IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mohankrishna Gallavali
06-16-2023

Outline

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

Executive Summary

Summary of methodologies:

In this project different classification algorithms were used to predict the launch success or failure taking the appropriate features into consideration using feature engineering techniques. KNN, Decision Tree, SVM, Logistic Regression these are used in this project.

Summary of all results:

For all above mentioned algorithms accuracy of the model were calculated and compared with others. Also, confusion matrix were drawn for each model and observed.

Out of all models Logistic Regression gives the best accuracy around 94% whereas remaining all gives

83%. May be there are some models who can get more accuracy. I will encourage to explore more.

Introduction

In this project I have used SpaceX data to analyze whether the SpaceX will reuse its first stage of launch. I have gathered data from SpaceX REST API. Also, I have used web scraping technique to retrieve the data from the Wikipedia webpage. All these collected data was cleaned by using some pre-processing techniques. After that EDA operations were performed for the feature engineering and Finally, different machine learning models were built, trained and tested and their respective accuracies were compared.

Problems I want to find answers:

Whether the launch was Success or Failure

Is any parameter influence the launch success i.e., Payload Mass, Launch Site etc.

Will SpaceX reuse its first stage or not

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

- Data was collected from different sources using REST API's and Data Scraping from web page

Perform data wrangling

- I have pre-processed the data. Impute the missing values with mean. Creating new columns based on conditions etc. The data gets cleaned for the better visualization

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

Perform predictive analysis using classification models

- KNN, SVM, Logistic Regression, Decision Tree are used in this project.

Data Collection

Data were collected from SpaceX REST API's and convert them into datasets using get requests and responses. The responses are in JSON format, so I transformed JSON results to Data Frames. Another technique is used for data collection is using Data Scraping from the web page. I have collected data from the Wikipedia by using soup object and extract the tabular data. Later it is converted to data set. In the next slide you will find the process flow for my data collecting methodology.

Data Collection API Flow

SpaceX REST API

Request

Response

JSON FILE

{ JSON }

Data Set

CSV

Data Collection – SpaceX API

Previous slide describes the workflow of data collection from SpaceX REST API.

The API used in this project for data collection is

https://api.spacexdata.com/v4

GitHub URL of the completed SpaceX API calls notebook

https://github.com/PynetDev/CapstoneProject/blob/master/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

From Wikipedia using web scraping technique. I have extracted data converts to dataset

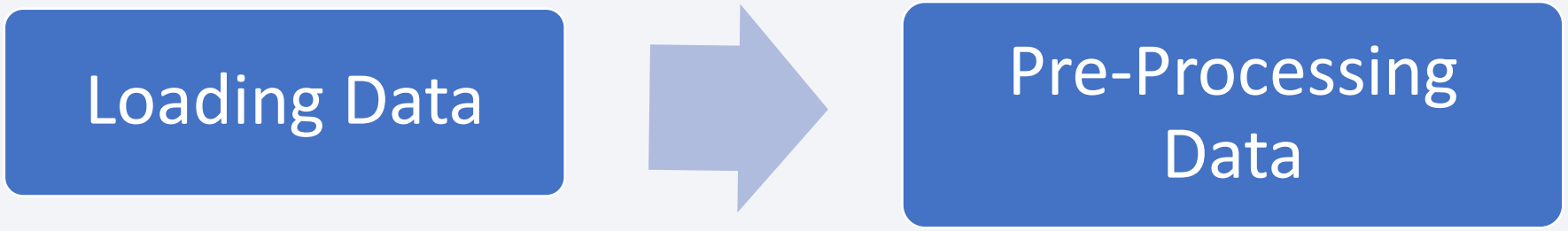
https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

GitHub URL of the completed web scraping notebook:

https://github.com/PynetDev/CapstoneProject/blob/master/jupyter-labs-webscraping.ipynb

Data Wrangling

After successful loading of data from different sources. I have used some preprocessed techniques to clean the data and fill the missing values. Also, created a new column in my dataset named **"CLASS" [Target Variable]** which has 0 -Failure and 1- Success



GitHub URL of your completed data wrangling related notebooks:

https://github.com/PynetDev/CapstoneProject/blob/master/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

After cleaning the data, the next stage is data visualization. It is important to know the relation between features whether they related linear or non-linear to each other. I have visualized some of the popular plots Scatter plot, Cat plot, Bar graph, Line graph etc. to know which features are more important and how they are related to target variable **"Class"**. Some of the following Observations were drawn

1. Different Launch Site have different success rate
2. With heavy payloads the successful landing or positive landing rate are more for orbits (Polar, LEO, LSS)
3. SpaceX Success Rate has been increasing since 2013

GitHub URL: https://github.com/PynetDev/CapstoneProject/blob/master/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

Following are the SQL queries implemented for EDA on SPACEXTBL (Table contains SpaceX data)

- Select Queries for Launch Site, Date for first successful launch on ground
- Total number of Successful and Failure missions
- Booster Versions carried maximum payload mass
- Count of Successful landing outcomes between specified dates

GitHub URL –

https://github.com/PynetDev/CapstoneProject/blob/master/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

I have used Folium Maps for the launch sites success rate visualization and some additional observations. Below are the folium objects I have used to achieve my requirements.

folium.Map() – Initialize the map with center

folium.Circle() – For circle marking on map

folium.Marker() – To mark the launch site locations using latitude and longitude

MarkerCluster() – To make clusters around launch site markers with red and green colors

MousePosition() – Gives corresponding latitude and longitude on the map

folium.Polyline() – Gives the distance line between launch site and desired location

GitHub URL Folium map:

https://github.com/PynetDev/CapstoneProject/blob/master/lab_jupyter_launch_site_location.jupyterlite%20(1).ipynb

Build a Dashboard with Plotly Dash

Plotly Dashboards are used for interactive visualization purpose. These are most user friendly. In this project pie charts, scatter plots were visualized. Html components like drop down lists and sliders were used to achieve the interactive plots. You should be able to use it to analyze SpaceX launch data, and answer the following questions:

1. Which site has the largest successful launches?
2. Which site has the highest launch success rate?
3. Which payload range(s) has the highest launch success rate?
4. Which payload range(s) has the lowest launch success rate?
5. Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest
6. launch success rate?

GitHub URL

https://github.com/PynetDev/CapstoneProject/blob/master/spacex_dash_app.py

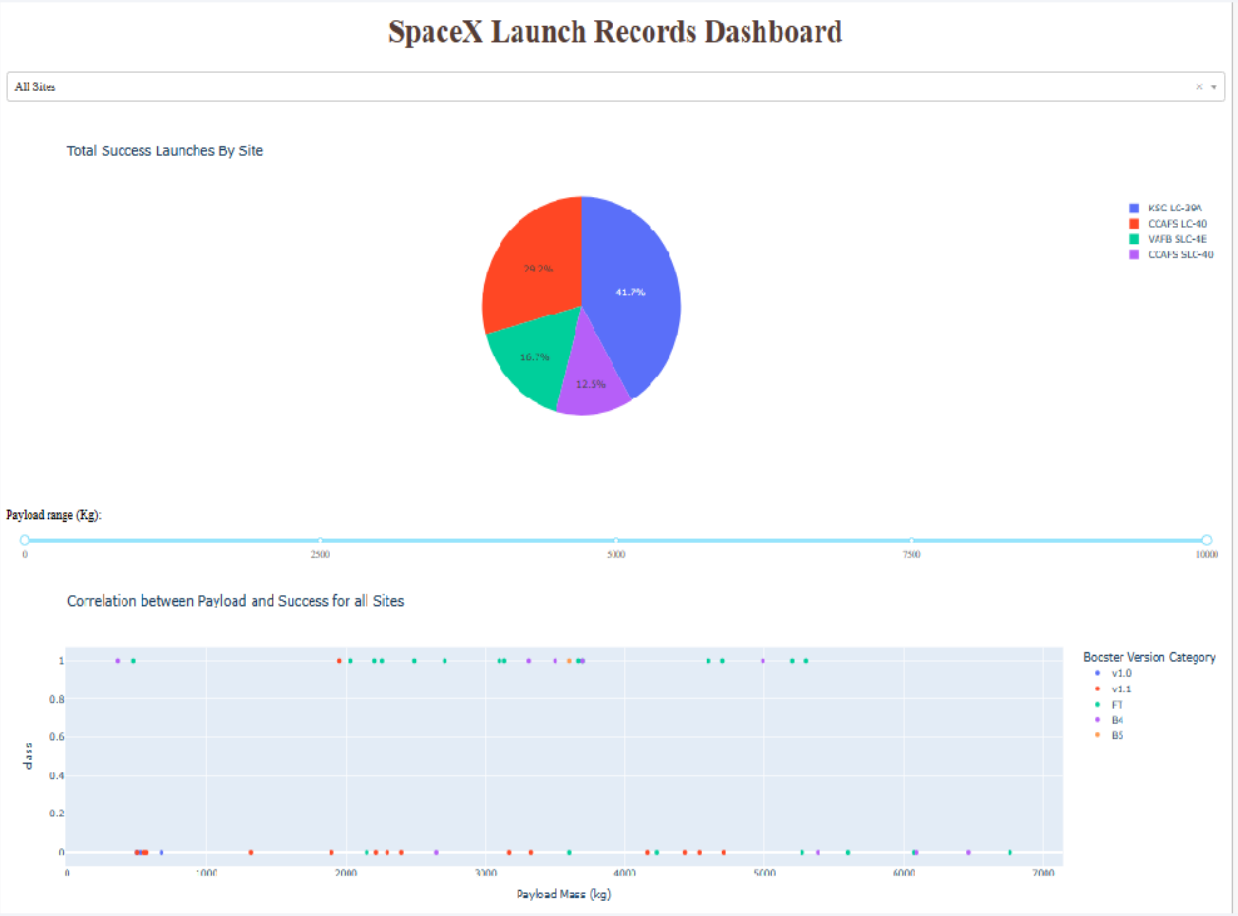
Predictive Analysis (Classification)

I have used different classification models for this SpaceX project and compared metric scores of each model and find the best model based up on the accuracy score. Also, In addition I have observed confusion matrix for each model.

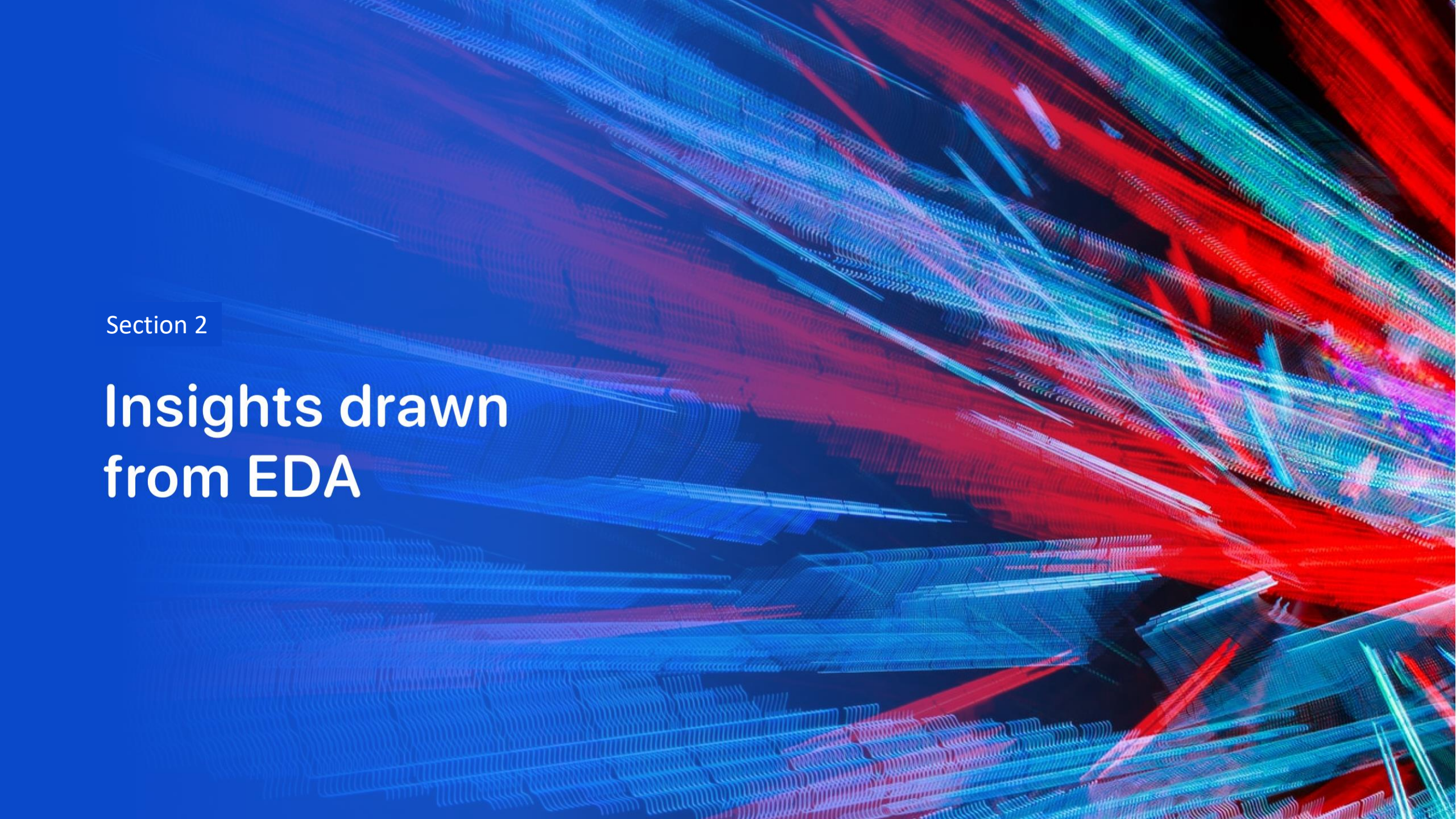
GitHub URL

https://github.com/PynetDev/CapstoneProject/blob/master/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

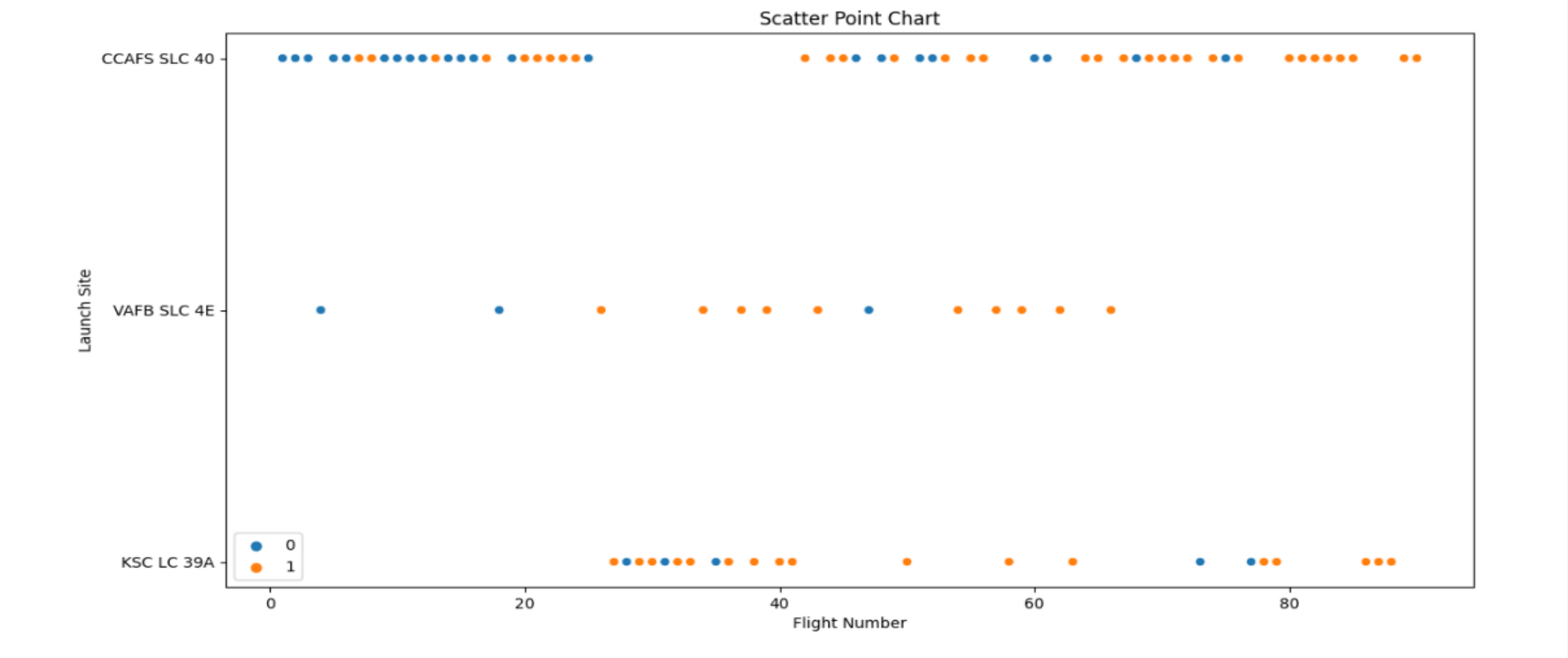
Results



- EDA is used for feature engineering where the important features are listed as - flight number, orbit type, launch site, payload mass, booster version these factors shows strong relation to the target variable(launch outcome).
- Interactive Plots summarize that KSLC – 39A has highest success rate among other launch sites. FT and B4 booster versions uses high payload mass only.
- In Predictive Analysis classification models like KNN, Logistic Regression, SVM, Decision Tree were used, and their accuracy scores were considered as measures for best fit model. Logistic Regression with 94%,Remaining with 83% accuracy was observed.

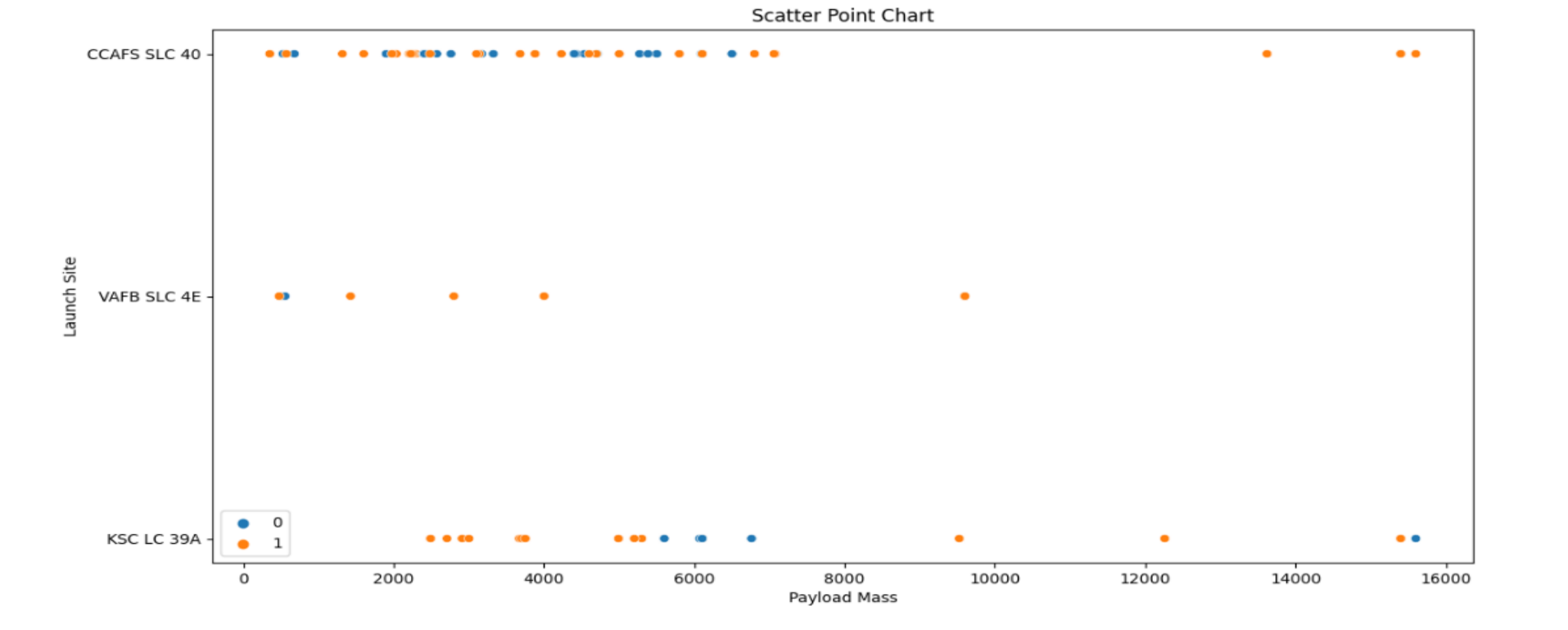

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



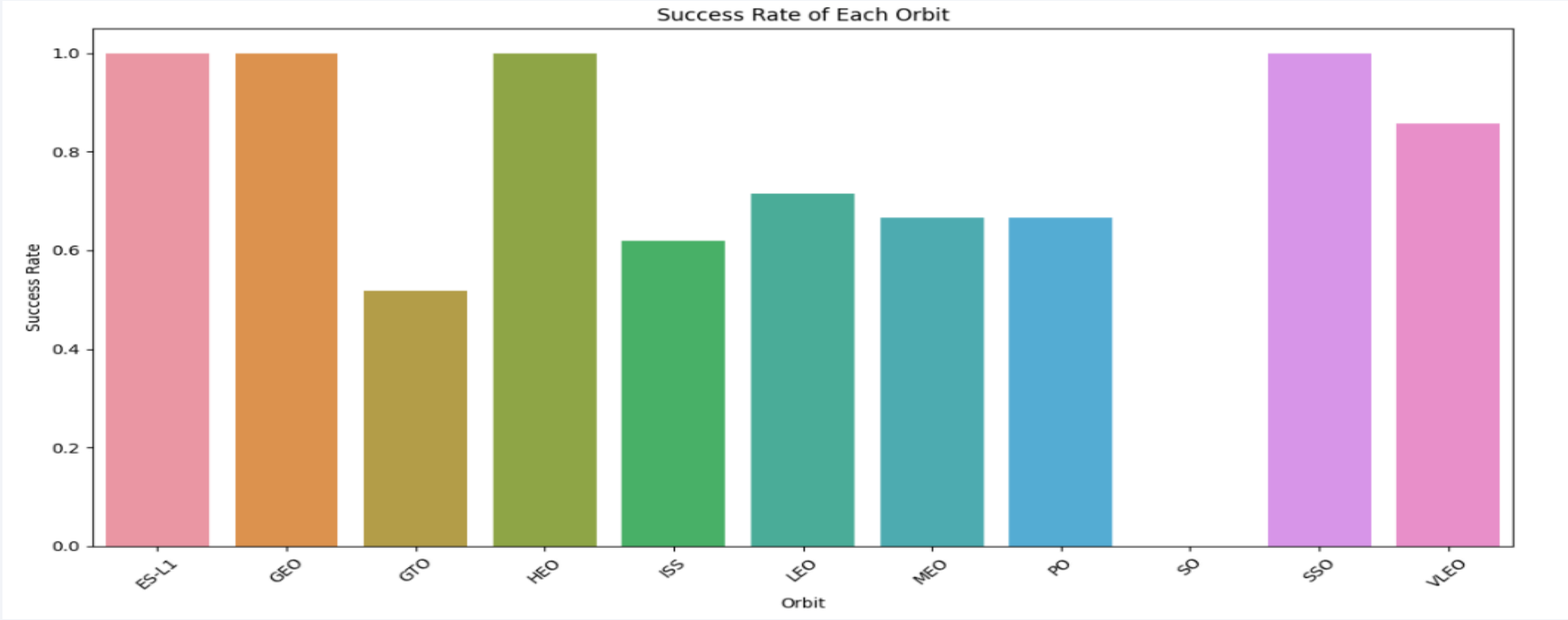
For Launch site VAFB SLC 4E and KSC LC 39A have some patterns high flight number high chances of success. In case of CCAFS SLC 40 the assumption becomes slightly weaker.

Payload vs. Launch Site



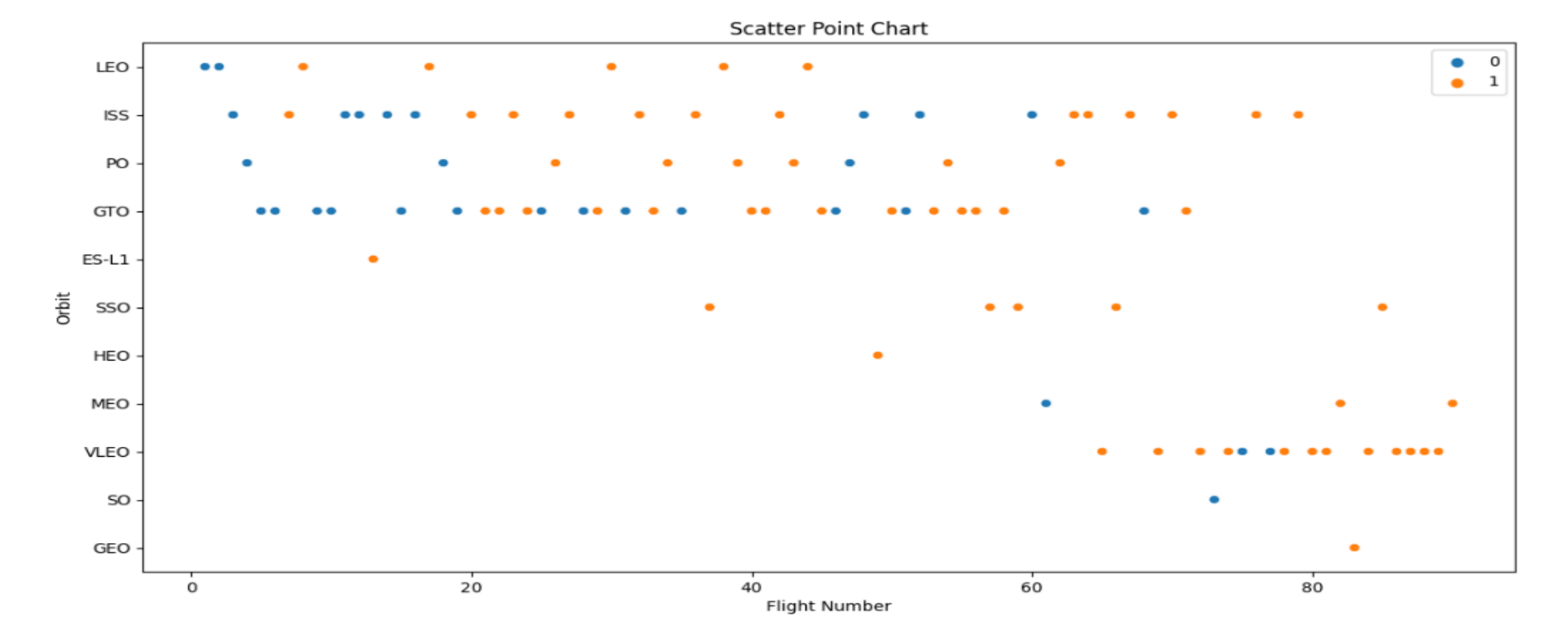
From the above graph we can draw the insight that irrespective of the launch site flights with high payload mass are successful. Payload mass greater than 8000 having only one failure

Success Rate vs. Orbit Type



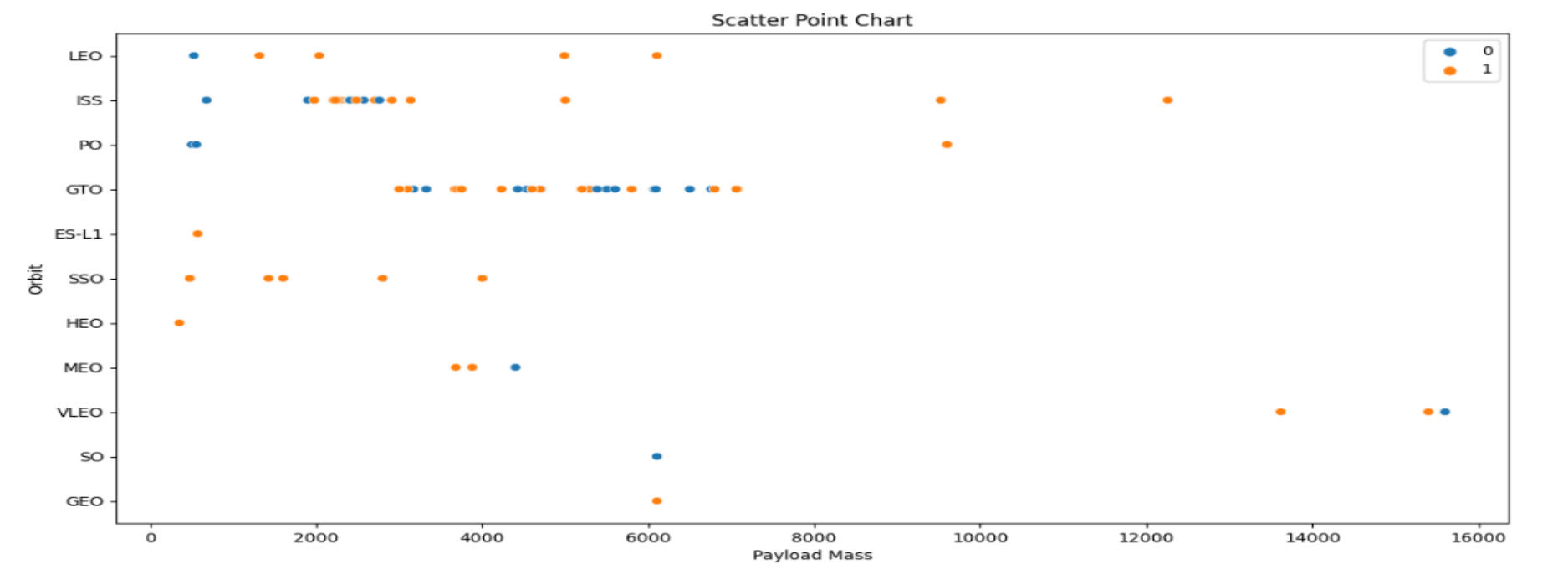
ES-L1, GEO, HEO, SSO Orbits Having High Success Rate Than Others

Flight Number vs. Orbit Type



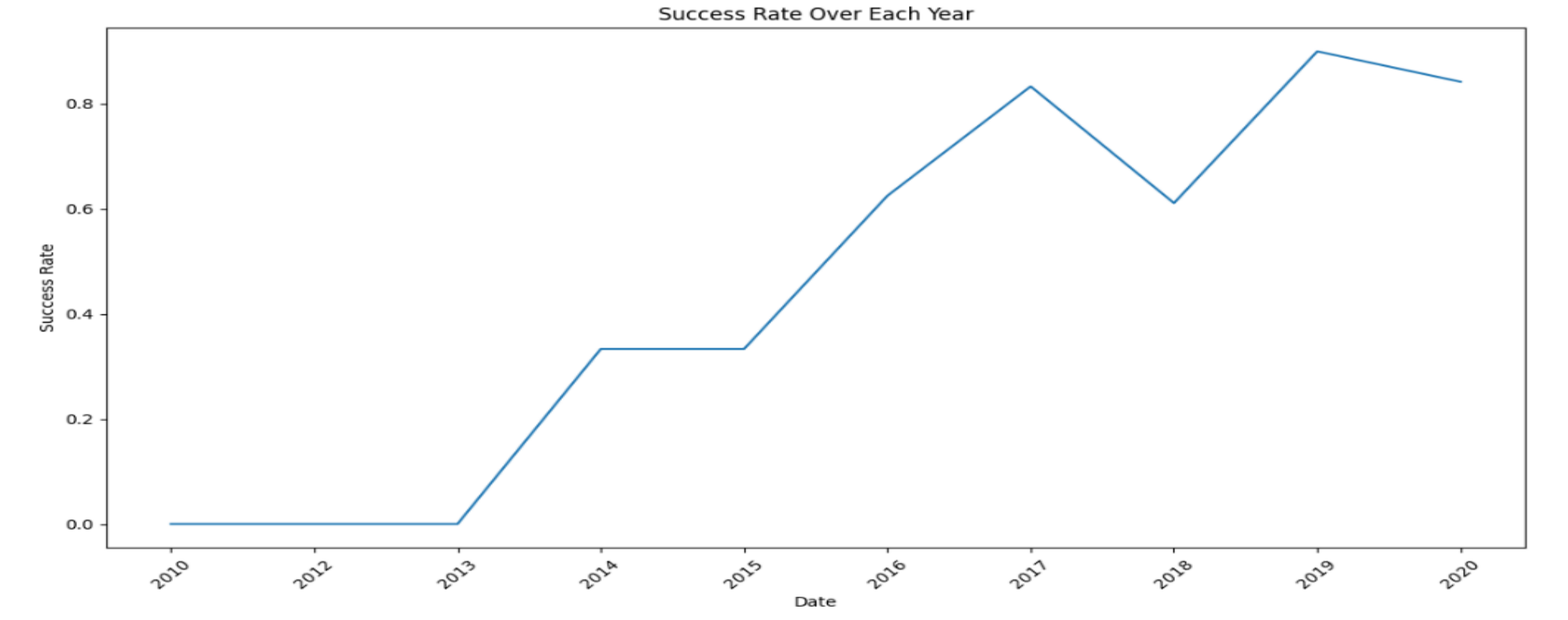
In LEO orbit success appears related to Flight Number, On the other hand there is no relation ship between Flight Number and GTO.

Payload vs. Orbit Type



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)

Launch Success Yearly Trend



The success rate of SpaceX since 2013 kept increasing till 2020

All Launch Site Names

```
[7]: %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTBL
 * sqlite:///my_data1.db
Done.
```

[7]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

The above SQL query is written using SQL magic. I have made a connection to database my_data1.db and created a table named SPACEXTBL where all SpaceX data is stored in records. The query returns launch site names in SpaceX data.

Launch Site Names Begin with 'CCA'

```
[8]: %sql SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5
 * sqlite:///my_data1.db
Done.
```

[8]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

The query returns top 5 records having Launch_Site column values starting with ‘CCA’. LIMIT is used to limit the query up to 5 records and LIKE is used for passing the search string pattern

Total Payload Mass

```
%sql SELECT Customer,sum("PAYLOAD_MASS__KG_") as Total FROM SPACEXTBL WHERE Customer == 'NASA (CRS)' GROUP BY Customer
```

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

Customer	Total
NASA (CRS)	45596.0

The query is used for calculating the Total Payload Mass by NASA (CRS). GROUP BY is used to group the columns and sum() is an aggregate function gives total mass.

Average Payload Mass by F9 v1.1

```
%sql Select avg("PAYLOAD_MASS__KG_") as [Average Payload Mass] from SPACEXTBL where "Booster_Version" like 'F9%'
```

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

Average Payload Mass
6138.287128712871

The query gives the average Payload Mass for the booster version F9 v1.1. avg() is another aggregate function gives average value.

First Successful Ground Landing Date

```
%sql select date from SPACEXTBL where Landing_Outcome == 'Success (ground pad)' Limit 1
```

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

Date
22/12/2015

The query gives first date of successful landing on the ground

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT "Booster_Version" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000 AND "Landing_Outcome" = 'Success (drone ship)'
```

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

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

This query gives booster version who are having payload mass between 4000 and 6000 and whose landing outcome is success on drone ship.

Total Number of Successful and Failure Mission Outcomes

```
%sql select "Mission_Outcome",count(*) as Total from SPACEXTBL group by "Mission_Outcome"
```

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

Mission_Outcome	Total
None	898
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

The query gives Total number of Success and Failure Outcomes

Boosters Carried Maximum Payload

```
%sql SELECT Distinct "Booster_Version" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG" = (SELECT MAX("PAYLOAD_MASS__KG") FROM SPACEXTBL)
```

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

Booster_Version
F9 v1.0 B0003
F9 v1.0 B0004
F9 v1.0 B0005
F9 v1.0 B0006
F9 v1.0 B0007
F9 v1.1 B1003
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1010
F9 v1.1 B1012
F9 v1.1 B1013
F9 v1.1 B1014

This query gives booster versions carrying maximum payload mass using sub query.

2015 Launch Records

```
%sql SELECT SUBSTR(Date, 4, 2) AS month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTBL WHERE SUBSTR(Date, 7, 4) = '2015' AND L
```

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

month	Landing_Outcome	Booster_Version	Launch_Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

This query will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

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

```
%%sql SELECT RANK() OVER (ORDER BY success_count DESC) AS rank, success_count, "Landing_Outcome" FROM (
        SELECT COUNT(*) AS success_count, "Landing_Outcome"
        FROM SPACEXTBL
        WHERE "Landing_Outcome" like '%Success%' AND "Date" BETWEEN '04-06-2010' AND '20-03-2017'
        GROUP BY "Landing_Outcome"
        ) AS subquery;
```

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

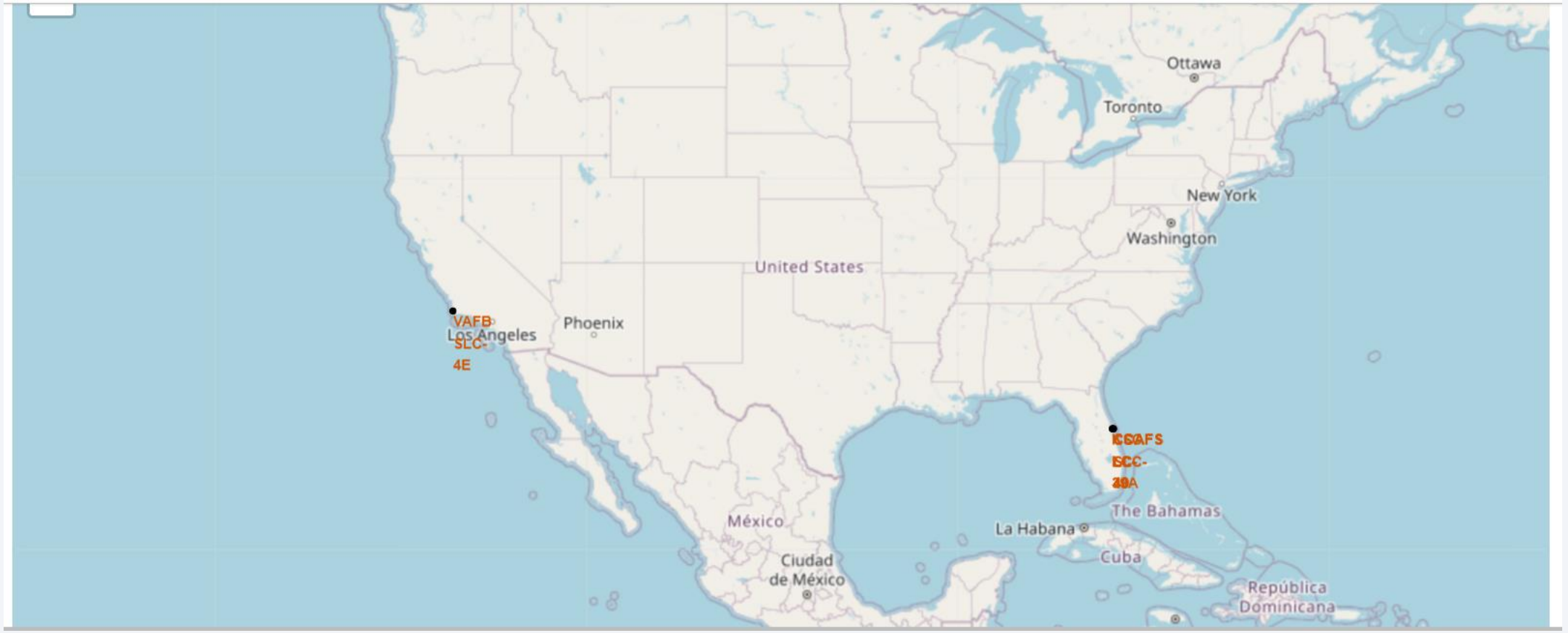
rank	success_count	Landing_Outcome
1	20	Success
2	8	Success (drone ship)
3	7	Success (ground pad)

The query gives count of all successful landing_outcomes between the dates 04-06-2010 and 20-03-2017 in descending order.

Section 3

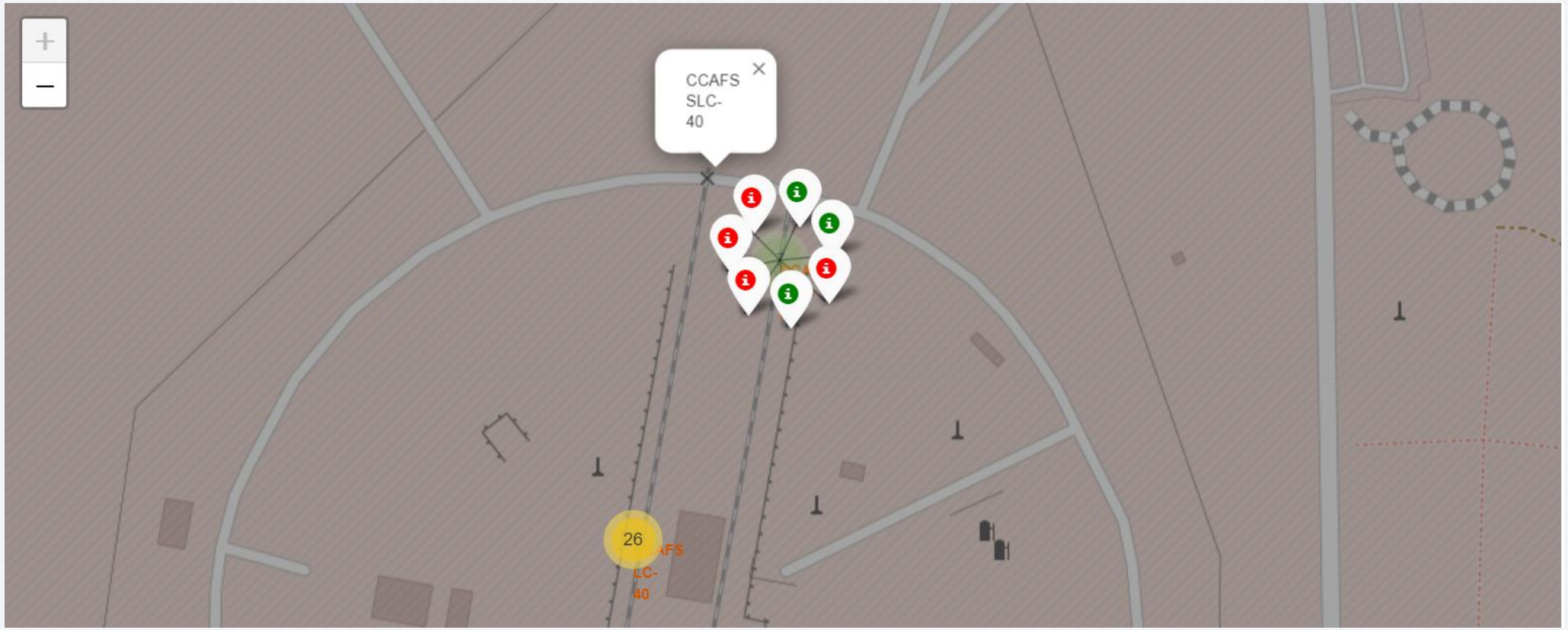
Launch Sites Proximities Analysis

Folium Map With Marker



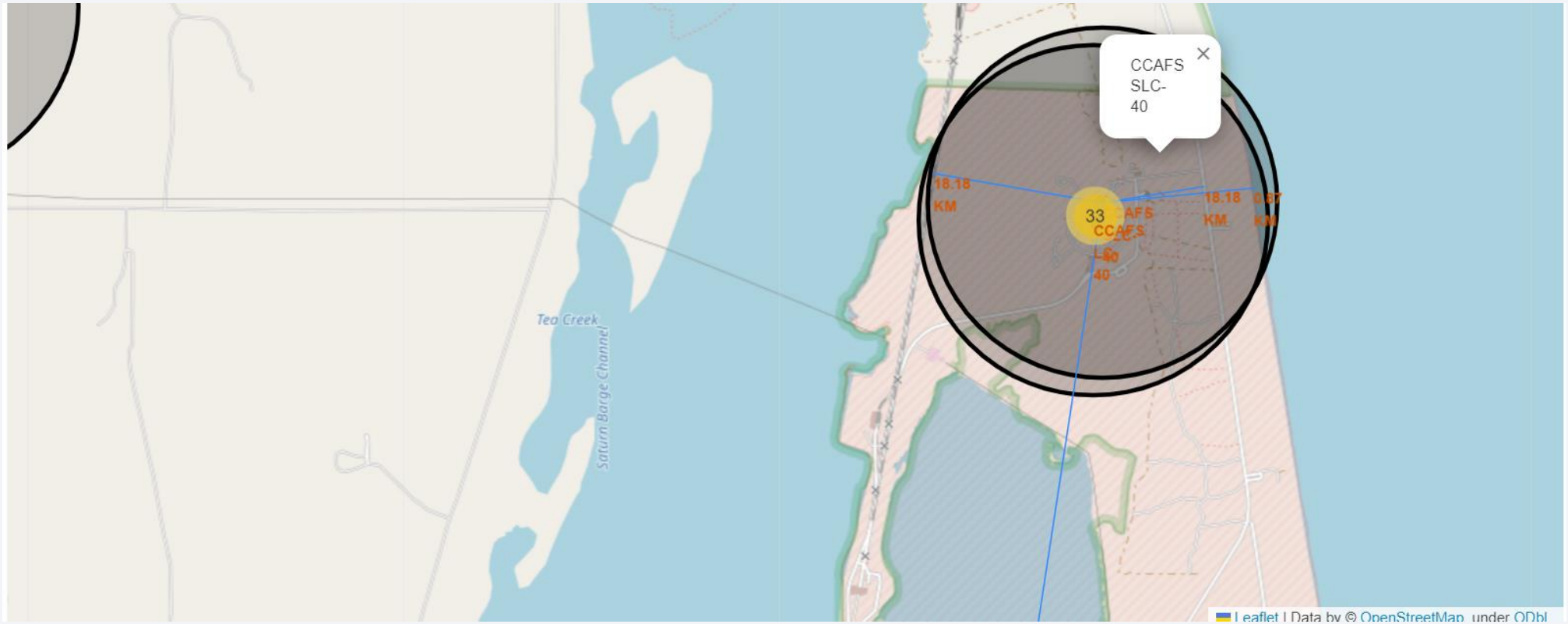
Folium map with red circle marker at the launch site locations

Folium Map With Marker Cluster

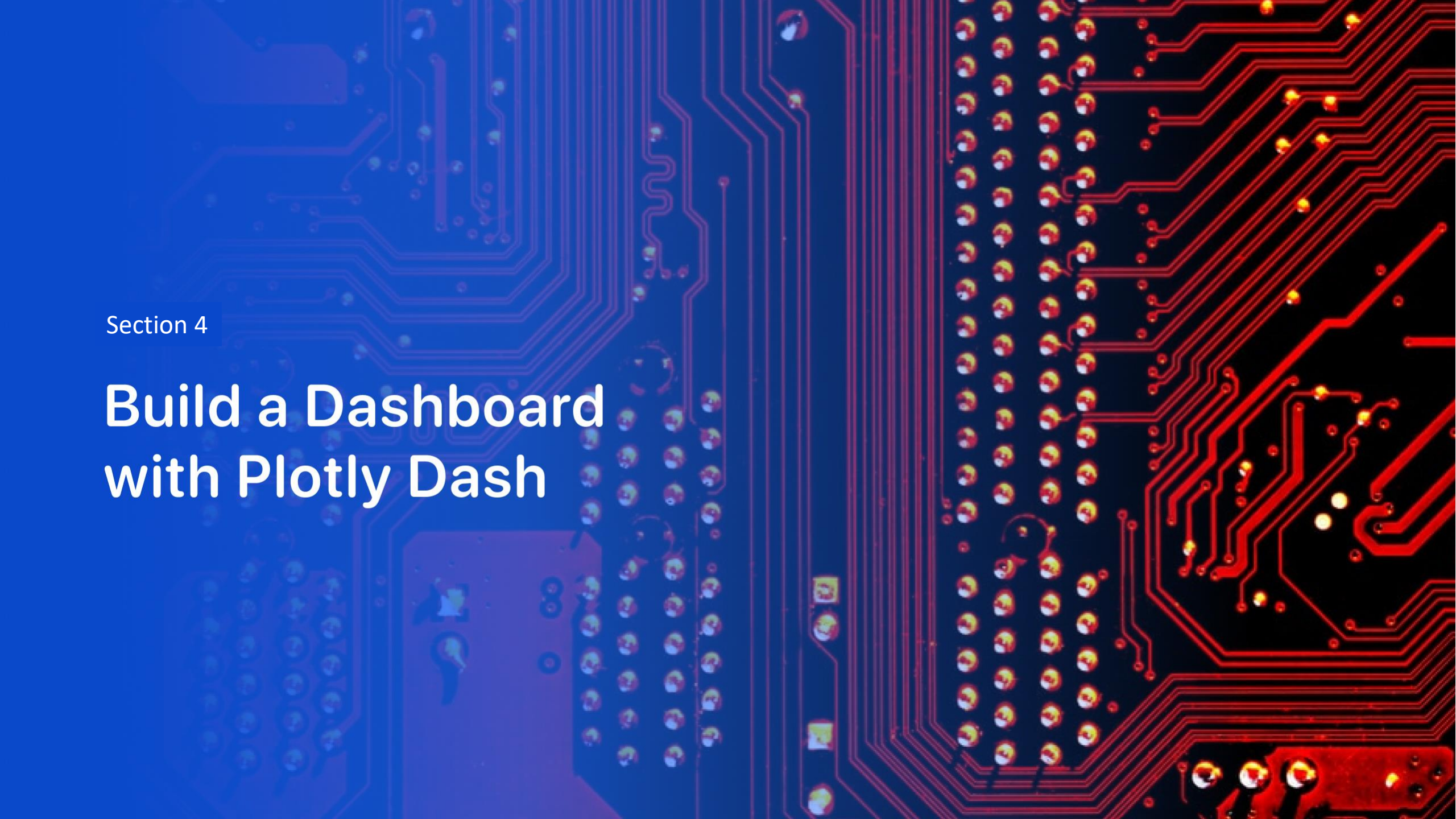


Folium map with marker cluster at launch site CCAFS SLC – 40. Green color for Success and Red color for Failure at launch site

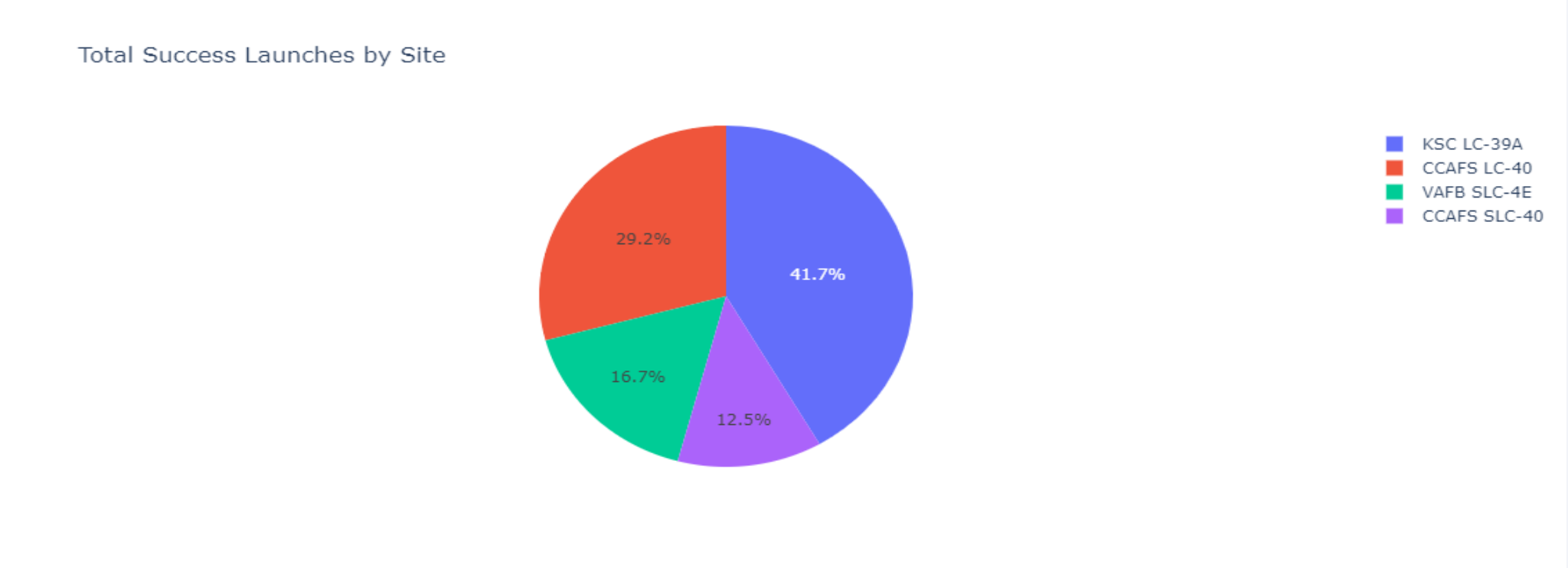
Folium Map With Poly Lines



Folium map with blue poly lines representing distance from the nearest coastal line, railway line, highway, city.

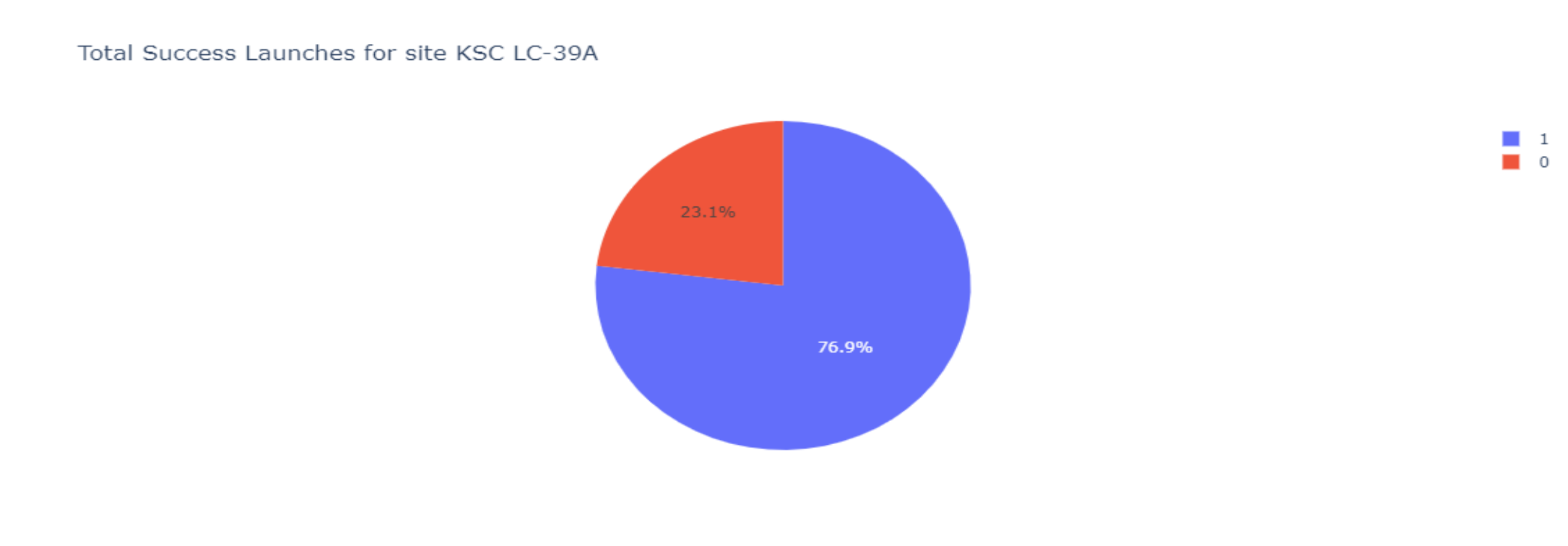

Section 4
Build a Dashboard with Plotly Dash

Dashboard of Successful Launches For Each Site



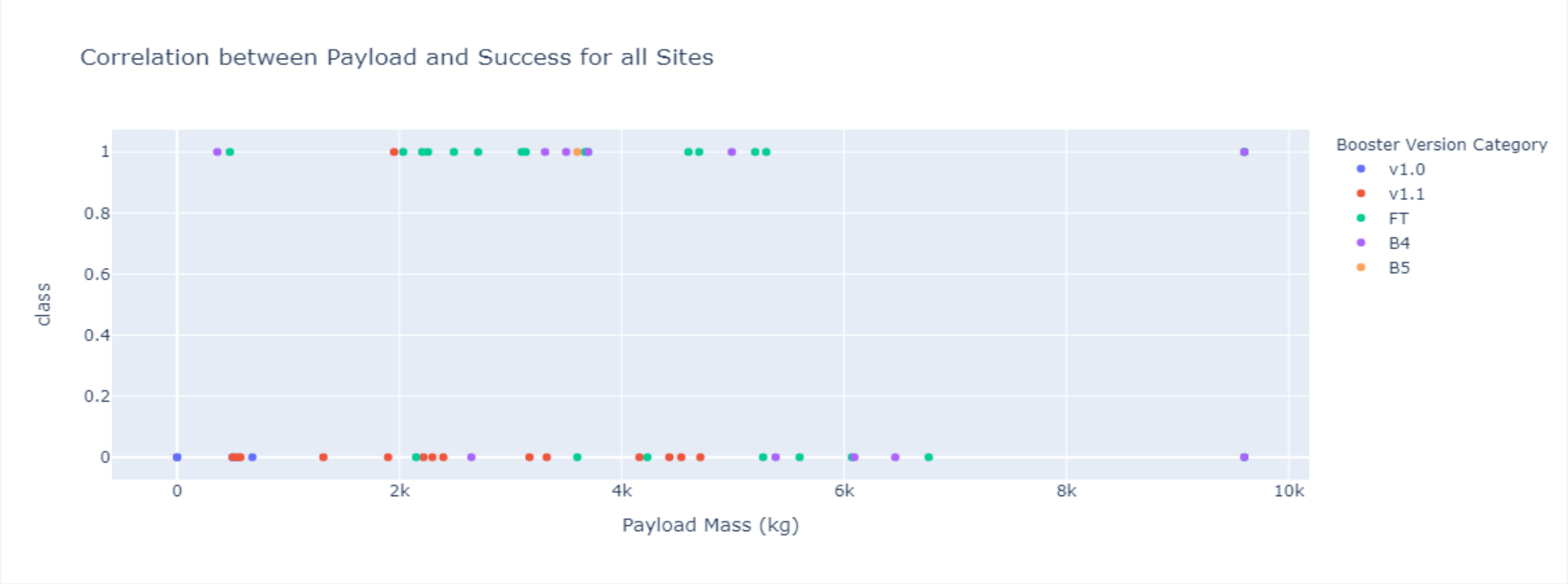
From the pie chart KSC LC – 39A has highest success rate than other launch sites

Dashboard of KSC LC – 39A



From the pie chart we can infer the KSC LC – 39A has 76.9% Success and 23.1% Failure for all its launches

Dashboard of Scatterplot Payload vs Success

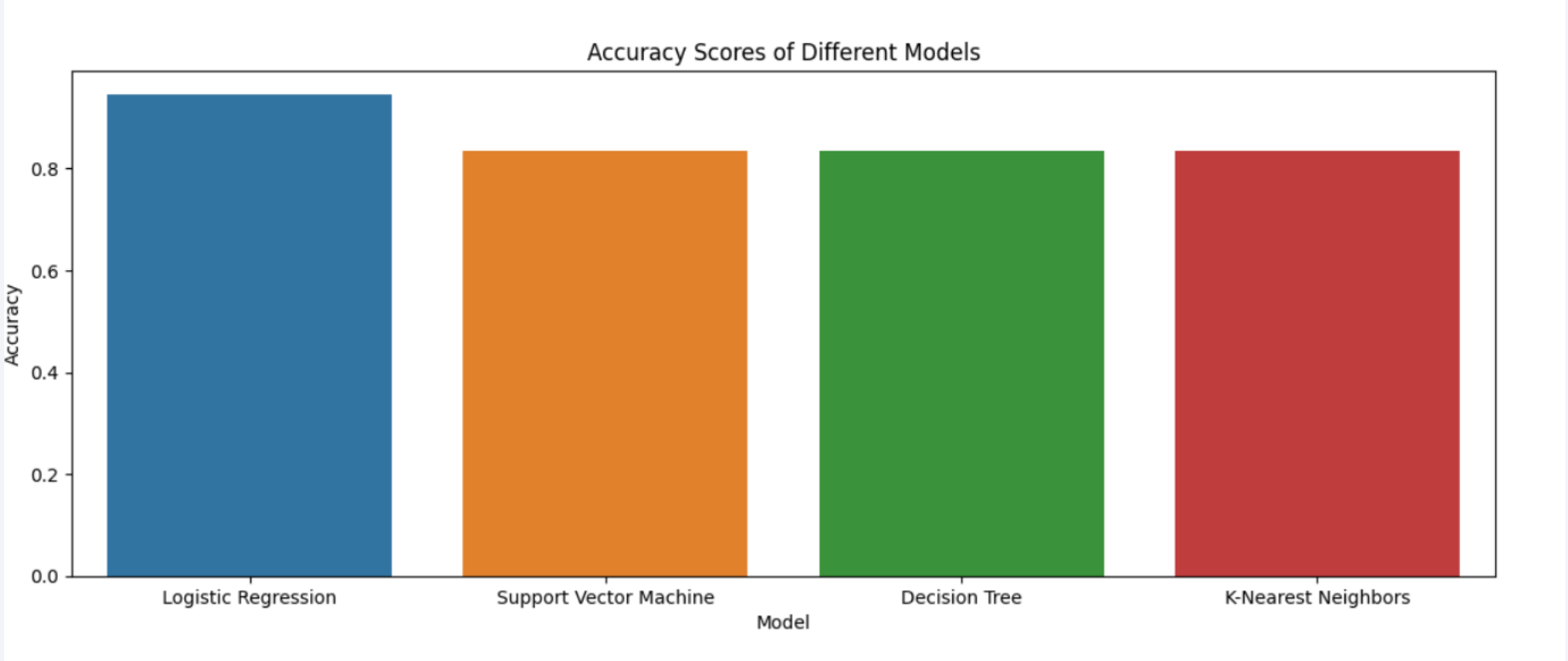


We can clearly observe that Booster Versions FT and B4 will consume high payload mass than other versions, Also they have the high success rate than other versions

Section 5

Predictive Analysis (Classification)

Classification Accuracy



Logistic Regression has 90% accuracy and Other models has 80%

Confusion Matrix

The confusion matrix for the K-Nearest Neighbors (KNN) model shows that out of 6 instances where the predicted class was "did not land," 5 were correctly classified, and 1 was misclassified as "land." Similarly, out of 12 instances where the predicted class was "land," all of them were correctly classified.

Conclusions

- Data Collection from various sources using web scraping from Wikipedia and extracting data from SpaceX REST API
- Data Wrangling techniques were used for data cleaning, filling missing values etc.
- Data Visualizations like scatter plot, cat plot, bar graph, line chart are used
- Advanced Visualizations like Folium Maps and Plotly dashboards were used for interactive plots.
- Classification models like KNN, SVM, Logistic Regression, Decision Tree were used for model building
- Model Evaluation was measured based on metrics accuracy score and confusion matrix for each model

Appendix-1

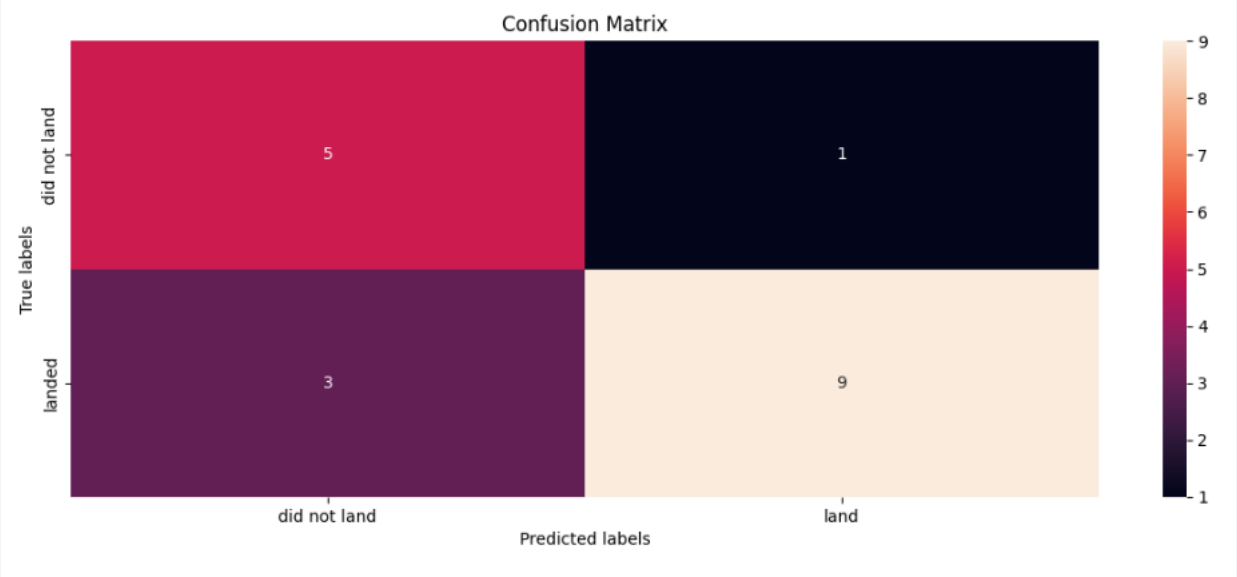
Additionally, I have tried a Naive Bayes Model and calculate the accuracy score 77.7%

```
from sklearn.naive_bayes import GaussianNB
parameters = {}
nb = GaussianNB()
nb_cv = GridSearchCV(nb, parameters, cv=10)
nb_cv.fit(X_train, Y_train)
print("Tuned Hyperparameters (Best Parameters):", nb_cv.best_params_)
print("Best Accuracy:", nb_cv.best_score_)
accuracy = nb_cv.score(X_test, Y_test)
print('Test Accuracy:', accuracy)
yhat = nb_cv.predict(X_test)
plot_confusion_matrix(Y_test, yhat)
```

```
Tuned Hyperparameters (Best Parameters): {}
Best Accuracy: 0.5678571428571428
Test Accuracy: 0.7777777777777778
```

Appendix-2

Confusion matrix for Naive Bayes Model.



References

- All coded files and this report in pdf format was uploaded in GitHub. You can refer through this link https://github.com/PynetDev/CapstoneProject

- REST APIs used for Data Collection - https://api.spacexdata.com/v4

- Wikipedia - https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

In this project I have used different classification algorithms like KNN, Logistic Regression, Decision Tree, SVM, Naive Bayes and compared accuracies of each model. However, there are other classification algorithms may give the best accuracy than this models. I would really appreciate if you try and explore.

Thank you!