IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Chakshu Grover
21th January 2022

Outline

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

Executive Summary

- Hey, in this we are looking at all the tech-Stack, operations, and methodologies required to complete this case study of the SpaceX's Falcon Launch Data and Prediction.
- Week 1 had the Data wrangling and the preprocessing part where we basically web scraped the data, and prepared it for the further application
- Week 2 was all about the first stage of Exploratory Data Analysis and Visualization to understand the data; and successfully driven
- Week 3 was all about the dashboarding and creating an interactive window for the understanding of the stakeholders.
- Week 4 had the final prediction model which used the above data and extracted the best fitting algorithm for the case.

Introduction

- The user case study revolved around the extraction of the data of the landing of the SpaceX rovers.
- we were provided with the multiple landing factor like Booster Model, lander options, Launch site, payload. And others
- For the future Customers for the space, the problem is to find the optimal Landing site for a future customers to minimize the risk of crash of the boosters as well as payload.
- For this case, we use web scraping for the extraction of the data, available online.

Section 1

Methodology

Methodology

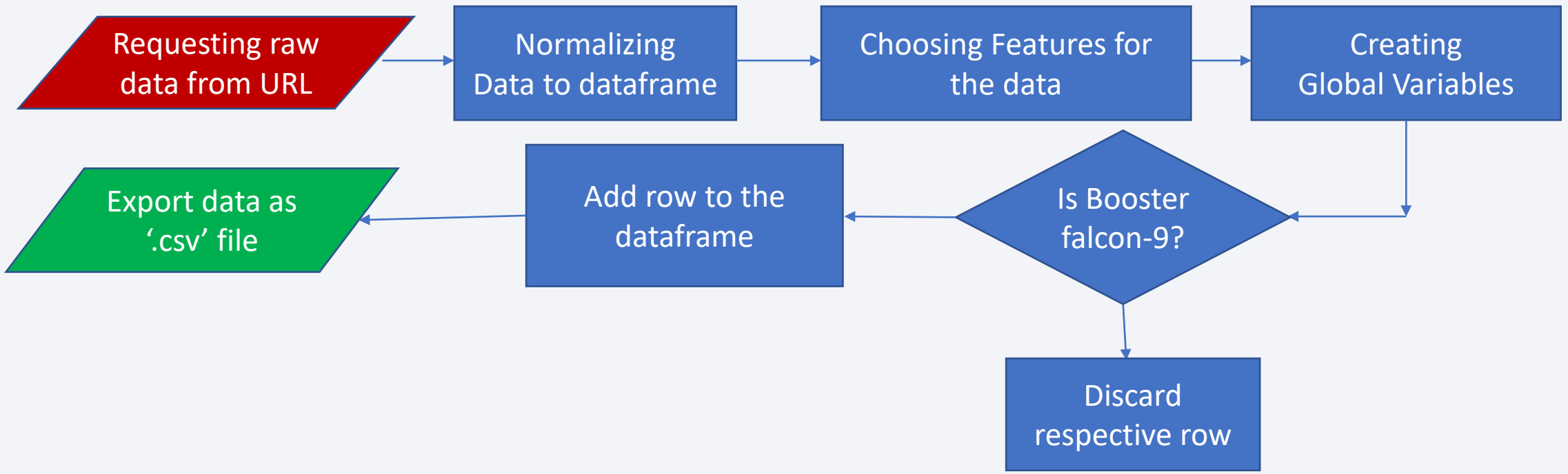
Executive Summary

- Data collection methodology:
  - Data was collected from Wikipedia data available open source and used pandas to convert the data into DataFrames.
- Perform data wrangling
  - We were interested in using only the ‘Falcon-9’ Booster, so we use filtering and extracted the only ‘Falcon-9’ data and chose only desired parameters.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We used GridSearchCV for the best parameters, and used LogisticRegression, KNN, DecisionTreeRegression and Support vector machine

Data Collection

- For the data collection, we use ‘requests’, ‘Pandas’, ‘numpy’ and ‘datetime’ libraries.
- The Tasks were:
  - 1. Requesting to the SpaceX API
  - 2. Save the requested data.
  - 3. Clean the requested Data
- The final form of data collected is a .csv file which contained the records of Booster named ‘Falcon-9’.

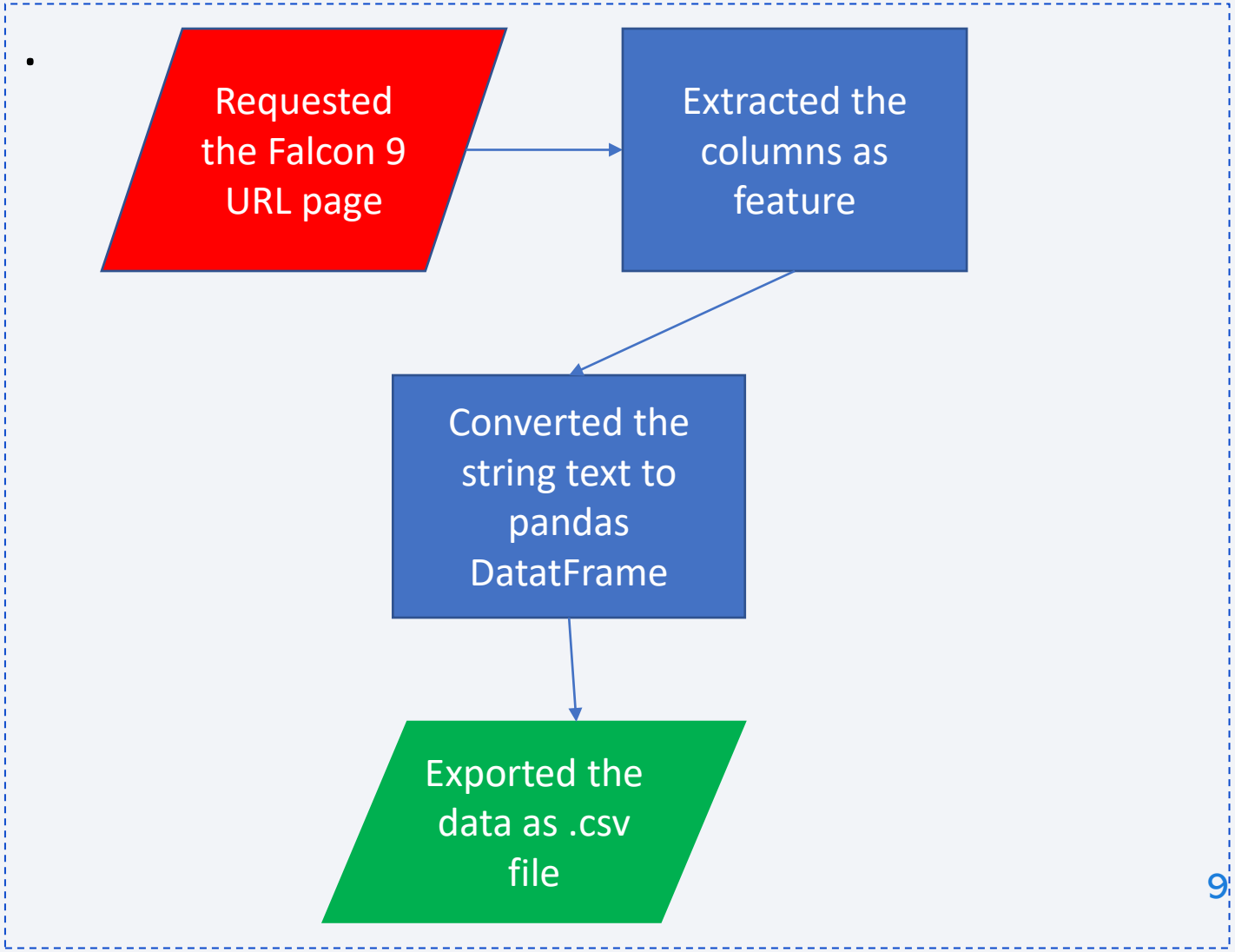
Data Collection – SpaceX API



- For the Notebook, visit: https://github.com/ChakshuGrover225/IBM_applied_capstone_project/blob/main/week%201%20-%20data%20collection%20and%20wrangling/1-1%20spacex%20data%20collection%20api.ipynb

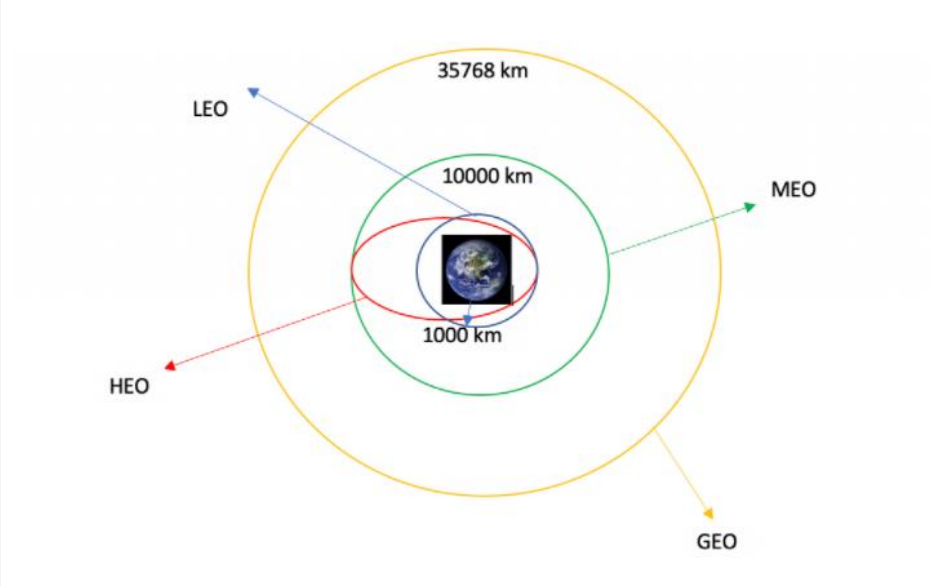
Data Collection - Scraping

- For web scraping, we used
  - Requests
  - BeautifulSoup
- Requested the Falcon9 launch from its URL.
- Extracted all Columns from HTML table header
- Create a data frame from Table
- Output- exported a .csv file for the scraped data.



Data Wrangling

- The basic EDA process is carried by the usual Numpy and pandas library.
- The Data is checked for the null values, and its type.
- Calculated count of each present Launch site.
- Calculated the number of occurrence of each orbit.
- Calculate the Outcome ratio for all orbits.
- Lastly, we created a landing Outcome label.



EDA with Data Visualization

- Here we visualized the relationship between Flight Number and Launch Site.
- Visualized the relationship between Payload and Launch Site
- Visualized the relationship between success rate of each orbit type and also other type of Visual Comparison between features.
- We carried out Feature Engineering and casted the type of data.
- Link for the Notebook: https://github.com/ChakshuGrover225/IBM_applied_capstone_project/blob/main/week%202%20-%20EDA%20and%20visualization/2-2%20eda%20dataviz.ipynb

EDA with SQL

We carried out another EDA using SQL and used the below Queries

- **Display the names of the unique launch sites in the space mission**
- **Display 5 records where launch sites begin with the string 'CCA'**
- **Display the total payload mass carried by boosters launched by NASA (CRS)**
- **Display average payload mass carried by booster version F9 v1.1**
- And others….
- https://github.com/ChakshuGrover225/IBM_applied_capstone_project/blob/main/week%202%20-%20EDA%20and%20visualization/2-1%20eda%20sql.ipynb

Build an Interactive Map with Folium

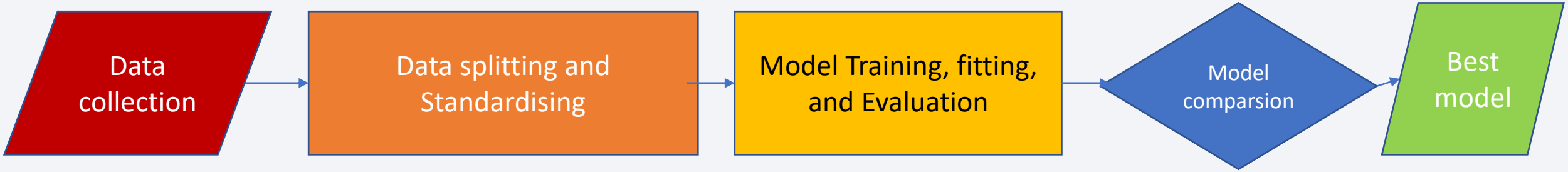
- In this lab, we tried to understand the geographical features of the Launch sites, which can determine the landings and maybe successful overall mission.
- We try to measure the shortest distance between the launch sites and the
  - Cities
  - Railroads
  - Highway
- URL :https://github.com/ChakshuGrover225/IBM_applied_capstone_project/blob/main/week%203%20-%20Dashboarding%20by%20Plotly/3-1%20launch_site_location%20dashboarding.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

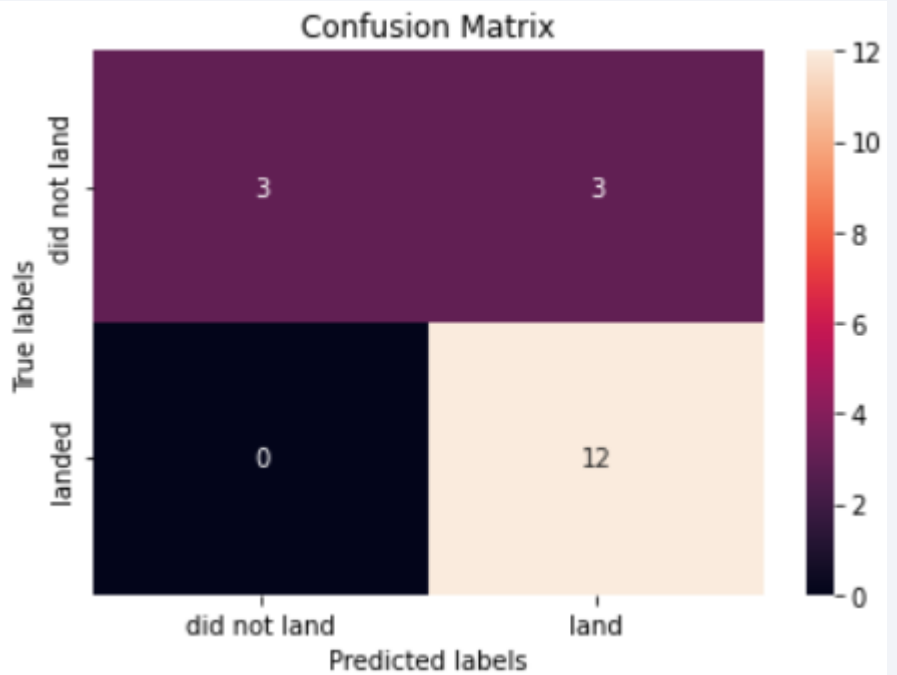
- We built a predictive analysis model that will determine the Outcome of a launch by learning the previous experiences.
- For the model, we use mathematical and statistical models like Decision Tree, K nearest neighbor classifier, Logistic Regression and Support vector Machine

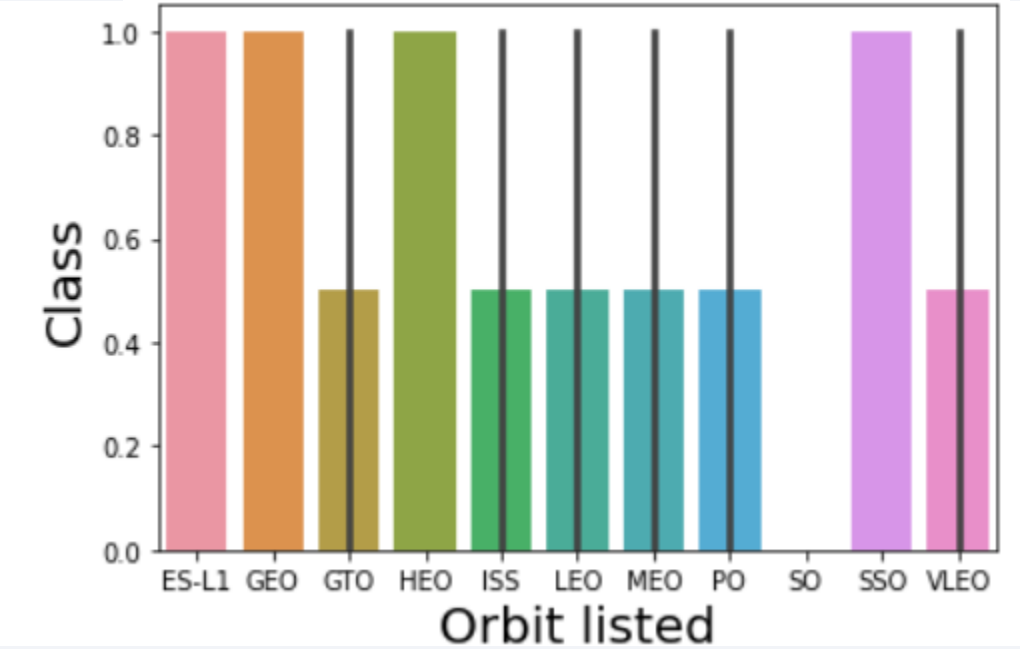


GitHub URL:
https://github.com/ChakshuGrover225/IBM_applied_capstone_project/blob/main/week%204%20-%20Predictive%20Modelling/4-1%20SpaceX%20ML%20Prediction.ipynb

Results

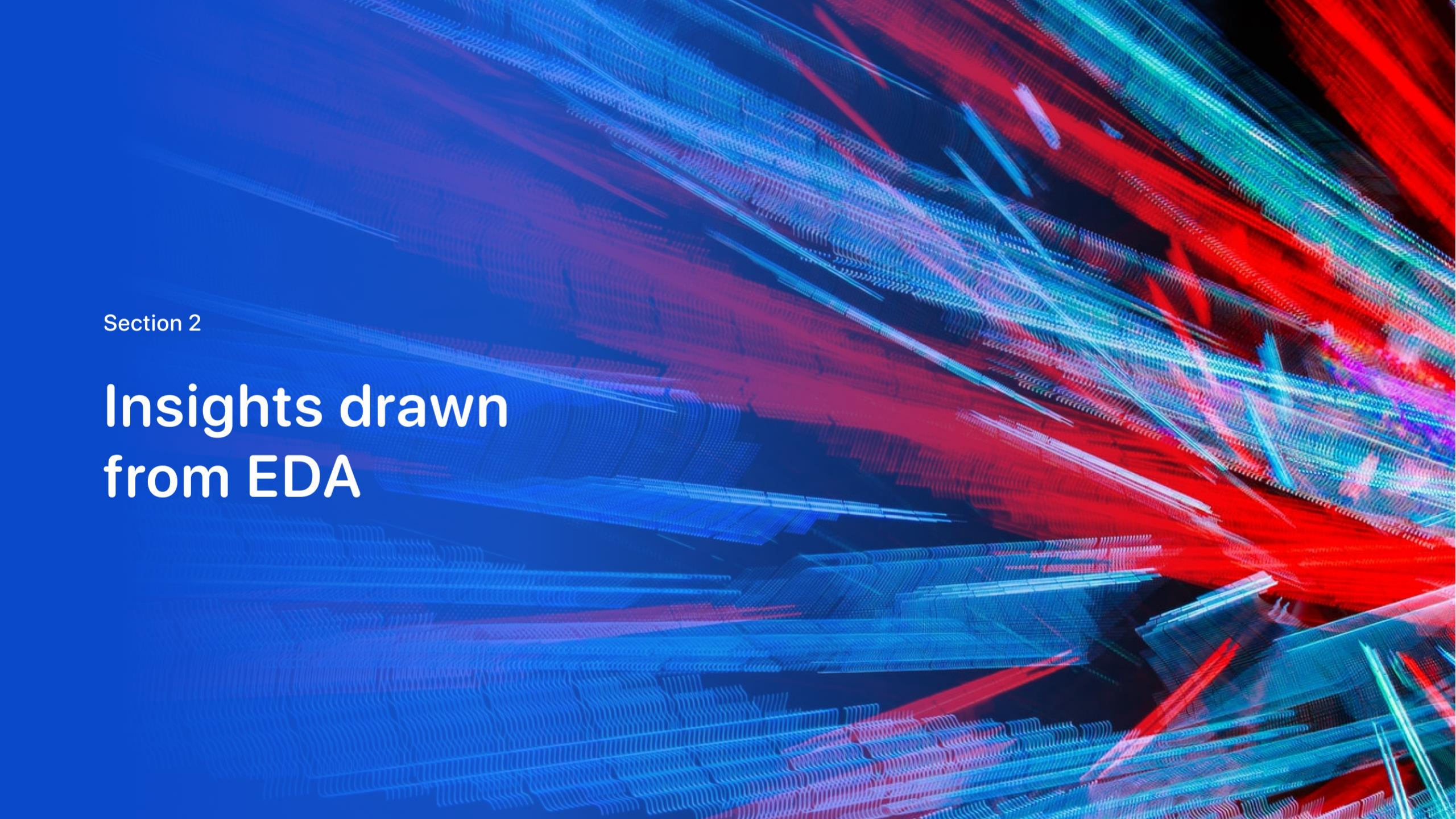
- Exploratory data analysis results
  - The data was messy.
  - The CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%
- Interactive analytics demo in screenshots
  - 1st: Confusion matrix for the Decision Tree
  - 2nd: Bar chart comparing Landing Class and Orbit.
- Predictive analysis results
  - Decision Tree is the best performing model in the predictive phase.





Section 2

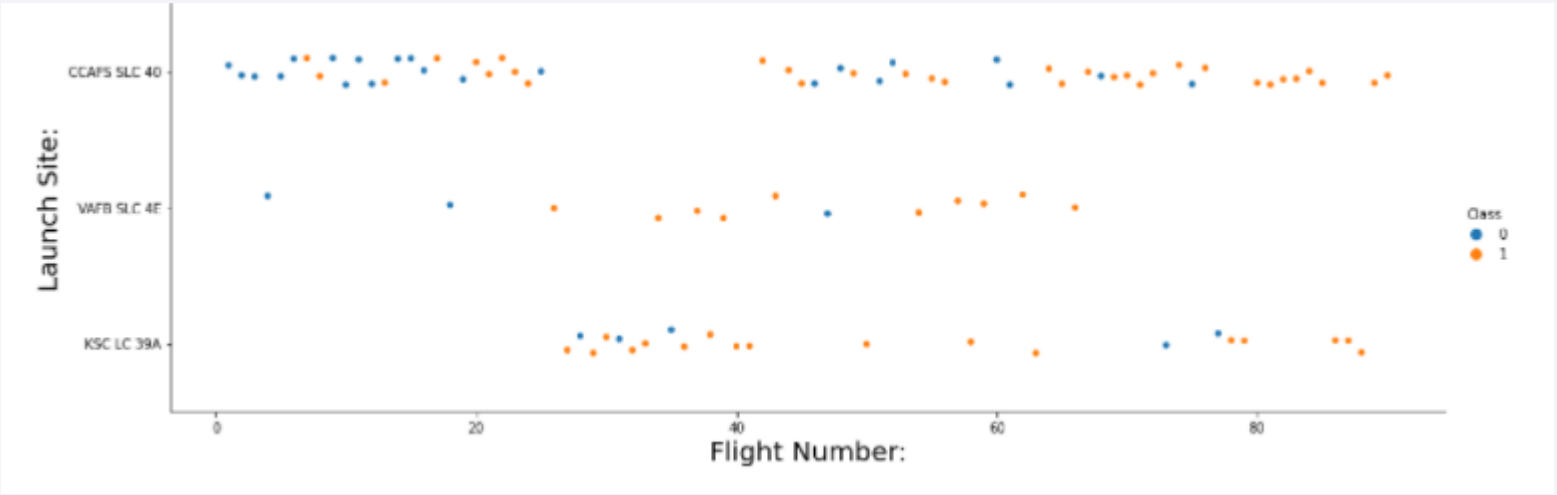
Insights drawn from EDA



Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site
- Show the screenshot of the scatter plot with explanations

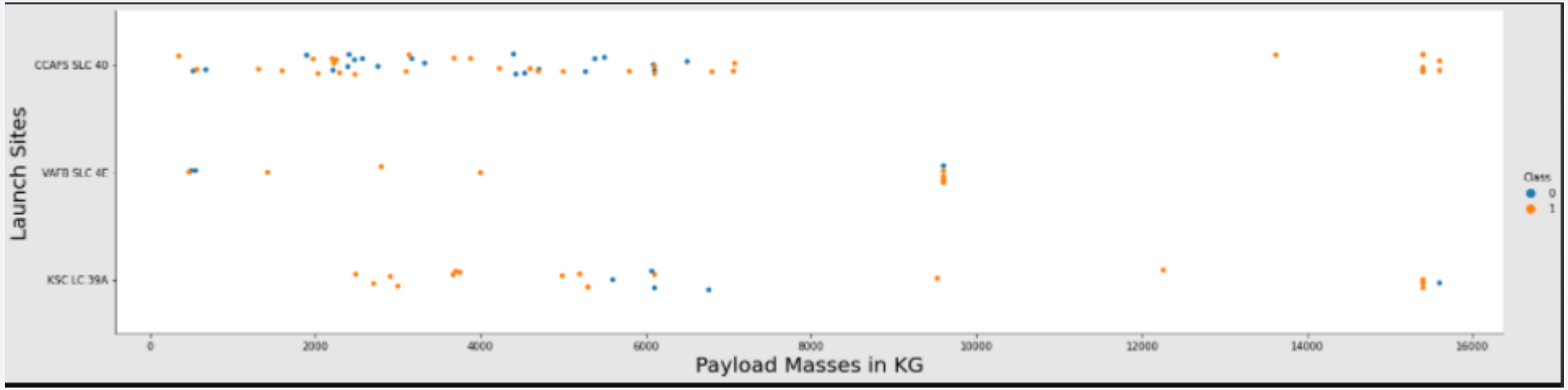
```
sns.catplot(x='FlightNumber',
            y='LaunchSite',
            hue='Class',
            data=df,
            aspect=3)

plt.xlabel('Flight Number:',fontsize=20)
plt.ylabel(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

```
sns.catplot(x='PayloadMass',
            y='LaunchSite',
            data=df,
            hue='Class',
            aspect=4)

plt.xlabel("Payload Masses in KG",fontsize=20)
plt.ylabel("Launch Sites",fontsize=20)
plt.show()
```

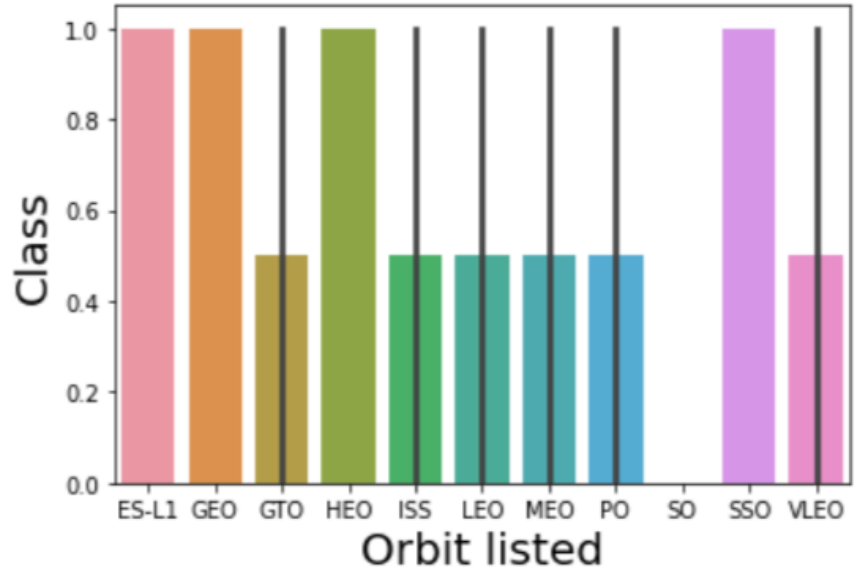


Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type

- Show the screenshot of the scatter plot with explanations

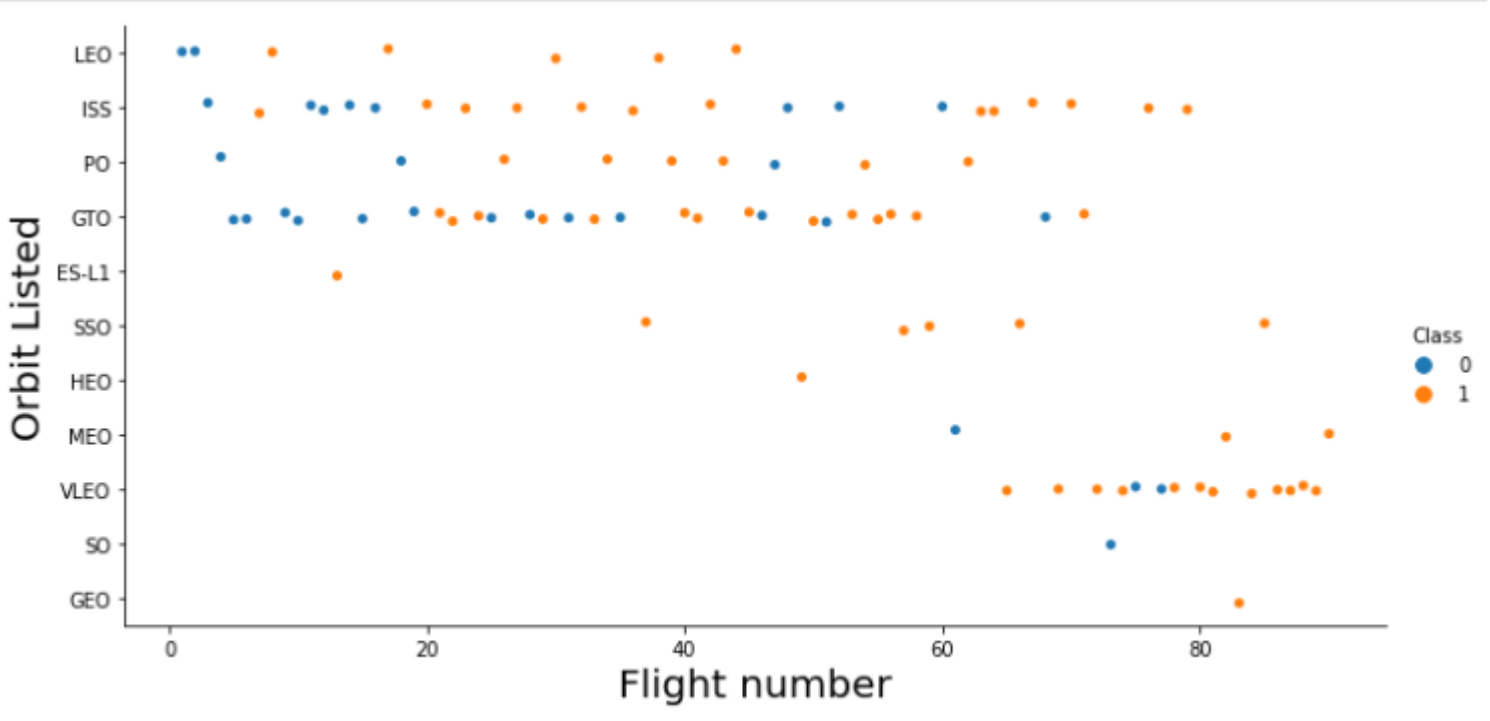
```
orbitData=df.groupby(['Orbit', 'Class'])['Class'].agg(['mean']).reset_index()
sns.barplot(x='Orbit',
            y='Class',
            data=orbitData)
plt.xlabel("Orbit listed",fontsize=20)
plt.ylabel("Class",fontsize=20)

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

```
sns.catplot(x='FlightNumber',
            y='Orbit',
            data=df,
            hue='Class',
            aspect=2)
plt.xlabel("Flight number",fontsize=20)
plt.ylabel("Orbit Listed", fontsize=20)
plt.show()
```

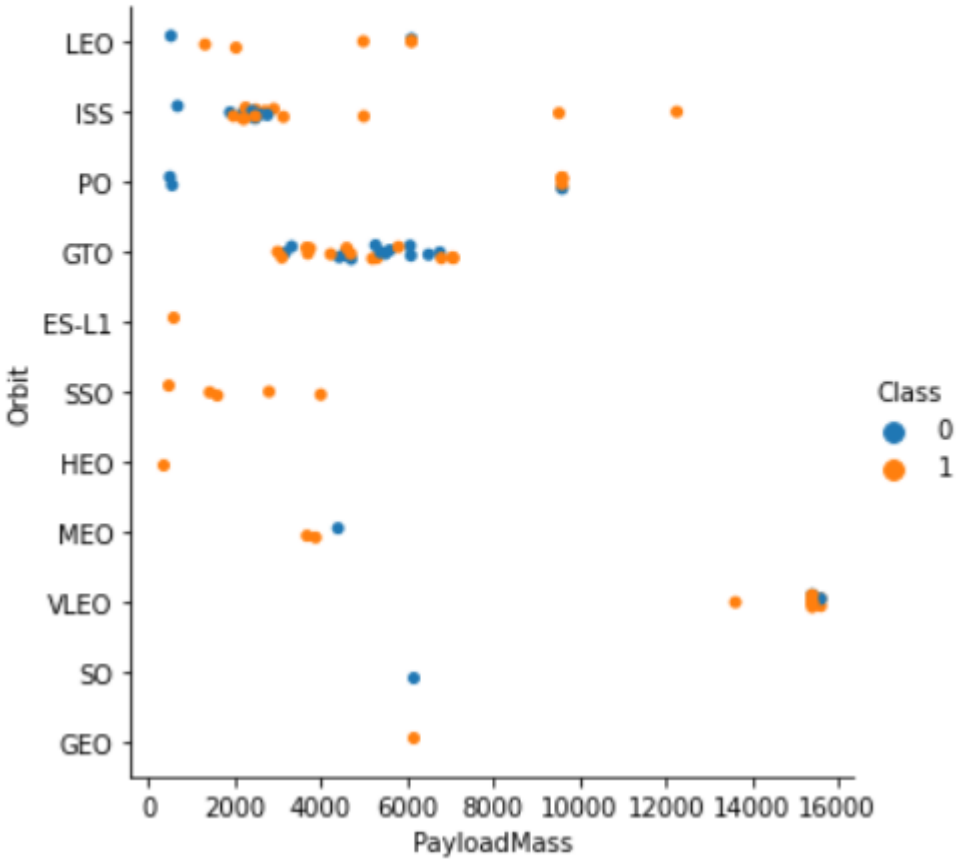


Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

- Show the screenshot of the scatter plot with explanations

```
sns.catplot(x='PayloadMass',
            y='Orbit',
            hue='Class',
            data=df
            )

plt.show()
```



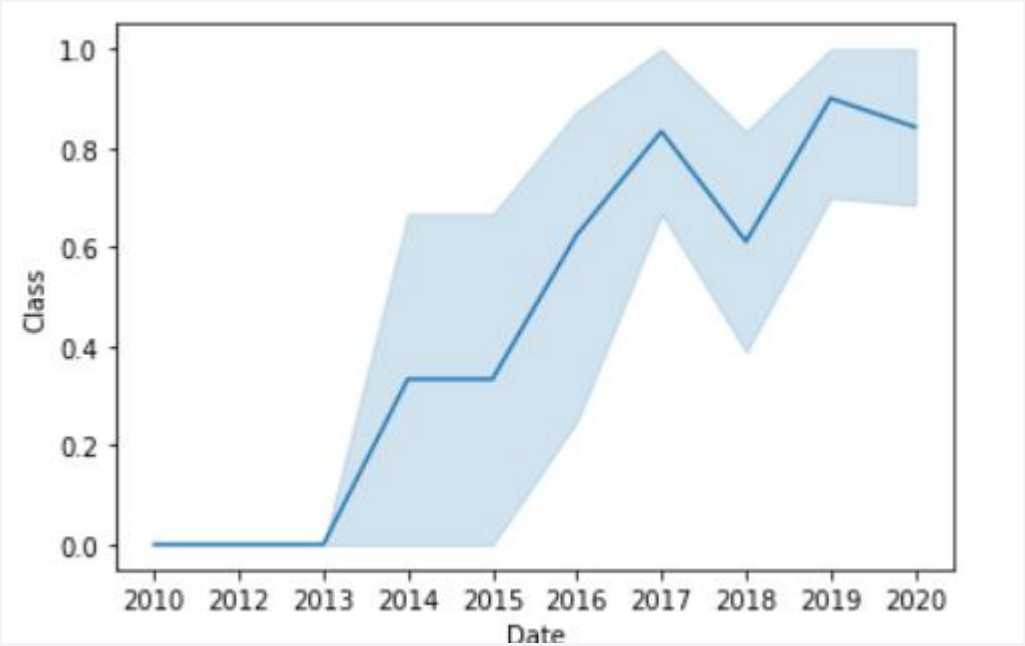
Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations

```
# A function to Extract years from the date

def Extract_year():
    for i in df["Date"]:
        year.append(i.split("-")[0])
    return year
```

```
year=[]
df1 = df.copy()
year = Extract_year()
df1["Date"] = year
df1.head()
```



All Launch Site Names

- Find the names of the unique launch sites

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

- Th distinct keyword in SQL present the common values out of all the record.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

- This query take ‘%’ as any characters and ‘CCA’ denoted means we need to see only the first characters to be like this, irrespective of what is on the other side.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

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 SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

```
%sql select min(DATE) from SPACEXTBL where Landing__Outcome = 'Success (ground pad)'
```

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 SPACEXTBL
where Landing__Outcome = 'Success (drone ship)' and
PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight)'
```

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql SELECT EXTRACT(MONTH, select min(DATE) from SPACEXTBL where Landing__Outcome = 'Success (ground pad)')
```

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

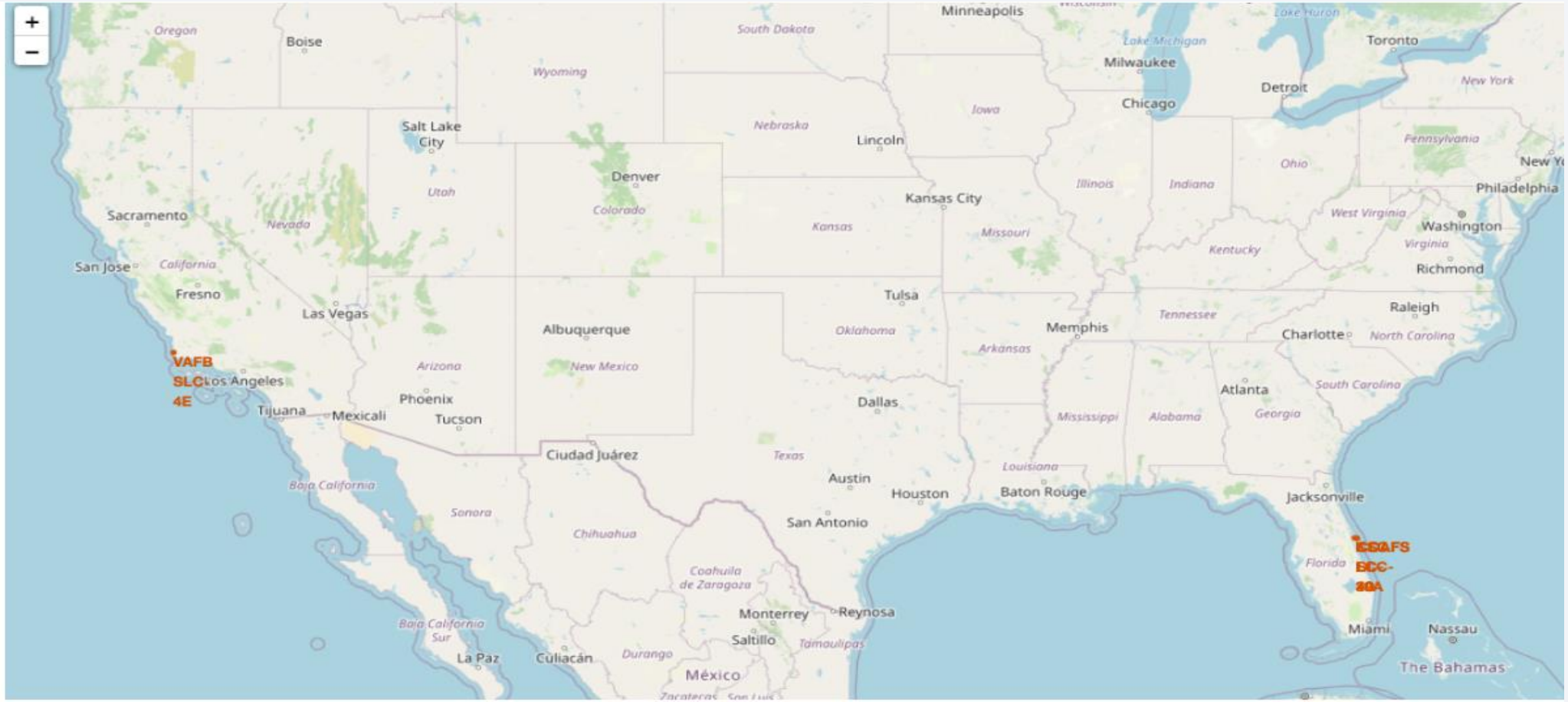
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

```
%sql select * from SPACEXTBL where Landing__Outcome like 'Success%' and (DATE between '2010-06-04' and '2017-03-20') order by date desc
```

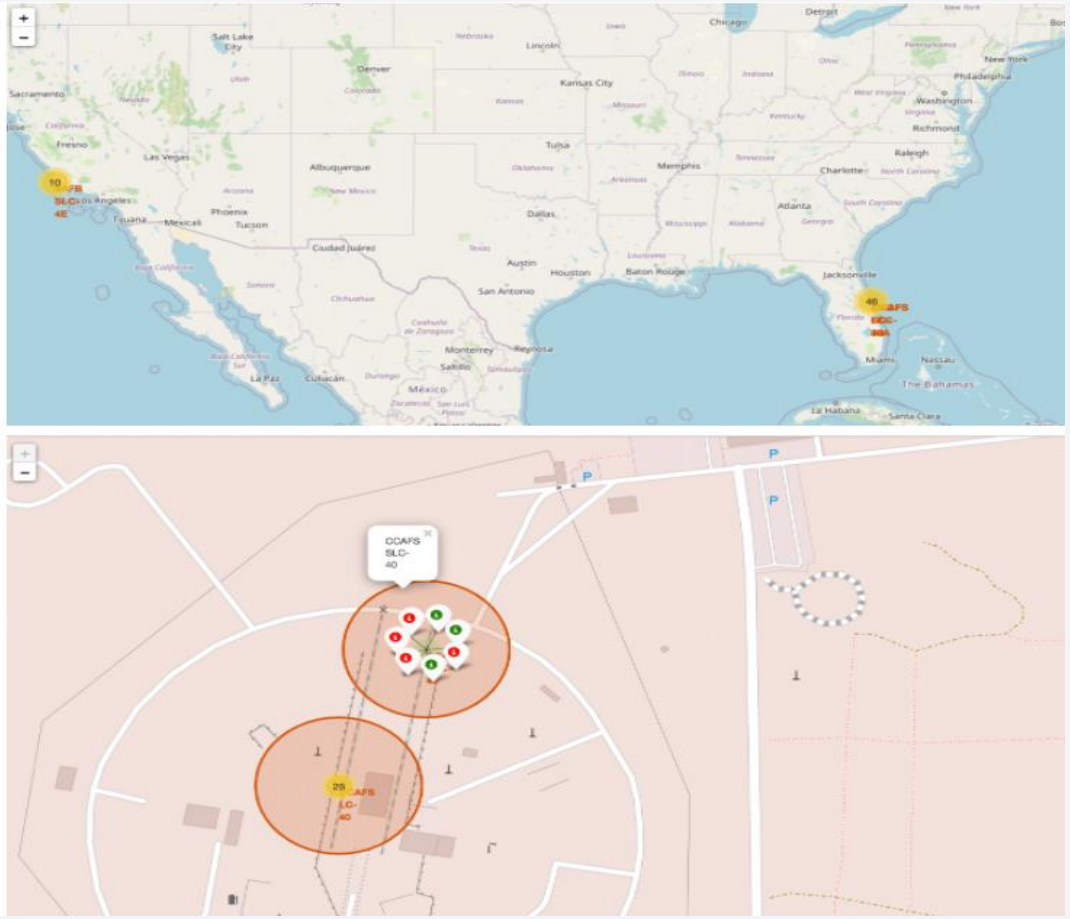
Section 4

Launch Sites Proximities Analysis

Mark all launch sites on a map



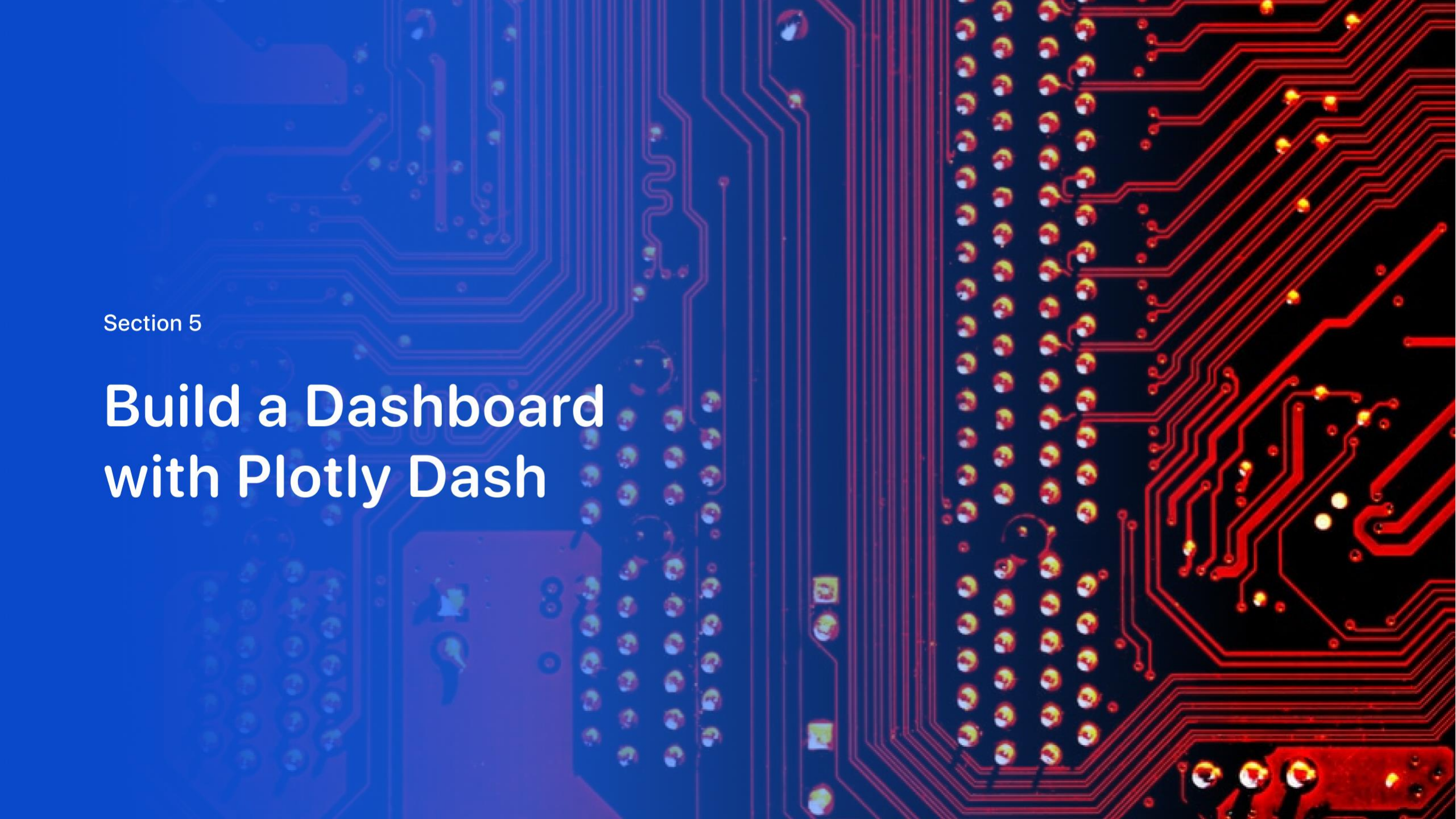
Mark the success/failed launches for each site on the map



Calculate the distances between a launch site to its proximities

Section 5

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 6

Predictive Analysis (Classification)

Classification Accuracy

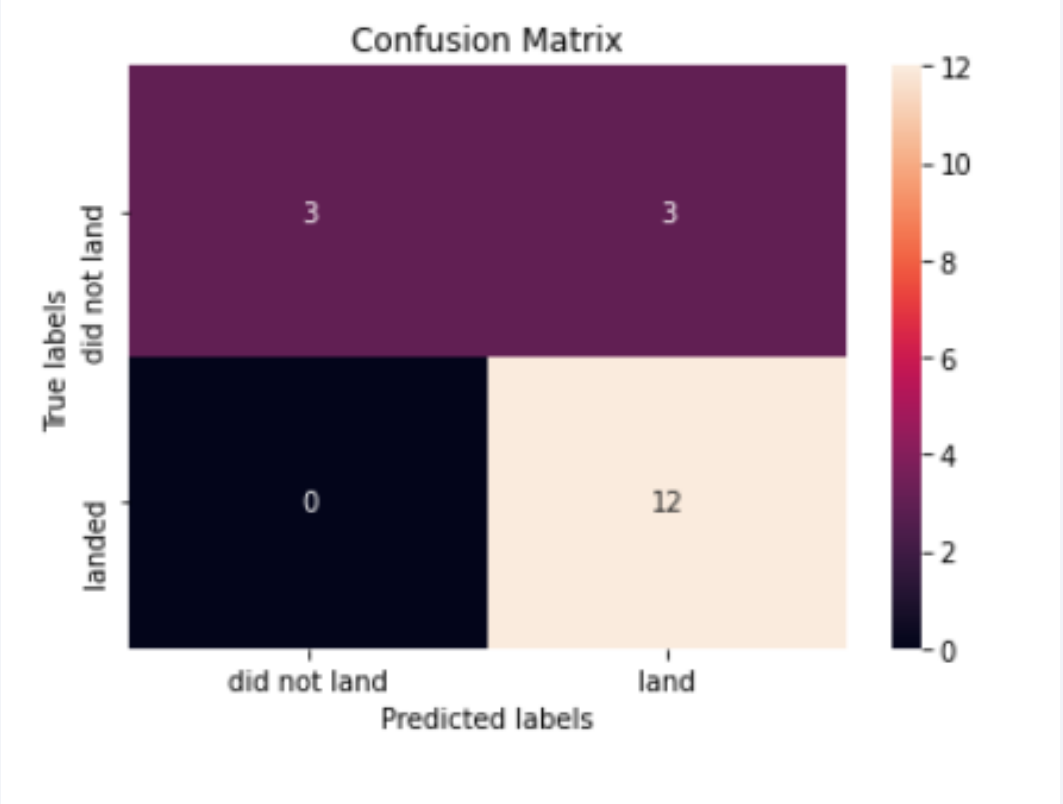
- Visualize the built model accuracy for all built classification models, in a bar chart
- Find which model has the highest classification accuracy


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

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_feature
s': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 5, 'splitter': 'best'}
accuracy : 0.8714285714285713
```

Confusion Matrix

- The Confusion matrix is showing the possible error values which a model can give since the model is not 100 percent accurate with the results



Conclusions

- The Landing depends on the Boosters types, size, orbit, and other valuable metrics.
- The Machine Learning can be quite helpful in the prediction of failure of project with the future customers.
- Data science and analytics can be highly efficient in value finding of each Launching site and metric
- This analyses can may further tell us the most suitable booster for any specific type of orbit or payload.
- …

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!