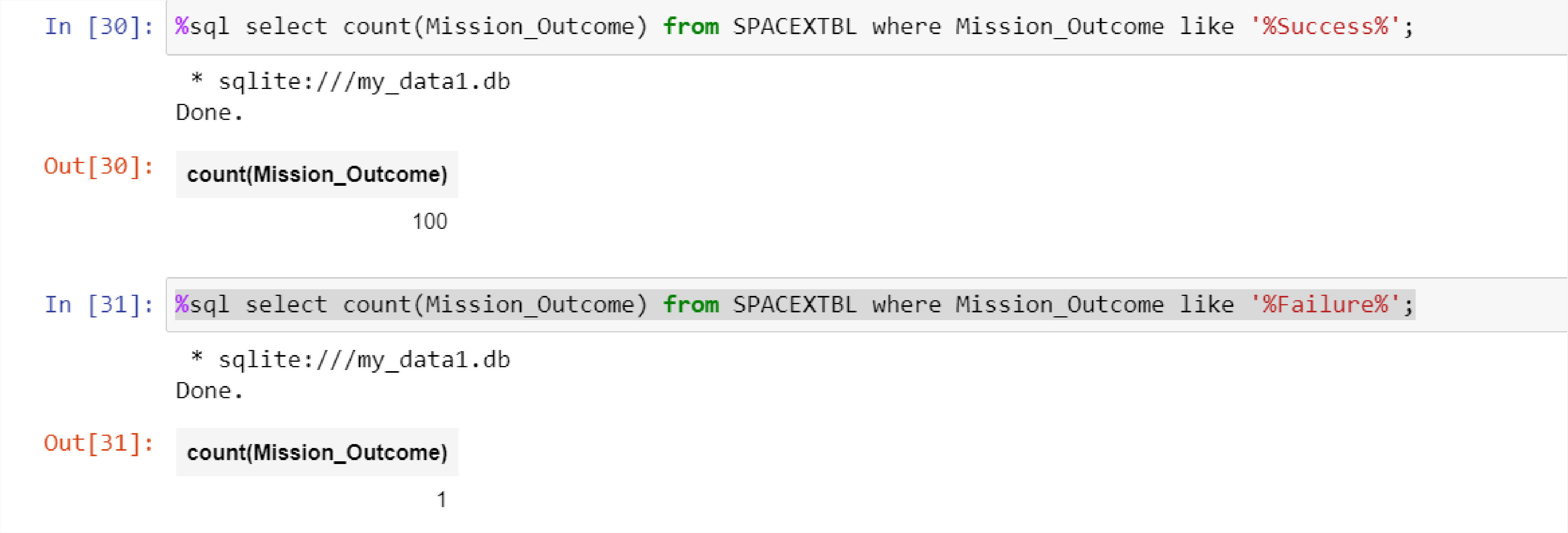
- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

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
- Present your query result with a short explanation here

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



```
In [30]: %sql select count(Mission_Outcome) from SPACEXTBL where Mission_Outcome like '%Success%';

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

Out[30]:

count(Mission_Outcome)
100

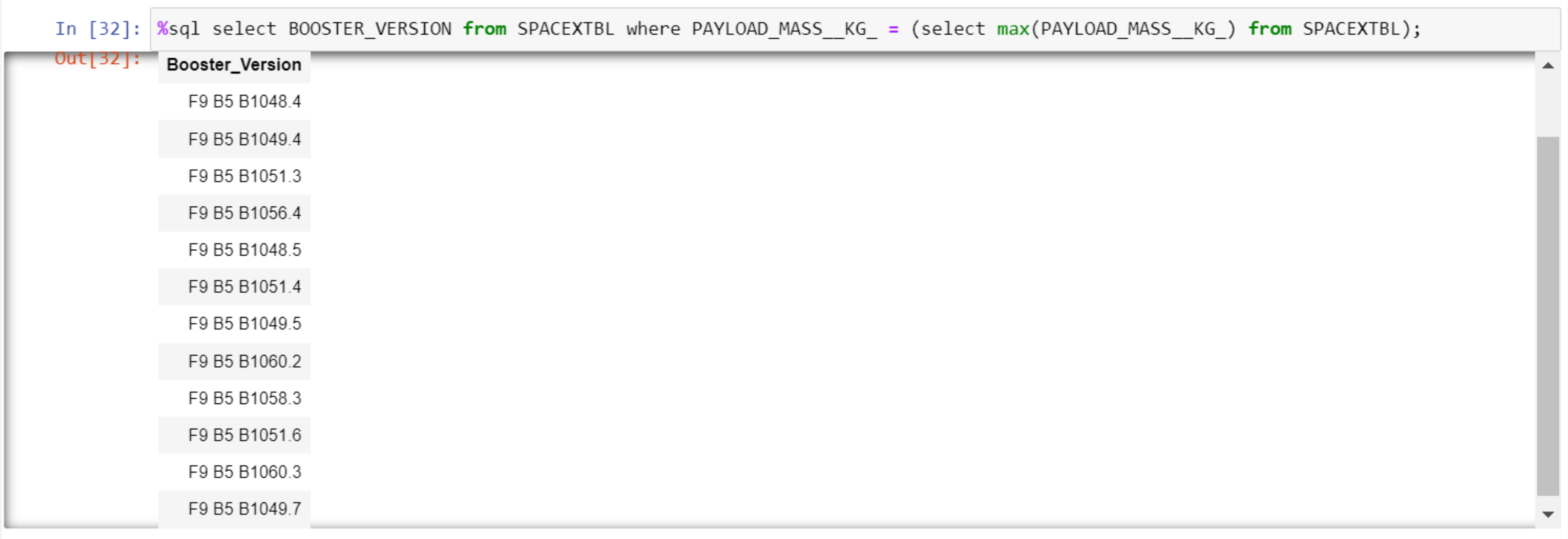
```
In [31]: %sql select count(Mission_Outcome) from SPACEXTBL where Mission_Outcome like '%Failure%';

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

Out[31]:

count(Mission_Outcome)
1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here



```
In [32]: %sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

Out[32]:

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

- Present your query result with a short explanation here

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

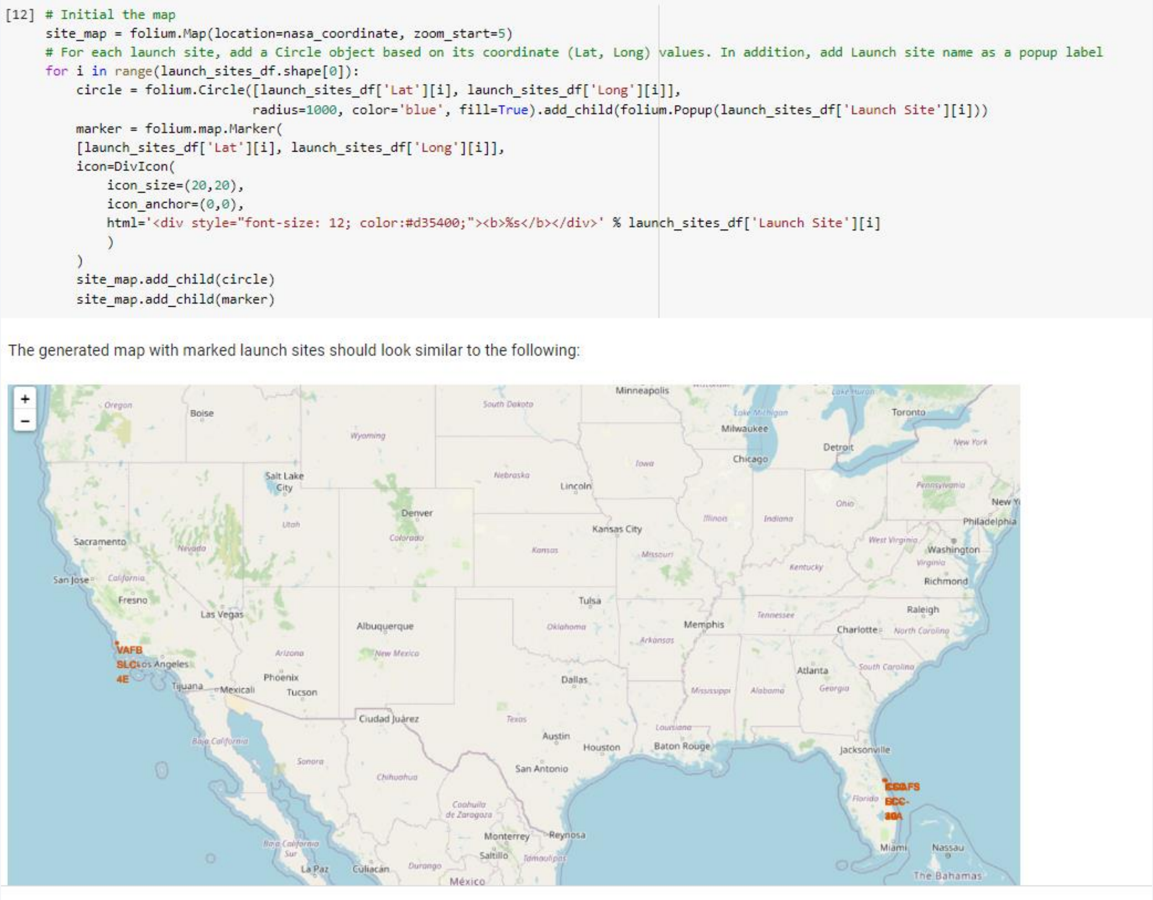
- Present your query result with a short explanation here

Section 3

Launch Sites Proximities Analysis

Mark all launch sites on a map

- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot: *launch sites are close to equator and on coastal lines*

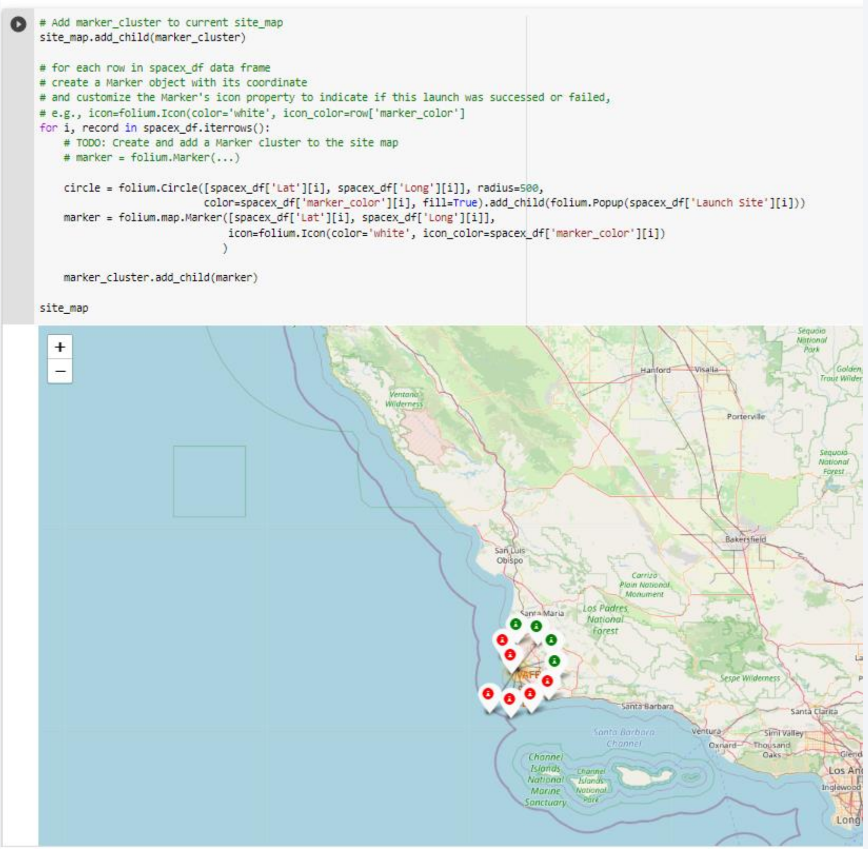


```
[12] # Initial the map
     site_map = folium.Map(location=nasa_coordinate, zoom_start=5)
     # For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition, add Launch site name as a popup label
     for i in range(launch_sites_df.shape[0]):
         circle = folium.Circle([launch_sites_df['Lat'][i], launch_sites_df['Long'][i]],
                                radius=1000, color='blue', fill=True).add_child(folium.Popup(launch_sites_df['Launch Site'][i]))
         marker = folium.map.Marker(
         [launch_sites_df['Lat'][i], launch_sites_df['Long'][i]],
         icon=DivIcon(
             icon_size=(20,20),
             icon_anchor=(0,0),
             html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % launch_sites_df['Launch Site'][i]
             )
         )
         site_map.add_child(circle)
         site_map.add_child(marker)
```

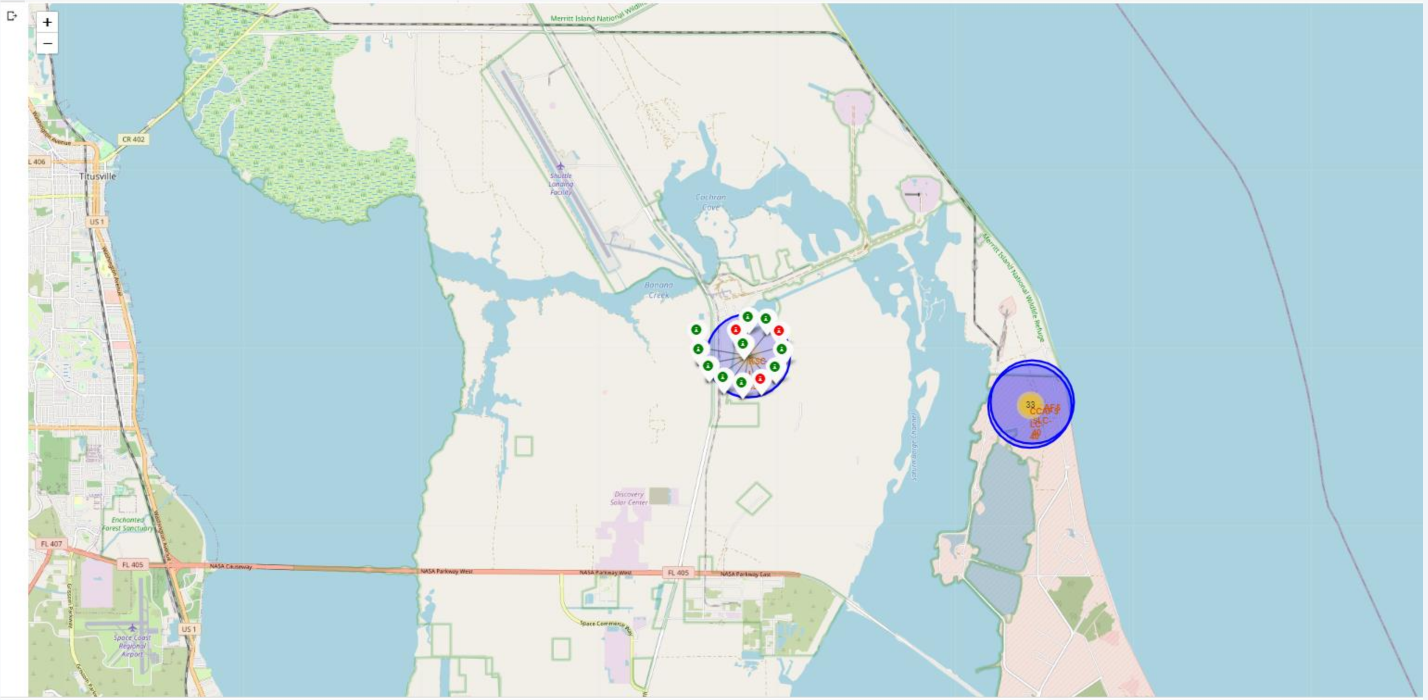
The generated map with marked launch sites should look similar to the following:

Color-labeled launch outcomes on the map

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot: *VAFB on west coast has low success rate*

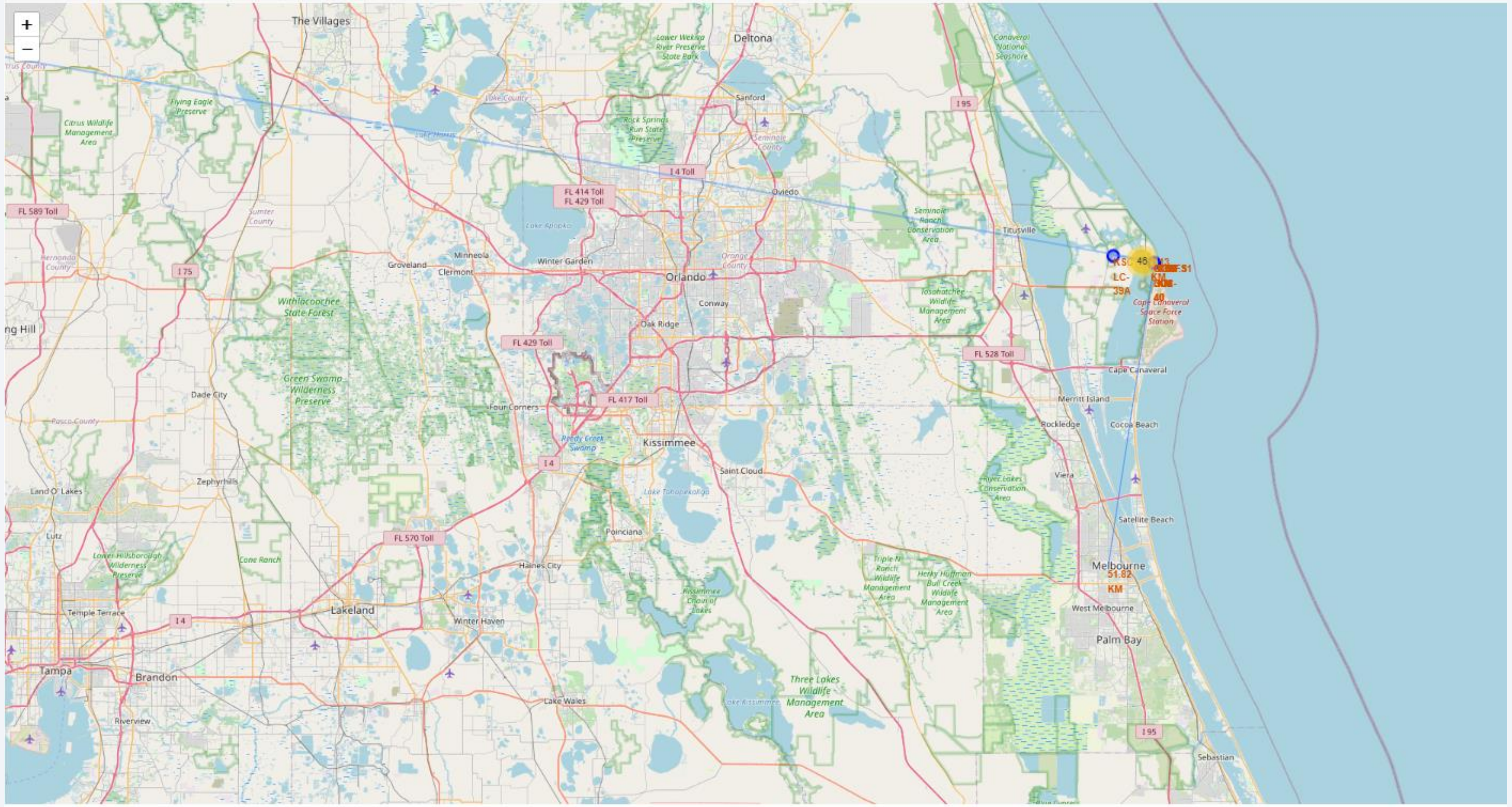


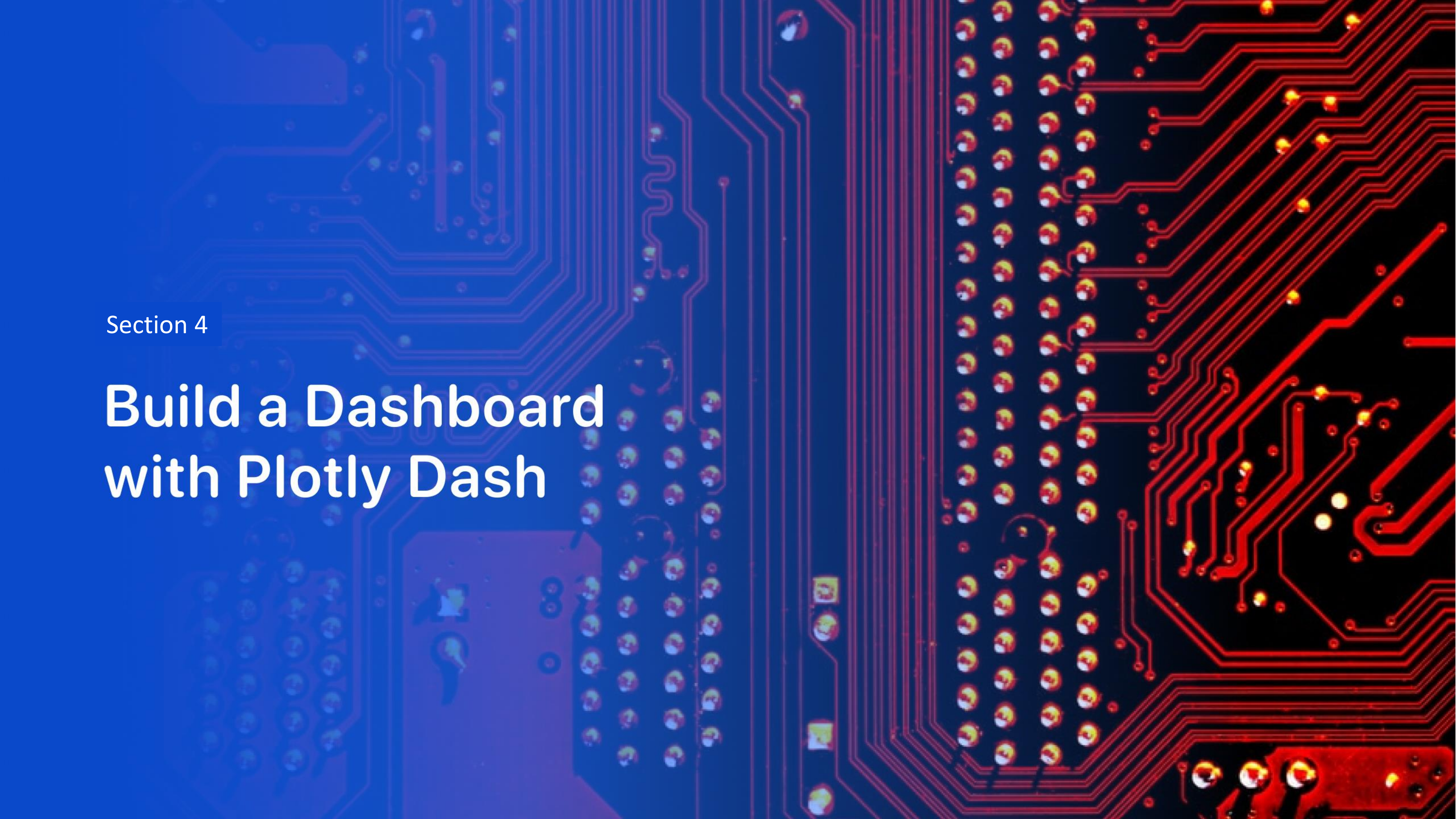
```
# Add marker_cluster to current site_map
site_map.add_child(marker_cluster)

# for each row in spacex_df data frame
# create a Marker object with its coordinate
# and customize the Marker's icon property to indicate if this launch was successed or failed,
# e.g., icon=folium.Icon(color='white', icon_color=row['marker_color']
for i, record in spacex_df.iterrows():
    # TODO: Create and add a Marker cluster to the site map
    # marker = folium.Marker(...)

    circle = folium.Circle([spacex_df['Lat'][i], spacex_df['Long'][i]], radius=500,
                          color=spacex_df['marker_color'][i], fill=True).add_child(folium.Popup(spacex_df['Launch Site'][i]))
    marker = folium.map.Marker([spacex_df['Lat'][i], spacex_df['Long'][i]],
                              icon=folium.Icon(color='white', icon_color=spacex_df['marker_color'][i])
                             )

    marker_cluster.add_child(marker)

site_map
```



<Folium Map Screenshot 3>

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- Explain the important elements and findings on the screenshot: *launch sites are close to highway and railroad, but far away from cities.*



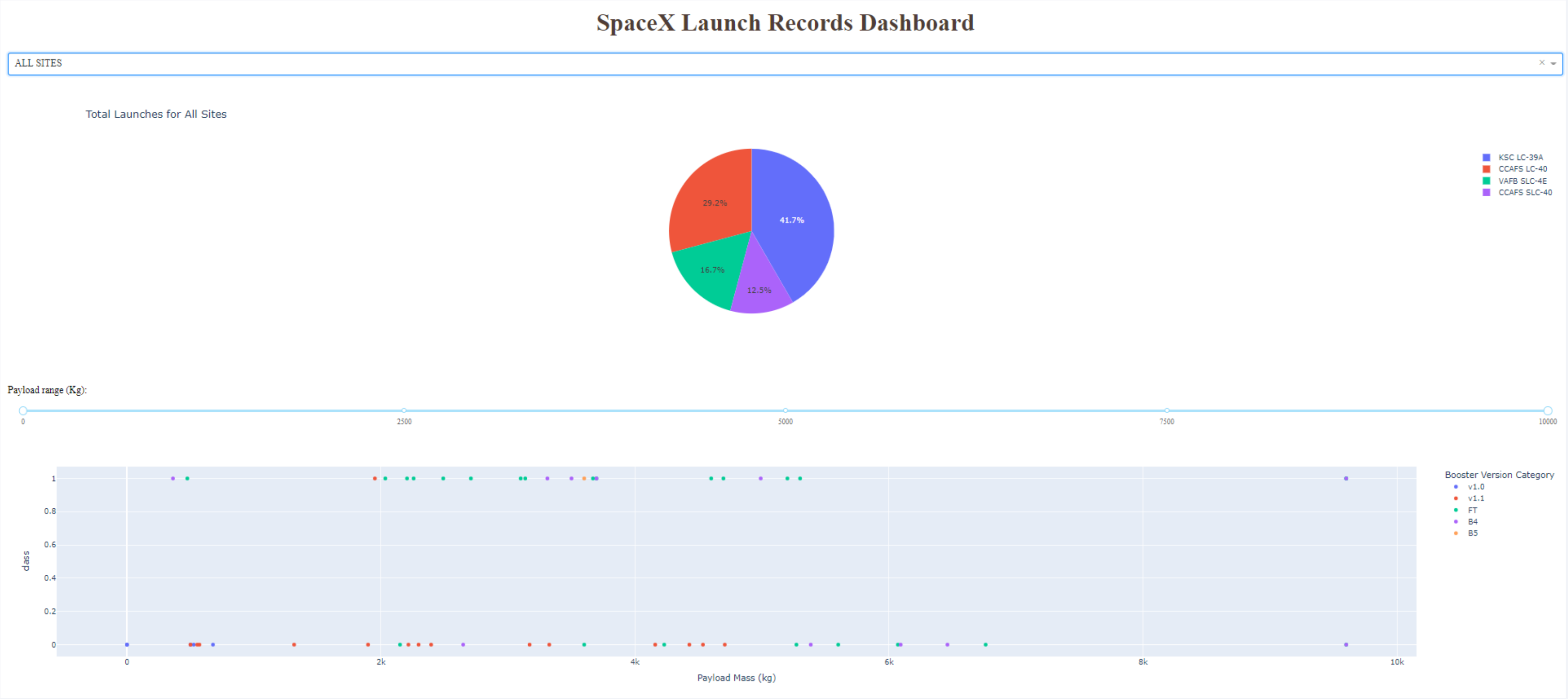


Section 4

Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot: *KSC has the highest success rate while VAFB has the lowest success rate.*

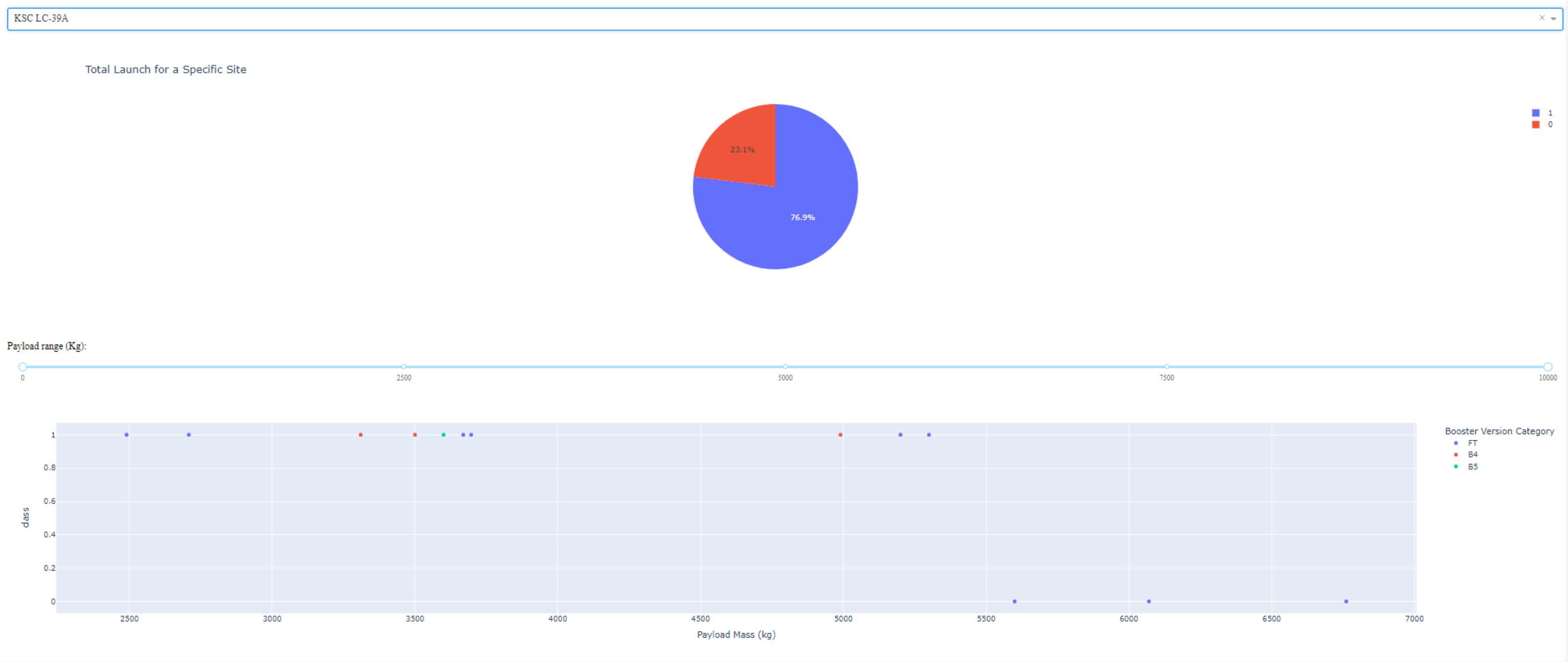


SpaceX Launch Records Dashboard

ALL SITES

Piechart for KSC LC-39A

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot:
  - Success rate was low for heavy payload (>5500 kg) for launch site KSC LC-39A



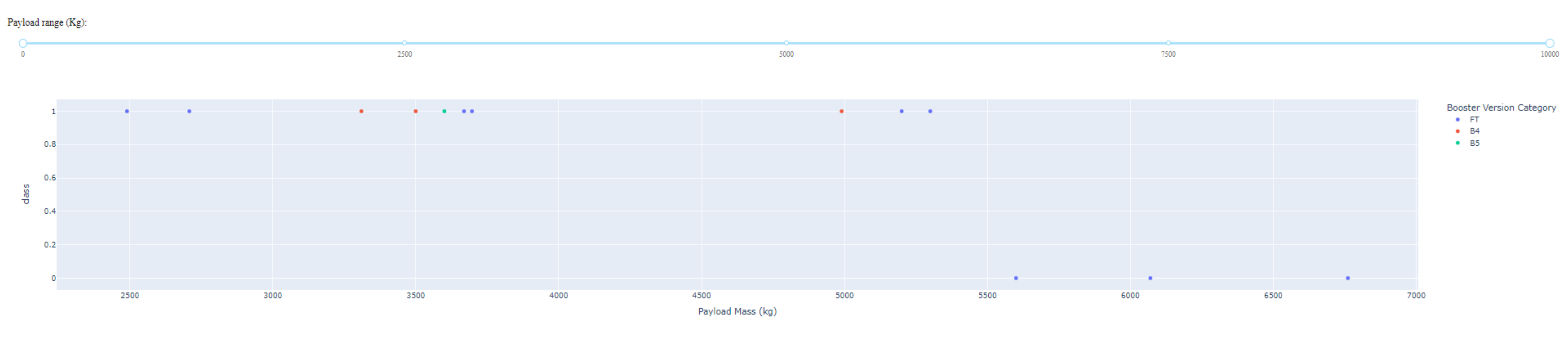
KSC LC-39A

Payload range (Kg):

Payload vs. Launch Outcome Scatter Plot

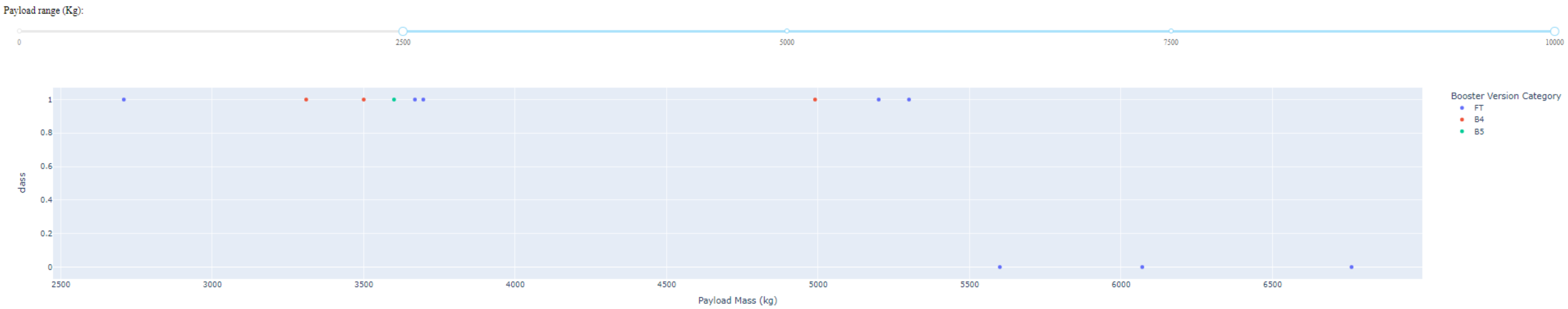
Continue to next slide

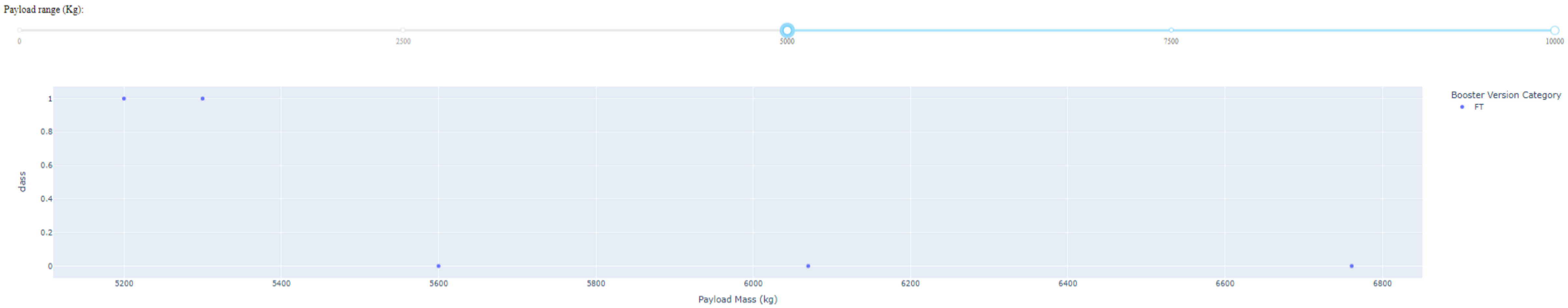
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.
  - Payload less than 5500 kg and booster version B5 have the largest success rate.



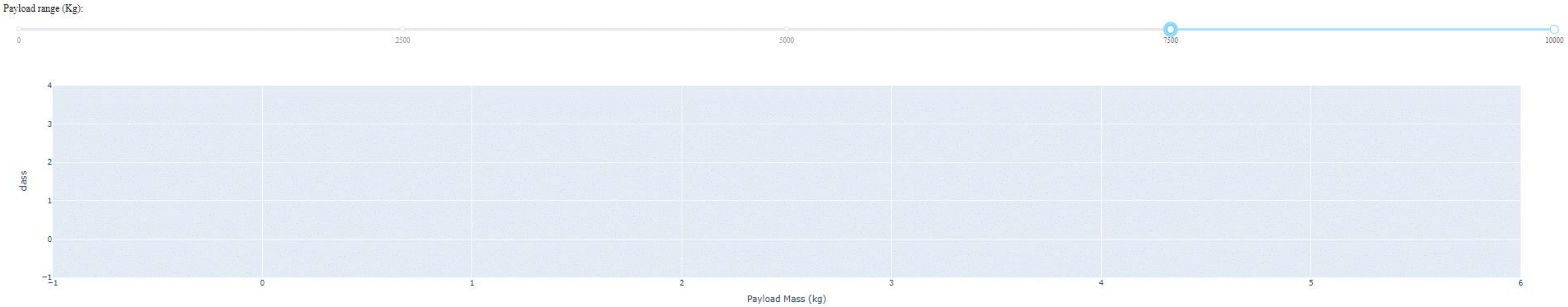
Payload vs. Launch Outcome Scatter Plot

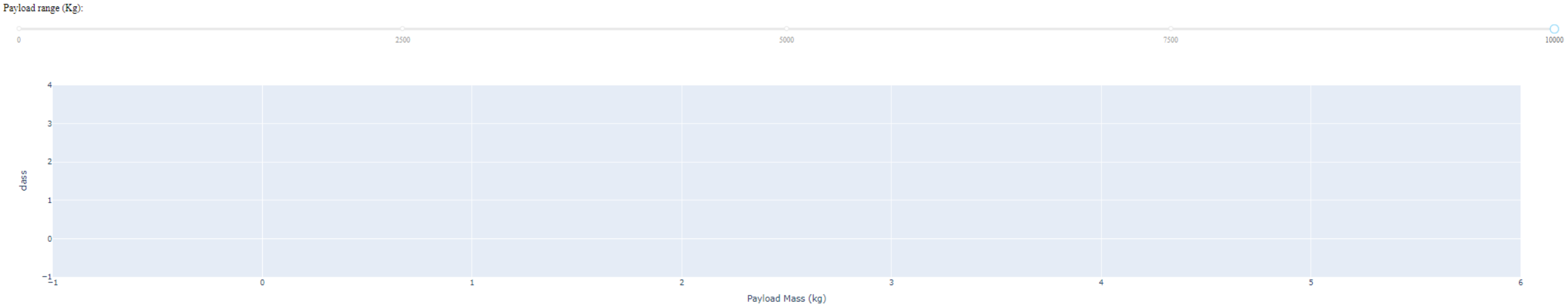
Continue to next slide





Payload vs. Launch Outcome Scatter Plot



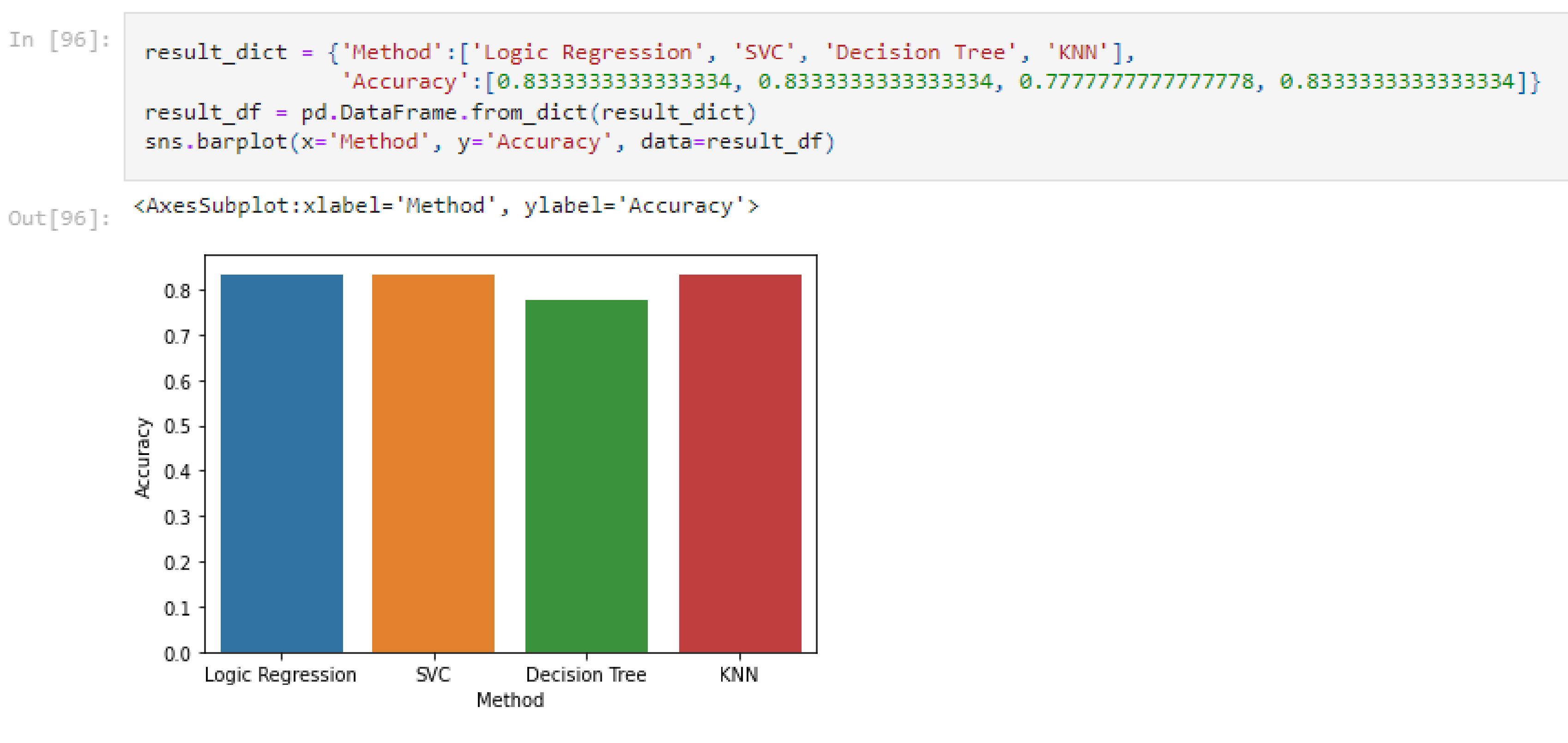


Section 5

Predictive Analysis (Classification)

Classification Accuracy

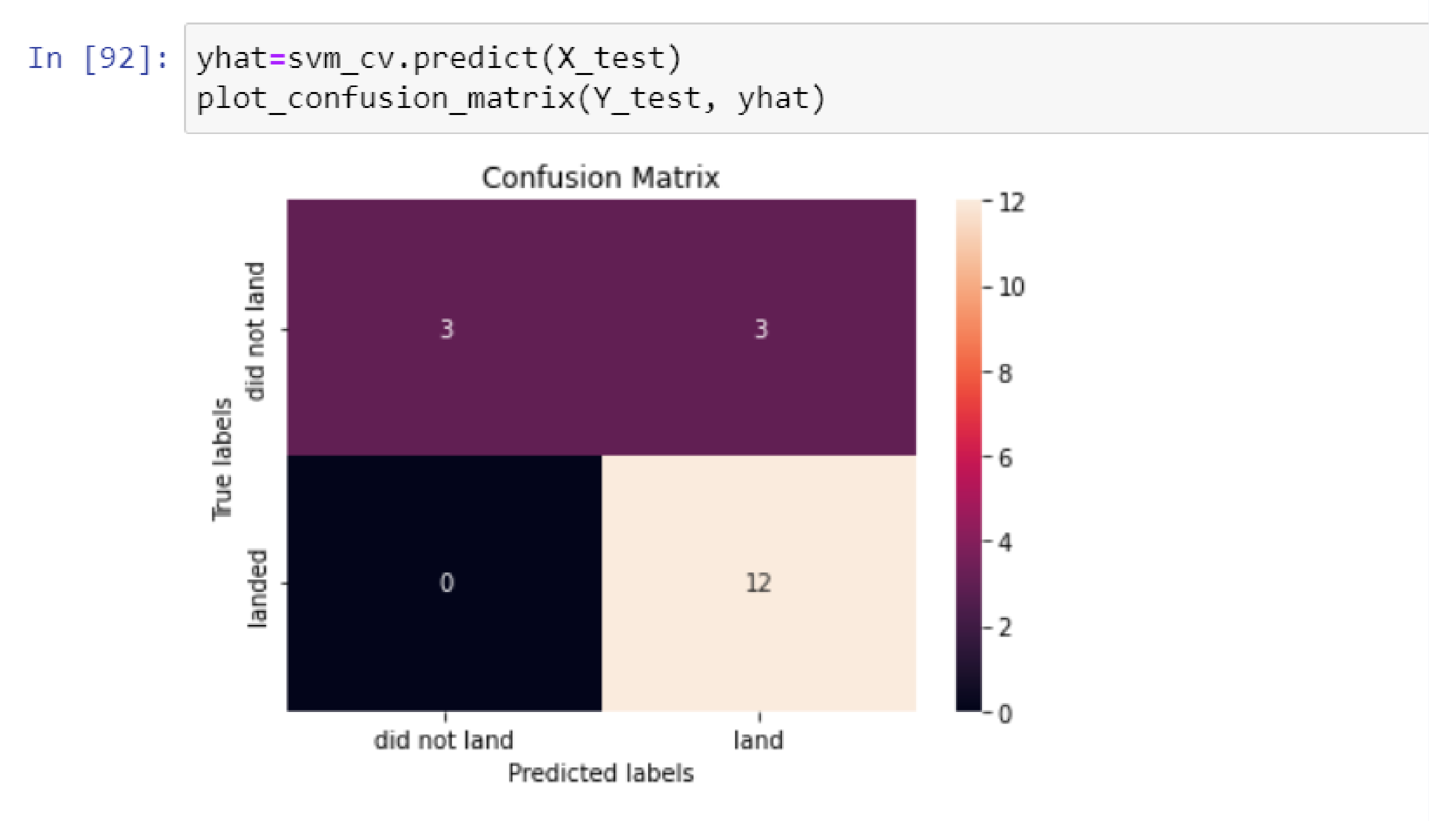
- Visualize the built model accuracy for all built classification models, in a bar chart
- Find which model has the highest classification accuracy: *SVC and KNN have the highest accuracy*.



```
In [96]:  result_dict = {'Method':['Logic Regression', 'SVC', 'Decision Tree', 'KNN'],
                          'Accuracy':[0.8333333333333334, 0.8333333333333334, 0.7777777777777778, 0.8333333333333334]}
          result_df = pd.DataFrame.from_dict(result_dict)
          sns.barplot(x='Method', y='Accuracy', data=result_df)

Out[96]: <AxesSubplot:xlabel='Method', ylabel='Accuracy'>
```

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation



In [92]:
```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test, yhat)
```

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!