IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

ELENI DAKOU
Wed Jul 20th 2022

Outline

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

Executive Summary

- **Summary of methodologies**

- Data Collection through API

- Data Collection with Web Scraping - Data Wrangling

- Exploratory Data Analysis with SQL

- Exploratory Data Analysis with Data Visualization

- Interactive Visual Analytics with Folium - Machine Learning Prediction

- **Summary of all results**

- Exploratory Data Analysis result

- Interactive analytics in screenshots

- Predictive Analytics result

Introduction

Project background and context

Space X 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 Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

Problems you want to find answers

- What factors determine if the rocket will land successfully?
- The interaction amongst various features that determine the success rate of a successful landing.
- What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
- Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
- One-hot encoding was applied to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
- create a column for the class
- Standardize the data
- Split into training data and test data
- Find the method performs best using test data

Data Collection

- **The data was collected :**

- Using the get request to the SpaceX API.

- Decoding the response content as a Json using .json() function call and turning it into a pandas dataframe using .json_normalize().

- Cleaning the data (checking for missing values and filling in missing values where was necessary).

-Perfoming web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.

- Extracting the launch records as HTML table, parsing the table and converting it to a pandas dataframe for future analysis

Data Collection – SpaceX API

- We used the Get request to the SpaceX API to collect data, then we clean the data and we lastly wrangle and format them.

- **The GitHub URL of the completed SpaceX API calls notebook is** https://github.com/ENtakou02/ELENIS-REPOSITORY/blob/master/jupyter-labs-spacex-data-collection-api%20(5).ipynb

1. Get request for rocket launch data using API

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

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

2. Use json_normalize method to convert json result to dataframe

```
In [12]: # Use json_normalize method to convert the json result into a dataframe

         # decode response content as json
         static_json_df = res.json()

In [13]: # apply json_normalize
         data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

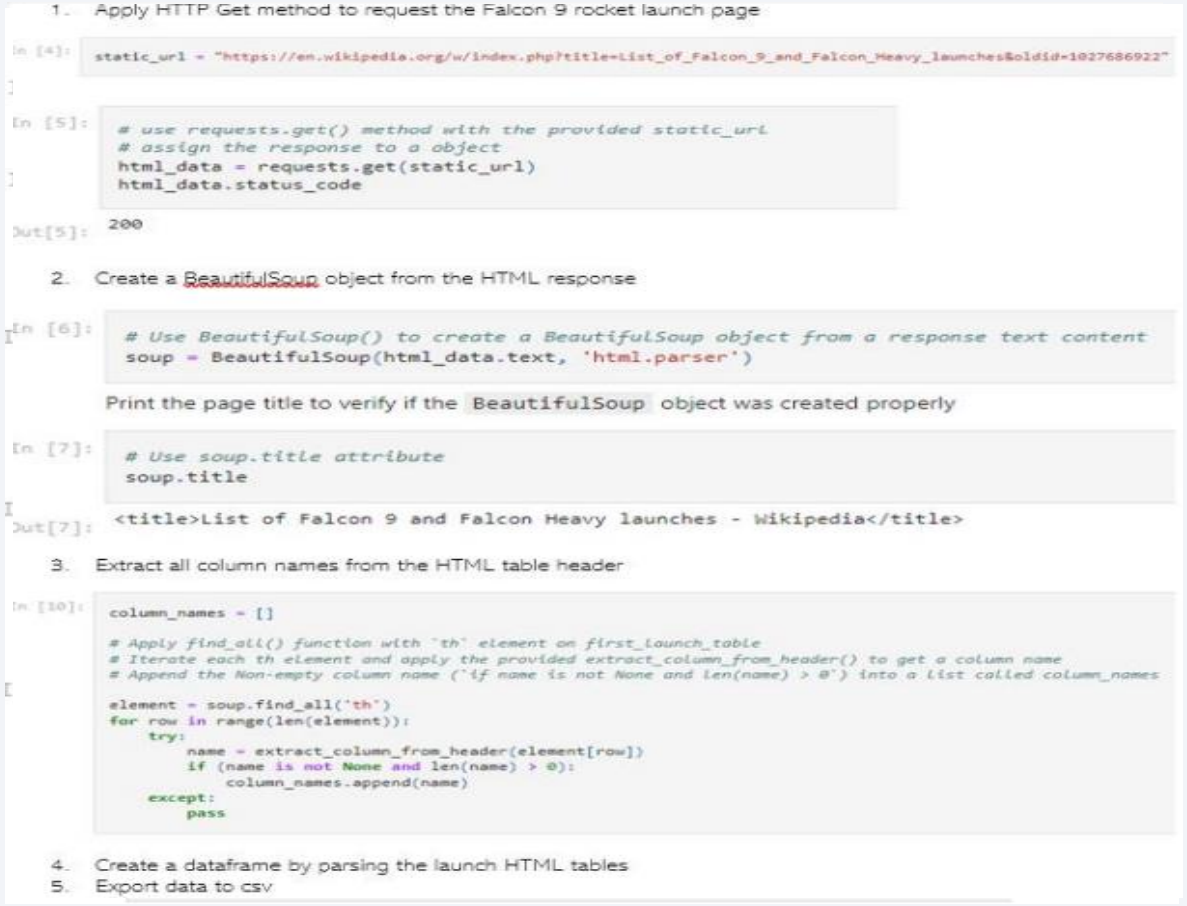
```
In [30]: rows = data_falcon9['PayloadMass'].values.tolist()[0]

         df_rows = pd.DataFrame(rows)
         df_rows = df_rows.replace(np.nan, PayloadMass)

         data_falcon9['PayloadMass'][0] = df_rows.values
         data_falcon9
```

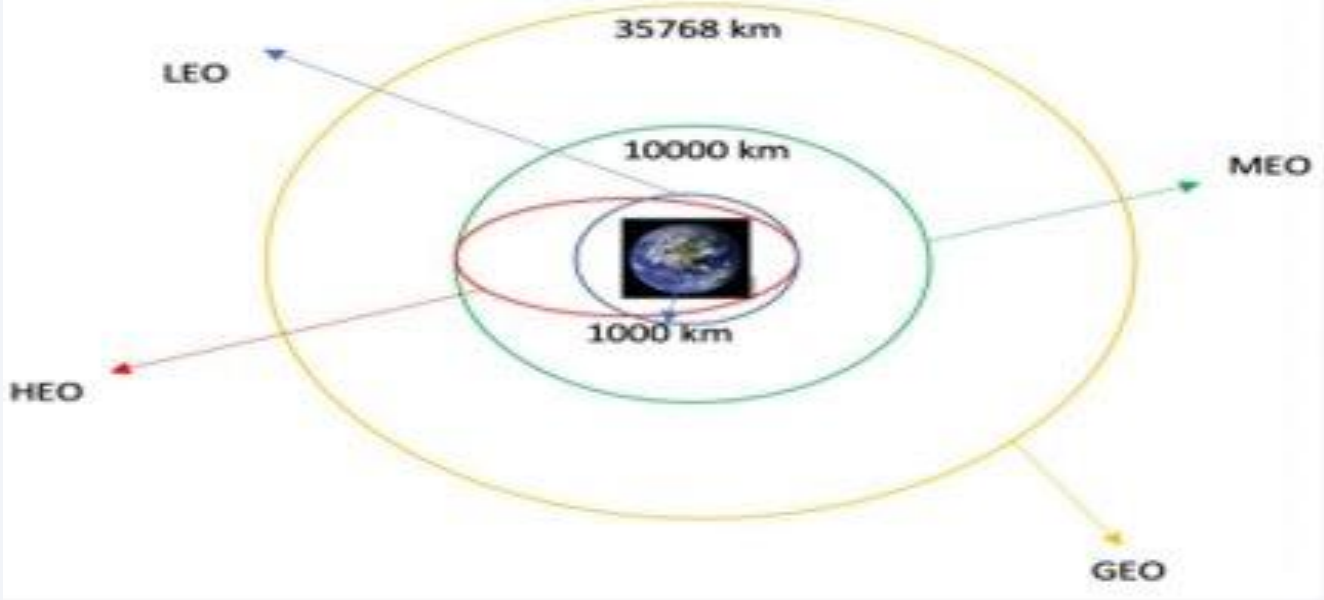
Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup and we parsed the table and converted it into a pandas dataframe (ex. df=pd.DataFrame.from_dict(launch_dict)
-
- **The GitHub URL of the completed web scraping notebook** is https://github.com/ENtakou02/ELENIS-REPOSITORY/blob/master/jupyter-labs-webscraping%20(4).ipynb



Data Wrangling

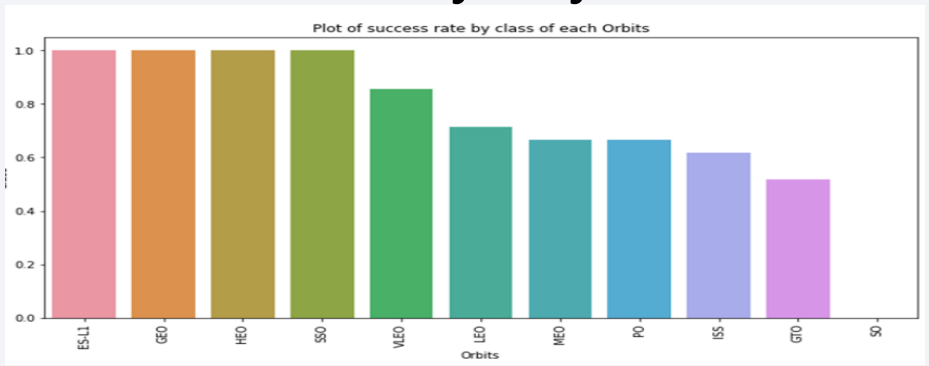
- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.

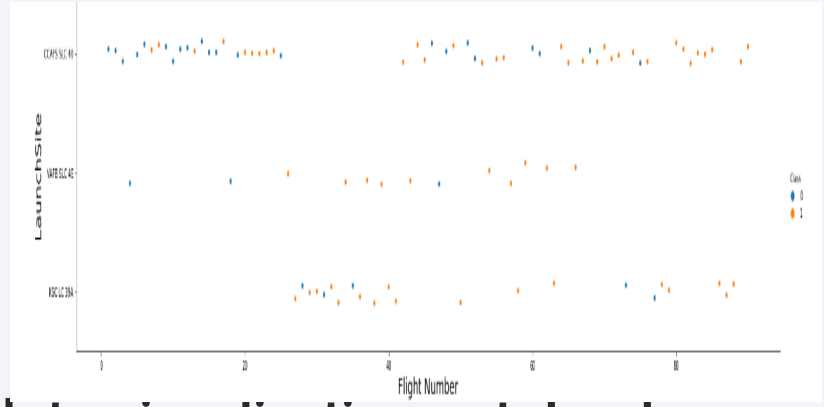


- **The GitHub URL of your completed data wrangling related notebooks is** https://github.com/ENtakou02/ELENIS-REPOSITORY/blob/master/labs-jupyter-spacex-Data%20wrangling%20(1).ipynb

EDA with Data Visualization

- We perform exploratory Data Analysis and Preparing Data Feature Engineering using Pandas and Matplotlib. Then, we explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.





- **The GitHub URL of your completed EDA with data visualization notebook is** https://github.com/ENtakou02/ELENIS-REPOSITORY/blob/master/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- **The SpaceX dataset was loaded into a PostgreSQL database .**
- **We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:**

- *The names of unique launch sites in the space mission.*
- *The total payload mass carried by boosters launched by NASA (CRS)*
- *The average payload mass carried by booster version F9 v1.1*
- *The total number of successful and failure mission outcomes*
- *The failed landing outcomes in drone ship, their booster version and launch site names.*

- **The GitHub URL of your completed EDA with SQL notebook is** https://github.com/ENtakou02/ELENIS-REPOSITORY/blob/master/jupyter-labs-eda-sql-coursera_sqllite%20(2).ipynb

Build an Interactive Map with Folium

- **We performed interactive visual analytics using Folium :**
- we marked all launch sites on a map, we mark the success/failed launches for each site on the map adding map objects such as markers, circles.
- we calculated the distances between a launch site to its proximities
- We used the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.

The GitHub URL of my completed interactive map with Folium map is https://github.com/ENtakou02/ELENIS-REPOSITORY/blob/master/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

- **The GitHub URL of my completed Plotly Dash lab** is https://github.com/ENtakou02/ELENIS-REPOSITORY/blob/master/spacex_dash_app.py **and the results** https://github.com/ENtakou02/ELENIS-REPOSITORY/blob/master/dashboard.odt

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas
- We performed exploratory Data Analysis and determine Training Labels
- We created a column for the class
- We standardized the data
- We split into training data and test data
- We found the best Hyperparameter for SVM, Classification Trees and Logistic Regression
- We found the method performs best using test data

```
Y = data['Class'].to_numpy()
```

```
X.mean(axis=0)
X.std(axis=0)
```

```
X_train, X_test, Y_train, Y_test
```

```
X_train, X_test, Y_train, Y_test =  train_test_split(X,Y,test_size=0.2,random_state=2)
```

```
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```

```
logreg_cv = GridSearchCV(lr, parameters, cv=10)
logreg_cv.fit(X_train, Y_train)

print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
```

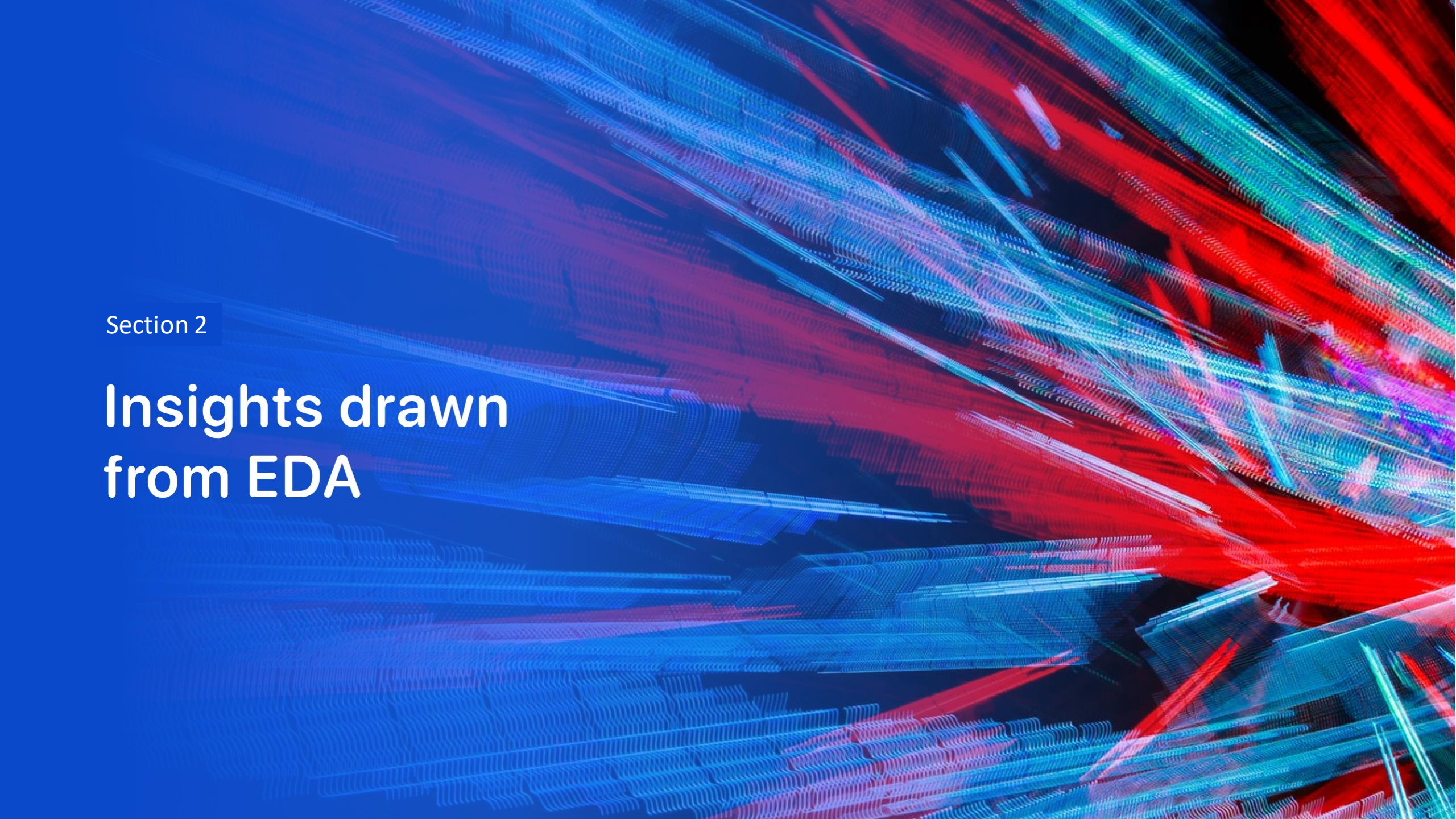
- **The GitHub URL of my completed predictive analysis lab is** https://github.com/ENtakou02/ELENIS-REPOSITORY/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

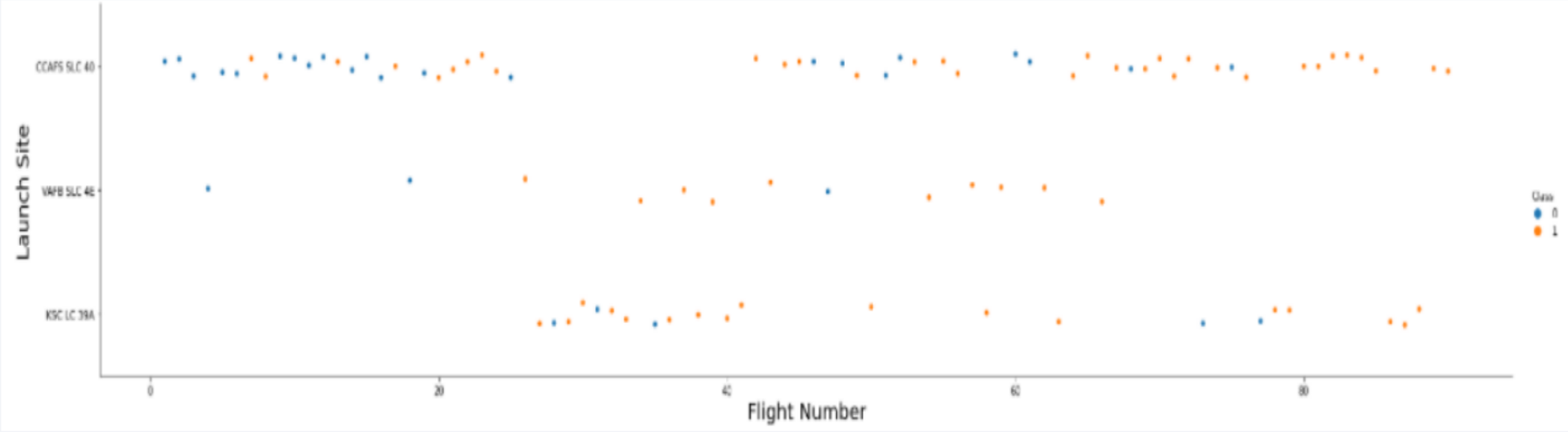
Insights drawn from EDA



Flight Number vs. Launch Site

- We found that the larger the flight amount at a launch site, the greater the success rate at a launch site.

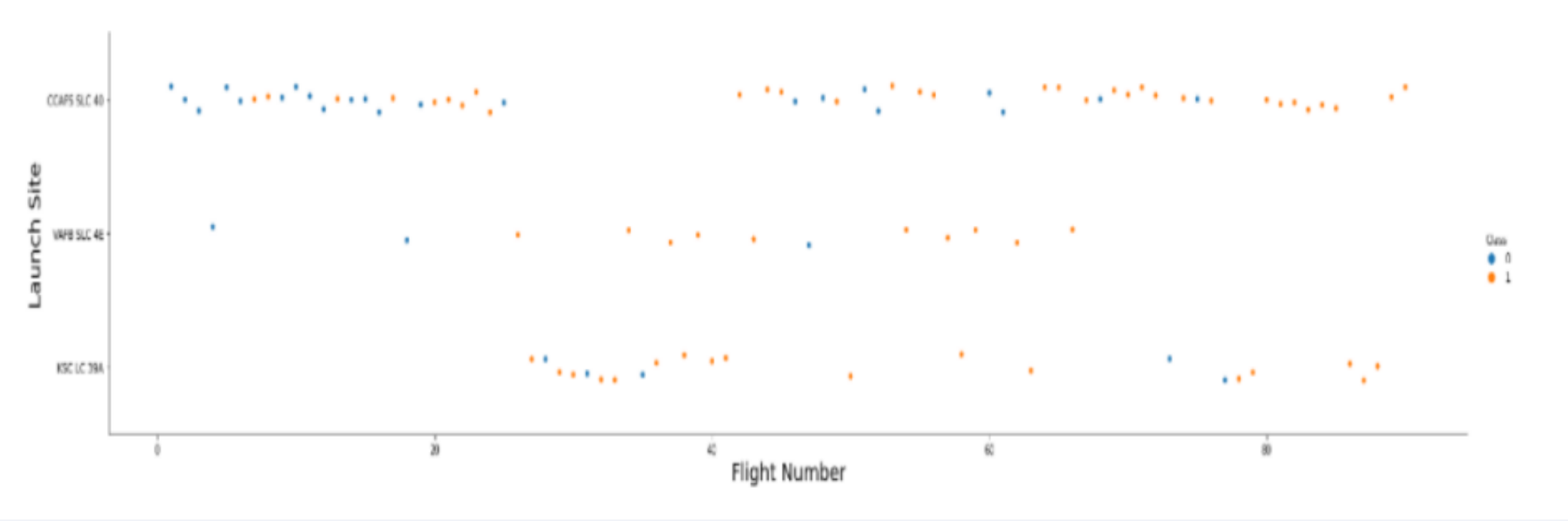
- The screenshot of the scatter plot



Payload vs. Launch Site

- We can see that the greater the payload for launch site CCAFS SLC 40 the higher the success rate for the rocket.

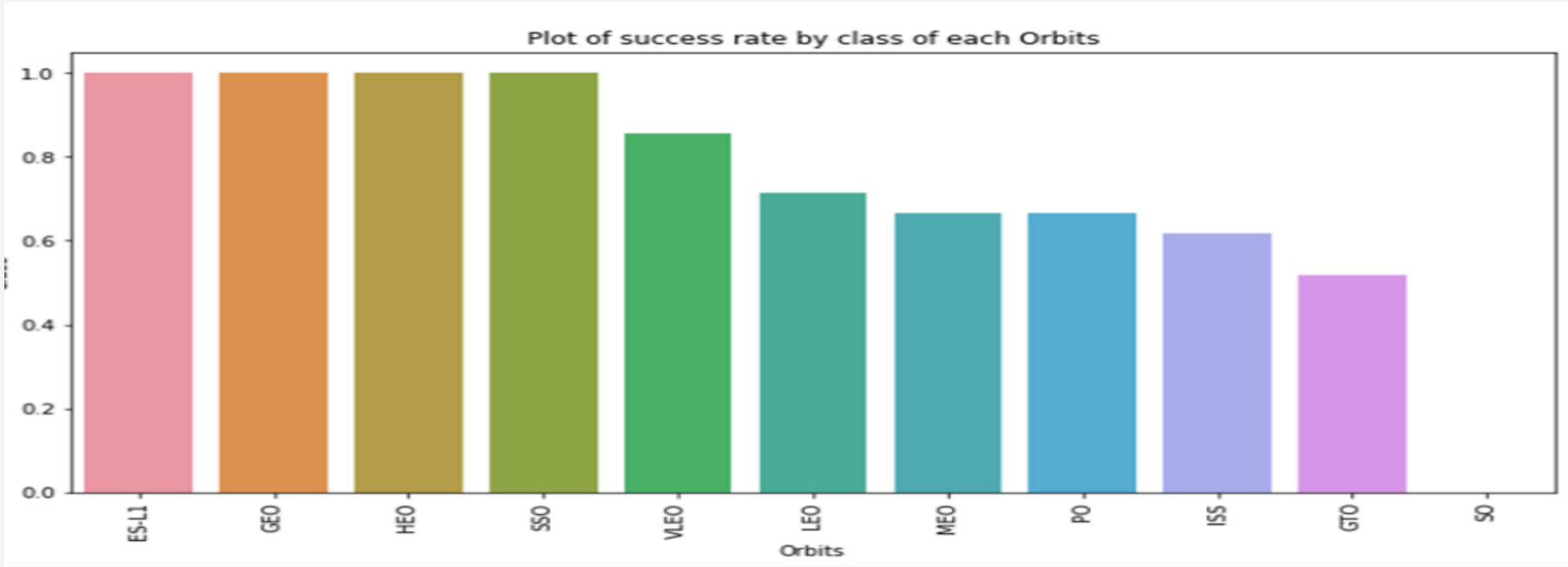
- The screenshot of the scatter plot



Success Rate vs. Orbit Type

- We can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

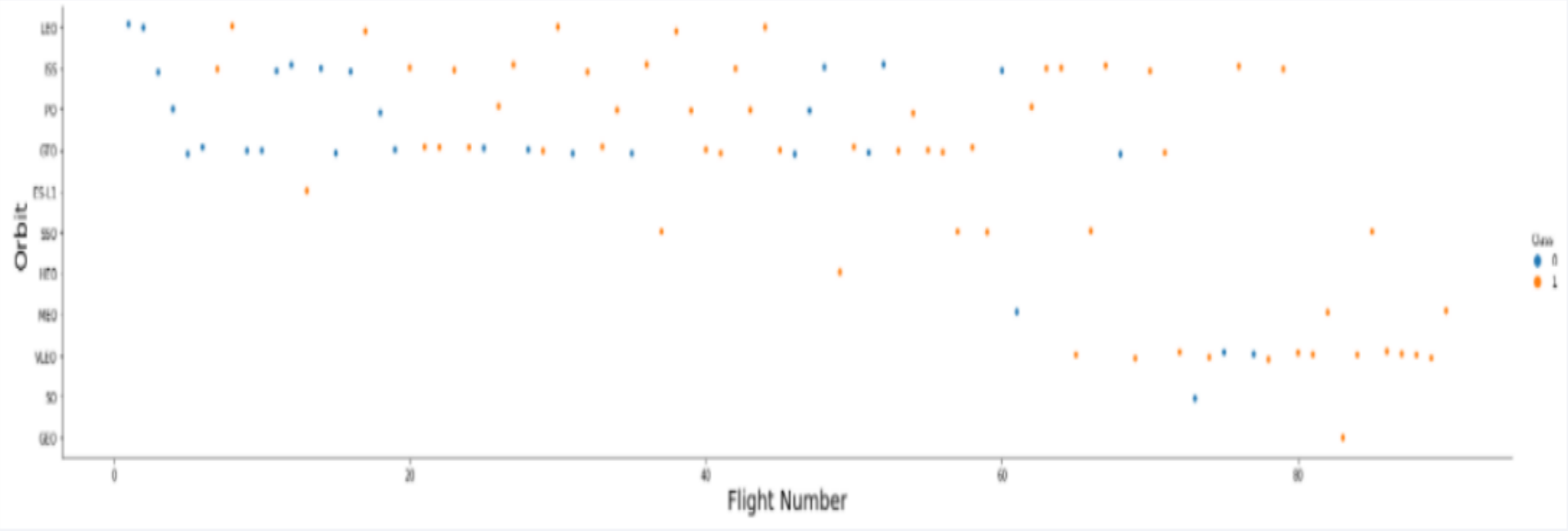
- The screenshot of the scatter Plot of success rate by class of each Orbits



Flight Number vs. Orbit Type

- We can see that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.

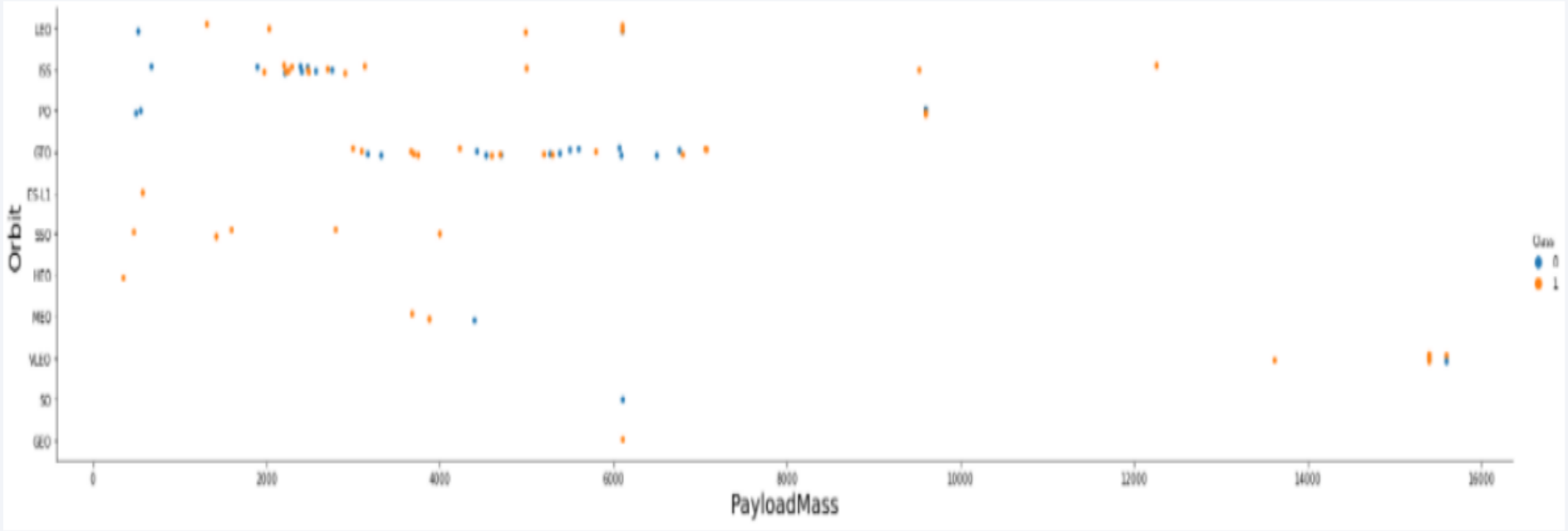
- The screenshot of the scatter plot .



Payload vs. Orbit Type

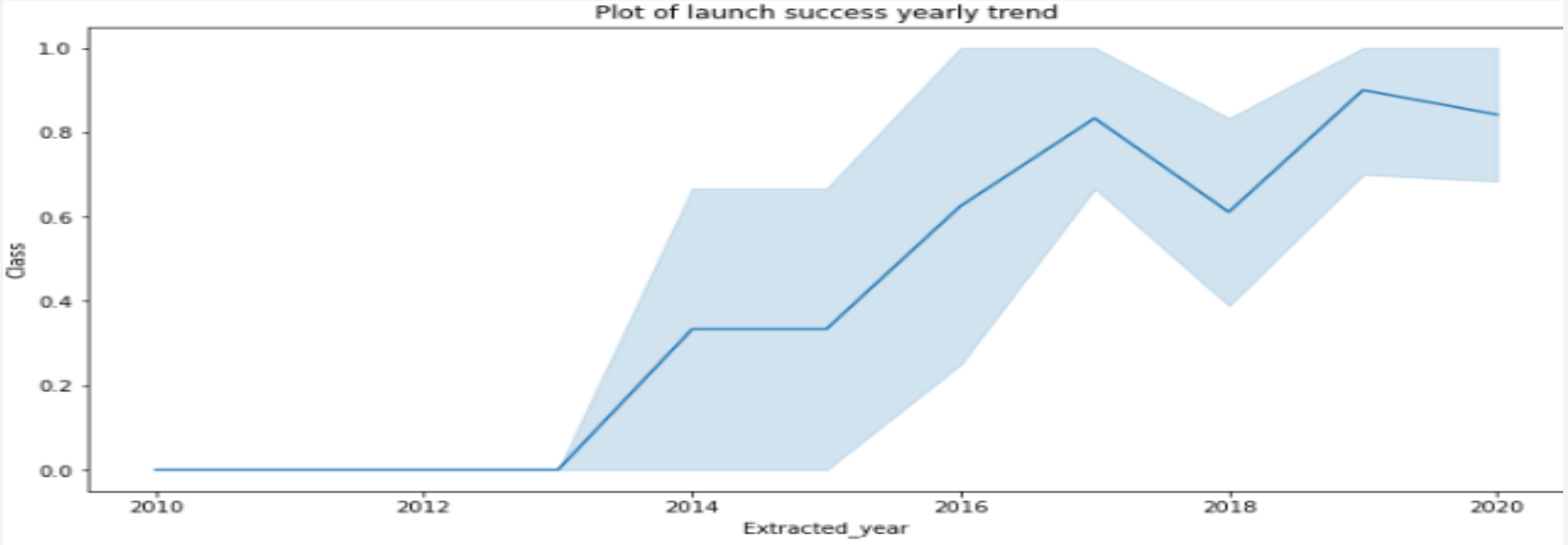
- We can see that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.

- The screenshot of the scatter plot .



Launch Success Yearly Trend

- We can observe that the sucess rate since 2013 kept increasing till 2020



All Launch Site Names

The names of all Launch Site are:

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

```
# Select relevant sub-columns: `Launch Site`, `Lat(Latitude)`, `Long(Longitude)`, `class`
spacex_df = spacex_df[['Launch Site', 'Lat', 'Long', 'class']]
launch_sites_df = spacex_df.groupby(['Launch Site'], as_index=False).first()
launch_sites_df = launch_sites_df[['Launch Site', 'Lat', 'Long']]
launch_sites_df
```

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

Launch Site Names Begin with 'CCA'

- We used the query above to display 5 records where launch sites begin with `CCA`

Display 5 records where launch sites begin with the string 'CCA'

```
%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- We calculated the total payload carried by boosters from NASA

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

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

payloadmass
619967

Average Payload Mass by F9 v1.1

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

Display average payload mass carried by booster version F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

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

payloadmass
6138.287128712871

First Successful Ground Landing Date

- We can see that the dates of the first successful landing outcome on ground pad was 1st March of 2013

```
2]: %sql select min(DATE) from SPACEXTBL;
 * sqlite:///my_data1.db
Done.
2]: min(DATE)
01-03-2013
```

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

We calculate the total number of successful and failure mission outcomes

```
%sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

missionoutcomes
1
98
1
1

Boosters Carried Maximum Payload

- We present the list the names of the booster which have carried the maximum payload mass

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

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

boosterversion
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

- We used the WHERE clause, to filter for mission outcome, their booster versions, and launch site names for year 2015

```
%sql SELECT substr(Date, 4, 2),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE from SPACEXTBL where substr(Date,7,4)='2015';
```

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

substr(Date, 4, 2)	Mission_Outcome	Booster_Version	Launch_Site
01	Success	F9 v1.1 B1012	CCAFS LC-40
02	Success	F9 v1.1 B1013	CCAFS LC-40
03	Success	F9 v1.1 B1014	CCAFS LC-40
04	Success	F9 v1.1 B1015	CCAFS LC-40
04	Success	F9 v1.1 B1016	CCAFS LC-40
06	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	CCAFS LC-40

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

- We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN **2010-06-04 to 2010-03-20.**

- We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

```
%sql SELECT LANDING_OUTCOME,COUNT(Landing_Outcome) as Count FROM SPACEXTBL WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017' GROUP BY Landing_Outcome O
```

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

Landing_Outcome	Count
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

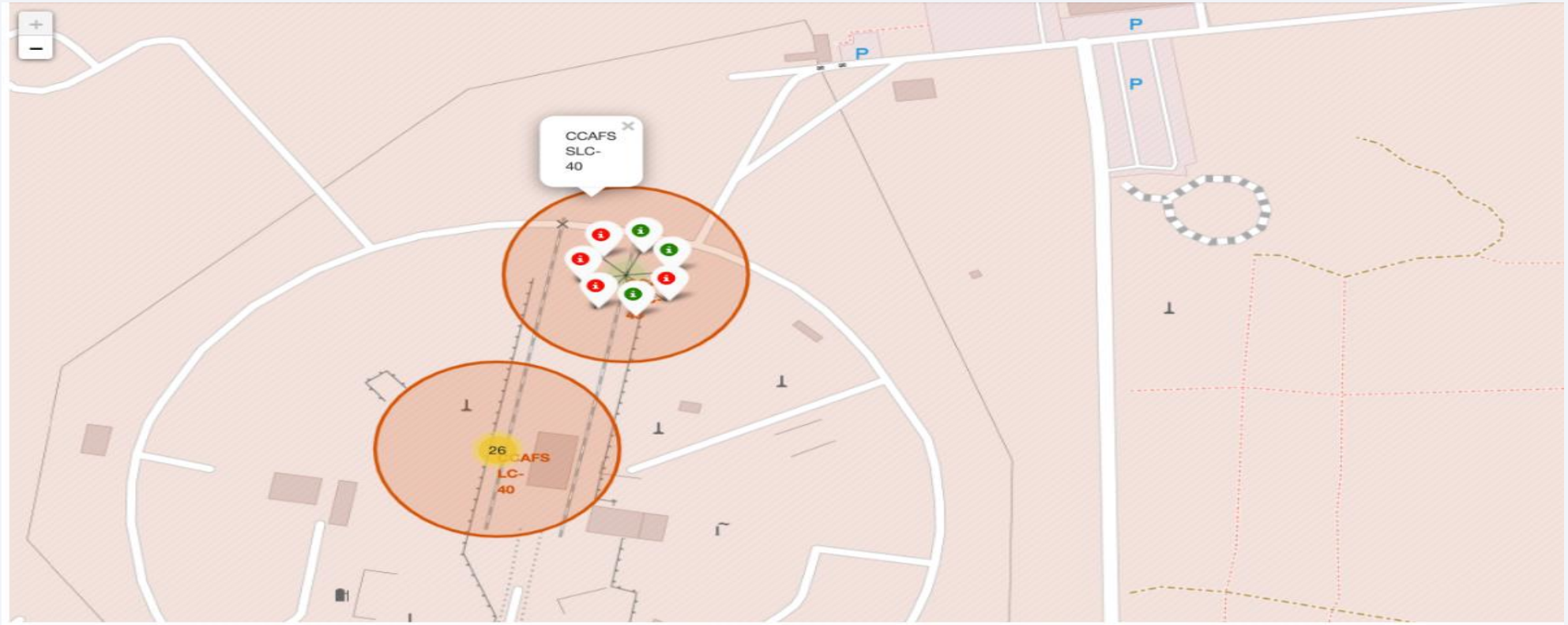

Section 3
Launch Sites
Proximities Analysis

Folium Map with all launch sites' location markers on a global map

- We can see that the Space X launch sites are in the US of America coasts (Florida and California).

Markers showing launch sites with color labels

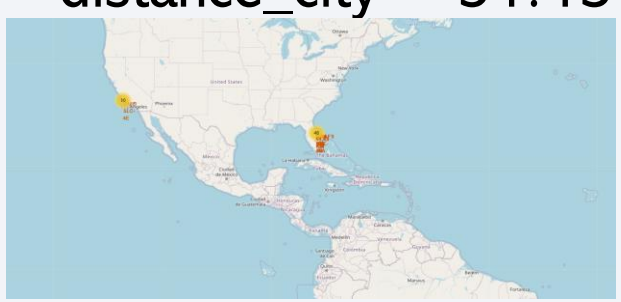
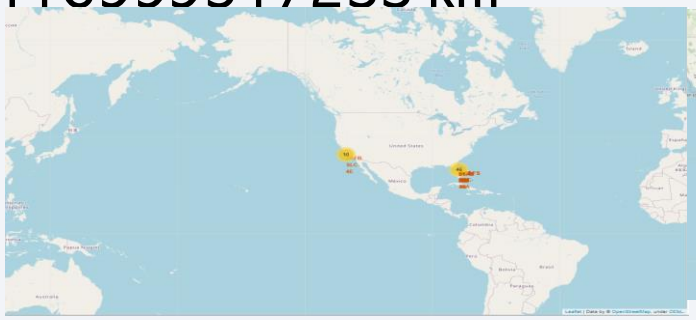
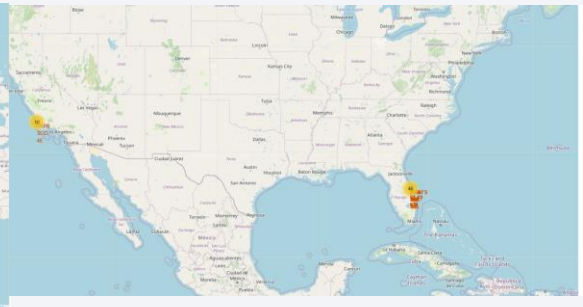
- Green Marker shows successful Launches and Red Marker shows Failures

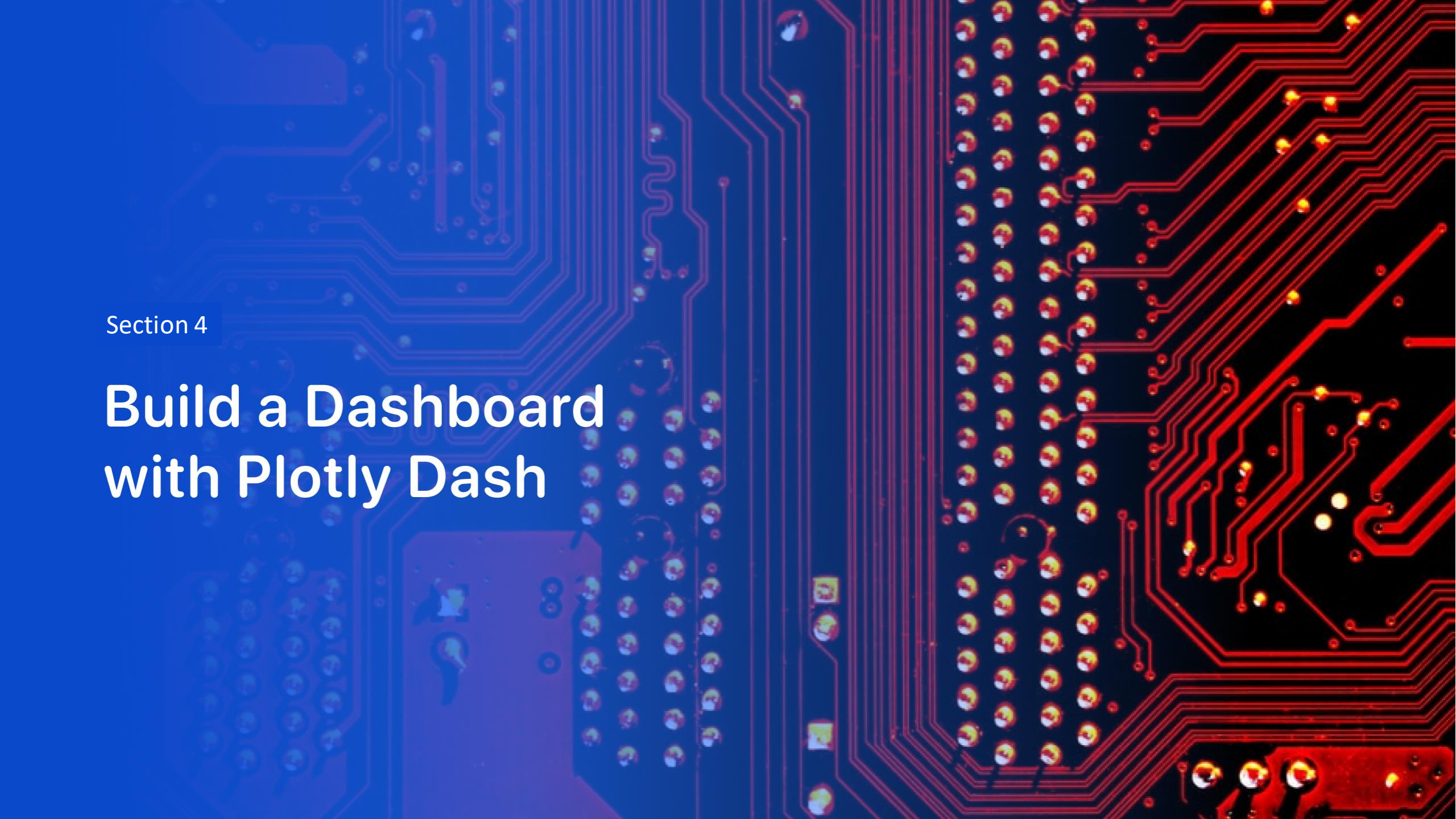


Launch Site distance to railway, highway, coastline

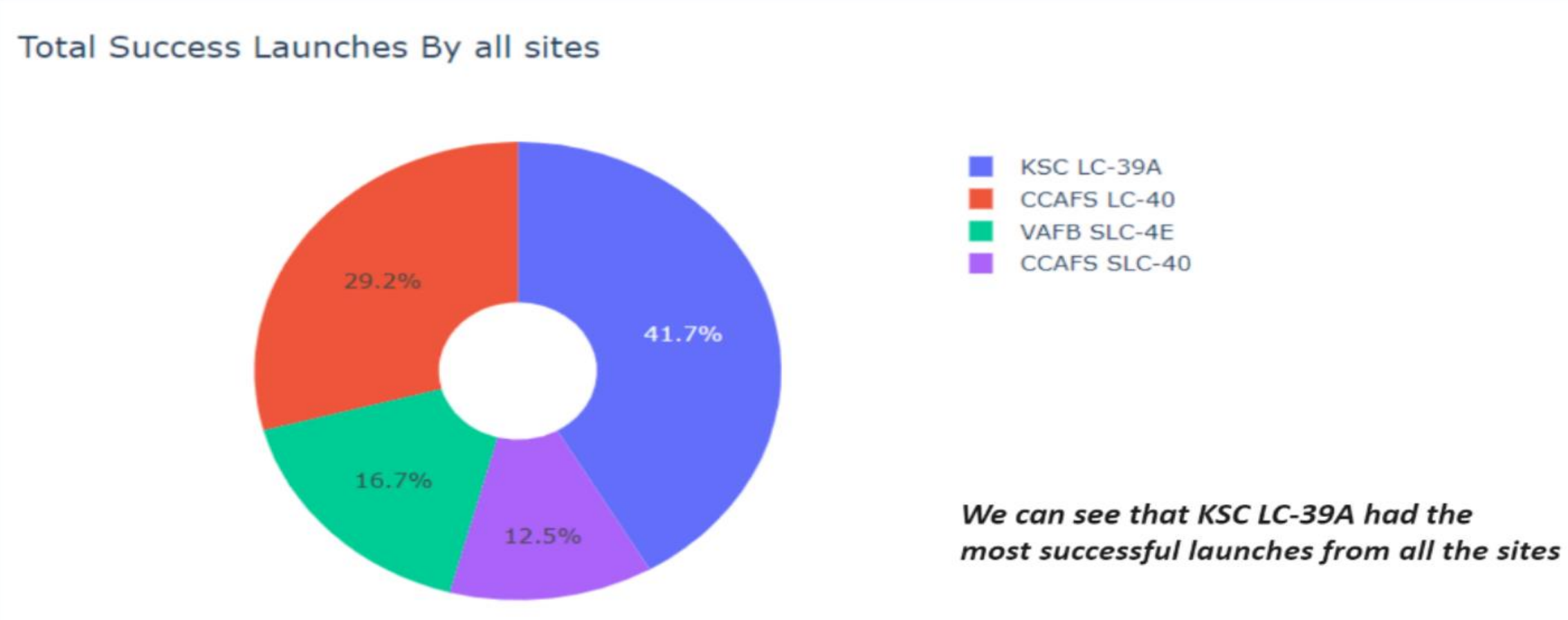
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

- distance_highway = 0.5834695366934144 km
- distance_railroad = 1.2845344718142522 km
- distance_city = 51.43416999517233 km

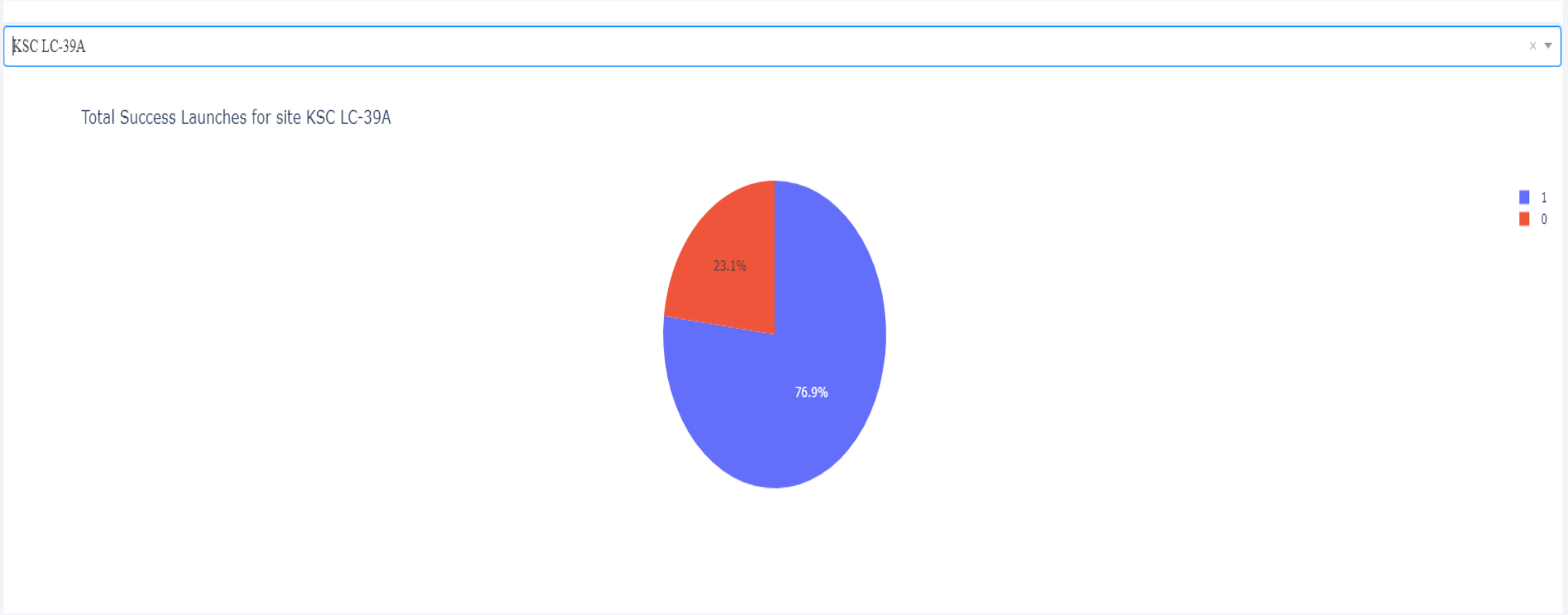

Section 4
Build a Dashboard with Plotly Dash

The Success Launches By all sites in a pie chart

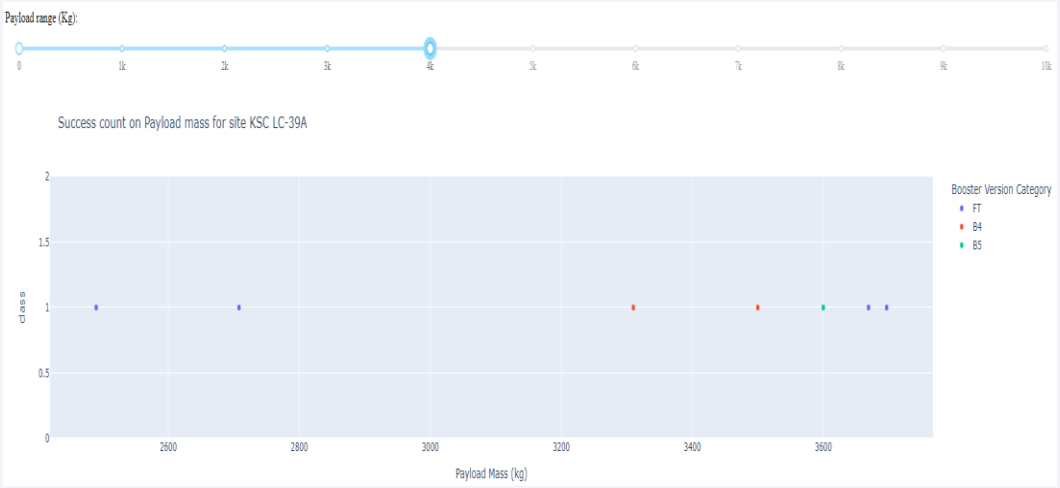


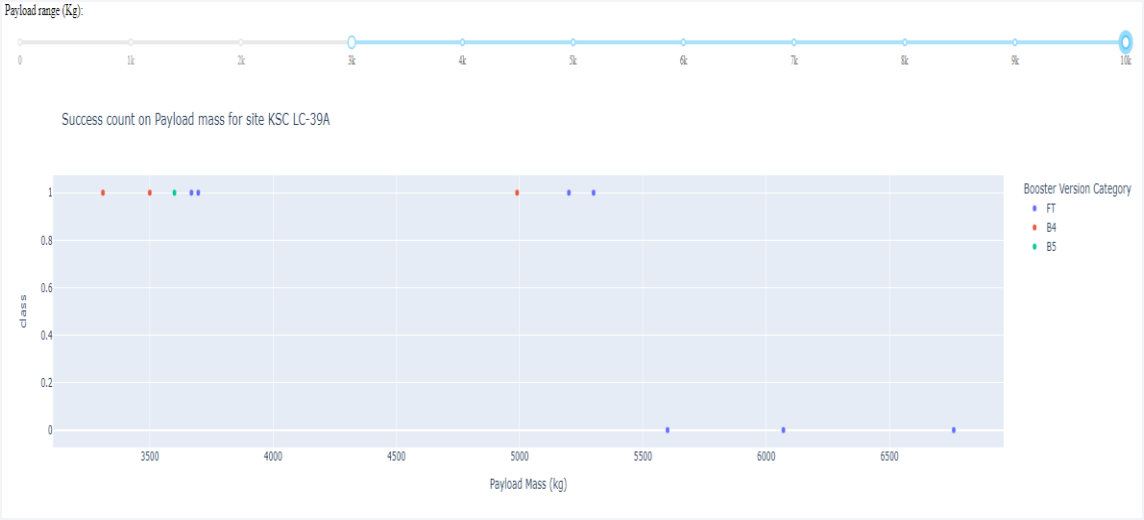
The Launch site with the highest launch success ratio in a pie chart

KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate



Scatter plot of Payload vs Launch outcome for all sites, with different payload selected in the range slider





Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy

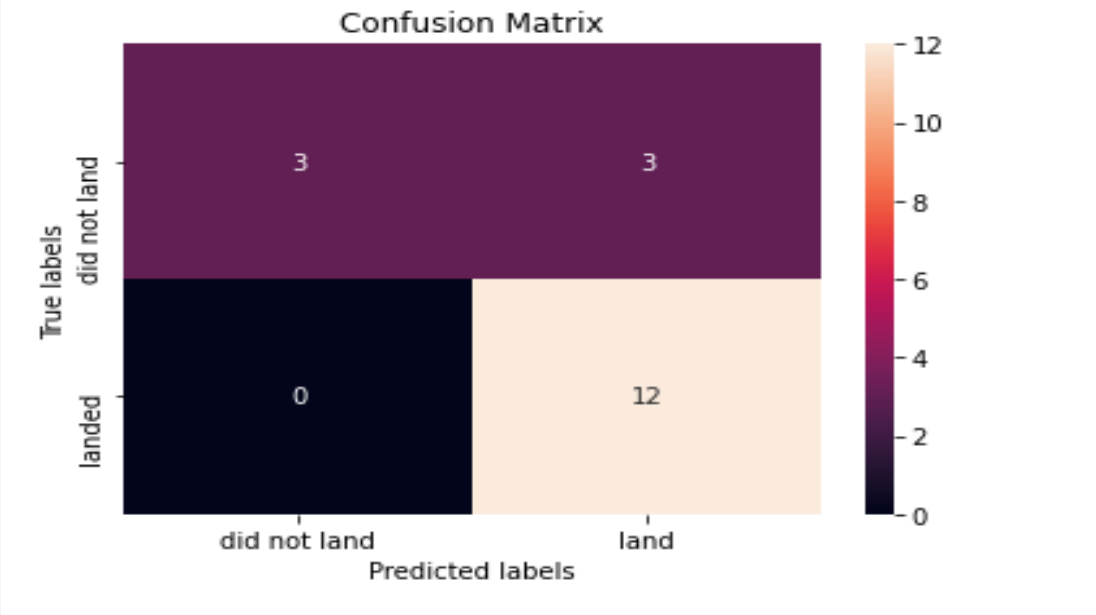
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

- Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives.



Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task

Thank you!