*Data science innovation in space science

NAME: ANUJA
DATE : 30/12/2023

Outline

Executive Summary

- **Summary of methodologies:**

In this project we study that, the companies are making space travel available for everyone but most successful is space X, so we will study about the space X, if space X will reuse the 1st stage. And also it requires low cost as compare to other companies.

So we use the methodologies:

1. Data collection.
2. Data wrangling.
3. Exploratory data analysis (EDA) using visualization and SQL.
4. Interactive visual analytics using Folium and Plotly Dash.
5. predictive analysis using classification models.

- **Summary of all results:**

By using this methodologies we collect the data , clean the data, store it in panda data frame and replace the null values, perform the SQL queries to study the data insights, then we plot the data in terms of graph for better understanding of data insights. Then we use different classification model for better accuracy of model. Then we sumarise the result to get the final conclusion and reach our aim.

Introduction

- **Project background and context:**

1.In the 1st step we will predict if the Falcon 9 first stage will land successfully. 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.so we can find the reason of low cost of space X.

2. In the 2nd step, we will collect data on the Falcon 9 first-stage landings and perform the data wrangling operations.

3. In 3rd step we will build a dashboard to analyze launch records interactively with Plotly Dashboard,then build an interactive map to analyze the launch site proximity with Folium.

4. In 4th step will use machine learning to determine if the first stage of Falcon 9 will land successfully.

5. In final step, we will compile all activities into one place and deliver data-driven insights to determine if the first stage of Falcon 9 will land successfully.

- **Problems you want to find answers:**

1.Find the cost of launch. And find if space X can reuse the 1st step to minimize the cost.

2.How to improve the data finding methods to collect the accurate data for the study.

3.How to launch the site proximity with folium.

4. How to find the best Hyperparameter for SVM, Classification Trees, and Logistic Regression. Then find the method that performs best using test data.

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**
- **Perform data wrangling**
- **Dealing with missing values**
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Perform interactive visual analytics using Folium and Plotly Dash**
- **Perform predictive analysis using classification model.**
- **How to build, tune, evaluate classification models**

Data Collection

- Procees to collect data set:
- Request and parse the SpaceX launch data using the GET request
- decode the response content as a Json using .json()   turn it into a Pandas dataframe using .json_normalize()
- use the API again to get information about columns rocket, payloads, launchpad, and cores.
- construct our dataset using the data
- Filter the dataframe to only include Falcon 9
- data_falcon9.isnull().sum().(use for the missing values)
- Calculate the mean value of PayloadMass column
- Replace the np.nan values with its mean value

Data Collection – SpaceX API

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

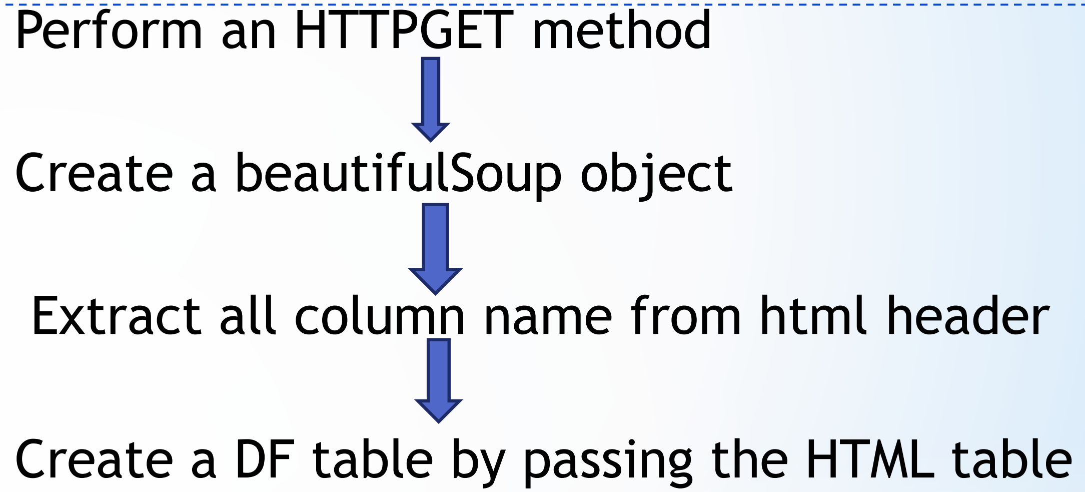
- *GitHub URL:https://github.com/anuja0591/aa/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

- * Request and parse the SpaceX launch data using the GET request
- * decode the response content as a Json using .json() turn it into a Pandas dataframe using .json_normalize()
- * use the API again to get information about columns rocket, payloads, launchpad, and cores.
- * construct our dataset using the data
- * Filter the dataframe to only include Falcon 9
- * data_falcon9.isnull().sum().(use for the missing values)
- * Calculate the mean value of PayloadMass column
- * Replace the np.nan values with its mean value

Place your flowchart of SpaceX API calls here

Data Collection - Scraping

* web scraping process using flowcharts



Perform an HTTPGET method

Create a beautifulSoup object

Extract all column name from html header

Create a DF table by passing the HTML table

* GitHub URL https://github.com/anuja0591/aa/blob/main/jupyter-labs-webscraping%20(5).ipynb

Place your flowchart of web scraping here

Data Wrangling

* Data wrangling process:
* Calculate the number of launches on each site by Useing the method value_counts() on the column LaunchSite to determine the number of launches on each site Calculate the number and occurrence of each orbit by Using the method .value_counts() to determine the number and occurrence of each orbit in the column Orbit

* Calculate the number and occurence of mission outcome of the orbits by Using the method .value_counts() on the column Outcome to determine the number of landing_outcomes.Then assign it to a variable landing_outcomes.

* Create a landing outcome label from Outcome column by Using the Outcome, create a list where the element is zero if the corresponding row in Outcome is in the set bad_outcome,otherwise, it's one. Then assign it to the variable landing_class:
* GitHub URL :**https://github.com/anuja0591/aa/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb**

EDA with Data Visualization

* what charts were plotted and why you used those charts:
* We plot out the FlightNumber vs. PayloadMassand overlay the outcome of the launch. We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.In the catplot We see that different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.
* then Use the function catplot to plot FlightNumber vs LaunchSite, there is no prominent relationship between them. Next we observe if there is any relationship between launch sites and their payload mass.
* We observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).
* we want to visually check if there are any relationship between success rate and orbit type. create a bar chart for the sucess rate of each orbit
* You should see 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.
* we can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and Orbit type we see With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.
* Then plot a line chart with x axis to be Year and y axis to be average success rate, to get the average launch success trend. can observe that the sucess rate since 2013 kept increasing till 2020.
* Use the function get_dummies and features dataframe to apply OneHotEncoder to the column Orbits, LaunchSite, LandingPad, and Serial.

* GitHub URL: **https://github.com/anuja0591/aa/blob/main/edadataviz.ipynb**

EDA with SQL

- **Display the names of the unique launch sites in the space mission**
- %%sql select DISTINCT launch_site from SPACEXTABLE;
- **Display 5 records where launch sites begin with the string 'CCA'**
- %%sql select * from SPACEXTABLE WHERE launch_site LIKE 'CCA%' LIMIT 5;
- **Display the total payload mass carried by boosters launched by NASA (CRS)**
- %%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';
- **Display average payload mass carried by booster version F9 v1.1**
- %%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1. ↪1';
- **List the date when the first succesful landing outcome in ground pad was acheived.**
- %%sql SELECT MIN(Date)FROM SPACEXTABLE WHERE Landing_Outcome = 'Controlled (ocean)';

EDA with SQL

* **List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000**
* %%sql select (Booster_Version) from SPACEXTABLE WHERE PAYLOAD_MASS__KG_ BETWEEN 4000␣ ↪AND 6000;
* **List the total number of successful and failure mission outcomes**
* %%sql SELECT count(Mission_Outcome) from SPACEXTABLE;
* **List the names of the booster_versions which have carried the maximum payload mass. Use a subquery**
* %%sql SELECT Booster_Version FROM SPACEXTABLE GROUP BY Booster_Version HAVING MAX(PAYLOAD_MASS__KG_);
* **List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.**
* %%sql select Date,Booster_version, Launch_Site, Landing_Outcome from SPACEXTABLE␣ ↪limit 10;
* **Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.**
* %%sql SELECT COUNT(Landing_Outcome) FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY COUNT(Landing_Outcome) DESC;
*
* **GITHUBURL:https://github.com/anuja0591/aa/blob/main/jupyter-labs-eda-sql-coursera_sqllite.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
* We first create a folium Map object, with an initial center location to be NASA Johnson Space Center at Houston, Texas.
* We use folium.Circle to add a highlighted circle area with a text label on a specific coordinate.add a circle for each launch site in data frame launch_sites.
* Now, we explore the map by zoom-in/out the marked areas , and find answer the following questions:
* Are all launch sites in proximity to the Equator line?
* Are all launch sites in very close proximity to the coast?
* now create markers for all launch records. If a launch was successful (class=1), then we use a green marker and if a launch was failed, we use a red marker (class=0)
* Then we to explore and analyze the proximities of launch sites.
* We first add a MousePosition on the map to get coordinate for a mouse over a point on the map. As such, while exploring the map, can easily find the coordinates of any points of interests (such as railway)
* Mark down a point on the closest coastline using MousePosition and calculate the distance between the coastline point and the launch site.
* After we plot distance lines to the proximities, we an answer the following questions easily: 1.Are launch sites in close proximity to railways? 2.Are launch sites in close proximity to highways? 3.Are launch sites in close proximity to coastline? 4.Do launch sites keep certain distance away from cities?
* **Github url:https://github.com/anuja0591/aa/blob/main/lab_jupyter_launch_site_location%20(1).ipynb**

Build a Dashboard with Plotly Dash

* Summarize what plots/graphs and interactions you have added to a dashboard
* we build a dashboard application with the Python Plotly Dash package.
* This dashboard application contains input components such as a dropdown list and a range
* slider to interact with a pie chart and a scatter point chart.
* After the dashboard is built, we can use it to find more insights from the SpaceX dataset
* more easily than with static graphs.
* Instead of presenting your findings in static graphs, interactive data visualization, or
* dashboarding, can always tell a more appealing story.

Add the GitHub URL:

Predictive Analysis (Classification)

* We perform the following steps to find the best model:
* Perform exploratory Data Analysis and determine Training Labels
* create a column for the class
* Standardize the data
* Split into training data and test data
* -Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
* Find the method performs best using test data
* We perform: 1. Create a logistic regression object then create a GridSearchCV object logreg_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.
* 2. Create a support vector machine object then create a GridSearchCV object svm_cv with cv - 10. Fit the object to find the best parameters from the dictionary parameters.
* 3. Create a decision tree classifier object then create a GridSearchCV object tree_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.
* 4. Create a k nearest neighbors object then create a GridSearchCV object knn_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.
* logistic Regression model performs the best among all hacing the accuracy 0.833333333334

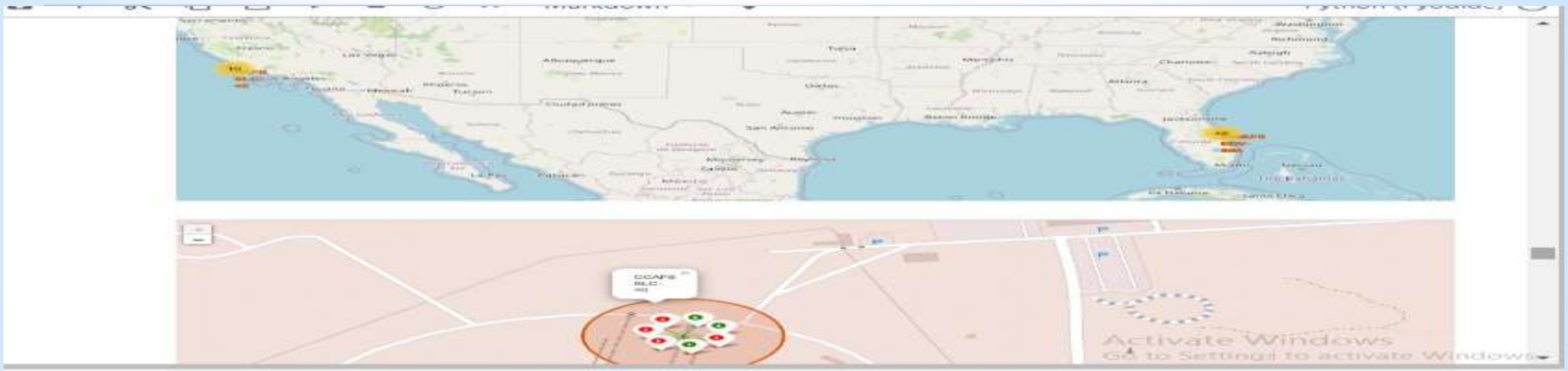
* **Github url:https://github.com/anuja0591/aa/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5%20(2).ipynb**

Results

- **Exploratory data analysis results: the names of the unique launch sites:**
- `: %%sql select DISTINCT launch_site from SPACEXTABLE;` Result is:
- `[('CCAFS LC-40',), ('VAFB SLC-4E',), ('KSC LC-39A',), ('CCAFS SLC-40',)]`
- 5 records where launch sites begin with `CCA`:
- `%%sql select * from SPACEXTABLE WHERE launch_site LIKE 'CCA%' LIMIT 5;`
- Calculate the total payload carried by boosters from NASA:
- `%%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';`
- Result is:
- `* sqlite:///my_data1.db Done. [27]: [(None,)]`
- Calculate 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';`
- Result is:`: [(2928.4,)]`
- Calculate the total number of successful and failure mission outcomes
- `%%sql SELECT count(Mission_Outcome) from SPACEXTABLE;`
- Result is:`[(101,)]`

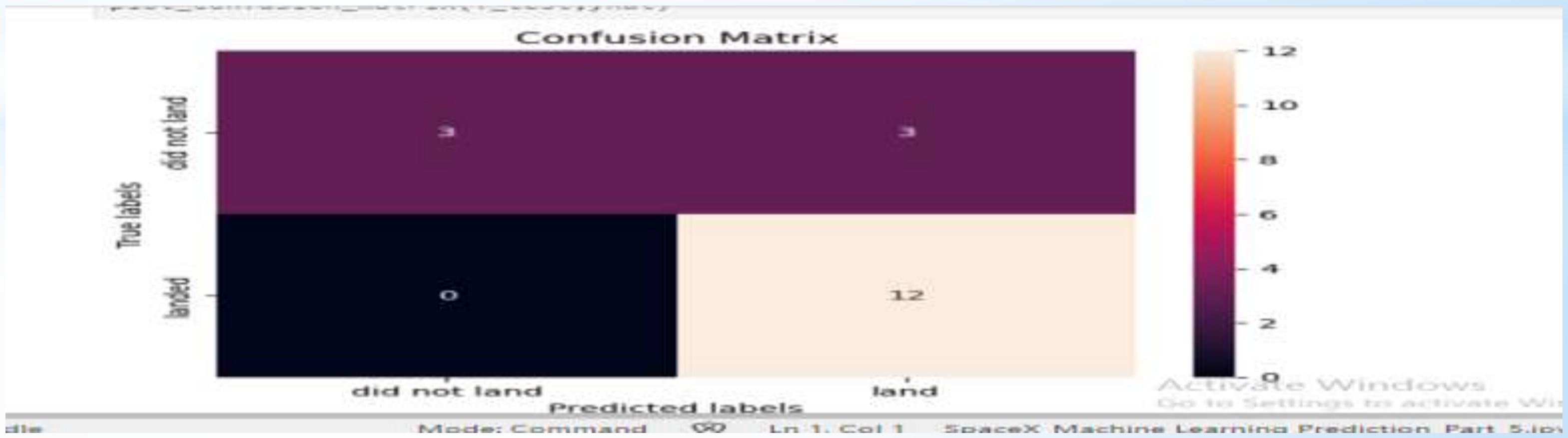
Results

Interactive analytics demo in screenshots



- Predictive analysis results:

Logistic regression is the best performing model because it has highest accuracy 0.833333334 than other model.

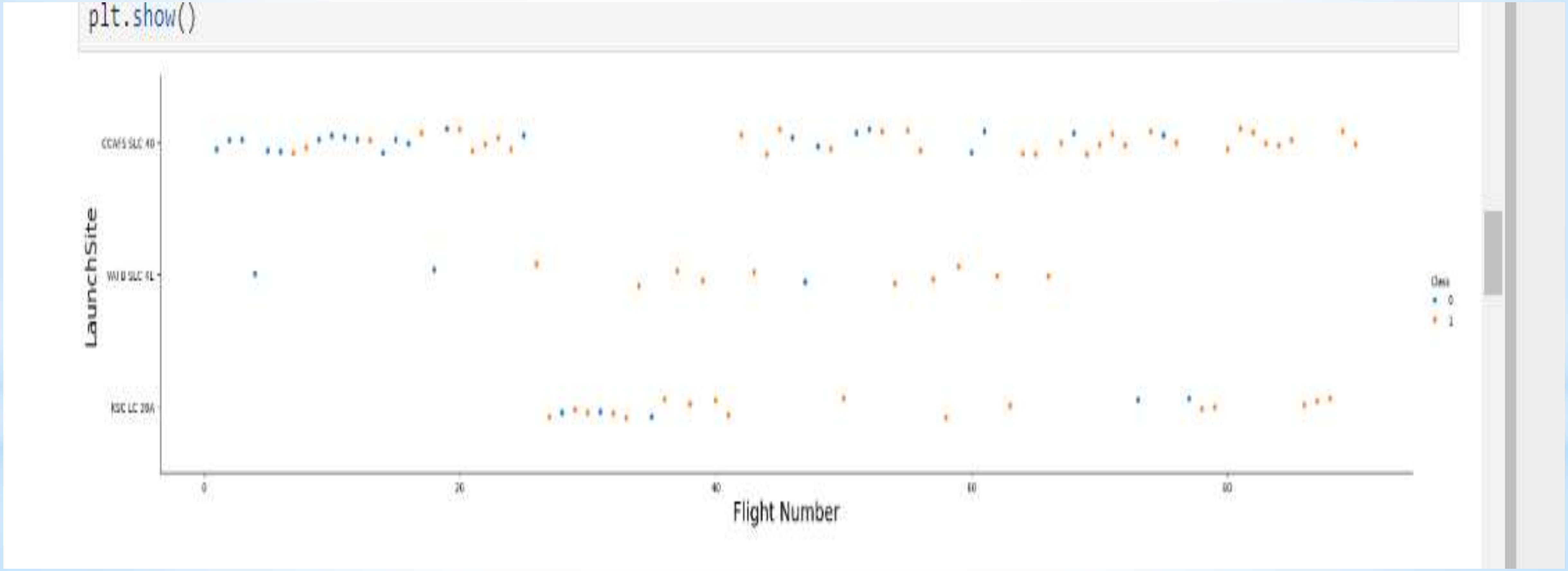


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

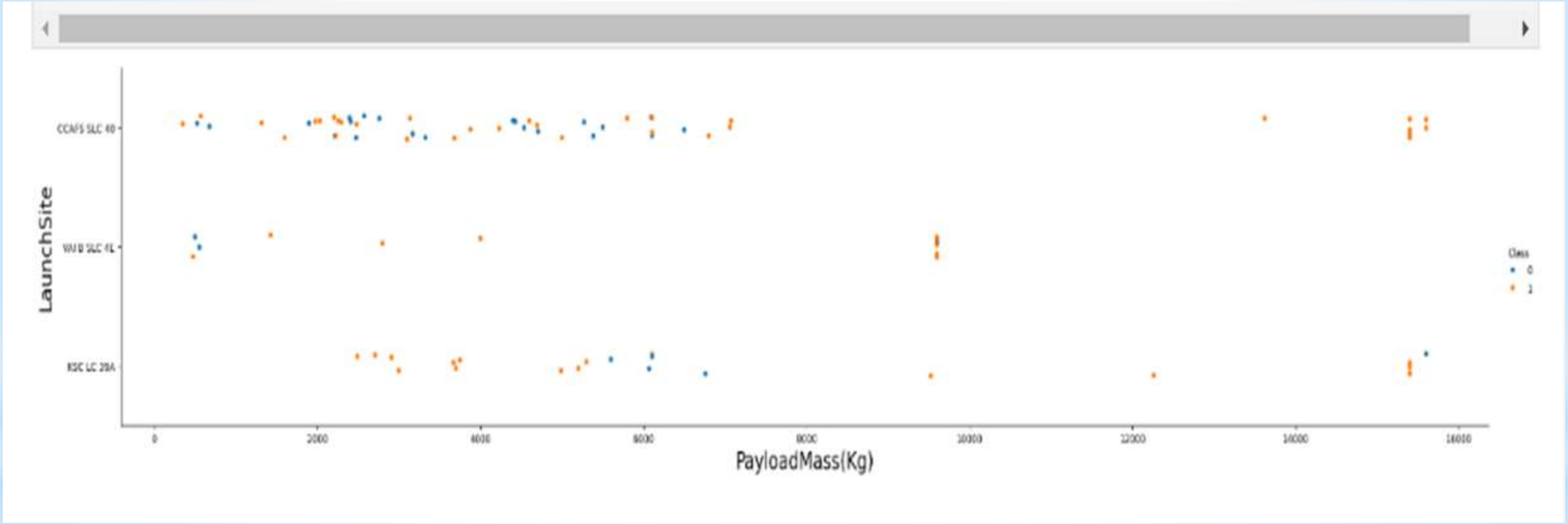
*a scatter plot of Flight Number vs. Launch Site:



```
plt.show()
```

Payload vs. Launch Site

* Show a scatter plot of Payload vs. Launch Site



Success Rate vs. Orbit Type

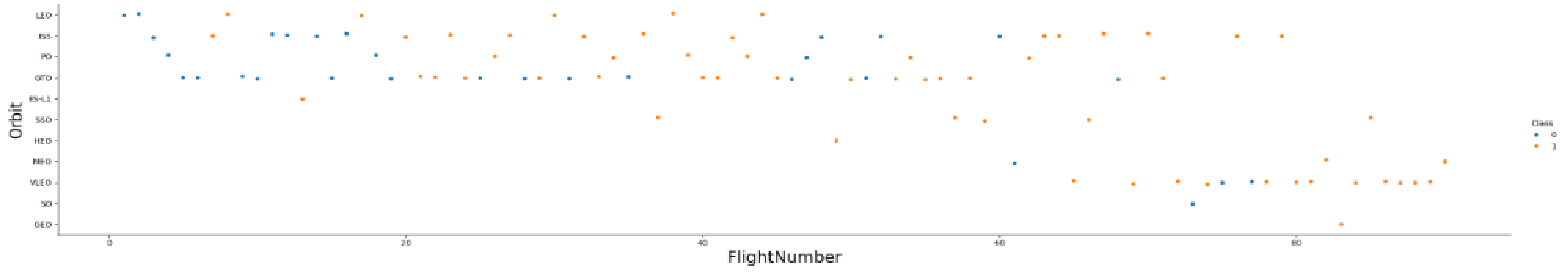
- bar chart for the success rate of each orbit type

```
plt.ylabel(y)
plt.show()
```

Flight Number vs. Orbit Type

* scatter point of Flight number vs. Orbit type

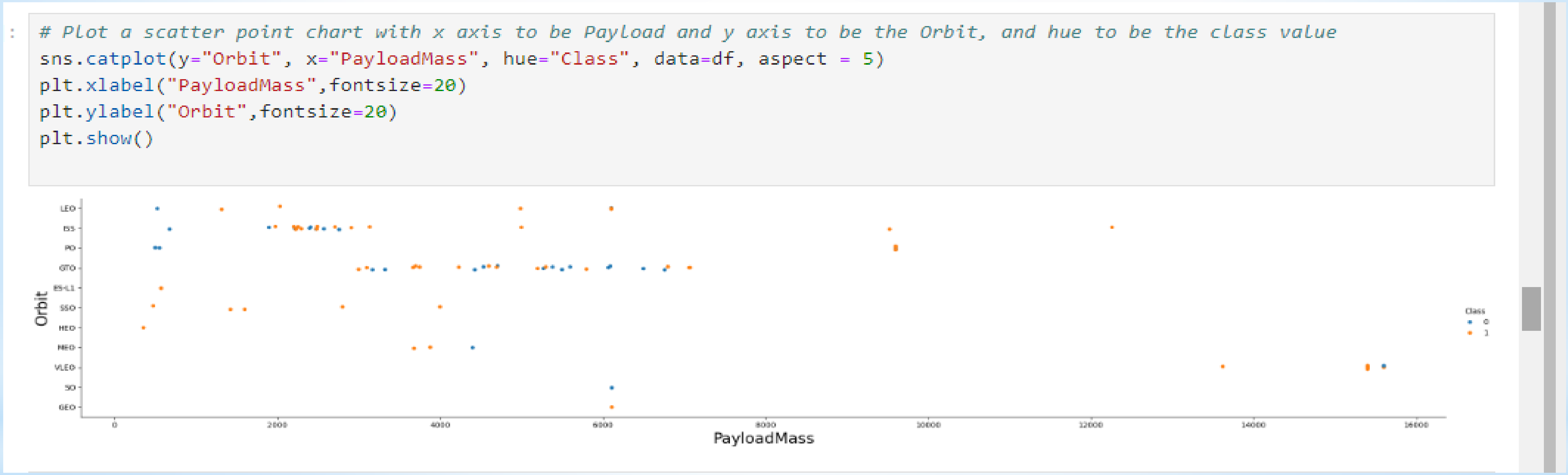
```
### TASK  4: Visualize the relationship between FlightNumber and Orbit type
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



* scatter plot with explanations:You should see 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

* Show a scatter point of payload vs. orbit type



```
# 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 = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

* scatter plot with explanations:With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
* However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

All Launch Site Names

* the names of the unique launch sites:
*: %%sql select DISTINCT launch_site from SPACEXTABLE;
*Result is:
*[('CCAFS LC-40',), ('VAFB SLC-4E',), ('KSC LC-39A',), ('CCAFS SLC-40',)]

Launch Site Names Begin with 'CCA'

* 5 records where launch sites begin with `CCA`:
* %%sql select * from SPACEXTABLE WHERE launch_site LIKE 'CCA%' LIMIT 5;
* Result is:
*
```
[('2010-06-04', '18:45:00', 'F9 v1.0 B0003', 'CCAFS LC-40', 'Dragon Spacecraft Qualification Unit', 0, 'LEO', 'SpaceX', 'Success', 'Failure (parachute)'), ('2010-12-08', '15:43:00', 'F9 v1.0 B0004', 'CCAFS LC-40', 'Dragon demo flight C1, two CubeSats, barrel of Brouere cheese', 0, 'LEO (ISS)', 'NASA (COTS) NRO', 'Success', 'Failure (parachute)'), ('2012-05-22', '7:44:00', 'F9 v1.0 B0005', 'CCAFS LC-40', 'Dragon demo flight C2', 525, 'LEO (ISS)', 'NASA (COTS)', 'Success', 'No attempt'), ('2012-10-08', '0:35:00', 'F9 v1.0 B0006', 'CCAFS LC-40', 'SpaceX CRS-1', 500, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'No attempt'), ('2013-03-01', '15:10:00', 'F9 v1.0 B0007', 'CCAFS LC-40', 'SpaceX CRS-2', 677, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'No attempt')]
```

Total Payload Mass

- Calculate the total payload carried by boosters from NASA:
- %%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';
- Result is:
- * sqlite:///my_data1.db Done. [27]: [(None,)]

Average Payload Mass by F9 v1.1

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

-
  ```
  %%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.
  ↪1';
  ```
- Result is:: `[(2928.4,)]`

First Successful Ground Landing Date

* the dates of the first successful landing outcome on ground pad
* %%sql SELECT MIN(Date)FROM SPACEXTABLE WHERE Landing_Outcome = 'Controlled (ocean)';
* Result is:[('2014-04-18',)]
* %%sql SELECT MIN(Date)FROM SPACEXTABLE WHERE Mission_Outcome = 'Success';
* Result is:
* [('2010-06-04',)]

Successful Drone Ship Landing with Payload between 4000 and 6000

* List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

* %%sql select (Booster_Version) from SPACEXTABLE WHERE PAYLOAD_MASS__KG_ BETWEEN 4000␣ ↪AND 6000;

* Result is:[('F9 v1.1',), ('F9 v1.1 B1011',), ('F9 v1.1 B1014',), ('F9 v1.1 B1016',), ('F9 FT B1020',), ('F9 FT B1022',), ('F9 FT B1026',), ('F9 FT B1030',), ('F9 FT B1021.2',), ('F9 FT B1032.1',), ('F9 B4 B1040.1',), ('F9 FT B1031.2',), ('F9 B4 B1043.1',), ('F9 FT B1032.2',), ('F9 B4 B1040.2',), ('F9 B5 B1046.2',), ('F9 B5 B1047.2',), ('F9 B5 B1046.3',), ('F9 B5B1054',), ('F9 B5 B1048.3',), ('F9 B5 B1051.2 ',), ('F9 B5B1060.1',), ('F9 B5 B1058.2 ',), ('F9 B5B1062.1',)]

Total Number of Successful and Failure Mission Outcomes

* Calculate the total number of successful and failure mission outcomes
* %%sql SELECT count(Mission_Outcome) from SPACEXTABLE;
* Result is:[(101,)]

Boosters Carried Maximum Payload

* List the names of the booster which have carried the maximum payload mass
* :%%sql SELECT Booster_Version FROM SPACEXTABLE GROUP BY Booster_Version HAVING MAX(PAYLOAD_MASS__KG_);
* Result is:[('F9 B4 B1039.2',), ('F9 B4 B1040.2',), ('F9 B4 B1041.2',), ('F9 B4 B1043.2',), ('F9 B4 B1039.1',), ('F9 B4 B1040.1',), ('F9 B4 B1041.1',), ('F9 B4 B1042.1',), ('F9 B4 B1043.1',), ('F9 B4 B1044',), ('F9 B4 B1045.1',), ('F9 B4 B1045.2',), ('F9 B5 B1046.1',), ('F9 B5 B1046.2',), ('F9 B5 B1046.3',), ('F9 B5 B1046.4',), ('F9 B5 B1047.2',), ('F9 B5 B1047.3 ',), ('F9 B5 B1048.2',), ('F9 B5 B1048.3',), ('F9 B5 B1048.4',), ('F9 B5 B1048.5',), ('F9 B5 B1049.2',), ('F9 B5 B1049.3',), ('F9 B5 B1049.4',), ('F9 B5 B1049.5',), ('F9 B5 B1049.6',), ('F9 B5 B1049.7 ',), ('F9 B5 B1051.2 ',), ('F9 B5 B1051.3',), ('F9 B5 B1051.4',), ('F9 B5 B1051.5',), ('F9 B5 B1051.6',), ('F9 B5 B1056.2 ',), ('F9 B5 B1056.3 ',), ('F9 B5 B1056.4',), ('F9 B5 B1058.2 ',), ('F9 B5 B1058.3 ',), ('F9 B5 B1058.4 ',), ('F9 B5 B1059.2',), ('F9 B5 B1059.3',), ('F9 B5 B1059.4',), ('F9 B5 B1060.2 ',), ('F9 B5 B1060.3',), ('F9 B5B1047.1',), ('F9 B5B1048.1',), ('F9 B5B1049.1',), ('F9 B5B1050',), ('F9 B5B1051.1',), ('F9 B5B1054',), ('F9 B5B1056.1 ',), ('F9 B5B1058.1 ',), ('F9 B5B1059.1',), ('F9 B5B1060.1',), ('F9 B5B1061.1 ',), ('F9 B5B1062.1',), ('F9 B5B1063.1',), ('F9 FT B1021.2',), ('F9 FT B1029.2',), ('F9 FT B1031.2',),

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015:
- %%sql select Date,Booster_version, Launch_Site, Landing_Outcome from SPACEXTABLE␣
↪limit 10;
- Result is:[('2010-06-04', 'F9 v1.0 B0003', 'CCAFS LC-40', 'Failure (parachute)'), ('2010-12-08', 'F9 v1.0 B0004', 'CCAFS LC-40', 'Failure (parachute)'), ('2012-05-22', 'F9 v1.0 B0005', 'CCAFS LC-40', 'No attempt'), ('2012-10-08', 'F9 v1.0 B0006', 'CCAFS LC-40', 'No attempt'), ('2013-03-01', 'F9 v1.0 B0007', 'CCAFS LC-40', 'No attempt'), ('2013-09-29', 'F9 v1.1 B1003', 'VAFB SLC-4E', 'Uncontrolled (ocean)'), ('2013-12-03', 'F9 v1.1', 'CCAFS LC-40', 'No attempt'), ('2014-01-06', 'F9 v1.1', 'CCAFS LC-40', 'No attempt'), ('2014-04-18', 'F9 v1.1', 'CCAFS LC-40', 'Controlled (ocean)'), ('2014-07-14', 'F9 v1.1', 'CCAFS LC-40', 'Controlled (ocean)')]

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

* Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order:
* %%sql
* SELECT COUNT(Landing_Outcome) FROM SPACEXTABLE
* WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
* GROUP BY Landing_Outcome
* ORDER BY COUNT(Landing_Outcome) DESC;
* Result:

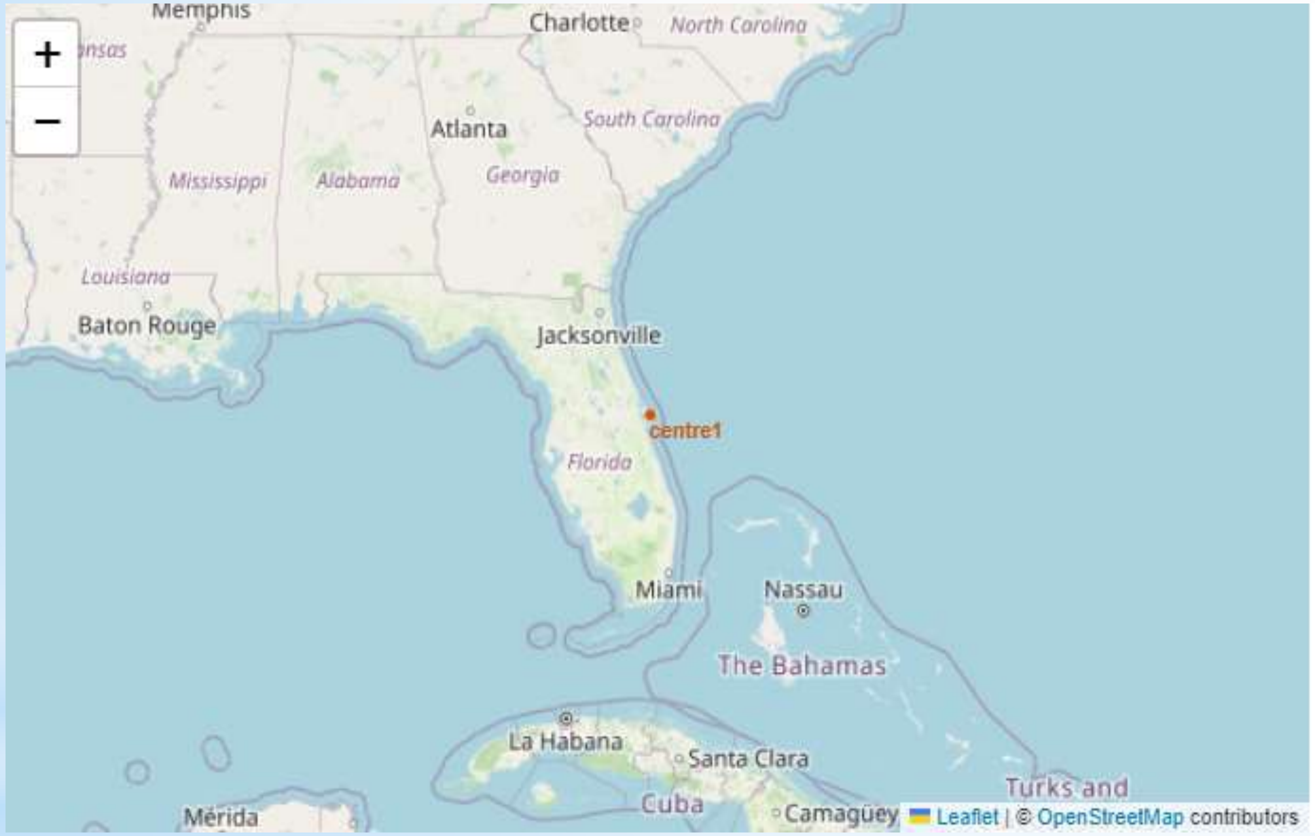
```
[(10,), (5,), (5,), (3,), (3,), (2,), (2,), (1,)]
```

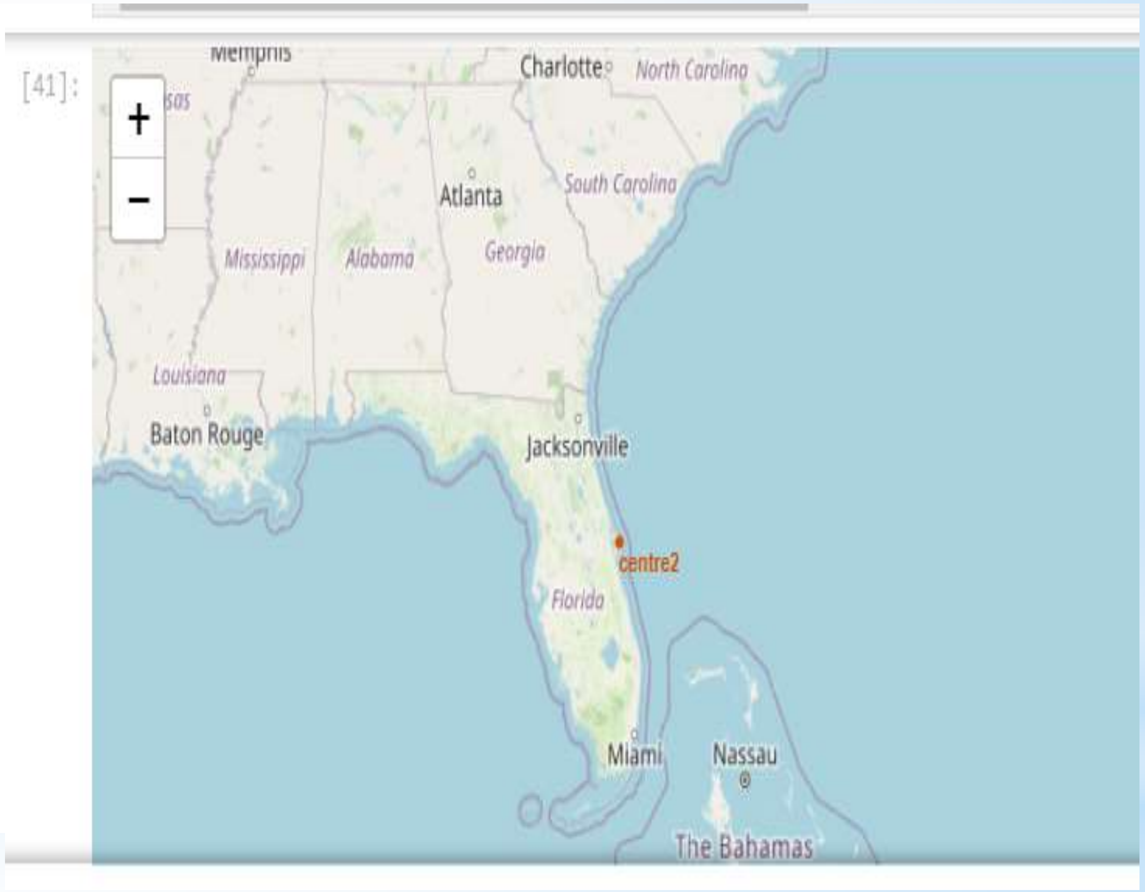
Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

*Folium map screenshot for all 4 locations .

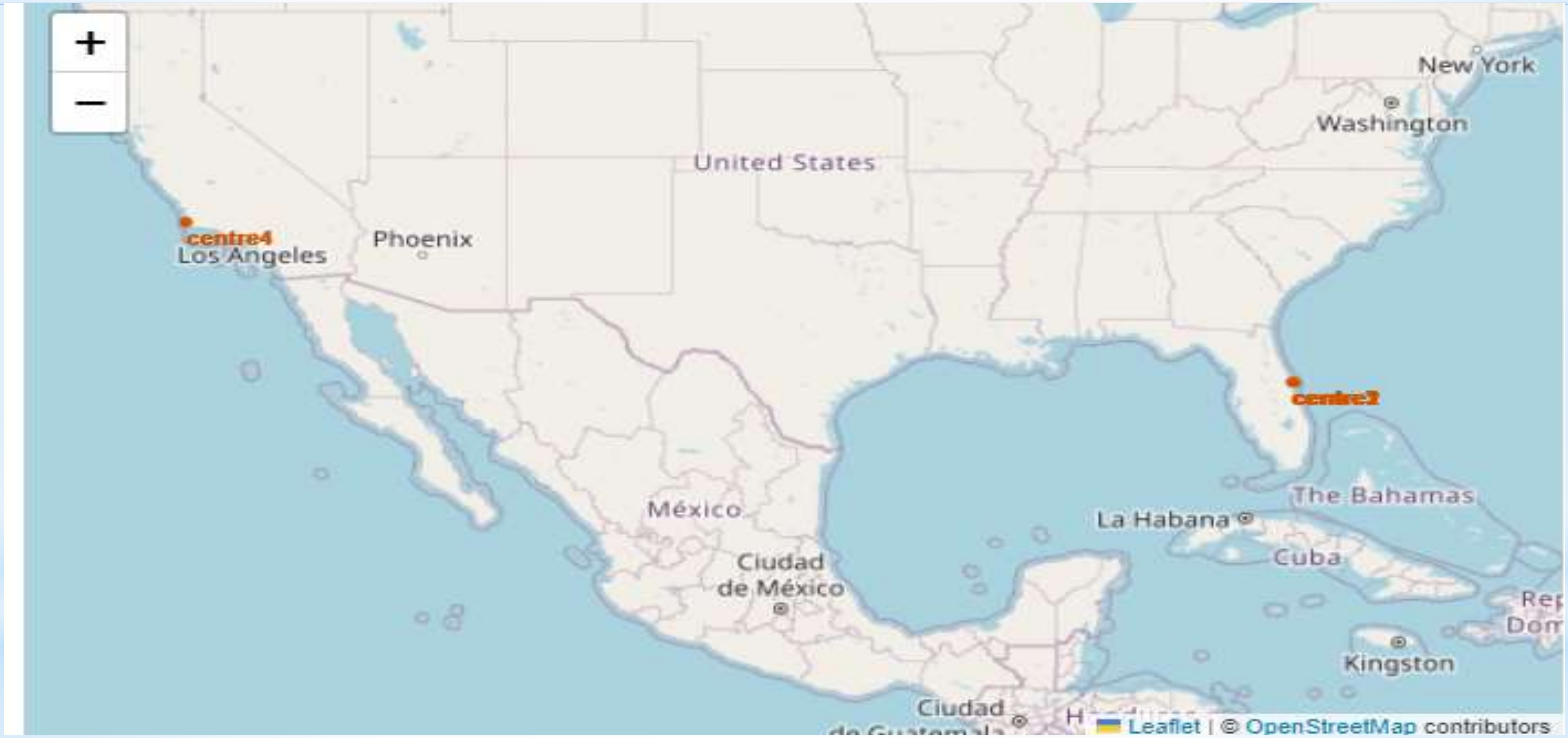




*Here the name is given center1 and center2 for the given liocation.

<Folium Map Screenshot 1>

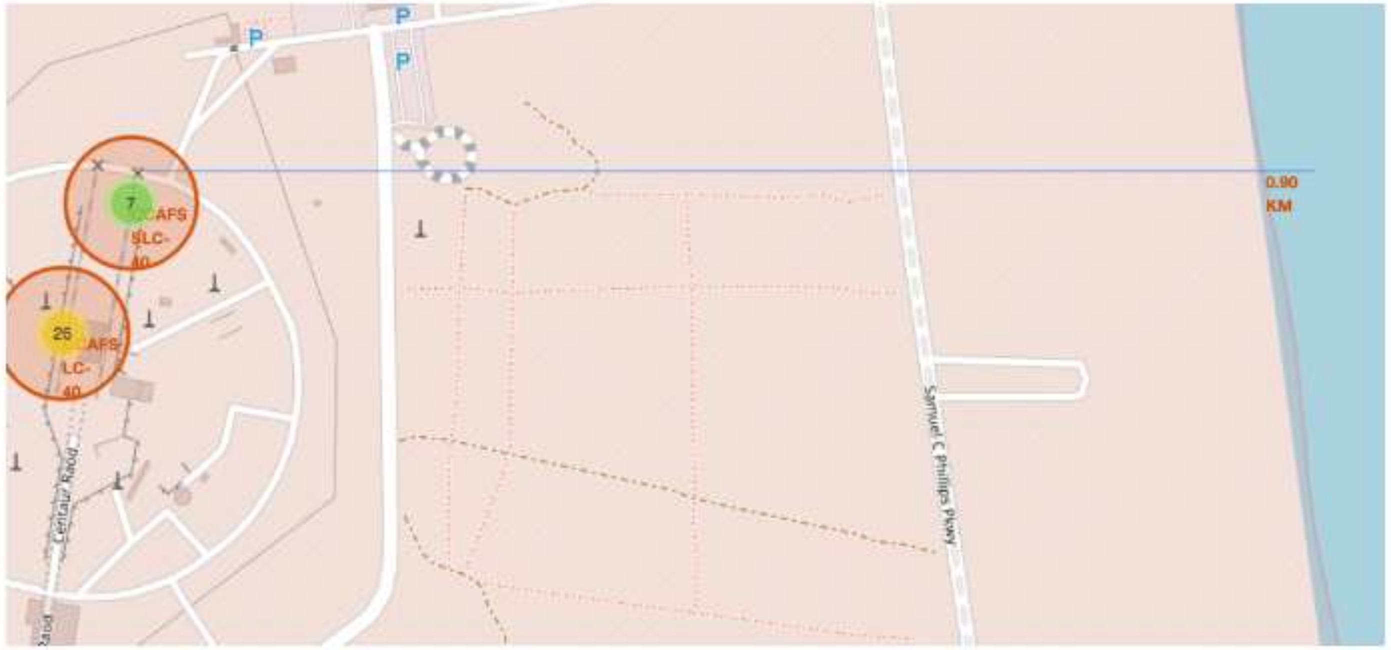
*Folium map screenshot for all 4 locations .



*Here the name is given center 3 and center 4.for the given location./

<Folium Map Screenshot 2>

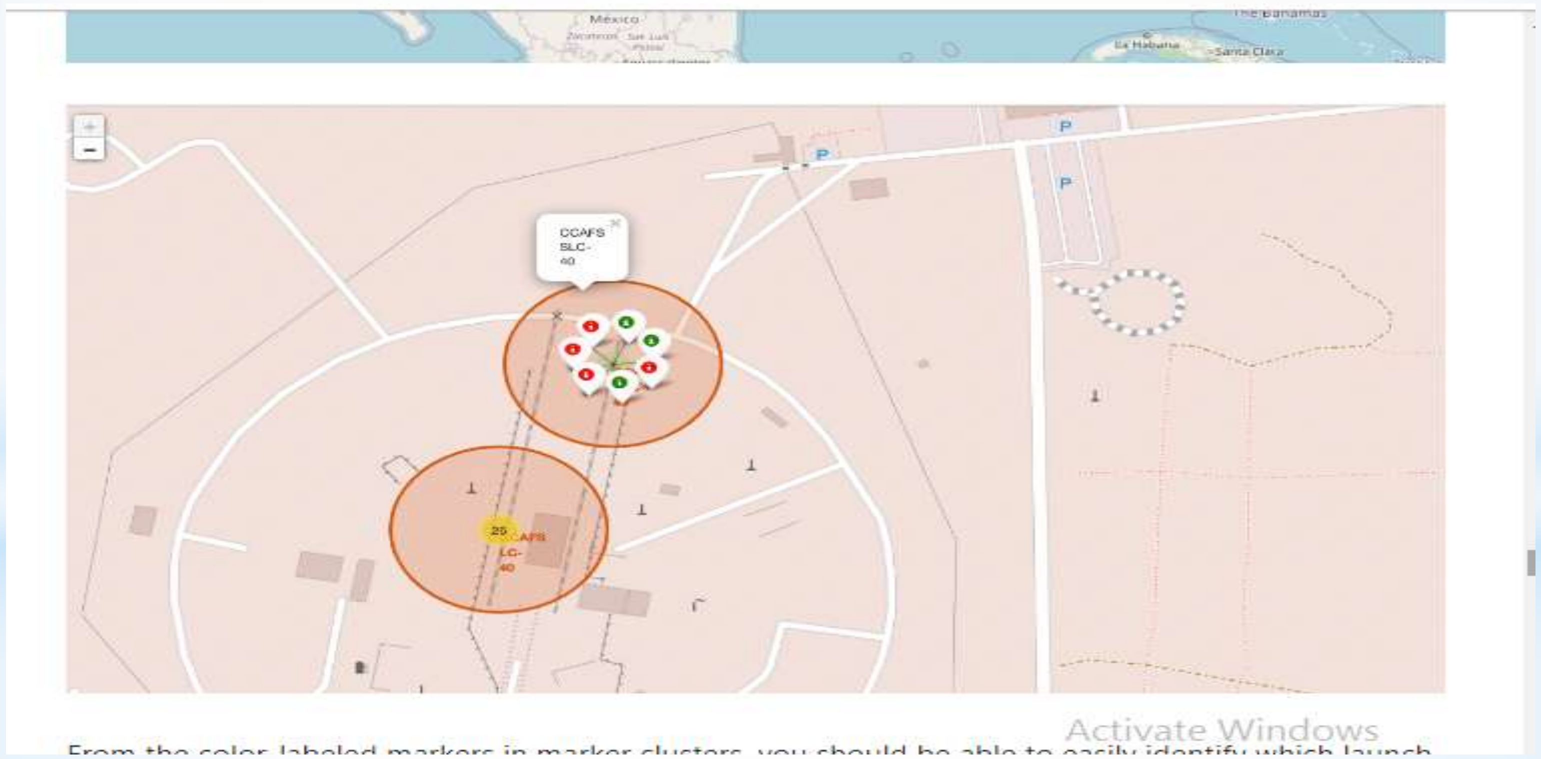
*folium map and make to show the color-labeled launch outcomes on the map



TODO: Similarly, you can draw a line betwee a launch site to its closest city, railway, highway, etc. You

<Folium Map Screenshot 3>

* generated folium map showing the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed



From the color labeled markers in marker clusters, you should be able to easily identify which launch

Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

* 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

<Dashboard Screenshot 2>

* 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

<Dashboard Screenshot 3>

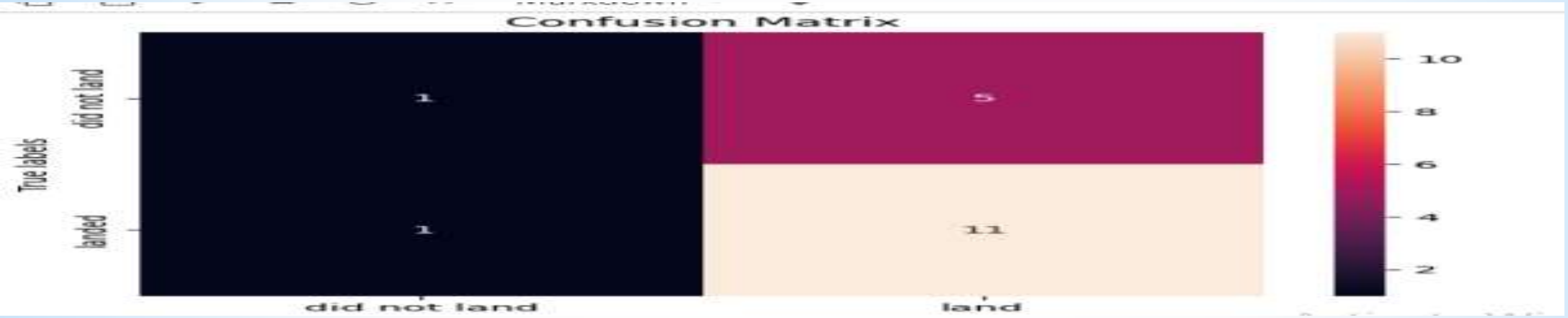
* Replace <Dashboard screenshot 3> title with an appropriate title
* Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
* Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

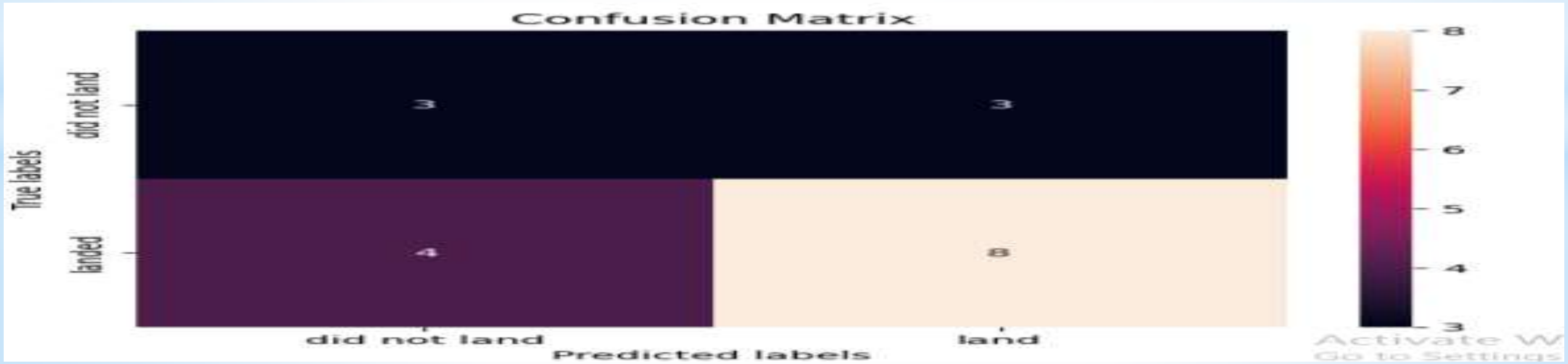
Predictive Analysis (Classification)

Classification Accuracy

Find which model has the highest classification accuracy

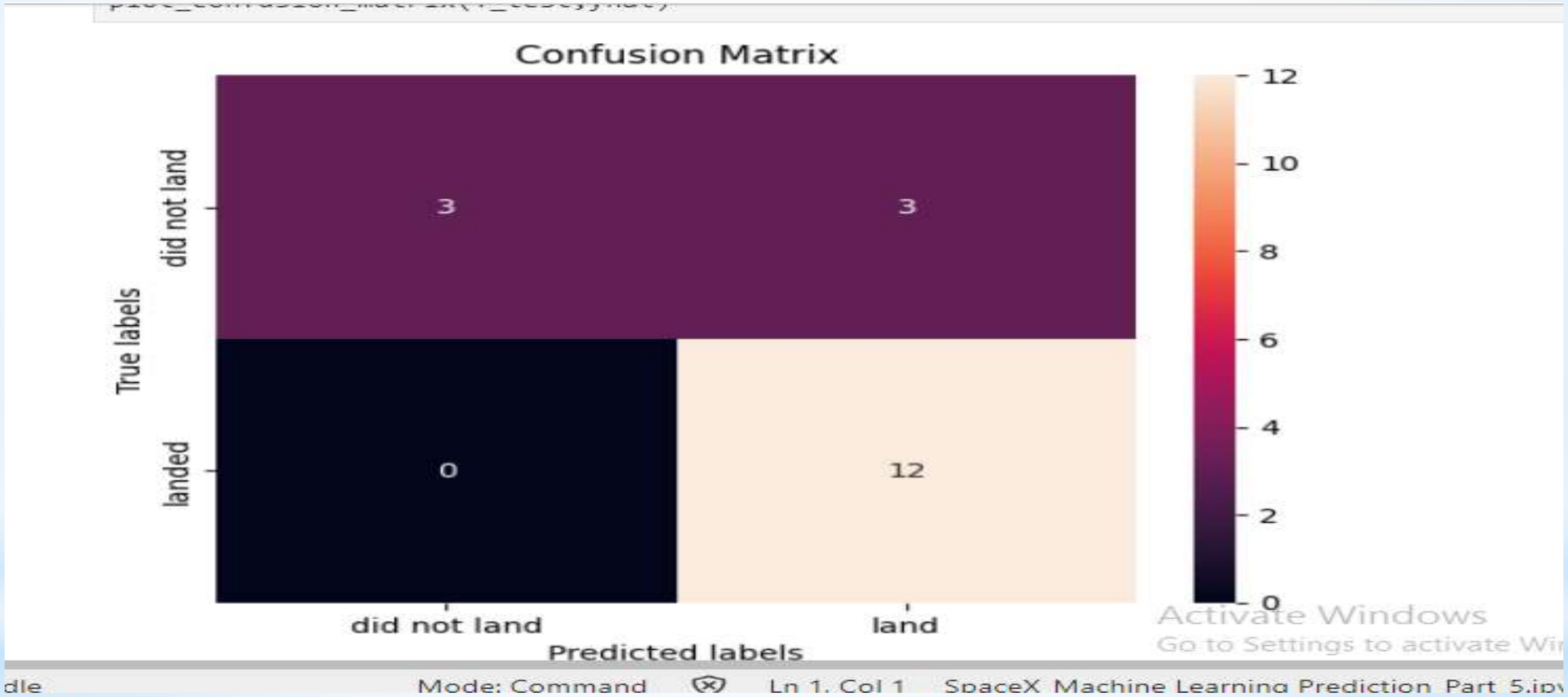


Decision tree classifier has the highest accuracy than knn classifier.



Confusion Matrix

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



* Logistic regression is the best performing model because it has highest accuracy 0.833333334 than other model.

Conclusions

* Point 1: In this project,at 1st stage we predict if the Falcon 9 first stage will land successfully. SpaceX savings is because it can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch.
* Point 2: in 2nd step we collect data on the Falcon 9 first-stage landings. We use a RESTful API and web scraping. we also convert the data into a dataframe and then perform data wrangling to clear the data.
* Point 3: we build a dashboard to analyze launch records interactively with Plotly Dash. We then build an interactive map to analyze the launch site proximity with Folium
* Point 4: then we,Split the data into training testing data,Train different classification models,Optimize the Hyperparameter grid search and Utilize your machine learning skills to build a predictive model to help a business function more efficiently.
* …

Appendix

- **PYTHON CODE SNIPPETS:**
- **Unzipping and Zipping Files by importing liabrary :import** zipfile
- **Using shutil library:**Often you don't want to extract or archive individual files from a .zip, but instead archive everything at once. The shutil library that is built in to python has easy to use commands for this:shutil.unpack_archive(output_filename,dir_for_extract_result,'zip')
- **Lambda Expressions, Map, and Filter:1.** The **map** function allows you to "map" a function to an iterable object. That is to say you can quickly call the same function to every item in an iterable, such as a list.**2. The filter function** returns an iterator yielding those items of iterable for which function(item) is true. Meaning you need to filter by a function that returns either True or False. Then passing that into filter (along with your iterable) and you will get back only the results that would return True when passed to the function.**3. Lambda Expressions** One of Pythons most useful (and for beginners, confusing) tools is the lambda expression. lambda expressions allow us to create "anonymous" functions. This basically means we can quickly make ad-hoc functions without needing to properly define a function using def.

Thank you!