IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

T.Venkata Manvitha
10-05-2024

Outline

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

Executive Summary

- In this course we will be applying our knowledge of data science and machine learning in real life scenario.
- We will analyze and visualize data using python.
- We will build and validate predictive learning model using python.
- Then we will create and share actionable insights to real life data problems.

Introduction

- In this project, we are going to determine various requirements in Capstone project.
- Few questions you will get answer from this project as follows:
- How do we collect data?
- How do we clean it?
- How the visualization is done to understand the data better?
- Which model is best suited to fit the data?
- Do SpaceX launches are all successful?

Section 1

Methodology

Methodology

Executive Summary

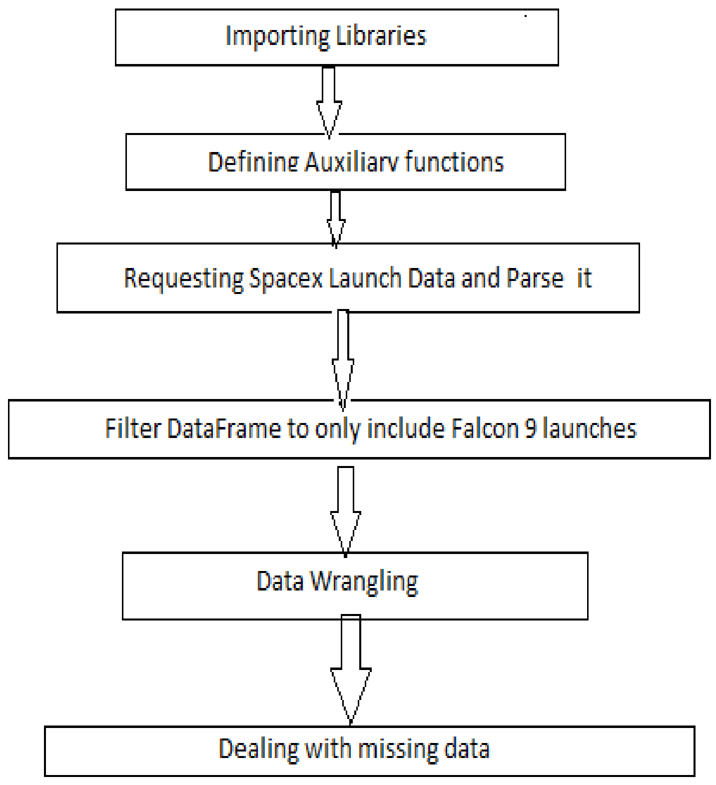
- Data collection methodology:
  - Data was collected using web scrapping and provided .csv and .xlsx files.
- Perform data wrangling
  - It was performed using different functions from pandas and numpy.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Different models like svm, knn, trees are used and analysed using confusion matrix.

Data Collection

- Data is collected using .csv and .xlsx files.
- We can access them using pandas.

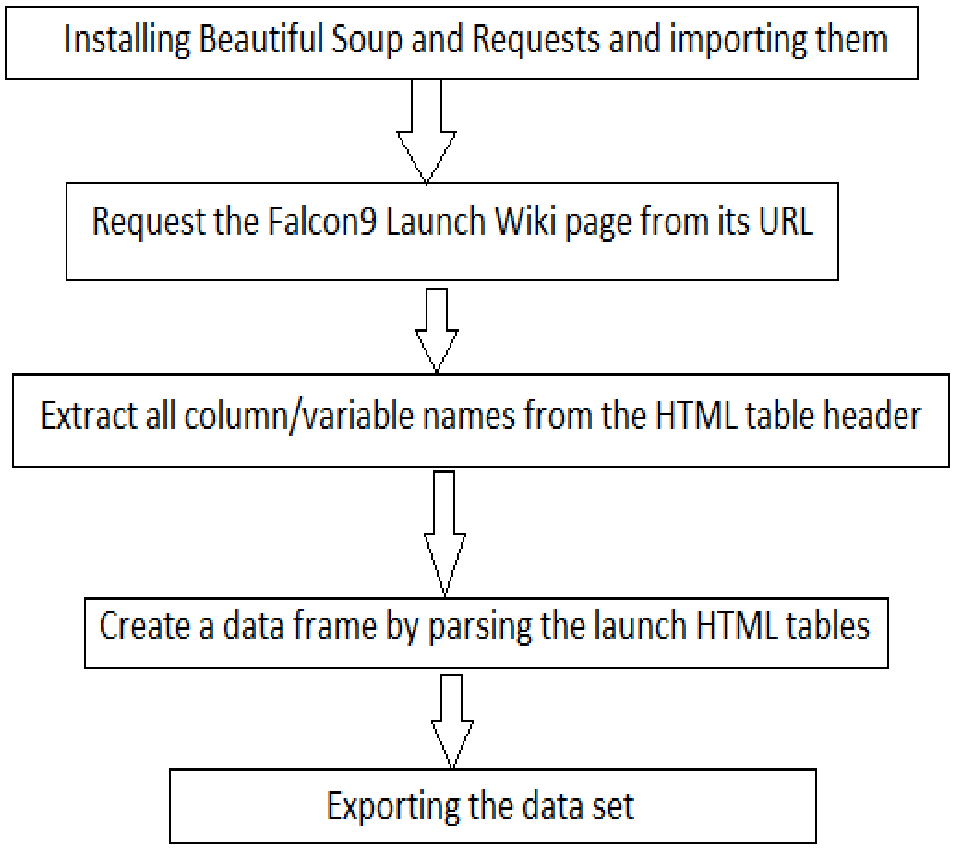
Data Collection – SpaceX API

- The collection is done through provided URL.
- https://github.com/Manvitha236/Peer/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



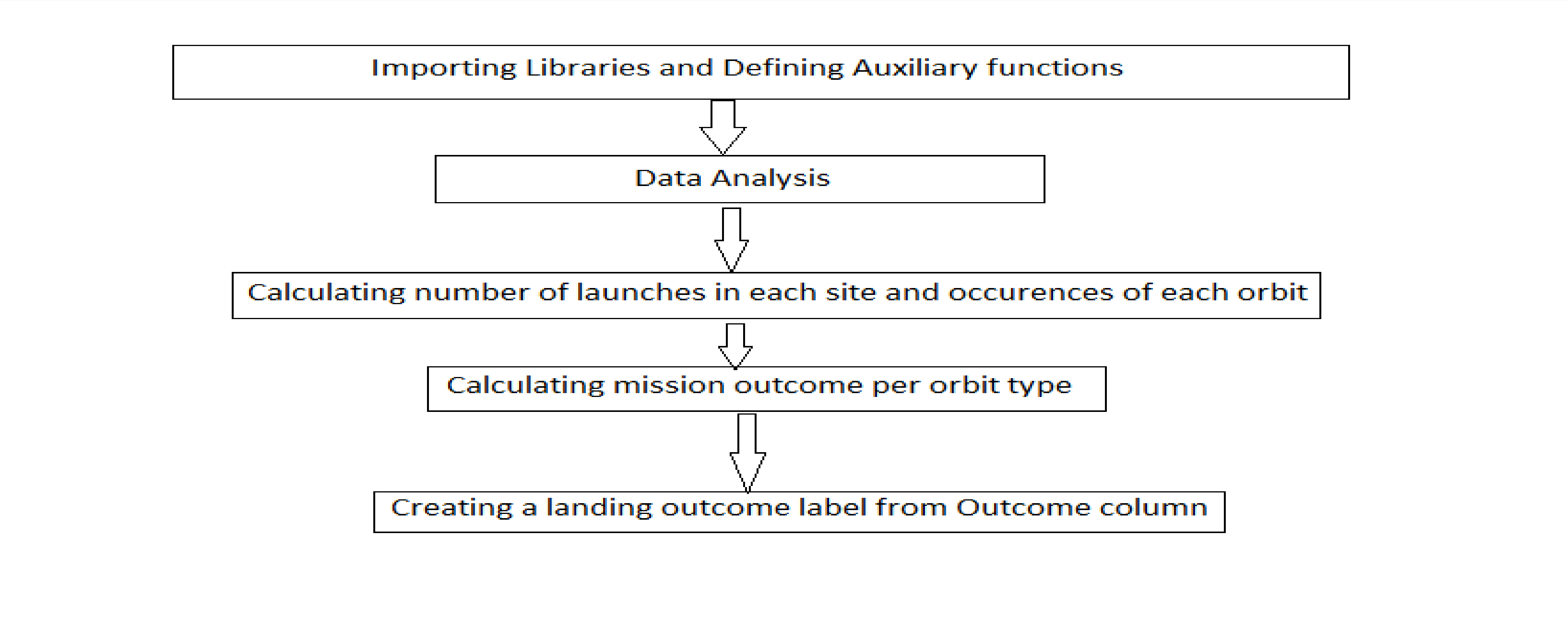
Data Collection - Scraping

- Web Scrapping is done by using BeautifulSoup and JSON.
- https://github.com/Manvitha236/Peer/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- https://github.com/Manvitha236/Peer/blob/main/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb



EDA with Data Visualization

- https://github.com/Manvitha236/Peer/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite%20(1).ipynb
- We have predicted if Falcon 9 first stage launched successfully or not by SpaceX.
- We have performed Exploratory Data Analysis and Feature Engineering here.
- Pandas and Matplotlib helped in achieving accurate prediction.

EDA with SQL

- https://github.com/Manvitha236/Peer/blob/main/jupyter-labs-eda-sql-edx_sqllite%20(1).ipynb
- We have created spacextbl table with data we have.
- We have gathered information about different landing outcomes we have and then performed few operations on payload mass.
- Then we have focused on success and failure of mission outcomes.
- Ranking of outcomes is done to analyze.
- Sqlite is used here.

Build an Interactive Map with Folium

- https://github.com/Manvitha236/Peer/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb
- We have marked all launch sites in a map.
- Then marked success and failures of each launch site in map with different colors.
- Distance from launch site to its proximities is calculated and marked.

Build a Dashboard with Plotly Dash

- We have performed four different tasks to a dashboard with plotly dash.
- Firstly, we have added a Launch Site Drop-down Input Component to select a specific launch site.
- Secondly, we have added a callback function to render success pie-chart based on selected site.
- Then, we have added a range slider to select payload.
- We have also added a callback function to render the success payload scatter chart scatter plot.

Predictive Analysis (Classification)

- https://github.com/Manvitha236/Peer/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(1).ipynb
- At first column creation for class is done through numpy array.
- Then we standardize the data.
- Data is got split into training and test data.
- Based on test data, the best performing method is determined among SVM, Classification trees, Logistic Regression.

Results

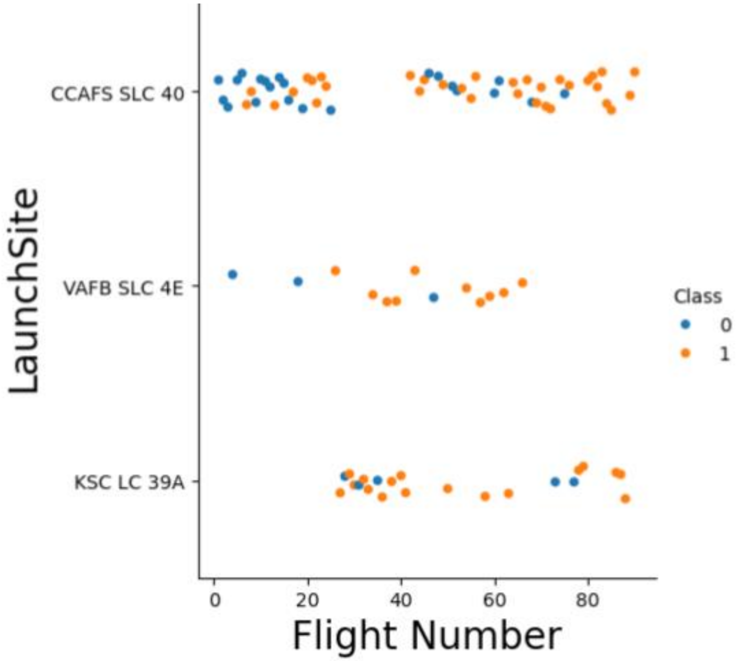
- Column class is created which determines the success and failure of landing outcomes by representing as 1 or 0.
- Collected data is obtained through json and web scrapping.
- Visually analyzed the data through folium and plotly dash.
- Best method is determined through predictive analysis.
- Decision tree classifier is determined to be the best.

Section 2

Insights drawn from EDA

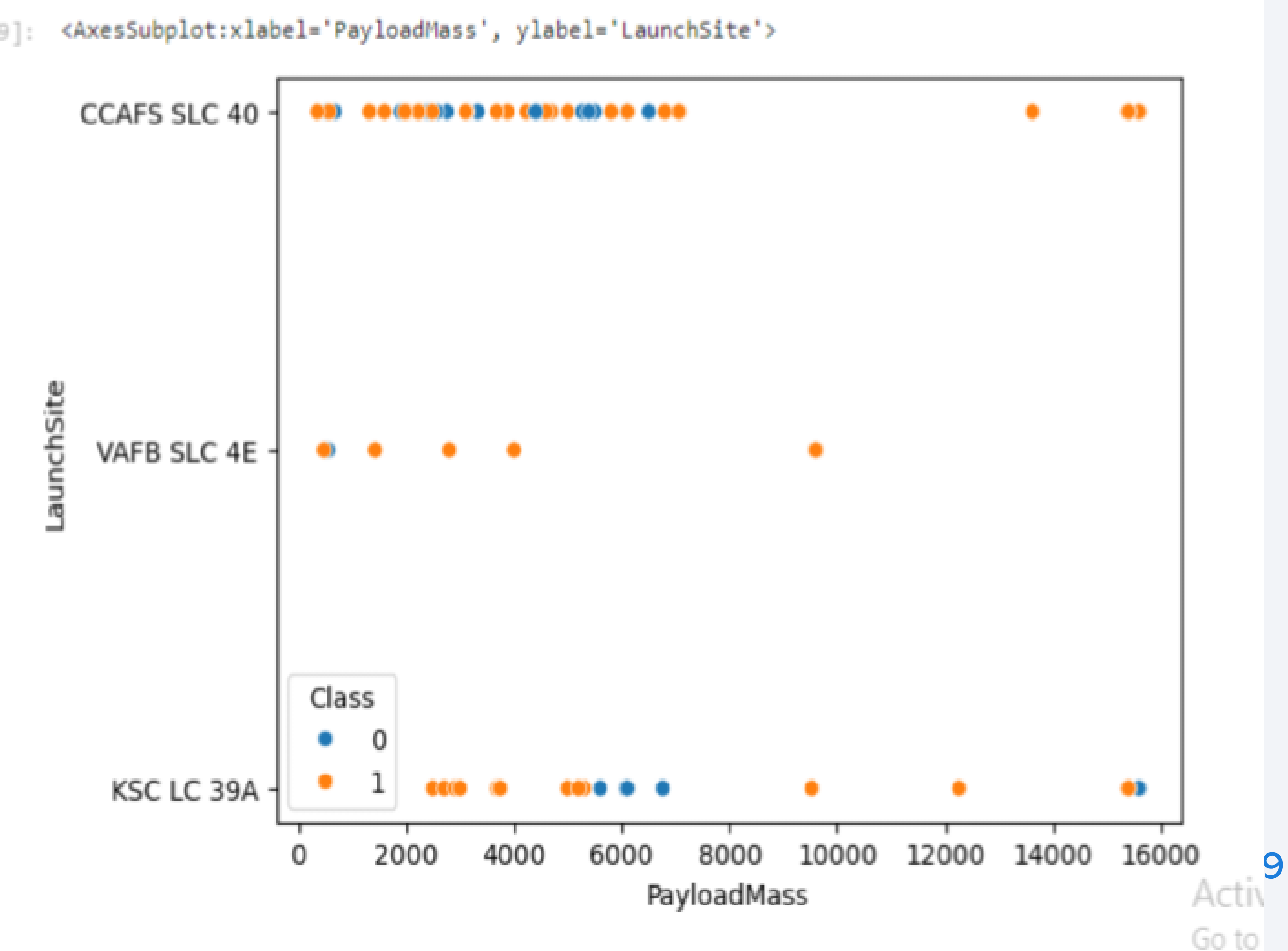
Flight Number vs. Launch Site

- Class 0 represents unsuccessful landing.
- Class 1 represents successful landing.
- 3 different launch sites along with their outcomes is provided in graph.



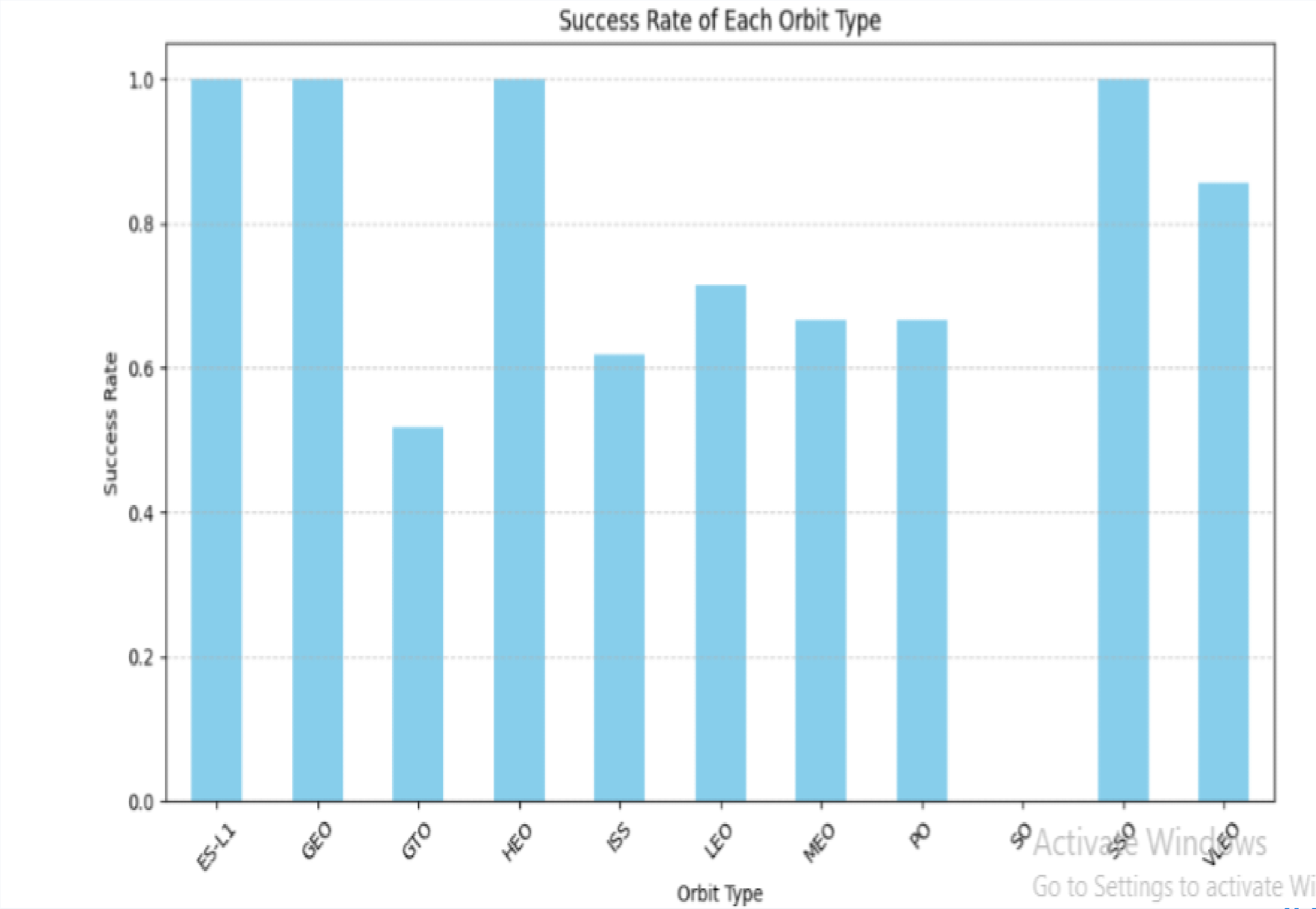
Payload vs. Launch Site

- From graph we can understand that for VAFB SLC 4E there are no rockets launched with more than 10000 mass.
- KSC LC 39A has a failure in attempting with high payload mass but CCAFS SLC 40 is very much successful in carrying high payload mass.



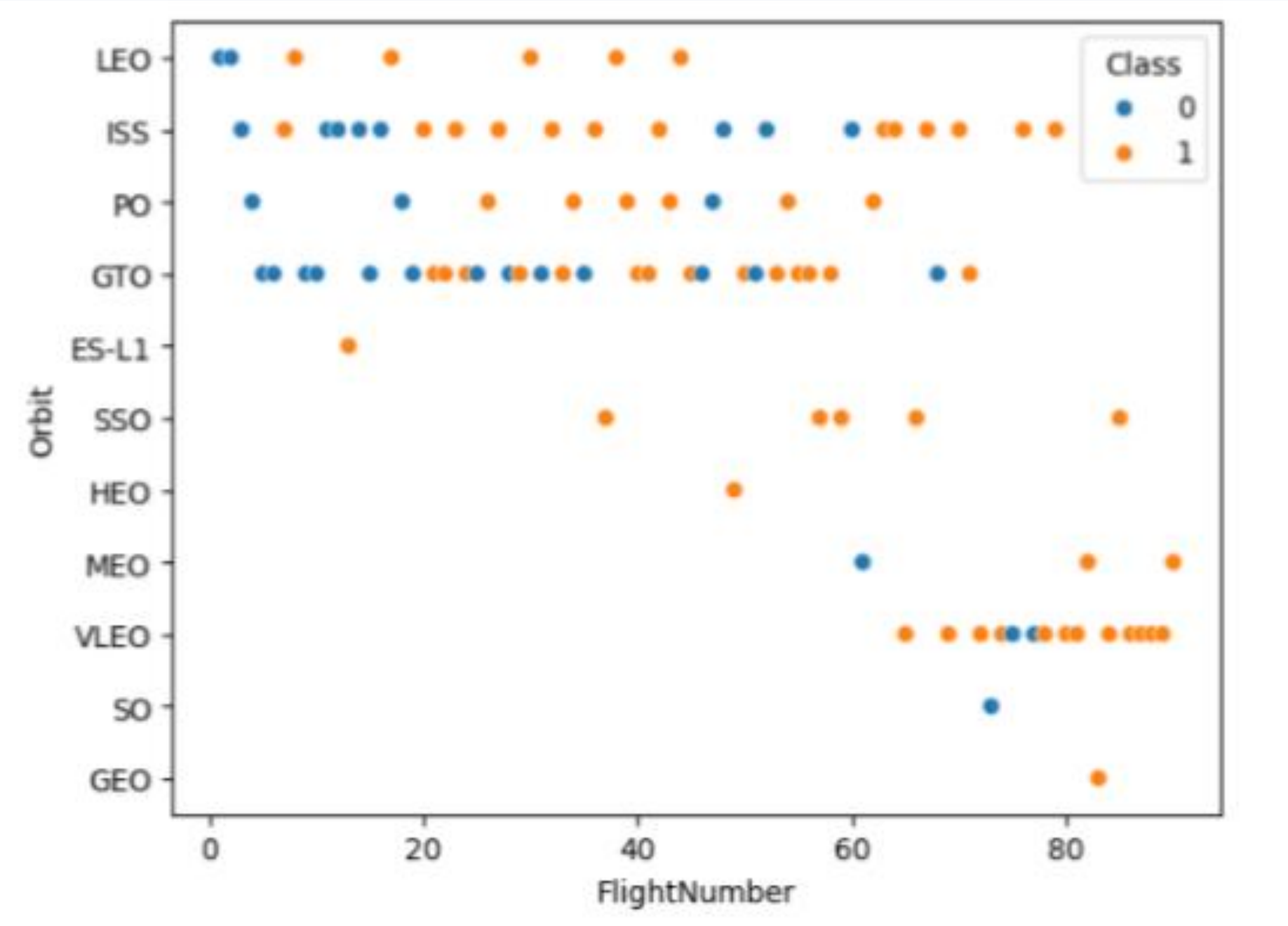
Success Rate vs. Orbit Type

- From graph, we can say which orbit type have high success rate.
- ES-L1, GEO, HEO and SSO has high success rate.



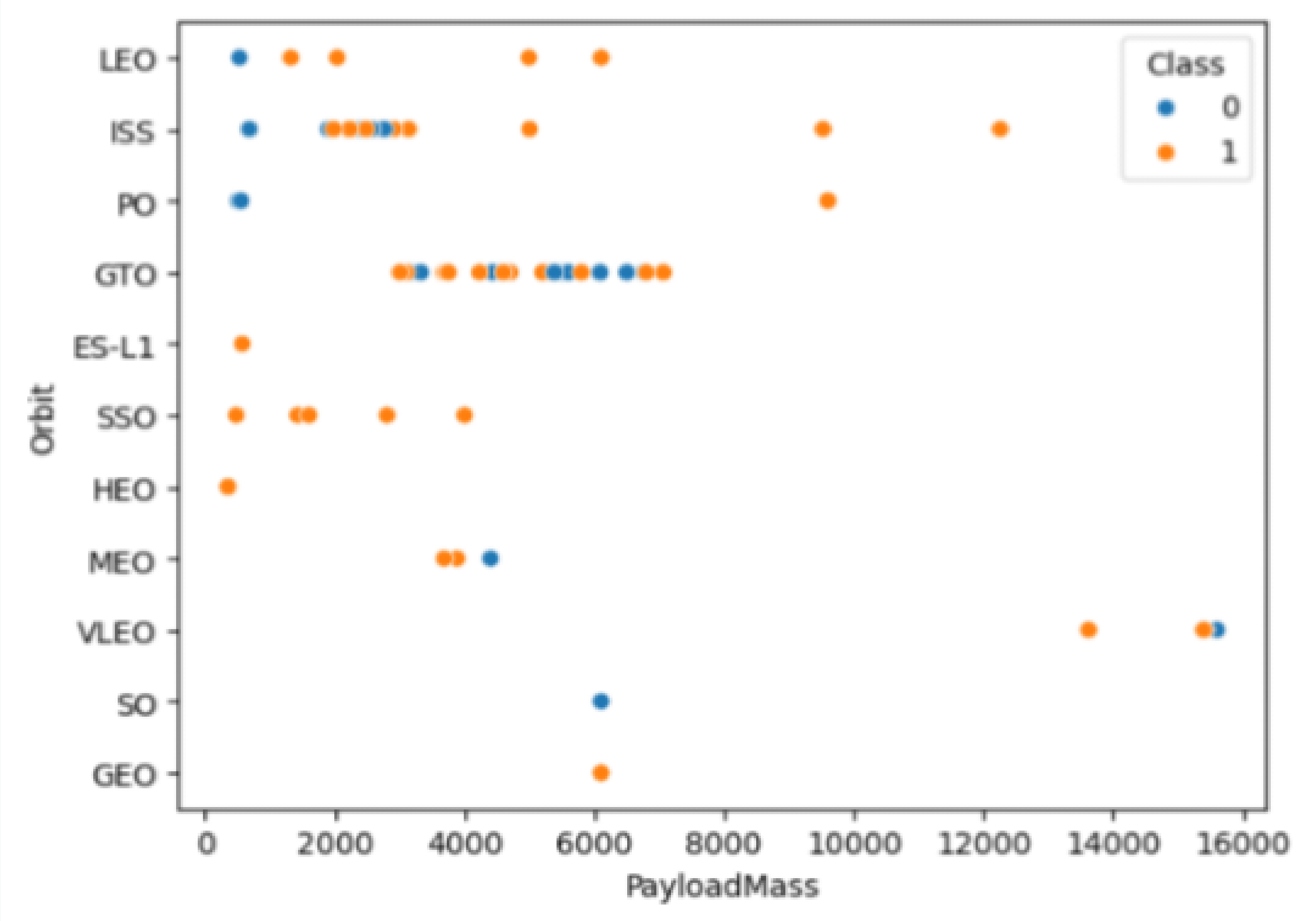
Flight Number vs. Orbit Type

- This plot is to find if there is any relation between flight number and orbit type.
- In LEO orbit, we can observe a relation between Orbit and Flight Number.
- But, if we see GTO orbit, there is no relationship between orbit and flight number.



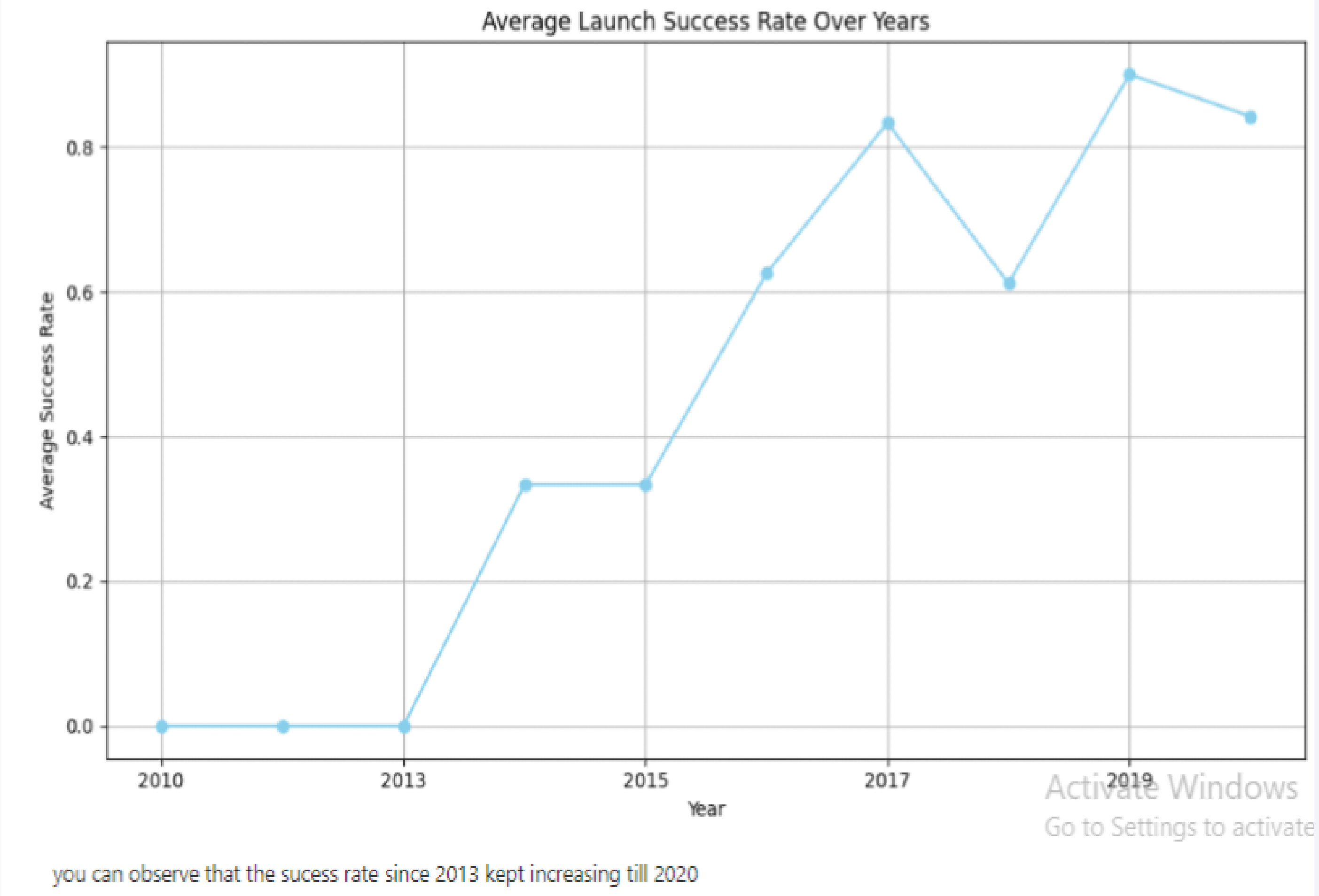
Payload vs. Orbit Type

- In PO Orbit, Our success rate is high when payload mass is more.
- This is same for ISS orbit too.
- But, in GTO we cannot actually specify particular relation between payload mass and orbit type.



Launch Success Yearly Trend

- In the graph we can observe the average success rate that achieved over few years of time.
- From 2013, success rate is higher.
- It kept increasing till 2020.



All Launch Site Names

- **SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL**
- This is query to find all the launch sites names.
- The output of the query gives 4 different launch site names.
- CCAFS LC-40
- VAFB SLC-4E
- KSC LC-39A
- CCAFS SLC-40
- Distinct is used to find non-duplicated values.

Launch Site Names Begin with 'KSC'

- **SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE "KSC%" LIMIT 5**
- The above query shows 5 records where launch site names begin with "KSC".
- Like is used to specify requirements in finding string.
- "%" is used to specify that there are few strings after the specified string.
- Limit is used to limit the number of records for printing.

Total Payload Mass

- **SELECT SUM(PAYLOAD_MASS_KG_) AS "TOTAL PAYLOAD MASS" FROM SPACEXTBL**
- The above query displays the total payload mass carried by boosters launched by NASA.
- Sum() is the aggregate function used in SQL to print the total sum of provided column.
- So to get the total payload mass we have to provide that column in the sum() function.

Average Payload Mass by F9 v1.1

- **SELECT AVG(PAYLOAD_MASS_KG_) AS "AVG PAYLOAD MASS" FROM SPACEXTBL WHERE "BOOSTER_VERSION" IS "F9 v1.1"**
- The above query provides the average payload mass carried by the booster version F9 v1.1
- Avg() is a function used to find the average of the column provided in it as parameter.
- As is used to provide the alias name.
- Where is used to provide any required condition.

First Successful Ground Landing Date

- **SELECT MIN(DATE) FROM SPCAEXTBL WHERE LANDING_OUTCOME IS "SUCCESS(DRONE SHIP)"**
- The above query provide first date where the landing outcome is successful on ground.
- Success(drone ship) is the success in landing on the drone ship.
- It specifies ground.
- So we have given it in where condition.

Successful Drone Ship Landing with Payload between 4000 and 6000

- **SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME IS 'SUCCESS(GROUND PAD)' AND PAYLOAD_MASS_KG_>4000 AND PAYLOAD_MASS_KG_<6000**
- The above query results list of names of booster versions which have success in ground pad and have payload mass greater than 4000 but less than 6000.
- The condition required, success in landing on ground pad is provided in where condition.
- Along with that required boundary conditions of payload mass is also given.

Total Number of Successful and Failure Mission Outcomes

- **SELECT MISSION_OUTCOME, COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME IN('SUCCESS','FAILURE') GROUP BY MISSION_OUTCOME**
- The above query results in total number of success and failure mission outcomes.
- The output is like success followed by 98, which means there are 98 successful mission outcomes.
- There is no output regarding failure which means there are no failures at all.

Boosters Carried Maximum Payload

- **SELECT BOOSTER_VERSION FROM SPACEXTBL PAYLOAD_MASS_KG_ IS (SELECT MAX(PAYLOAD_MASS_KG_) FROM SPACEXTBL)**
- The above query results in listing all the names of the booster versions which have carried the maximum payload mass.
- We have used a subquery to attain this.
- In subquery we have selected maximum payload mass.
- In main query we have booster versions which have carried maximum payload mass.

2017 Launch Records

- SELECT CASE WHEN SUBSTR(DATE,6,2)='01' THEN 'JANUARY' WHEN SUBSTR(DATE,6,2)='02' THEN 'FEBRUARY' WHEN SUBSTR(DATE,6,2)='03' THEN 'MARCH' WHEN SUBSTR(DATE,6,2)='04' THEN 'APRIL' WHEN SUBSTR(DATE,6,2)='05' THEN 'MAY' WHEN SUBSTR(DATE,6,2)='06' THEN 'JUNE' WHEN SUBSTR(DATE,6,2)='07' THEN 'JULY' WHEN SUBSTR(DATE,6,2)='08' THEN 'AUGUST' WHEN SUBSTR(DATE,6,2)='09' THEN'SEPTEMBER' WHEN SUBSTR(DATE,6,2)='10'THEN'OCTOBER' WHEN SUBSTR(DATE,6,2)='11' THEN 'NOVEMBER' WHEN SUBSTR(DATE,6,2)='12' THEN 'DECEMBER' END AS MONTH_NAME, LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING_OUTCOME IS 'SUCCESS(GROUND PAD)' AND SUBSTR(DATE,0,5)='2017'
- The above query gives month names along with successful landing outcomes in ground pad along with booster versions and launch sites in 2017.

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

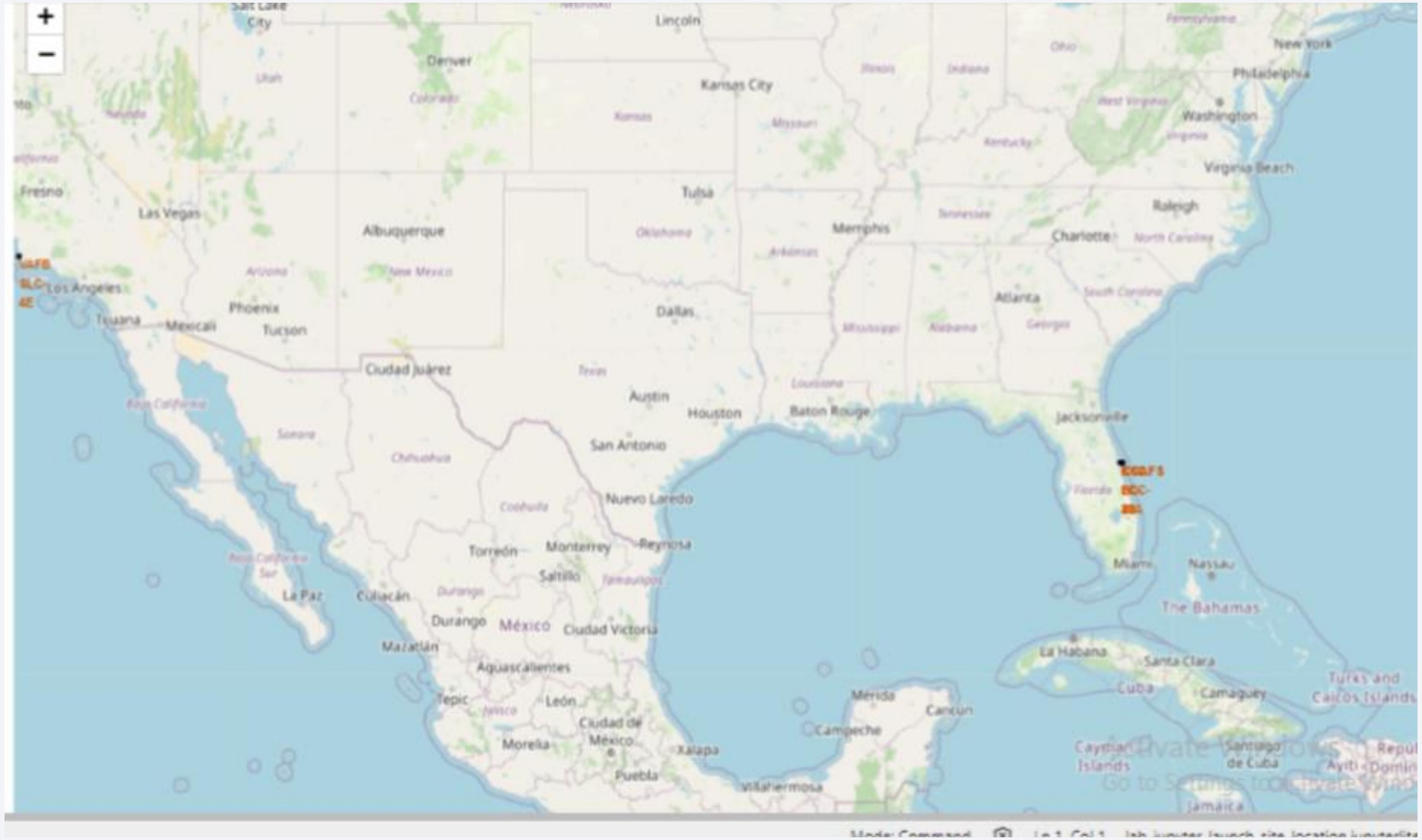
- **SELECT LANDING_OUTCOME, COUNT(*) FROM SPACEXTBL WHERE DATE>='2010-06-04' AND DATE<='2017-03-20' AND LANDING_OUTCOME IN ('FAILURE (PARACHUTE)','NO ATTEMPT','UNCONTROLLED (OCEAN)','CONTROLLED (OCEAN)','FAILURE (DRONE SHIP)','PRECLUDED (DRONE SHIP)','SUCCESS (GROUND PAD)','SUCCESS (DRONE SHIP)','SUCCESS','FAILURE','NO ATTEMPT') GROUP BY LANDING_OUTCOME ORDER BY COUNT(*) DESC**
- The above query is used to rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order.
- Count() function is used to count the records in the column given in it as parameter.
- Order by is used in placing given data rank wise.

Section 3

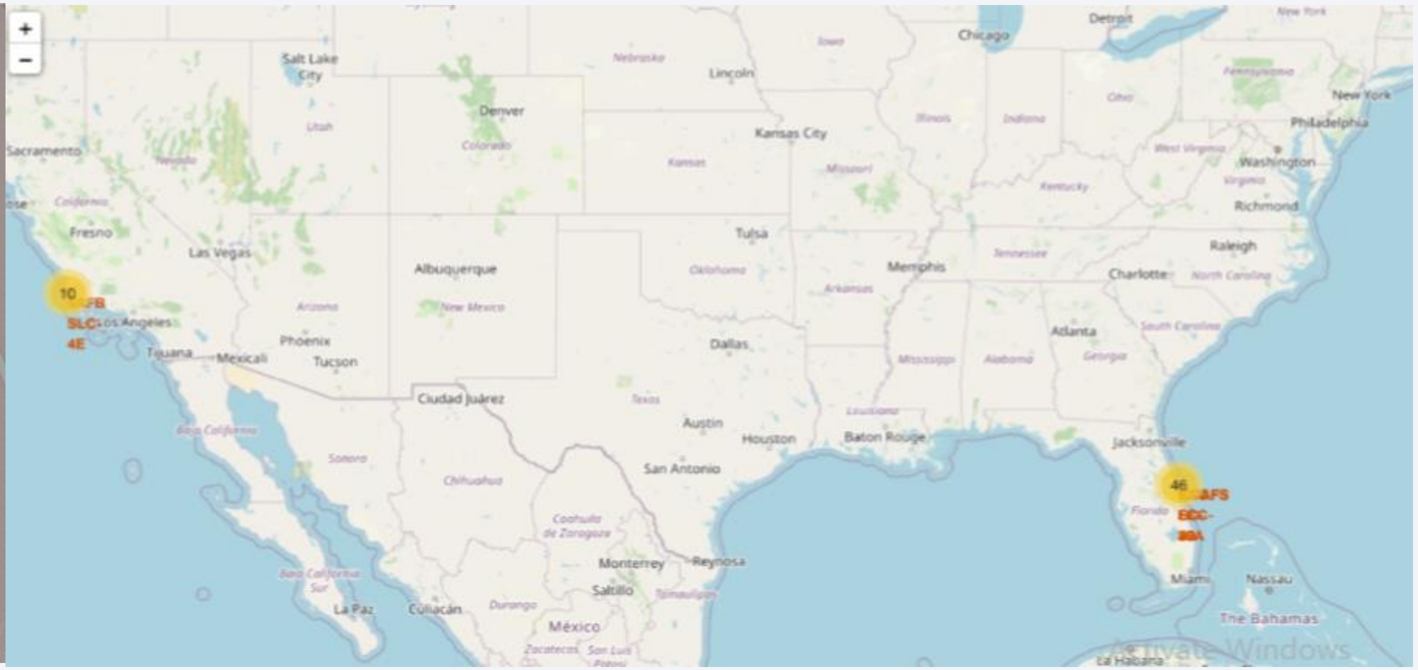
Launch Sites Proximities Analysis

Launch Sites on map

The map besides shows the launch sites.
We have 4 different launch sites.
In the map VAFB SLC-4E is in west side.
Remaining three launch sites CCAFS LC-40, CCAFS SLC-40 and KSC LC-39A are on the eastern side of the map.



Markers for all launch records



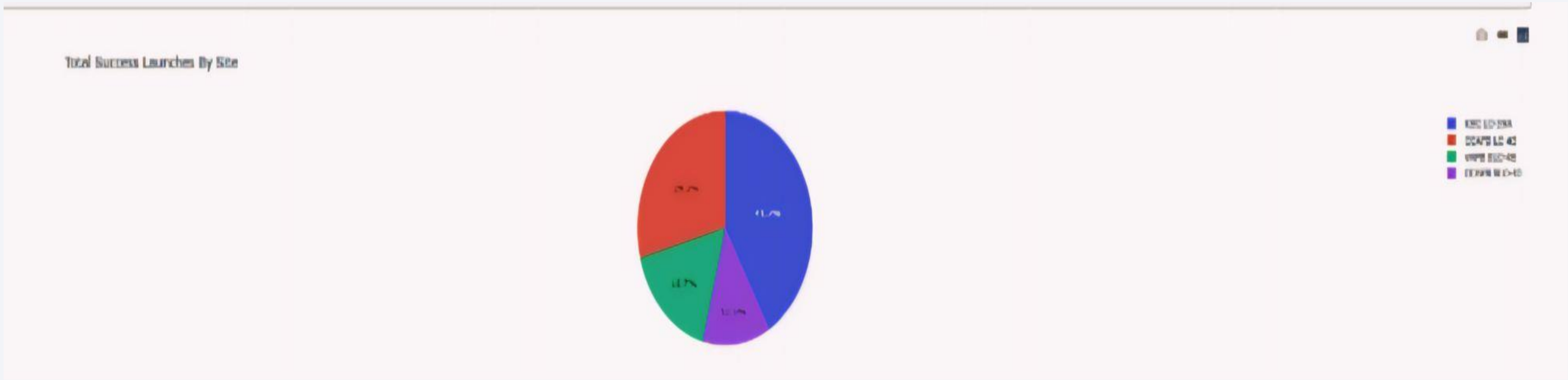
The above screenshots shows colors at each launch site based on the success or failure of the mission.

Launch site to it's proximities

Section 4

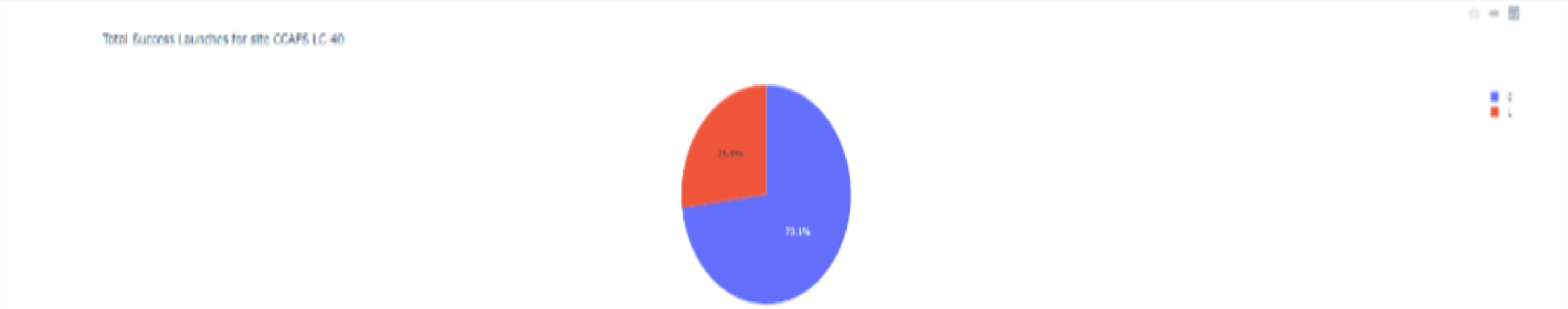
Build a Dashboard with Plotly Dash

Launch Success count of all sites



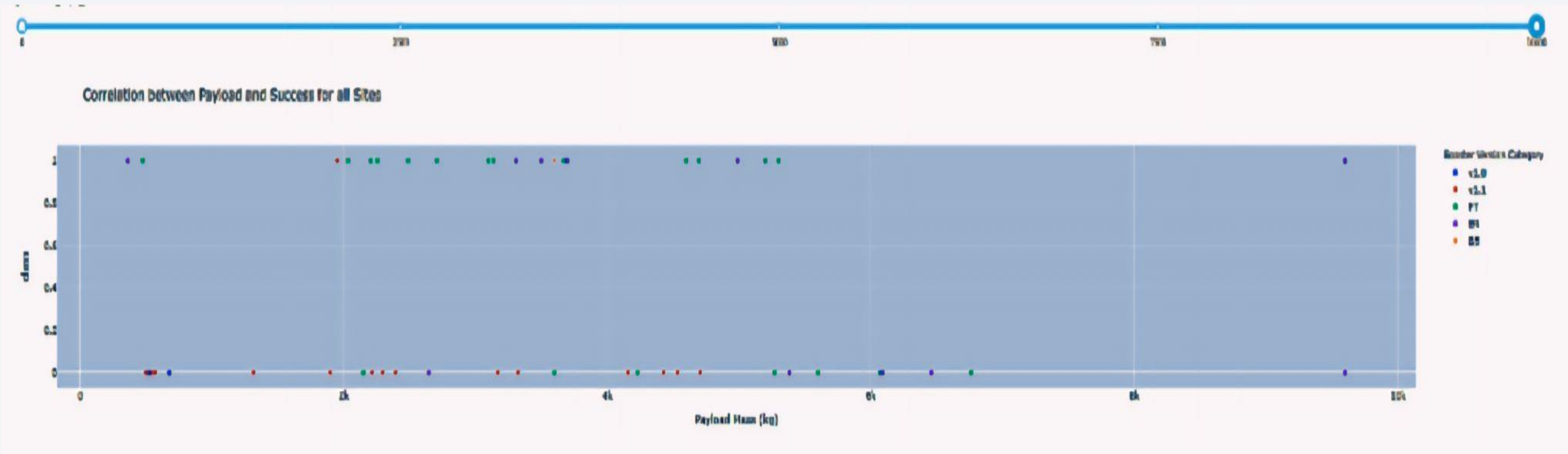
From the figure we can say that KSC LC-39A have the most success than the other launch sites.

Pie chart of a launch site



The above pie chart shows one launch site specifically. Here red color represents 0 which means failure and 1 represented by blue which means success.

Correlation between payload and success for all sites



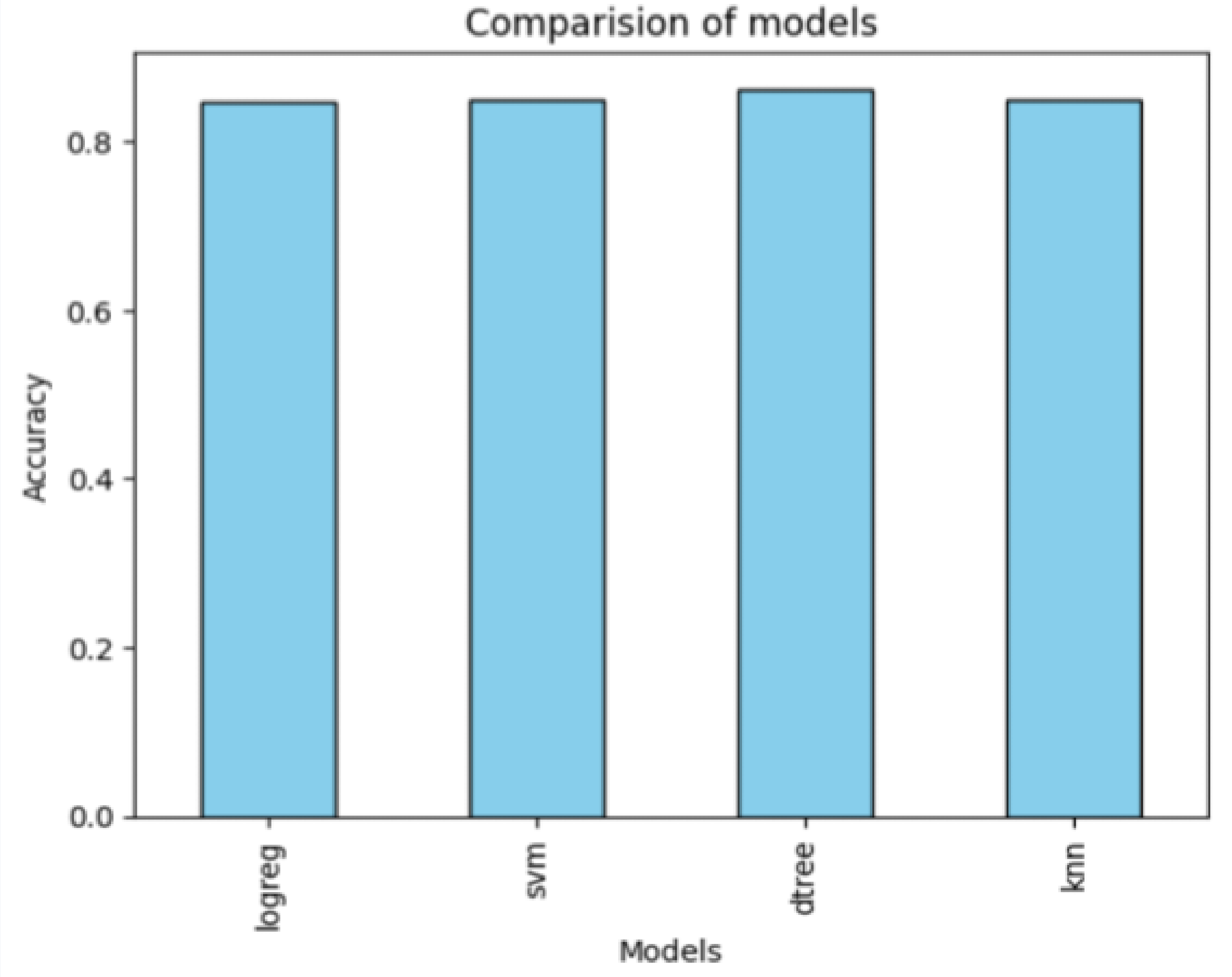
The above slider used to get different ranges of payload mass. According to that the scatter plot changes.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

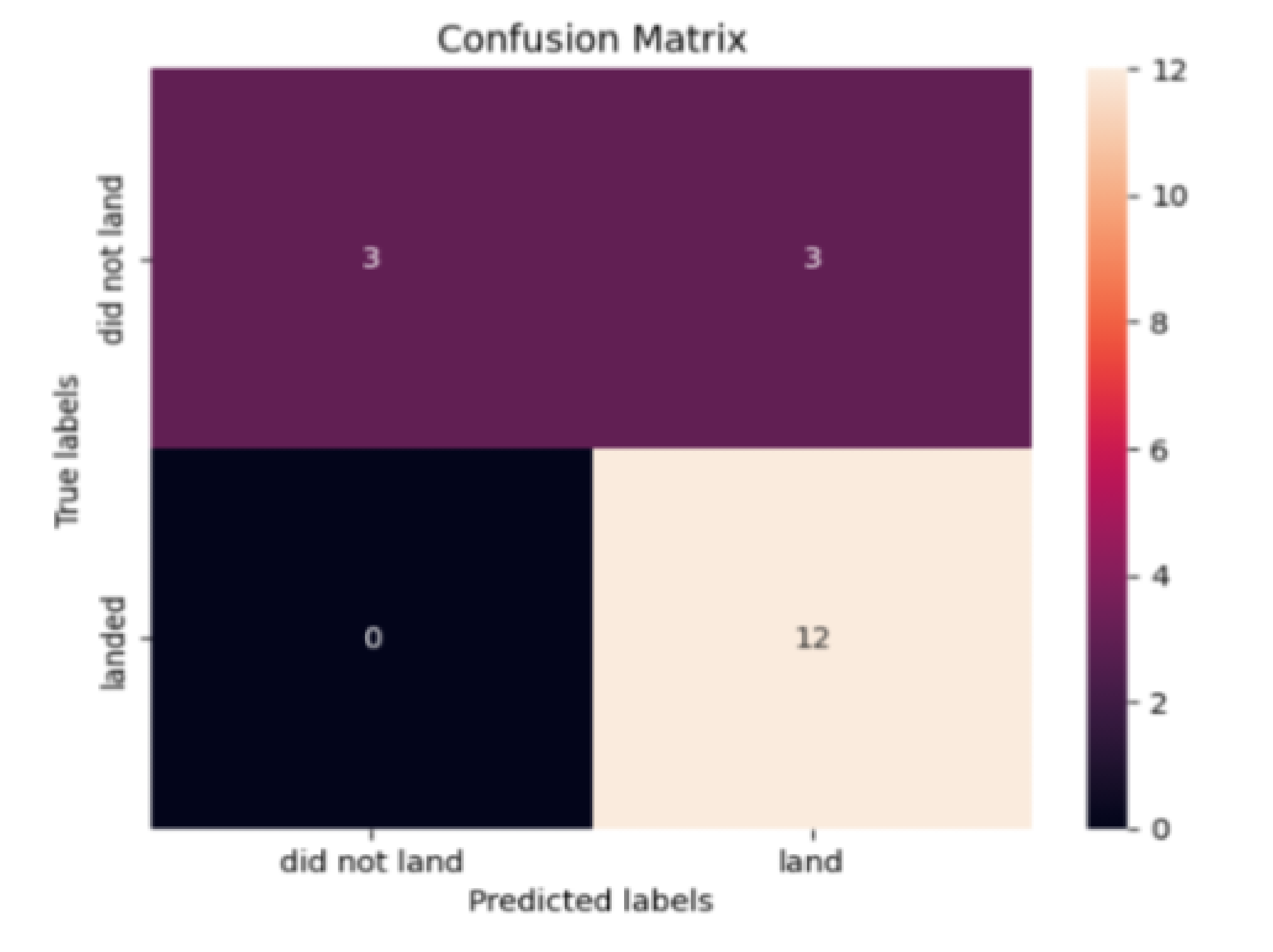
- The bar graph shows accuracy of different models.
- We can see that highest accuracy is in decision tree classifier model.
- Therefore decision tree classifier model is the best model.



Confusion Matrix

This is the confusion matrix of best performing model,

i.e., decision tree classifier.



Conclusions

- We have concluded that logistic regression model have accuracy of 0.8464285714285713.
- Grid search cv object has accuracy of 0.8482142857142856
- Decision tree classifier has accuracy of 0.8625
- Knn model has accuracy of 0.8482142857142858.
- Hence, we can conclude that Decision Tree Classifier is best performing model.

Appendix

- Below is the python code snippet that we have used to draw the bar graph of different models and their accuracies.

```
import matplotlib.pyplot as plt
import pandas as pd
data={'model':['logreg','svm','dtree','knn'],'accuracy':[0.846428,0.848214,0.8625,0.848214]}
df=pd.DataFrame(data)
ax=df.plot(kind="bar",x="model",y="accuracy",color='skyblue',edgecolor='black',legend=False)
plt.title("Comparision of models")
plt.xlabel("Models")
plt.ylabel("Accuracy")
plt.show()
```

Thank you!