IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Author: Vaseem Raja
Date: 10-Sep-2024

Outline

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

Executive Summary

- Goal is to use the SpaceX data and predict whether SpaceY will attempt to land a rocket successfully or will result in a failure.
- Decision tree classifier yields the best accuracy score of 0.375

Introduction

Project background and context

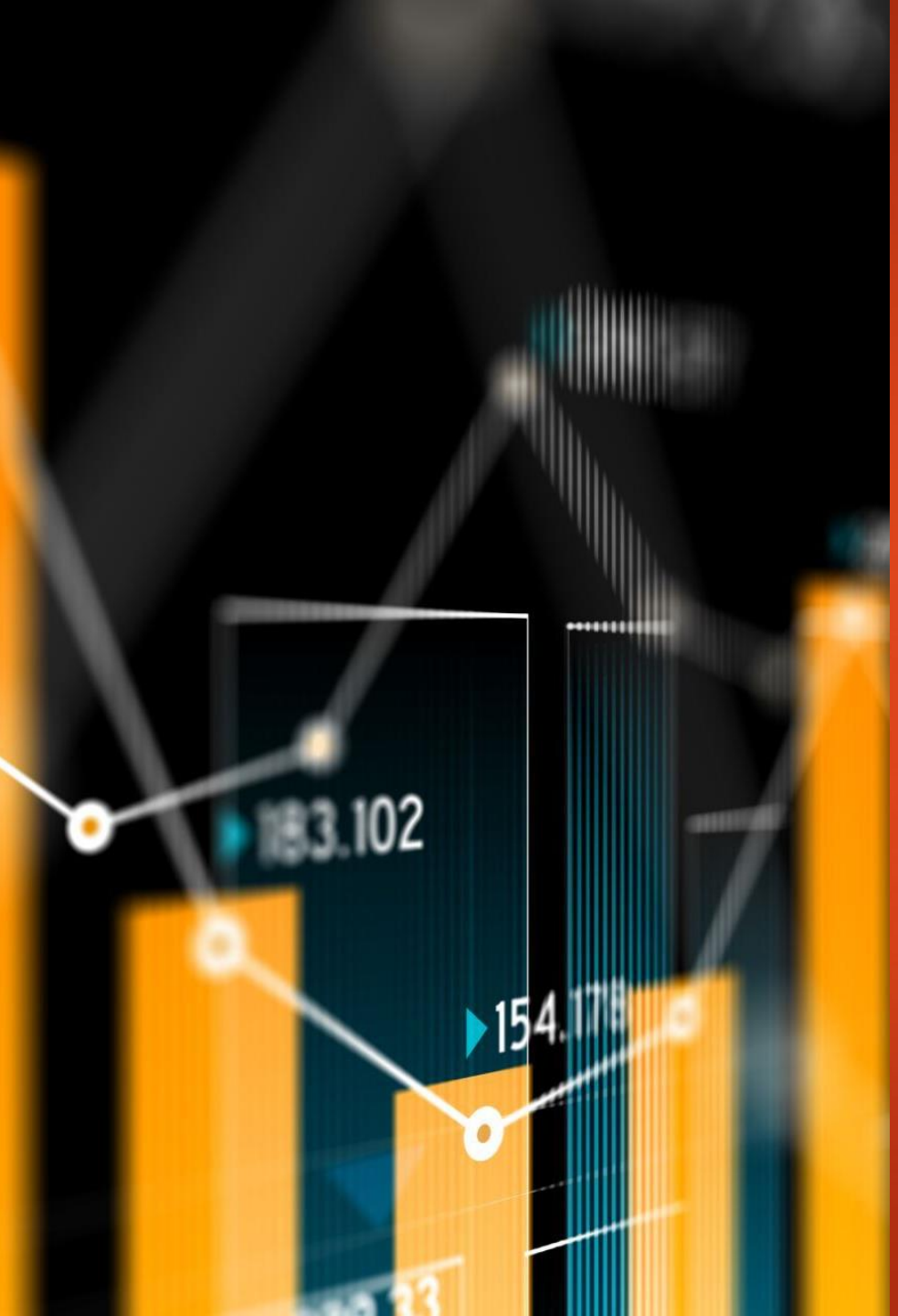
- The project objective is to work with SpaceX launch data that is gathered from an API, specifically the SpaceX REST API, which provides the information about the Falcon 9 rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- Another popular data source for obtaining the launch data is through web scraping from the related Wiki pages.

Problems you want to find answers

- The goal is to use this data to predict whether SpaceX will attempt to land a rocket successfully or will result in a failure.

Section 1

Methodology



Methodology

- Executive Summary
- Data collection methodology:
  - SpaceX REST API calls
  - Web scraping Wiki pages
- Perform data wrangling
  - Sampling the data and dealing with Nulls
- Performed exploratory data analysis (EDA) using visualization and SQL
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models

Data Collection

- Data sets are collected through an SpaceX REST API call request which will provide the data to determine if the Falcon 9 first stage landed successfully, so that the cost of a launch can be optimized.
- Historical launch records are collected using the web scraping technique from a Wikipedia page titled "List of Falcon 9 and Falcon Heavy launches".

Data Collection – SpaceX API

- SpaceX launch data is gathered from an API, specifically the SpaceX REST API, which will give us data about launches, including the information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome and the results can be viewed by calling the .json() method.
- SpaceX REST API endpoints used are api.spacexdata.com/v4/launches/past
- GitHub URL:
- https://github.com/VaseemRJ/Space-Y/blob/main/SpaceX%20Falcon%209%20-%20Collecting%20the%20data.ipynb

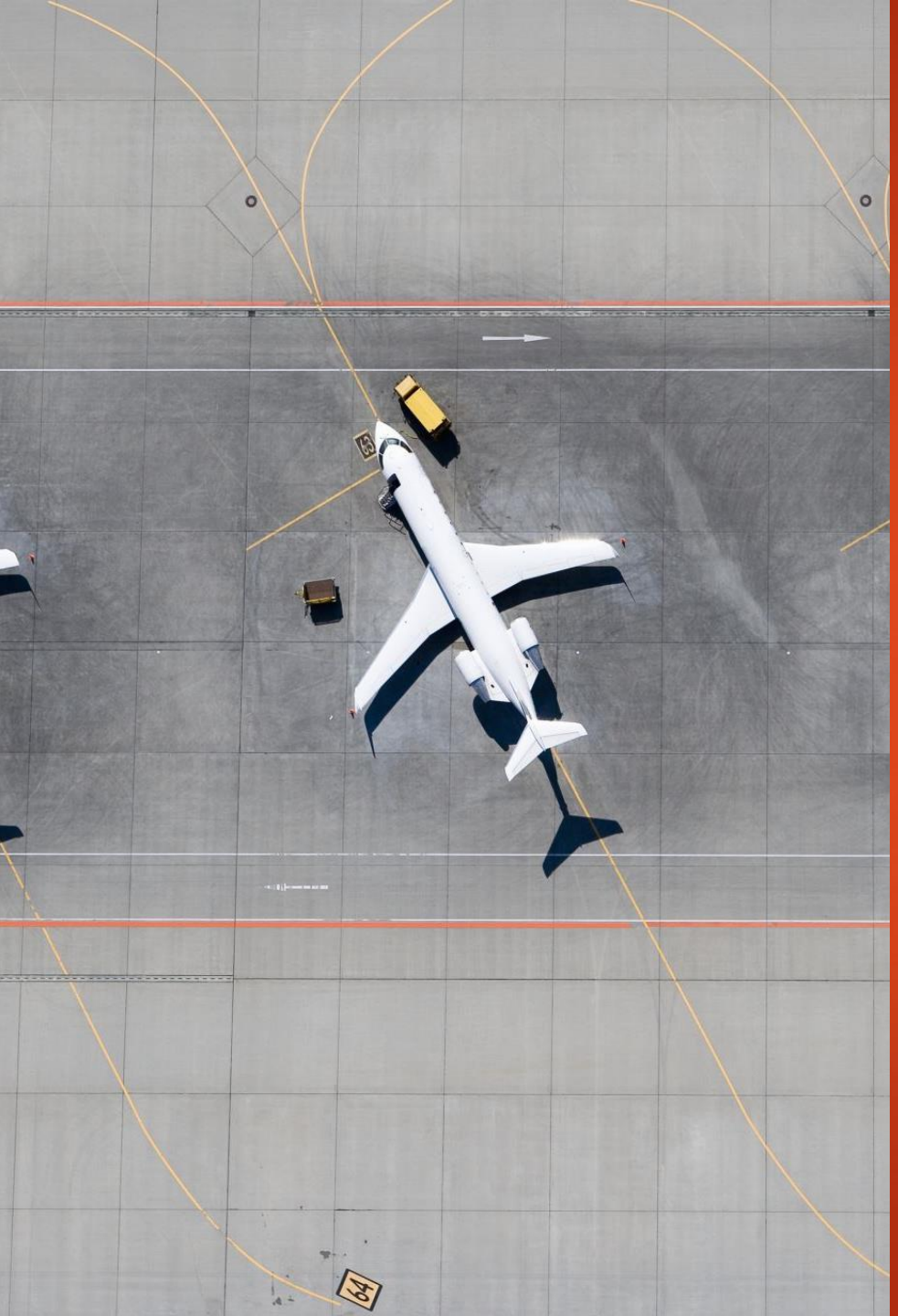
Data Collection – Web Scraping

Web scraping technique is used to scrape the related Wiki pages to obtain Falcon 9 Launch records, and to parse the data from the tables in the sites and convert them into a Pandas data frame for further visualization and analysis.

Web scraping is done using BeautifulSoup method

GitHub URL:

https://github.com/VaseemRJ/Space-Y/blob/main/SpaceX%20Falcon%209%20-%20Web%20Scrapping.ipynb



Data Wrangling

- Data Wrangling is done using an API, Sampling Data, and Dealing with Nulls
- Some of the attributes like Flight Number, Date, Booster version, Payload mass Orbit, Launch Site, Outcome are reviewed for the status of the first stage Flights, and Grid Fins are reviewed for landing Reused, Legs are reviewed for Landing pad, Block, Reused count, Serial, Longitude and latitude of launch.
- GitHub URL:
- https://github.com/VaseemRJ/Space-Y/blob/main/SpaceX%20Falcon%209%20-%20Data%20Wrangling.ipynb

EDA with Data Visualization

- Scatter plot: FlightNumber vs. PayloadMassand, to overlay the outcome of the launch
- Scatter plot: Flight Number vs. Launch Site, to find the patterns
- Bar chart: Class vs. Orbit, to visualize if there is any relationship between success rate and orbit type
- Scatter plot: FlightNumber vs.Orbit type, to see if there is any relationship between FlightNumber and Orbit type
- Line chart: Year vs. Class, to observe that the sucess rate since 2013 kept increasing till 2020
- Summarize what charts were plotted and why you used those charts
- GitHub URL:
- https://github.com/VaseemRJ/Space-Y/blob/main/SpaceX%20Falcon%209%20-%20Exploring%20and%20Preparing%20Data.ipynb

EDA with SQL

- Created dummy variables for categorical columns
- Casted all numeric columns to float64
- GitHub URL:
- https://github.com/VaseemRJ/Space-Y/blob/main/SQL%20Notebook%20for%20Peer%20Assignment.ipynb

Build an Interactive Map with Folium

- Map objects that were used are ---
- Folium.Marker, MarkerCluster object, MousePosition, PolyLine
- The objects were added to --
  - Understand if all launch sites in proximity to the Equator line
  - Understand if all launch sites in very close proximity to the coast
- GitHub URL:
- https://github.com/VaseemRJ/Space-Y/blob/main/SpaceX%20Falcon%209%20-%20Visual%20Analytics.ipynb

Build a Dashboard with Plotly Dash

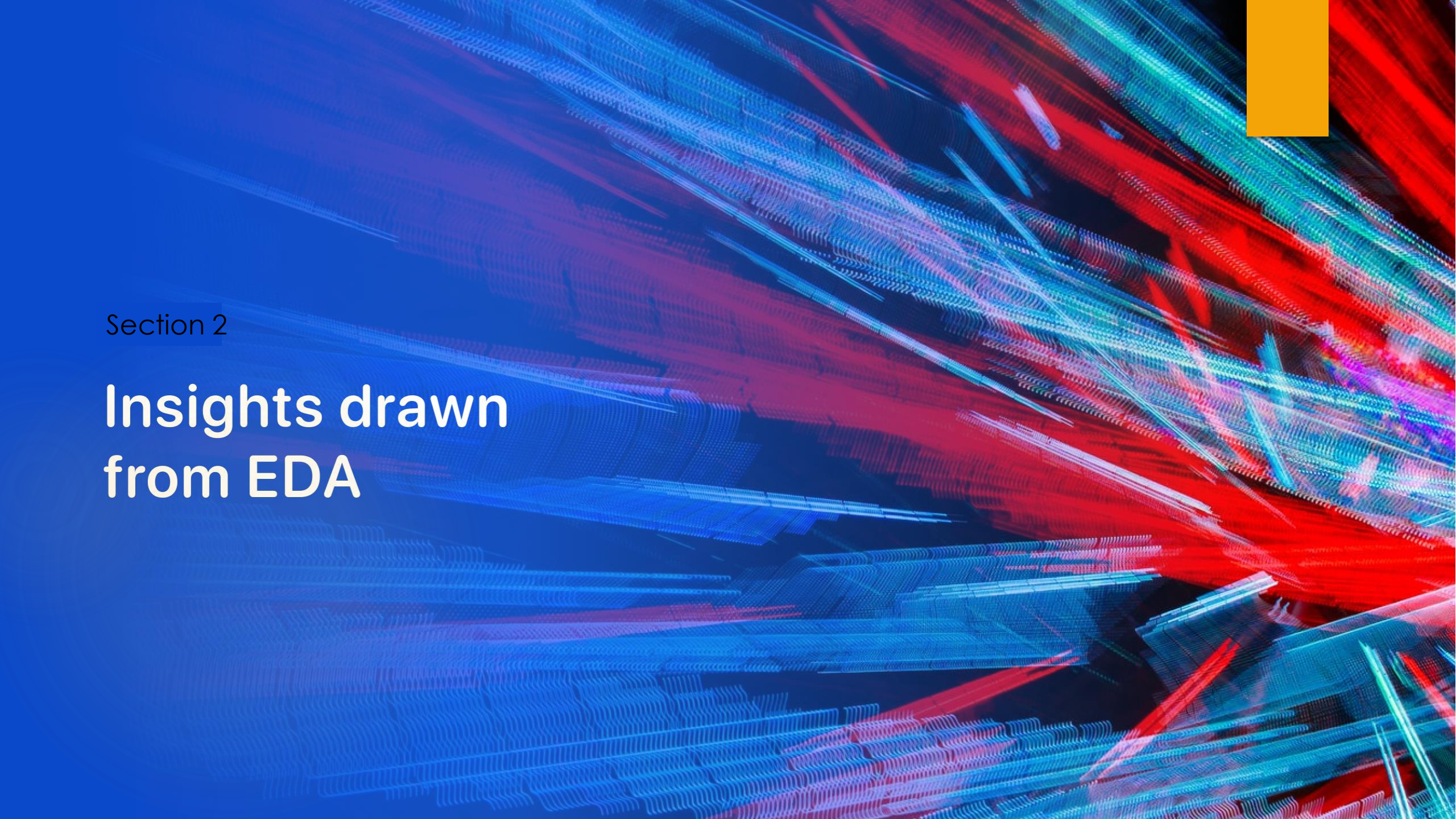
- The dashboard application contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart.
- Dashboard highlights some of the insights like:
- Sites that have the largest successful launches
- Sites that have the highest launch success rate
- Payload range(s) that have the highest launch success rate
- Payload range(s) that have the lowest launch success rate
- F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) that have the highest launch success rate
- GitHub URL: https://github.com/VaseemRJ/Space-Y/blob/main/SpaceX_Falcon_9_Dashboard_application.py

Predictive Analysis (Classification)

- Performed the exploratory Data Analysis and determined the Training Labels
  - Create a column for the class
  - Standardized the data
  - Split the data into training data and test data
- Best Hyperparameter was evaluated using SVM, Classification Trees and Logistic Regression
- Classification Trees Method performs best using test data
- GitHub URL: https://github.com/VaseemRJ/Space-Y/blob/main/SpaceX%20Falcon%209%20ML%20Prediction.ipynb

Results

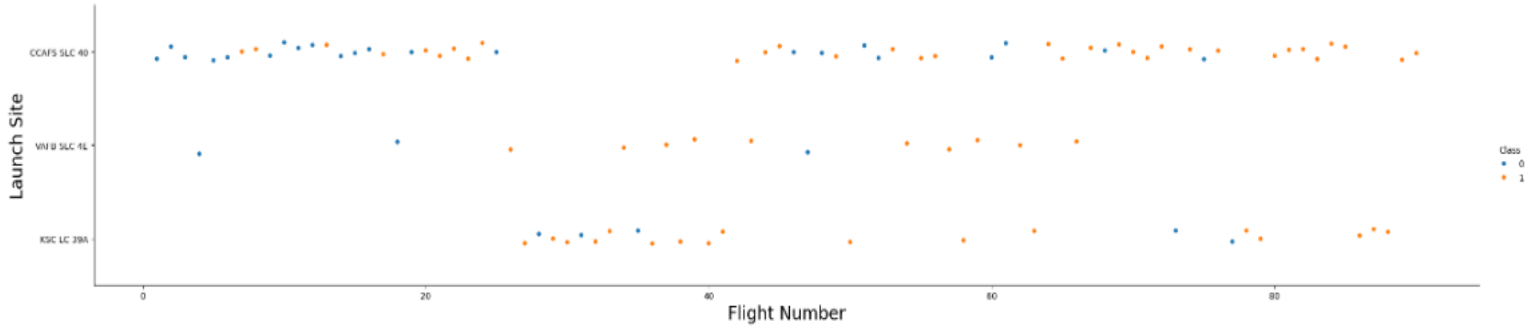
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- Scatter plot: Flight Number vs. Launch Sites, showing the patterns found in the scatter point plots.

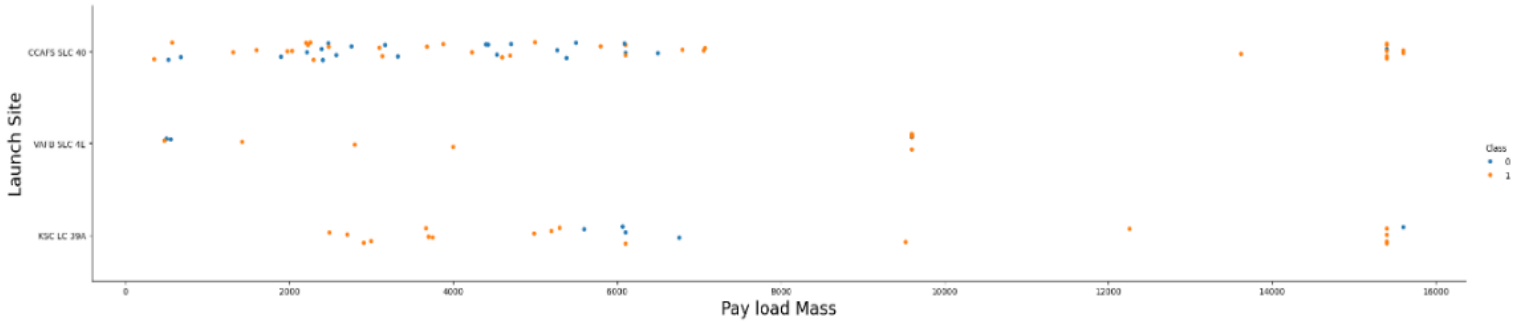
```
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Payload vs. Launch Site

- Scatter plot: Payload vs. Launch Site, to observe if there is any relationship between launch sites and their payload mass.
- Observation: It is noticed that in the VAFB-SLC launchsite, there are no rockets launched for heavypayload mass (greater than 10000)

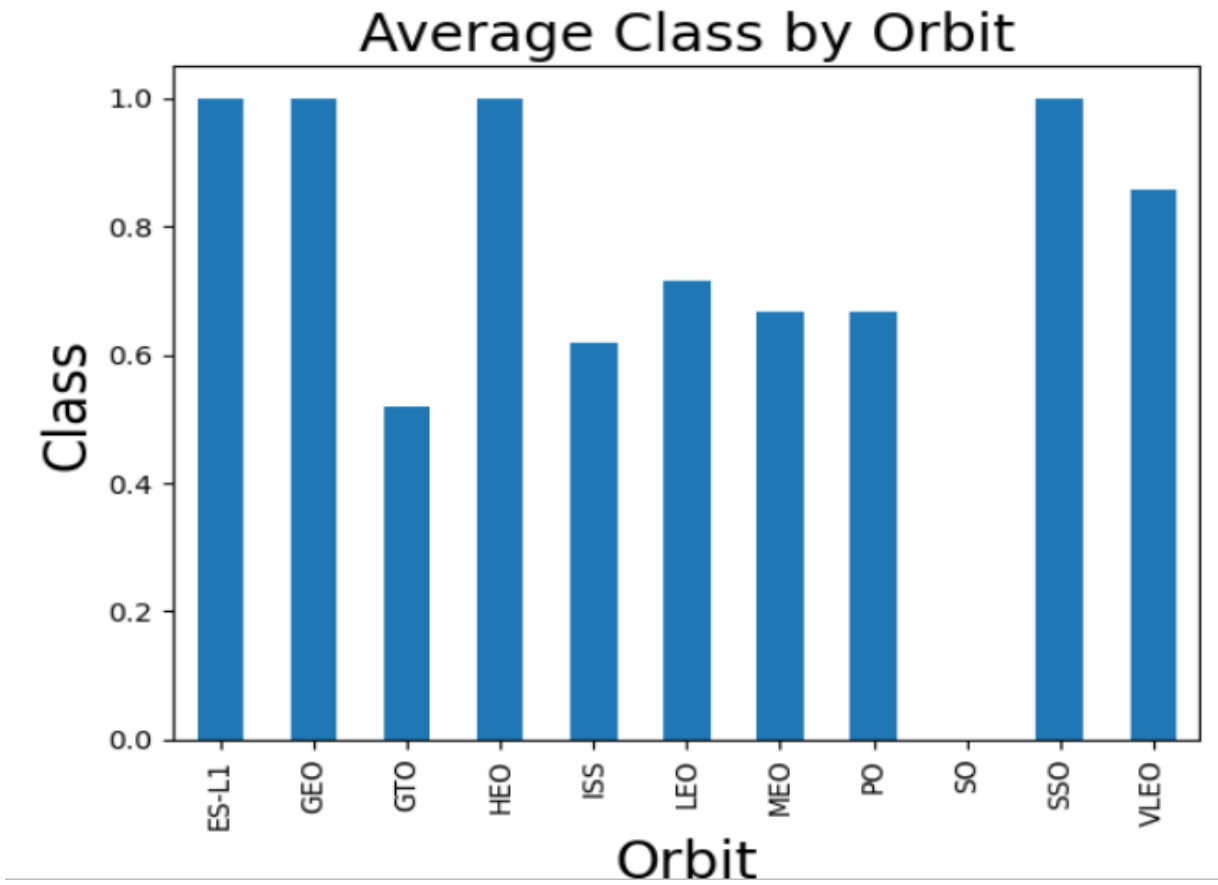
```
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Success Rate vs. Orbit Type

- Bar chart: Success Rate vs. Orbit Type, to visualize if there is any relationship between success rate and orbit type.
- It is also to identify which orbits have the highest success rates.

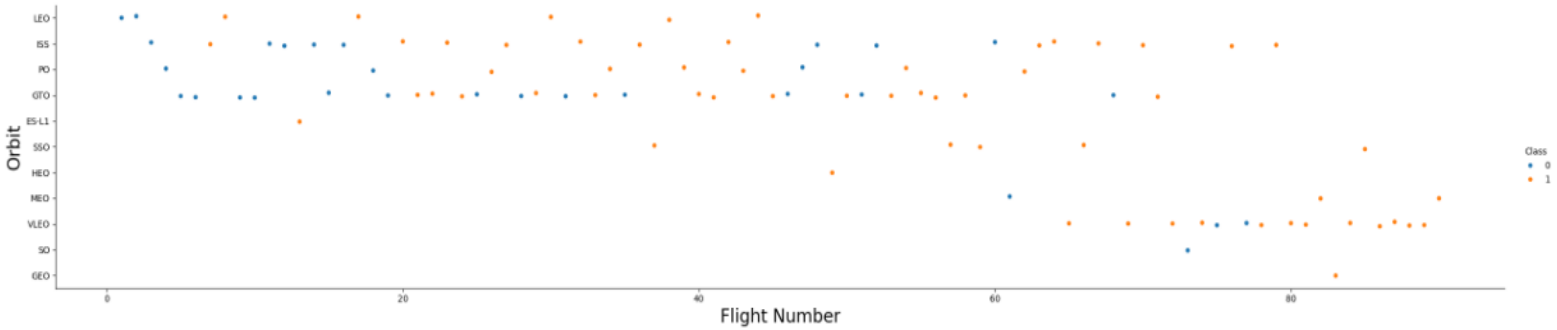
```
df_bar = df.groupby('Orbit')['Class'].mean()
df_bar.plot(kind = 'bar')

plt.xlabel("Orbit",fontsize=20)
plt.ylabel("Class",fontsize=20)
plt.title("Average Class by Orbit", fontsize=20)
plt.show()
```



Flight Number vs. Orbit Type

- Scatter plot: Flight Number vs. Orbit Type, this is to observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success

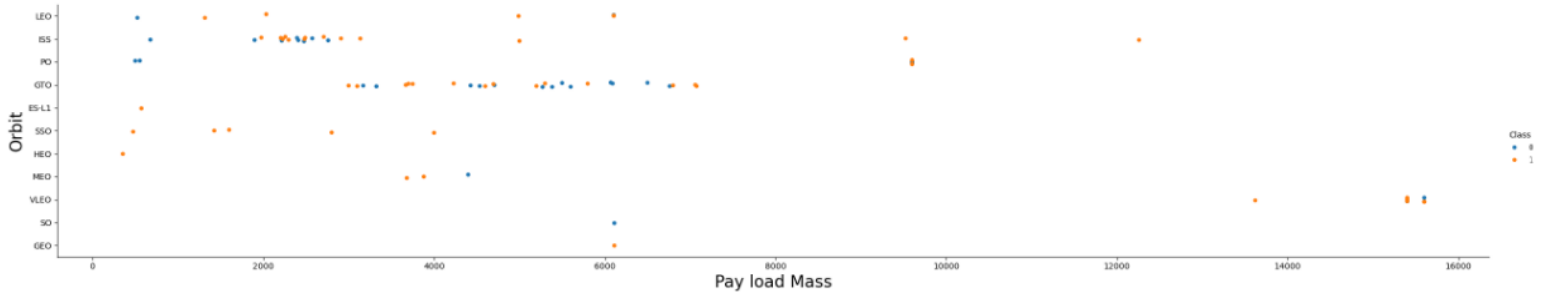
```
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Payload vs. Orbit Type

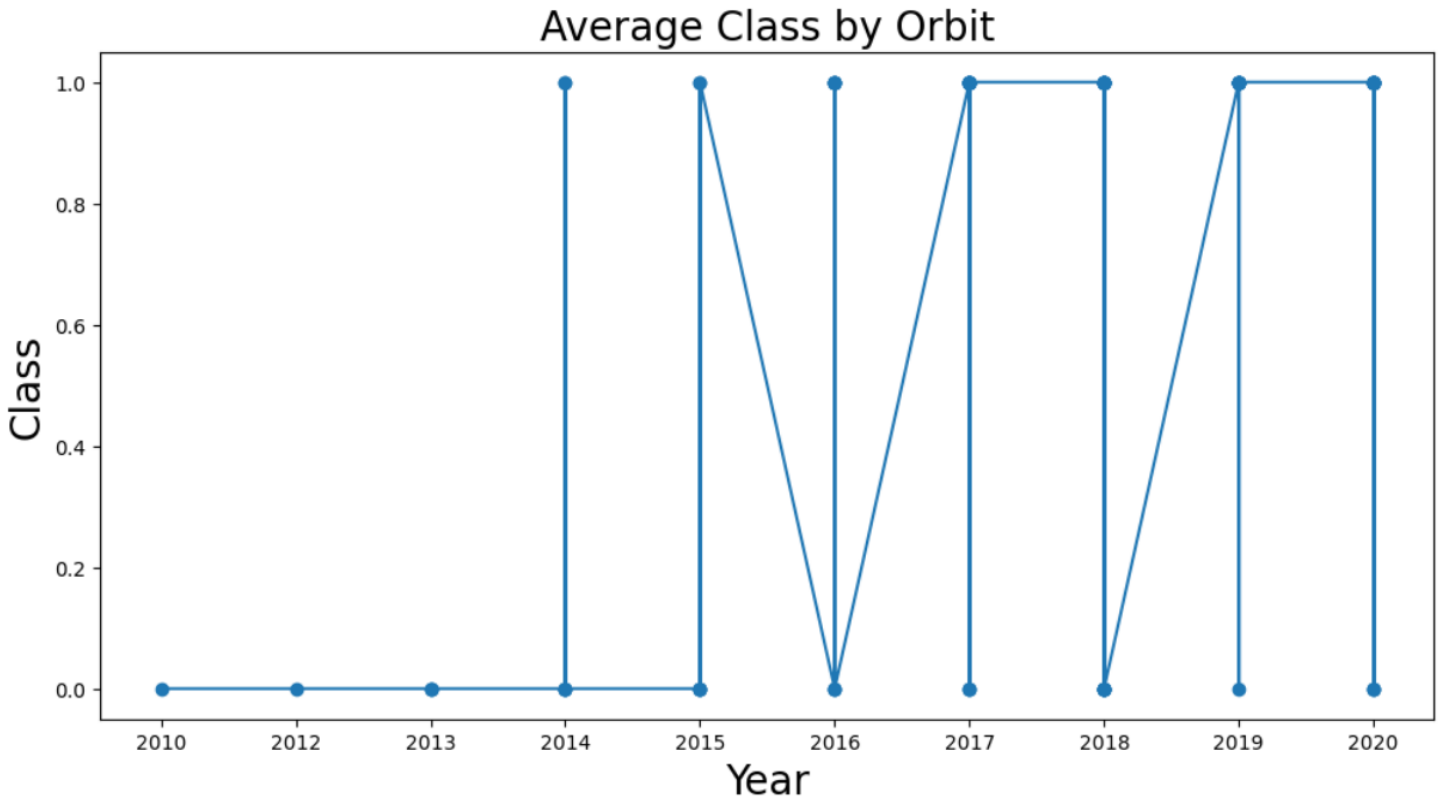
- Scatter plot, Payload vs. Orbit Type, it is to visualize that with heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

```
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Launch Success Yearly Trend

- Line chart: Year vs. Success, it is observed that the success rate since 2013 kept increasing till 2020



All Launch Site Names

THE QUERY IS TO DISPLAY THE NAMES OF THE UNIQUE LAUNCH SITES IN THE SPACE MISSION

```
%sql select distinct(Launch_Site) from SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

THE QUERY IS TO DISPLAY THE LAUNCH SITES THAT BEGIN WITH `CCA`, AND LISTING 5 RECORDS ONLY

```
%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
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

THE QUERY IS TO CALCULATE THE TOTAL PAYLOAD CARRIED BY BOOSTERS FROM NASA (CRS) LAUNCH SITE

```
%sql select sum(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

THE QUERY IS TO CALCULATE THE AVERAGE PAYLOAD MASS CARRIED BY BOOSTER VERSION F9 V1.1

```
%sql select avg(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version like 'F9 v1.1%'
```

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

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

THE QUERY IS TO LIST THE DATES OF THE FIRST SUCCESSFUL LANDING OUTCOME ON GROUND PAD

```
%sql select date, Landing_Outcome from SPACEXTABLE where Landing_Outcome like '%ground%' ORDER BY date ASC Limit 1
```

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

Date	Landing_Outcome
2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- The query is to 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 * from SPACEXTABLE where Landing_Outcome like'%drone%' and Mission_Outcome = 'Success' and PAYLOAD_MASS__KG_ betw
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2016-03-04	23:35:00	F9 FT B1020	CCAFS LC-40	SES-9	5271	GTO	SES	Success	Failure (drone ship)
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- The query is to calculate the total number of successful and failure mission outcomes

```
%sql select count(Mission_Outcome) from SPACEXTABLE where Mission_Outcome = 'Success'
```

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

count(Mission_Outcome)
98

Boosters Carried Maximum Payload

- The query is to list the names of the booster which have carried the maximum payload mass

```
%sql select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select Max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

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

Booster_Version
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

2015 Launch Records

- The queyr is to list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql SELECT Date, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE Landing_Outcome like '%Failure%' and L
```

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

Date	Landing_Outcome	Booster_Version	Launch_Site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- The query is to 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 Landing_Outcome = 'Failure (drone ship)' and date between '2010-06-
```

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

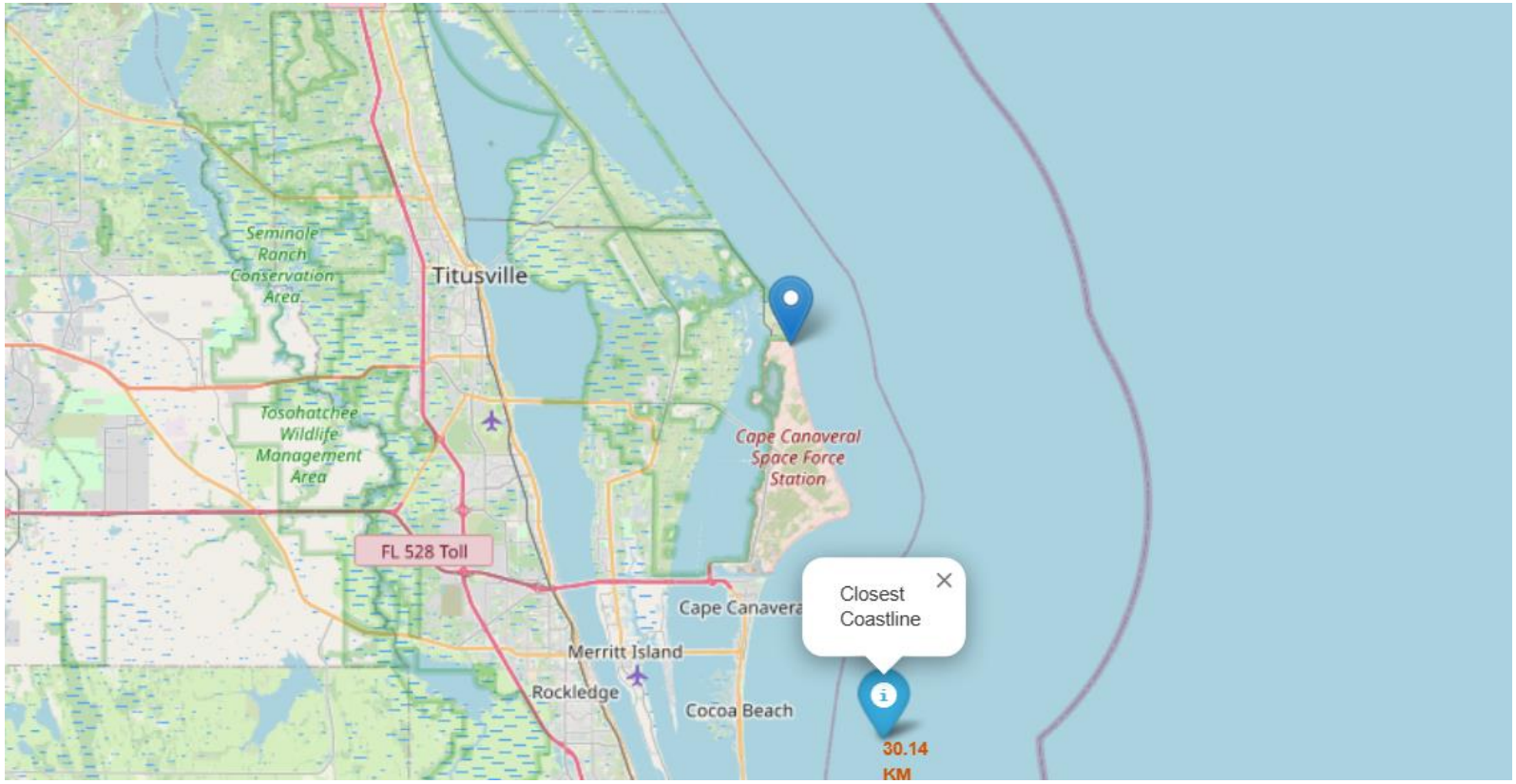
count(Landing_Outcome)
5

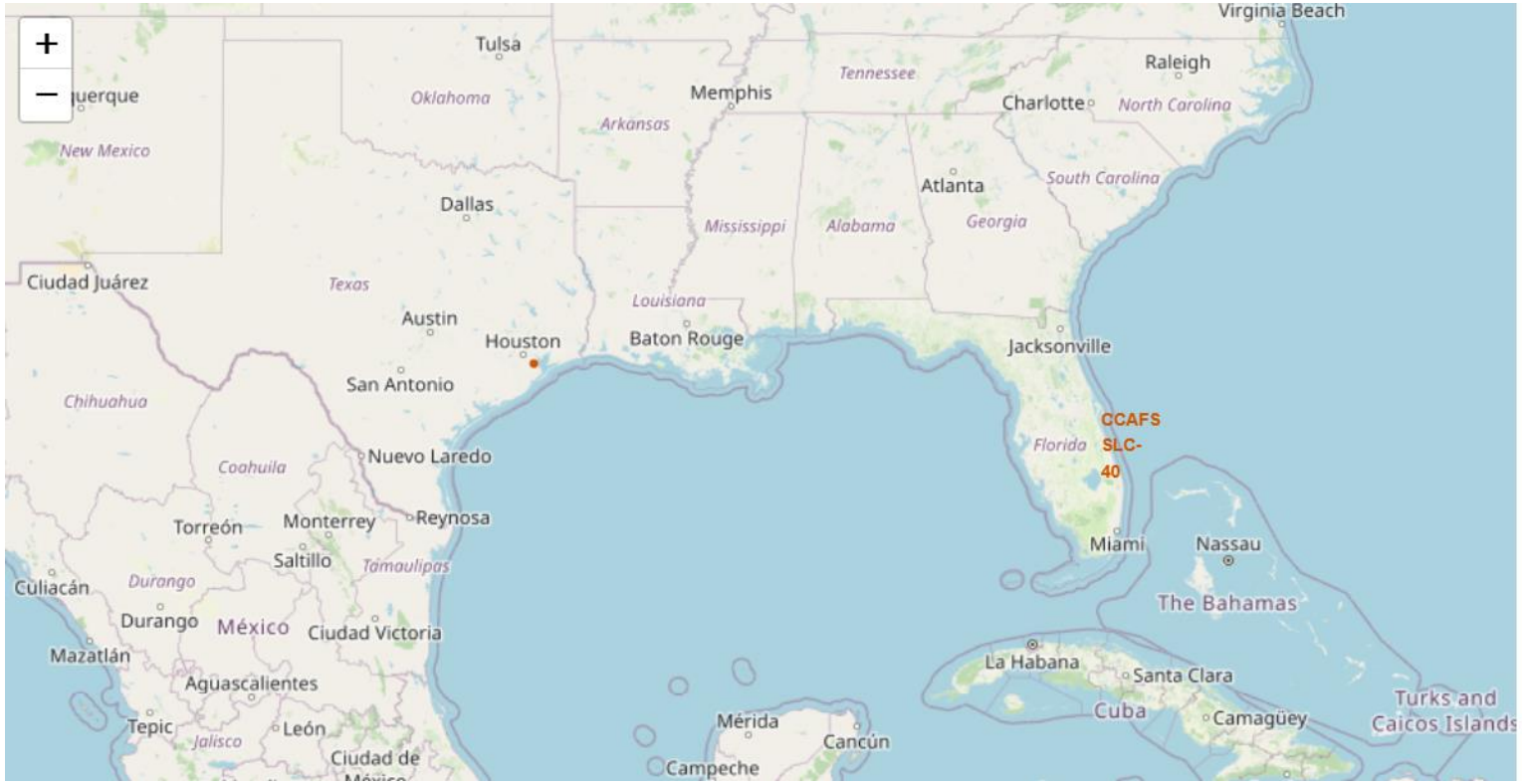
Section 3

Launch Sites Proximities Analysis

Mark down a point on the closest coastline

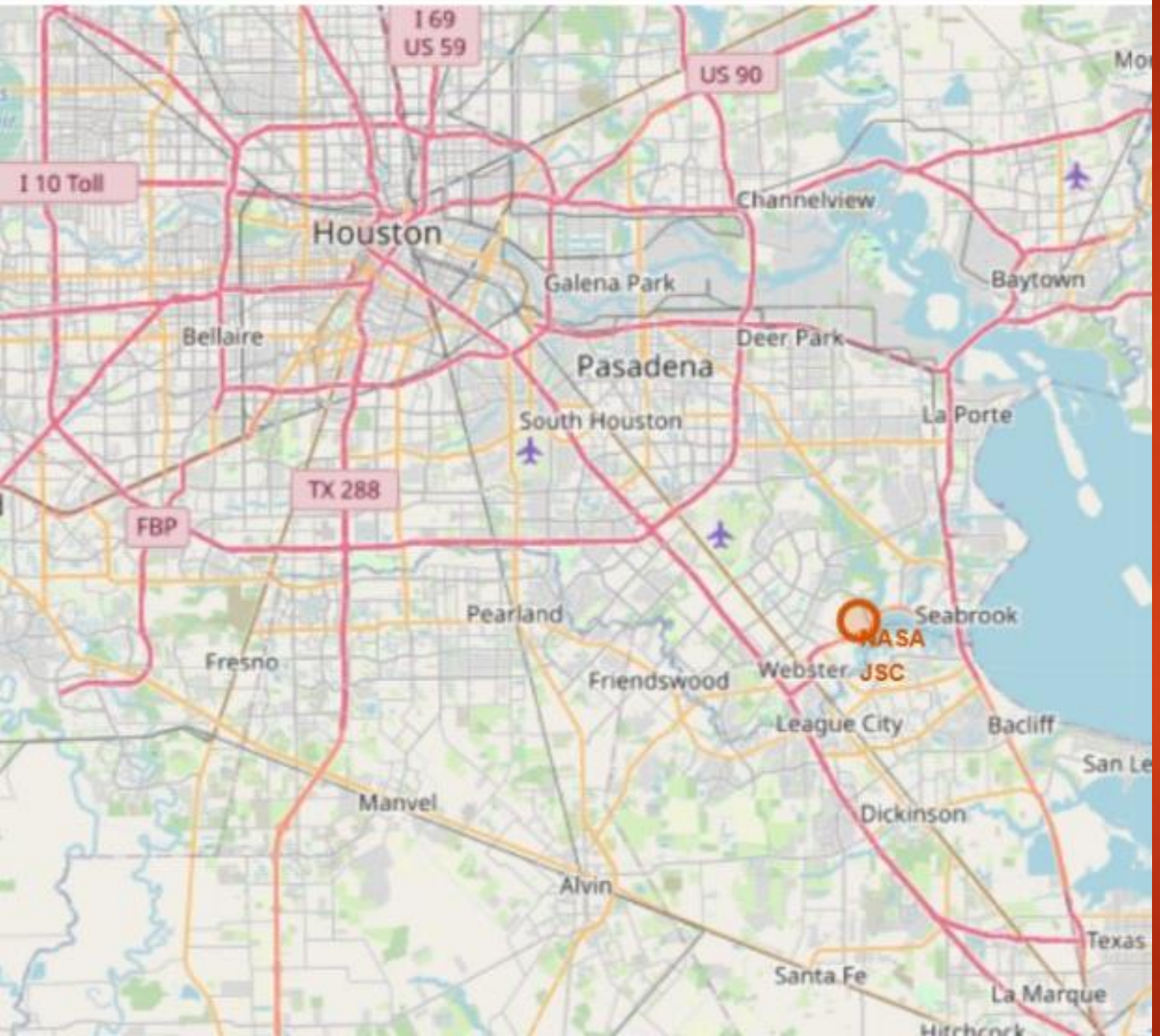
- To mark down a point on the closest coastline using MousePosition and calculate the distance between the coastline point and the launch site





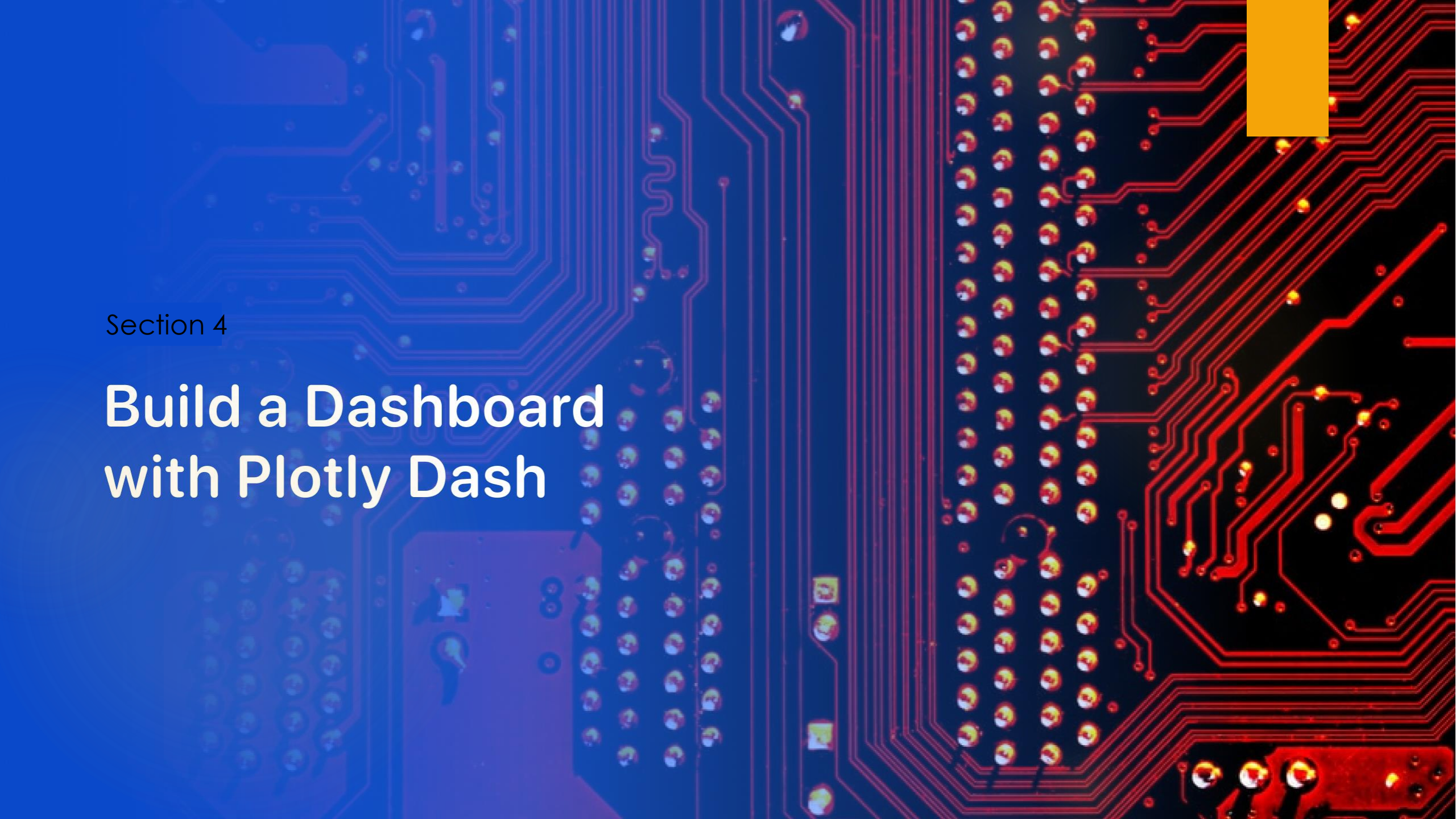
Cluster Map

TO MARK A CLUSTER



NASA Johnson Space Center at Houston, Texas

TO INITIATE CENTER LOCATION TO BE NASA JOHNSON SPACE CENTER AT HOUSTON, TEXAS


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

- Logistic Regression = 0.87
- Confusion Matrix = 0.44
- Decision Tree = 0.375
- KNN = 0.375

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

```
tuned hpyerparameters :(best parameters)  {'C': 1, 'gamma': 0.01, 'kernel': 'rbf'}
accuracy : 0.8767857142857143
```

```
from sklearn.metrics import jaccard_score
jaccard_score(y_test, yhat,pos_label=0)
```

```
0.4444444444444444
```

```
jaccard_score(y_test, yhat,pos_label=0)
```

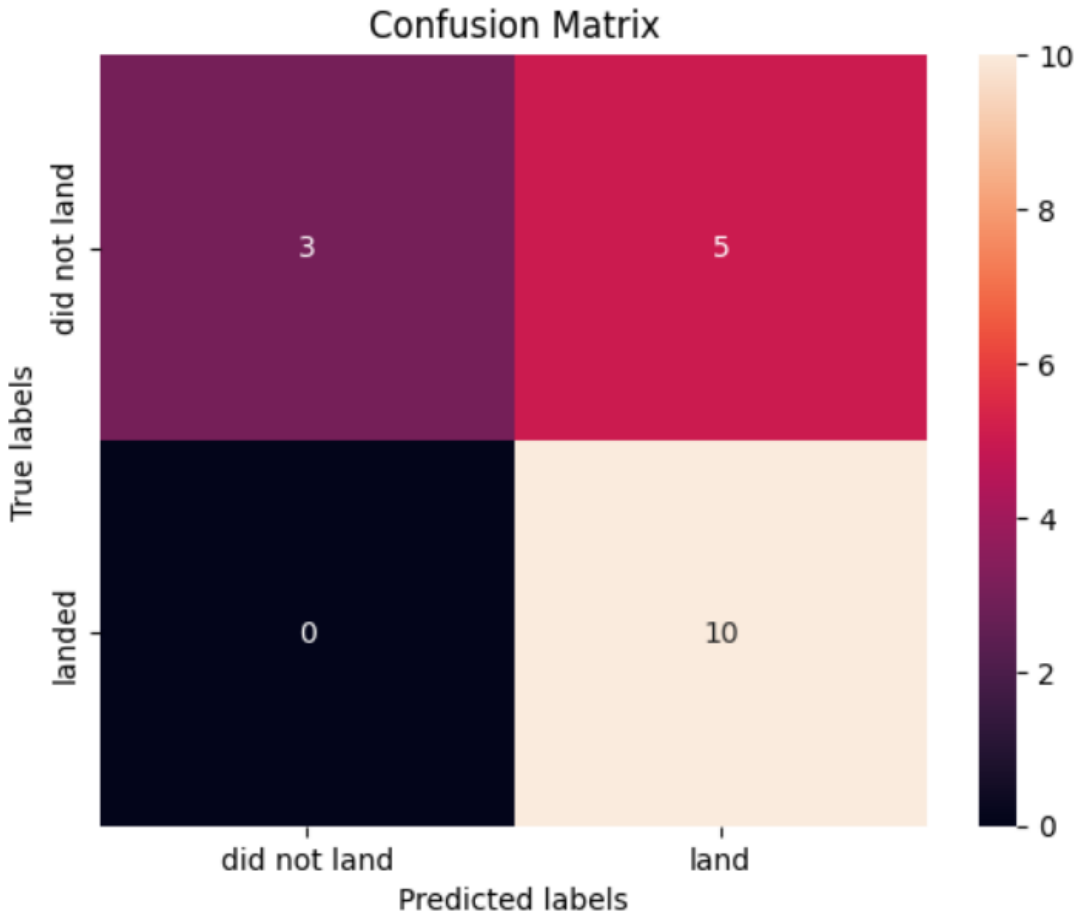
```
0.375
```

```
jaccard_score(y_test, yhat,pos_label=0)
```

```
0.375
```

Confusion Matrix

- Confusion matrix of the best performing model where landed shows highest success ratio of 10 without any short comings.



Conclusions

- The objective was to determine if the first stage of Falcon 9 will land, secondly to determine the cost of a launch
- SpaceX launch data was gathered from the SpaceX REST API and historical data was collected from Wiki pages
- Total number of success and failure attempts were studied with boosters carrying the maxium payload
- Machine Learning algorithm were evaluated to quantify the best accuracy
- It was concluded that the Decision Tree Classifier yeilds the best score of **0.375**

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!