IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ferdous Akbary
Dec 19th, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Collecting the Data with an API – SpaceX lunch data (SpaceX REST API)
  - Wrangling data using API
  - Use Python's BeautifulSoup package to scrape Falcon 9 launch data from Wiki pages.
  - Exploratory Analysis Using SQL.
  - Exploratory Data Analysis for Data Visualization.
  - Interactive Visual Analytics and Dashboards using Plotly and Folium.
  - Machine learning Predictive Analysis
- Summary of all results
  - Requested/parsed SpaceX launch data using a GET request.
  - Visualized relationship between different parameters, visualized launch success yearly trend, and dummy variables for categorical columns.
  - Launch sites marked on the Folium map, success/failed launches indicated, distances calculated, launch site drop-down added, callback for success outcome pie chart implemented, payload range slider included, and callback for payload-outcome scatter plot.
  - Data standardized, train/test split, identified best hyperparameters for each model, test data accuracy calculated.

Introduction

- Project background and context

  In this capstone, we will predict if the Falcon 9 first stage will land successfully. SpaceX 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 SpaceX 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 SpaceX for a rocket launch. In this lab, you will collect and make sure the data is in the correct format from an API. The following is an example of a successful and launch.

- Problems you want to find answers
  - Prediction of Successful First Stage Landing?
  - Cost Implications of Successful Landings?
  - Data Collection and Format?
  - Exploratory Analysis?
  - Interactive Visual Analytics?
  - Machine Learning Model Performance?

Section 1

Methodology

Methodology

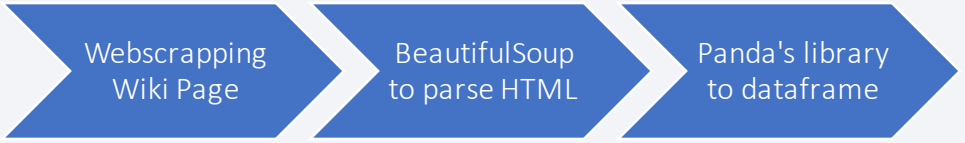
Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX Rest API and web-scrapping SpaceX launch Wiki page.
- Perform data wrangling
  - One-hot encoding (0 or 1) was applied on features of the dataset.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Describe how data sets were collected.
  - Requested SpaceX launch data by making SpaceX rest API call, parsed the SpaceX launch data using the GET request. Use Normalization to convert Json to dataframe.
  - Filtered the dataframe to only include Falcon 9 launches.
  - Relaced None values in the PayloadMass with the mean.
  - Used BeautifulSoup library to parse Wiki page containing SpaceX launch data.
  - Then extracted HTML Launch data tables and converted it to dataframes using Panda's library.
- Flowcharts
  - API Call

Webscrapping



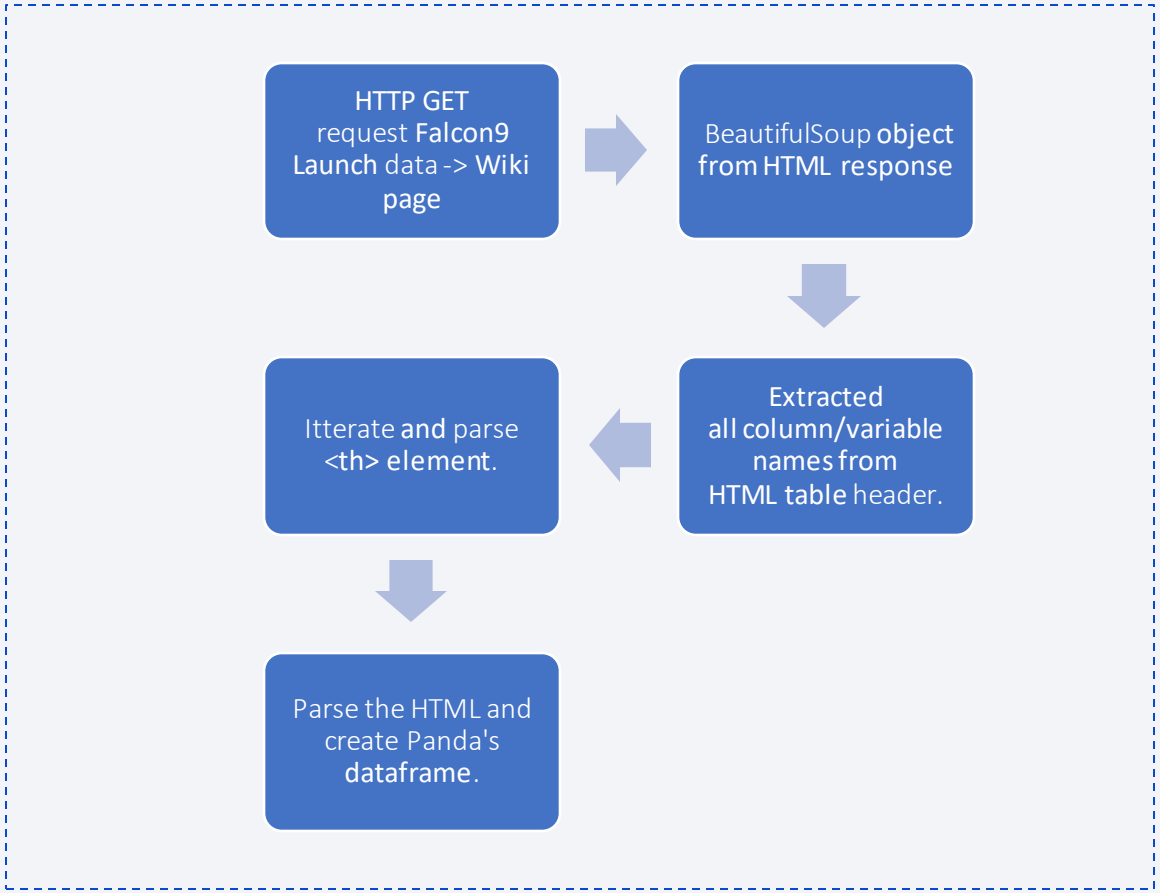
Data Collection – SpaceX API

- Made a GET request to SpaceX Rest API, decoded using .json(), converted to Panda's dataframe using .json_normalize() and then made a second API call to store the data based on IDs in lists to create a new dataframe.
- GitHub URL of the completed SpaceX API calls notebook: https://github.com/AkbFerdy/CAPSTONE-SpaceX-Falcon-9/blob/main/spacex-data-collection-api.ipynb

Data Collection - Scraping

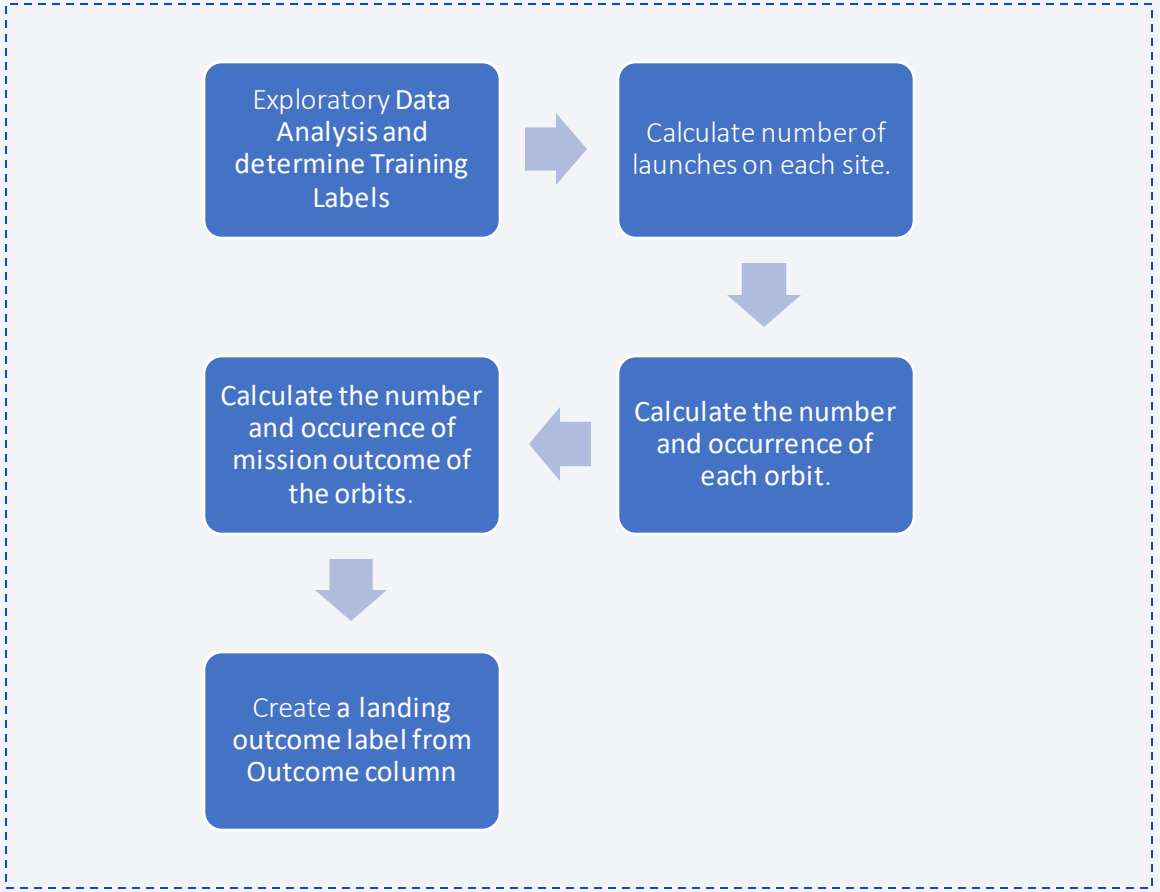
- Using HTTP GET method requested Falcon9 Launch from its Wiki page. Created a BeautifulSoup object from HTML response. Extracted all column/variable names from HTML table header, itterated and parsed all the relevant data (<th> element) and created a panda's dataframe by parsing the HTML launch data.

- GitHub URL of the completed web scraping notebook: https://github.com/AkbFerdy/CAPSTONE-SpaceX-Falcon-9/blob/main/webscraping.ipynb



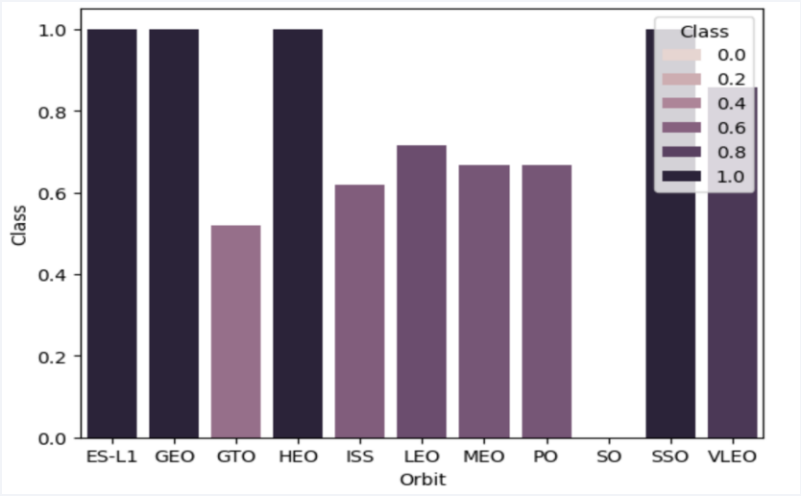
Data Wrangling

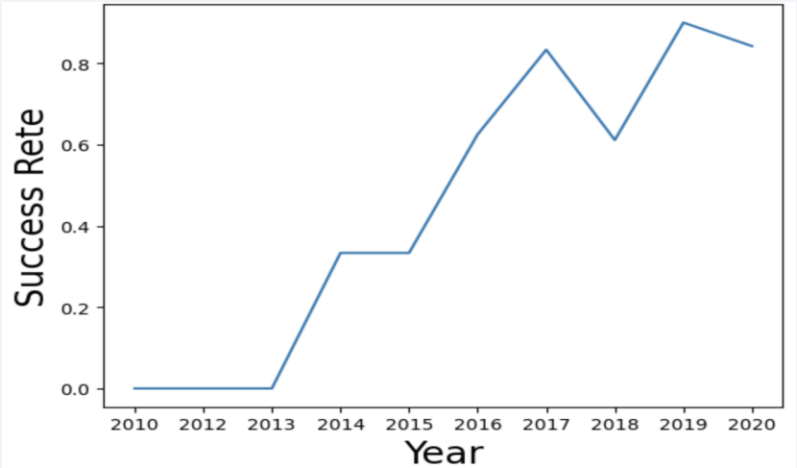
- Exploratory Data Analysis (EDA) to find patterns in the data and label for training supervised models.
- Calculated the number of launches on each site, occurrence of each orbit, occurence of mission outcome of the orbits, and landing outcome label from Outcome column.
- GitHub URL of your completed data wrangling: https://github.com/AkbFerdy/CAPSTONE-SpaceX-Falcon-9/blob/main/spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- Relationships between Flight Number & Launch Site, Payload & Launch Site, success rate of orbit type, Flight Number & Orbit type, Payload & Orbit type, and Launch success yearly trend were visualized using seaborn to find patterns and correlations. (i.e., two visuals on the right. Refer to the nootbook for the rest).
- GitHub URL: https://github.com/AkbFerdy/CAPSTONE-SpaceX-Falcon-9/blob/main/eda-dataviz.ipynb.jupyterlite.ipynb





EDA with SQL

- After the dataset was downloaded the following queries were performed
  - 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
  - List the date when the first succesful landing outcome in ground pad was acheived.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
  - 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.
- GitHub URL: https://github.com/AkbFerdy/CAPSTONE-SpaceX-Falcon-9/blob/main/eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

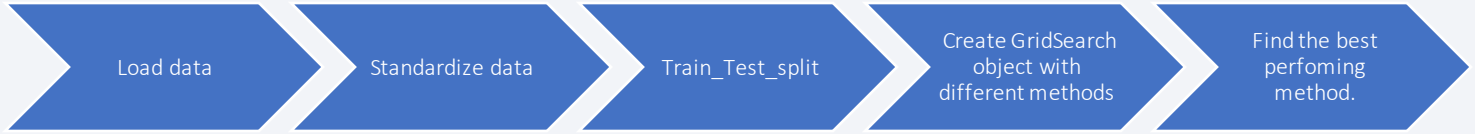
- Marked all launch sites on a Folium map. marked the success/failed launches for each site on the Map.
- Draw circles on the Folium map of green for success and red for failed launch. Then we used clustering to to cluster the number of success/failed launch per lauch site.
- Calculated the distances between a launch site to its proximities.
- We added these objects to find out are launch sites in close proximity to railways, cities, highways, and costlines.
- GitHub URL: https://github.com/AkbFerdy/CAPSTONE-SpaceX-Falcon-9/blob/main/Launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Plots/graphs and interactions added to a dashboard:
  - Add a Launch Site Drop-down Input Component.
  - Add a callback function to render success-pie-chart based on selected site dropdown.
  - Add a Range Slider to Select Payload.
  - Add a callback function to render the success-payload-scatter-chart scatter plot.
- Created this Plotly Dash application for users to perform interactive visual analytics on SpaceX launch data in real-time.
- GitHub URL: https://github.com/AkbFerdy/CAPSTONE-SpaceX-Falcon-9/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Summary:
  - Load the data, Created a NumPy array from the column Class in data.
  - Standardize the data in X then reassign it to the variable X using the transform provided below.
  - train_test_split to split the data X and Y into training and test data.
  - Create a GridSearchCV object using logistic regression, support vector machine, decision tree classifier, and k nearest neighbors. Then we calculated the accuracy of each and found out that the decision tree classifier method performs well on training set but on test data, the rest of them are even.



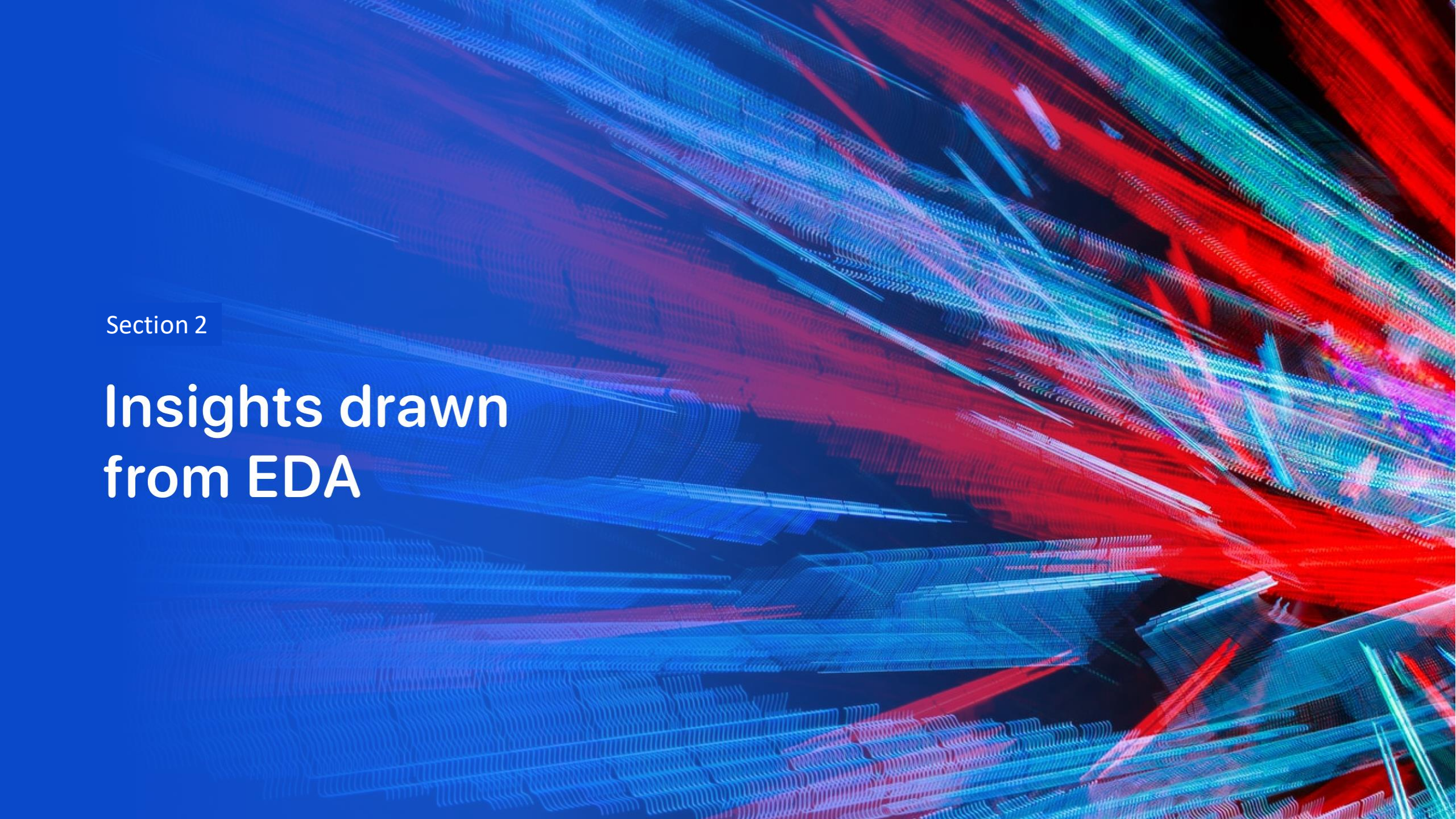
- GitHub URL: https://github.com/AkbFerdy/CAPSTONE-SpaceX-Falcon-9/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

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

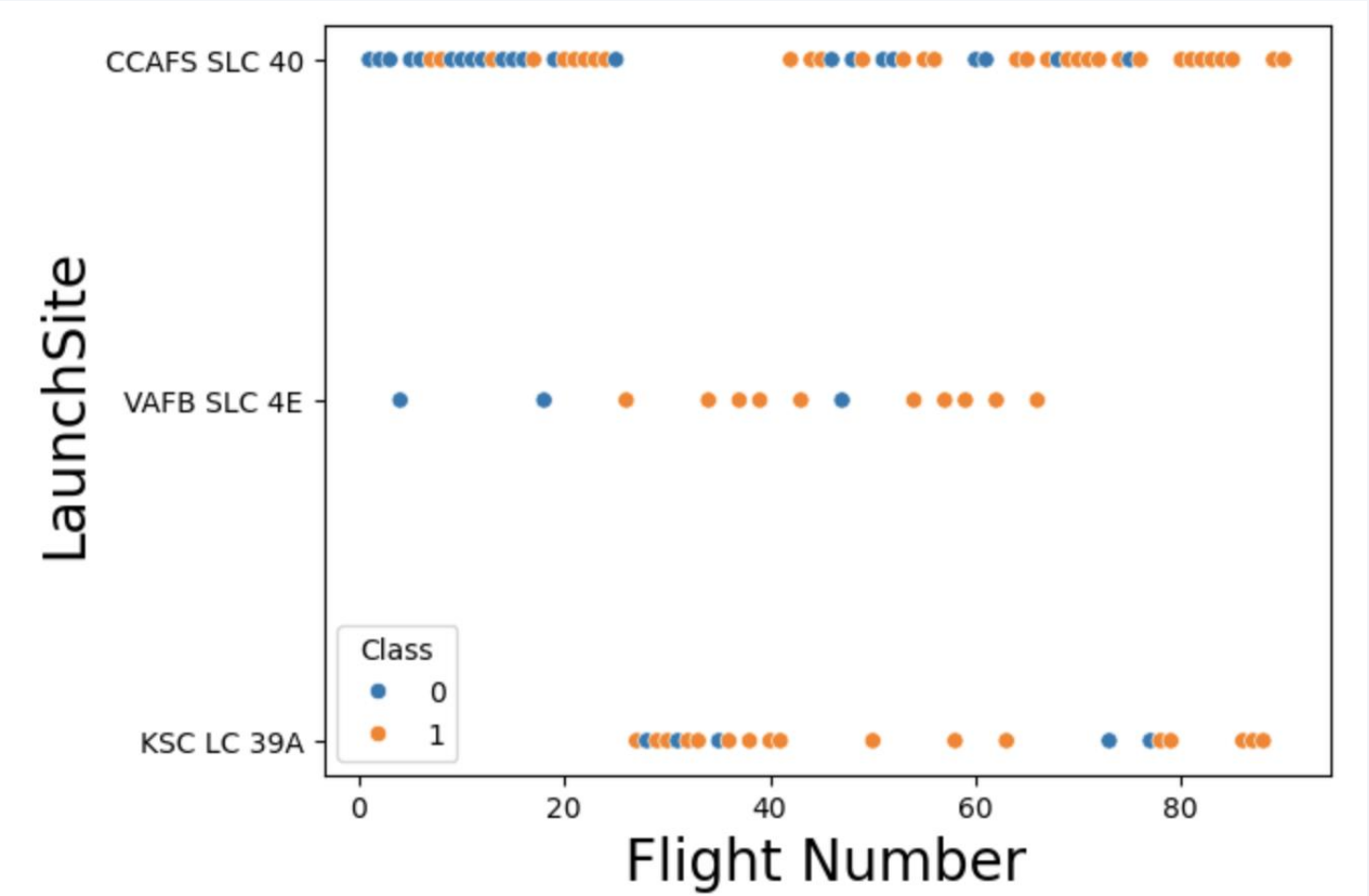
Section 2

Insights drawn from EDA



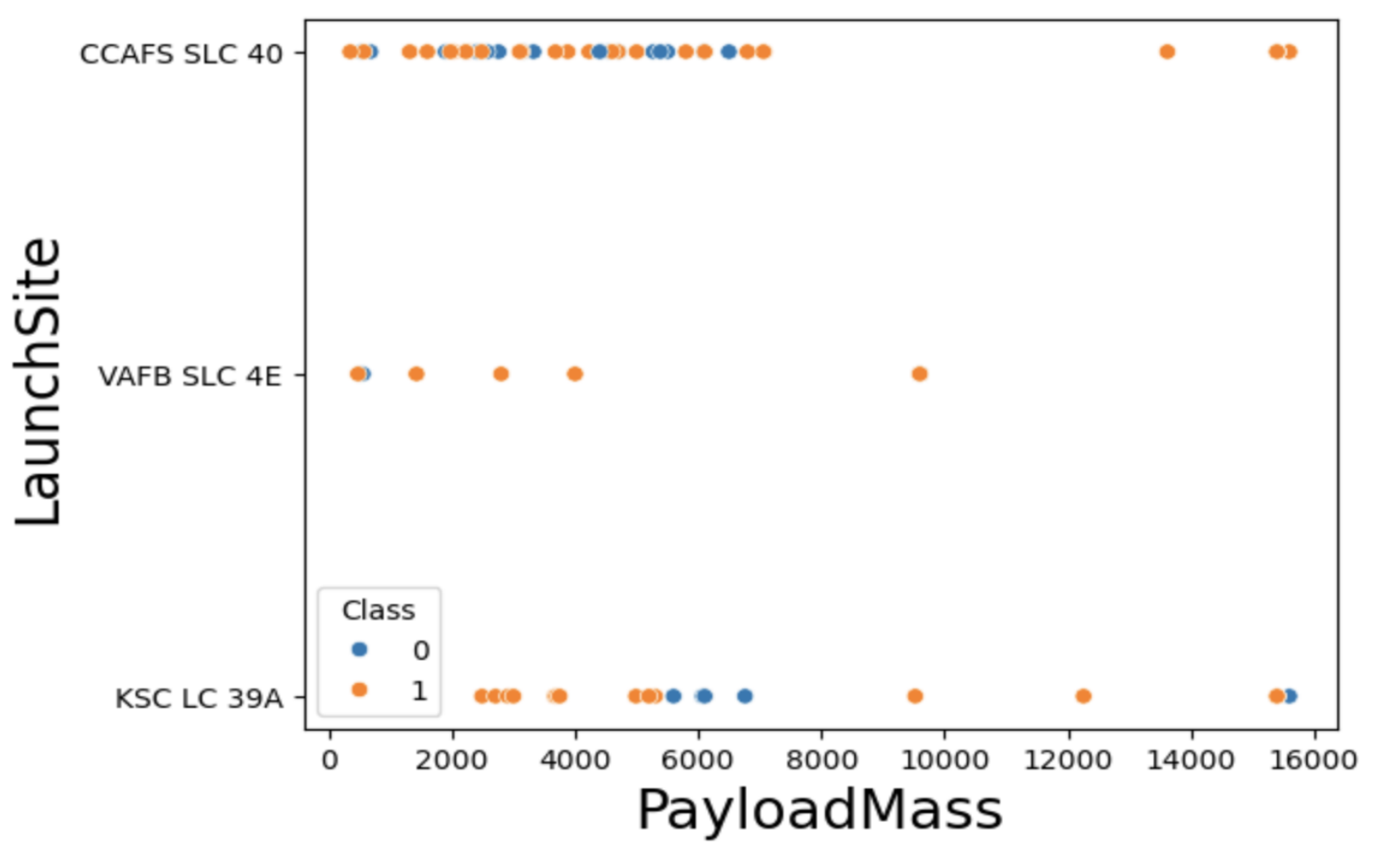
Flight Number vs. Launch Site

- A scatter plot of Flight Number vs. Launch Site shows a direct correlation between flight number and lauch sites. (higher flight number = higer succes launch).



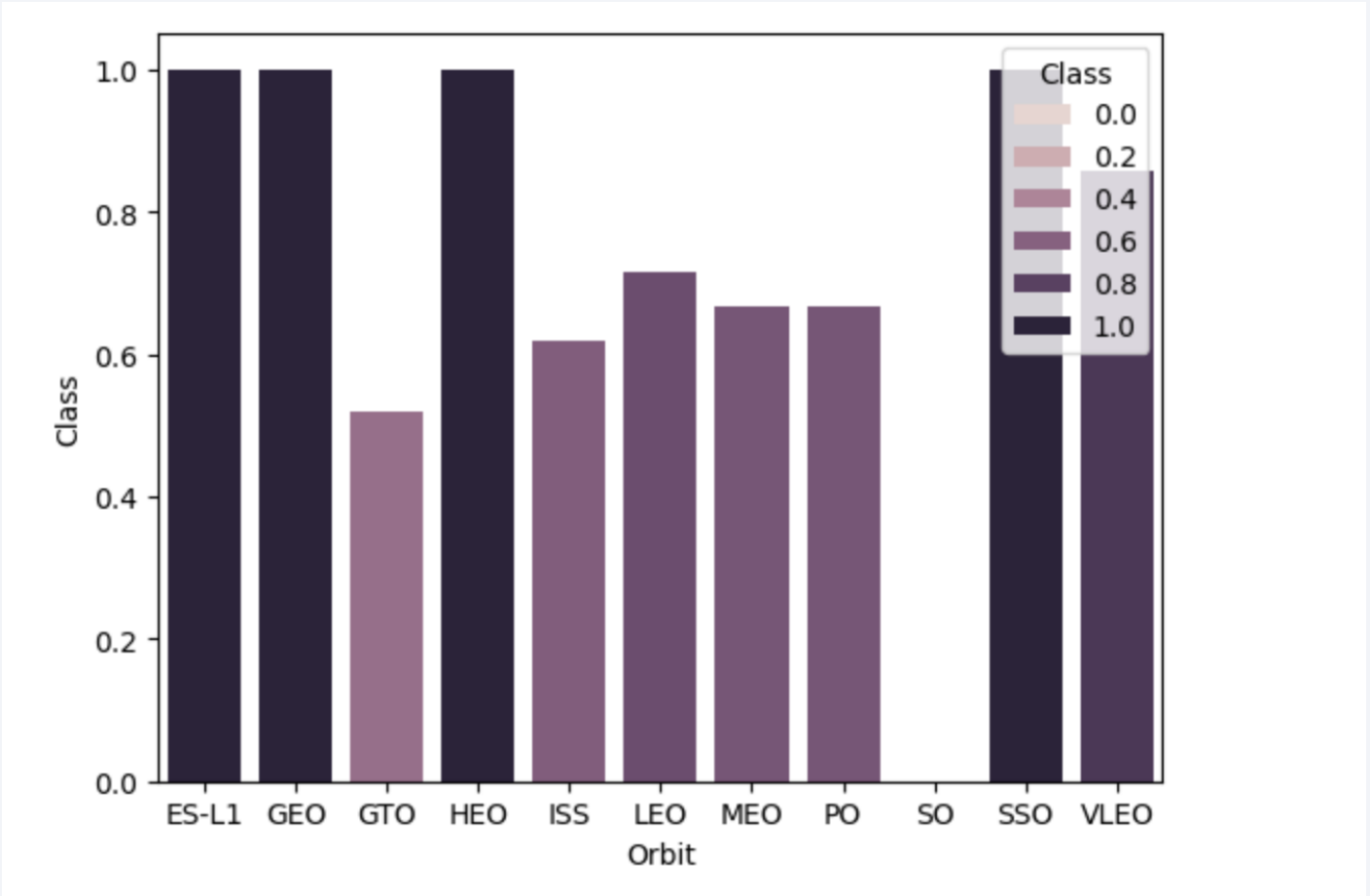
Payload vs. Launch Site

- A scatter plot of Payload vs. Launch Site shows the success rate higher for pass below 5000 and above 7000. For KSC LC 39A.



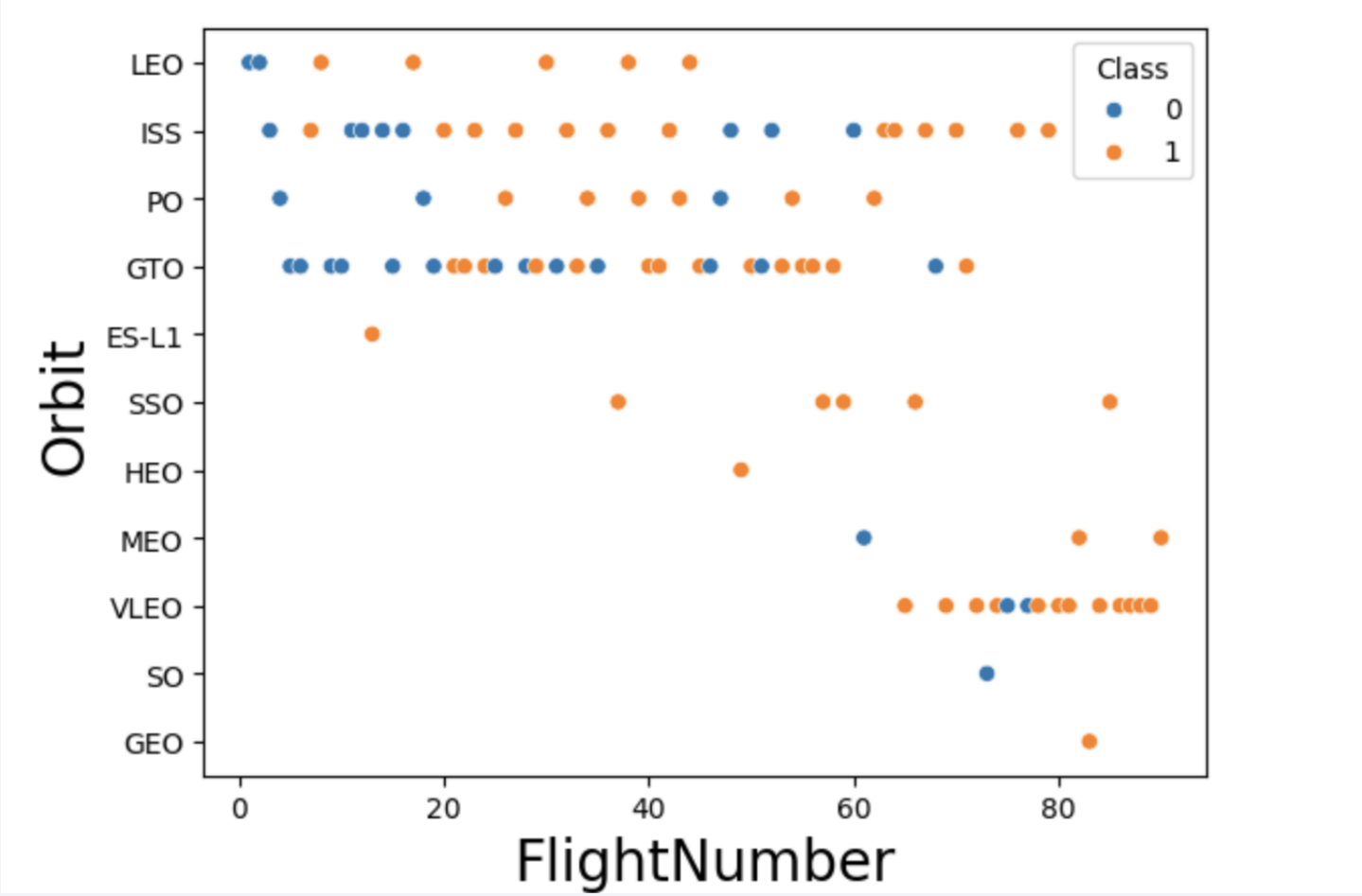
Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type shows super high success rate for ES-L1, GEO, HEO, and SSO. While GTO with the lowest success rate.



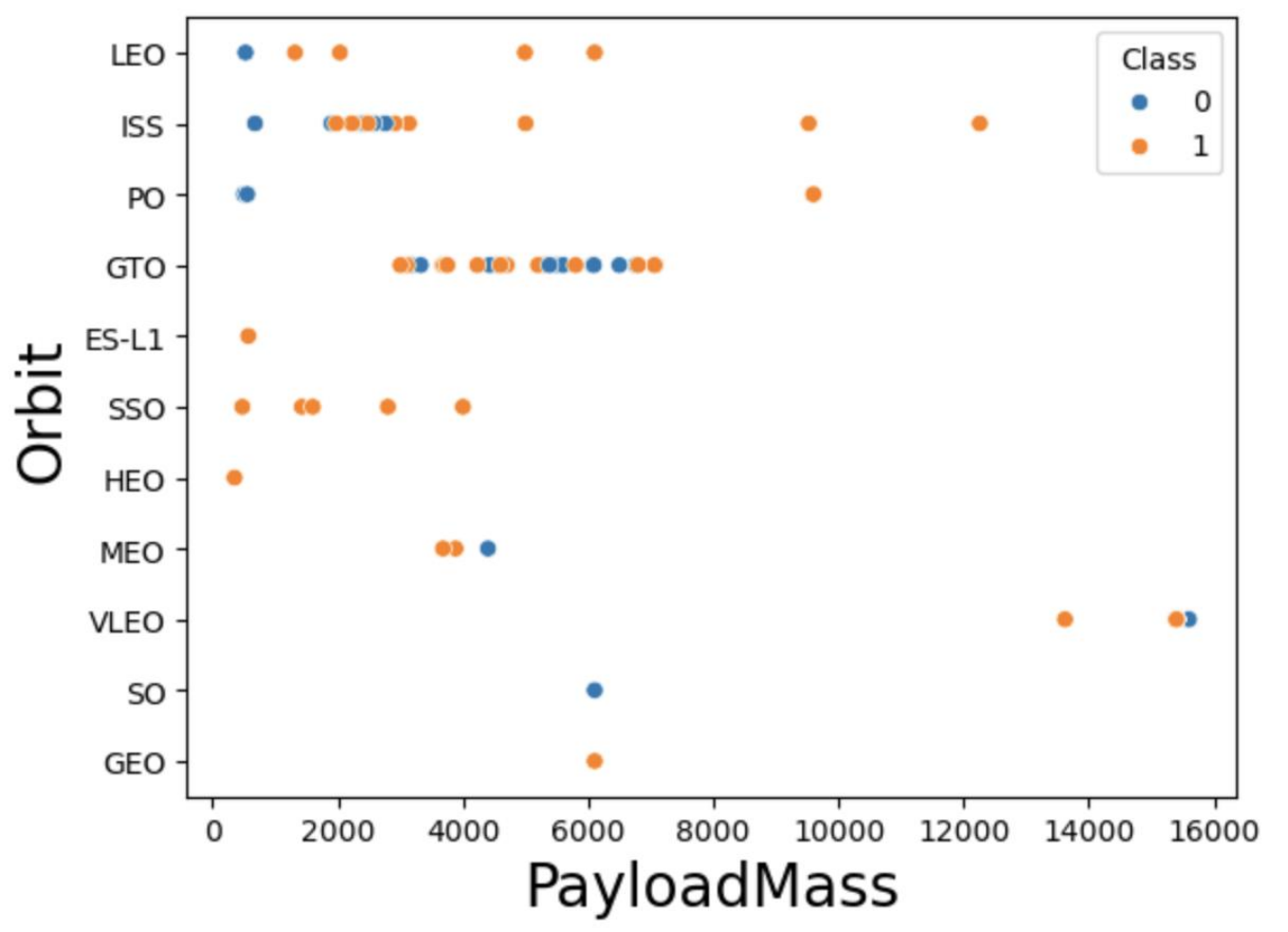
Flight Number vs. Orbit Type

- A scatter point of Flight number vs. Orbit type shows a clear corlation between orbit and flightnumber, as success rate goes up as flightnumber goes up but the rest our all random.



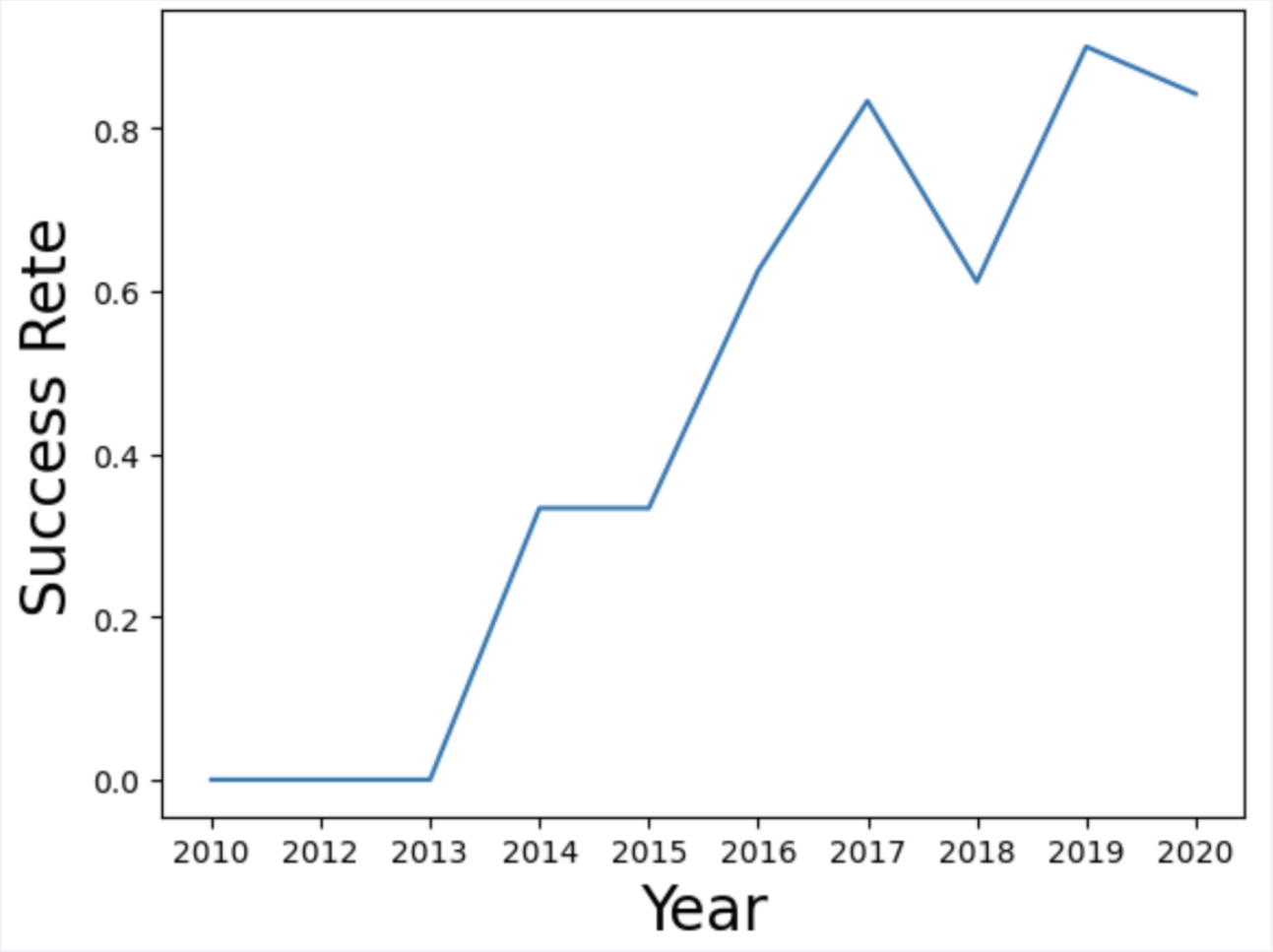
Payload vs. Orbit Type

- A scatter point of payload vs. orbit type shows direct coralation between PayloadMass and success rate for orbits LEO, ISS, SSO, and PO.



Launch Success Yearly Trend

- A line chart of yearly average success rate shows the sucess rate since 2013 kept increasing till 2020.



All Launch Site Names

- A SQL query was made using DISTINCT to find all the unique launch sites.

Task 1

Display the names of the unique launch sites in the space mission

```
%sql SELECT DISTINCT(Launch_Site) FROM SPACEXTBL;
```

```
 * 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'

- Using LIKE and LIMIT, the first 5 records where launch sites begin with `CCA` was displayed using the SQL query below.

Task 2

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

- Usign SUM and Customer == 'NASA (CRS) the total payload carried by boosters from NASA was calculated in the SQL query below.

Task 3

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS Payload_Mass_by_NASA_kg FROM SPACEXTBL WHERE Customer == 'NASA (CRS)';
```

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

Payload_Mass_by_NASA_kg
45596

Average Payload Mass by F9 v1.1

- Using AVG and Booster_Version == 'F9 v1.1'. The average payload mass carried by booster version F9 v1.1was calculated in the SQL query below.

Task 4

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_Payload_F9 FROM SPACEXTBL WHERE Booster_Version == 'F9 v1.1';
```

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

AVG_Payload_F9
2928.4

First Successful Ground Landing Date

- Usig the MIN(Data) and Mission_Outcome == 'Success', the dates of the first successful landing outcome on ground pad was found in SQL query below.

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
]: %sql SELECT MIN(Date) AS First_succes FROM SPACEXTBL WHERE Mission_Outcome == 'Success';
```

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

```
]:
```

First_succes
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

- By restricting PAYLOAD to >4000 and <6000 with Landing_outcome == 'succes (drone ship)', the names of boosters which have successfully landed on drone ship were found in SQL query below.

```
%sql SELECT Booster_Version AS succes_boosters FROM SPACEXTBL WHERE (Landing_Outcome) == 'Success (drone ship)' AND (PAYLOAD_MASS__KG_)
<6000 AND (PAYLOAD_MASS__KG_) >4000 ;
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Using Group by Mission_outcome the success count were found but it wasn't super clear, so I made some adjusement and did two separete queries (7a & 7b), where Mission_outcome == 'success' and != 'success.

Task 7

List the total number of successful and failure mission outcomes

```
print('98 is supposed to be sucess count')
%sql SELECT COUNT(Mission_Outcome) AS succes_count FROM SPACEXTBL GROUP BY Mission_Outcome;
```

```
98 is supposed to be sucess count
 * sqlite:///my_data1.db
Done.
```

succes_count
1
98
1
1

Task 7 but redone (7a*)

List the total number of successful outcomes

```
%sql SELECT COUNT(Mission_Outcome) AS succes_count FROM SPACEXTBL WHERE Mission_Outcome == 'Success';
```

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

succes_count
98

Task 7 but redone (7b*)

List the total number of failure mission outcomes

```
%sql SELECT COUNT(Mission_Outcome) AS fail_count FROM SPACEXTBL WHERE Mission_Outcome != 'Success';
```

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

fail_count
3

Boosters Carried Maximum Payload

- Using Subquery Max Payload was selected From Booster_version to find the list of names of boosters which have carried the maximum payload mass.

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ == (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- Using substr (Date, 0,5) = '2015' and Landing_outcome == 'Failure (drone ship)' the list of failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 were found with the SQL query below.

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT Date, Launch_Site, substr(Date, 6,2) AS Month, Landing_Outcome FROM SPACEXTBL WHERE Landing_Outcome == 'Failure (drone ship)' AND substr(Date,0,5)='2015';
```

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

Date	Launch_Site	Month	Landing_Outcome
2015-01-10	CCAFS LC-40	01	Failure (drone ship)
2015-04-14	CCAFS LC-40	04	Failure (drone ship)

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

- The rank of the count was found where landing_outcomes was set to == (Failure (drone ship) or == Success (ground pad)) between the date 2010-06-04 and 2017-03-20, and GROUP BY landing_outcome in descending order by default.

Task 10

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 Date, Landing_Outcome, COUNT(Landing_Outcome)FROM SPACEXTBL\
WHERE Landing_Outcome == 'Failure (drone ship)' OR Landing_Outcome == 'Success (ground pad)' \
AND Date BETWEEN '2010-06-04' and '2017-03-20' GROUP BY Landing_Outcome;
```

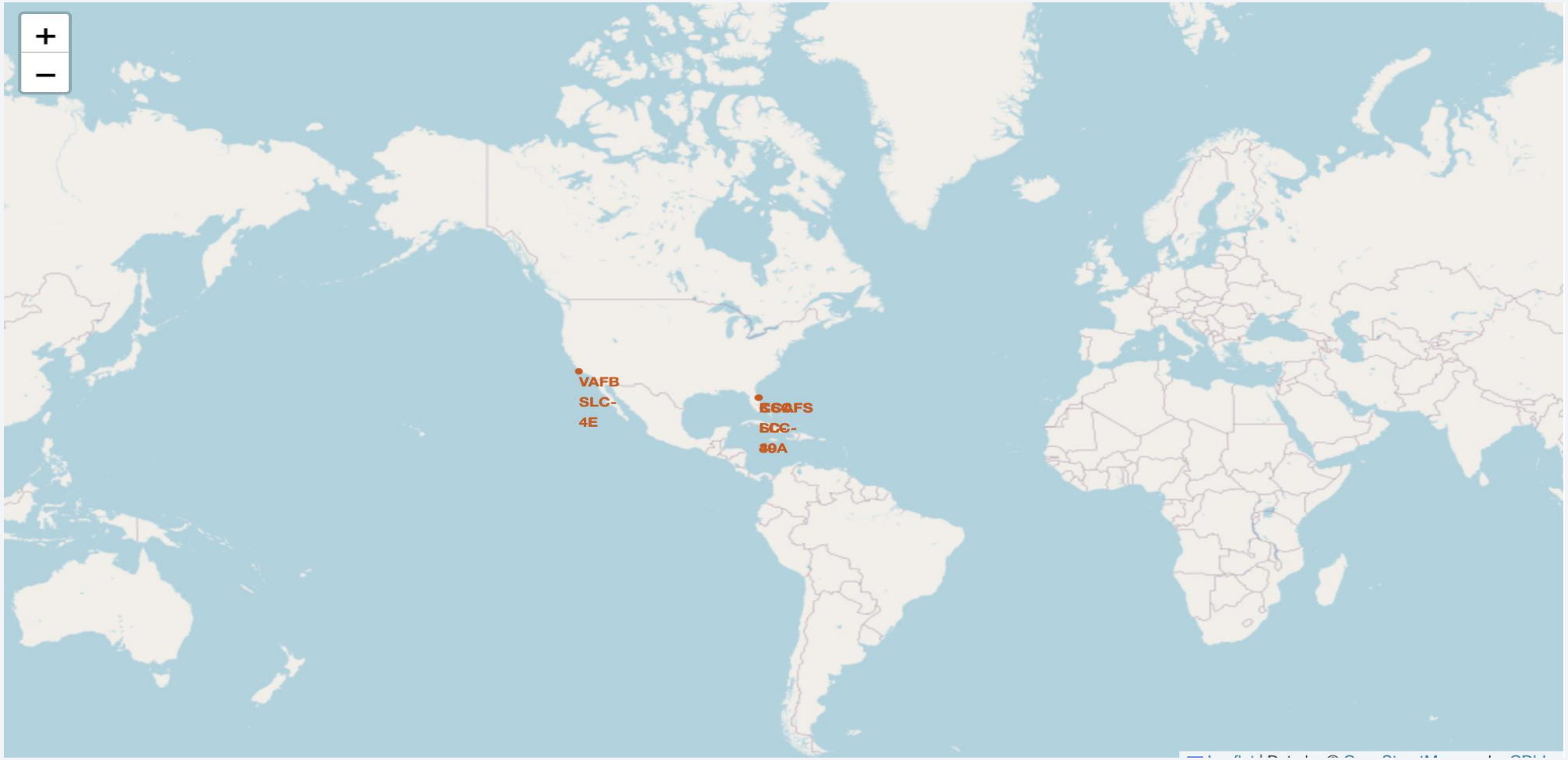
```
 * sqlite:///my_data1.db
Done.
```

Date	Landing_Outcome	COUNT(Landing_Outcome)
2015-01-10	Failure (drone ship)	5
2015-12-22	Success (ground pad)	3


Section 3
Launch Sites
Proximities Analysis

Map with marked launch sites for SpaceX

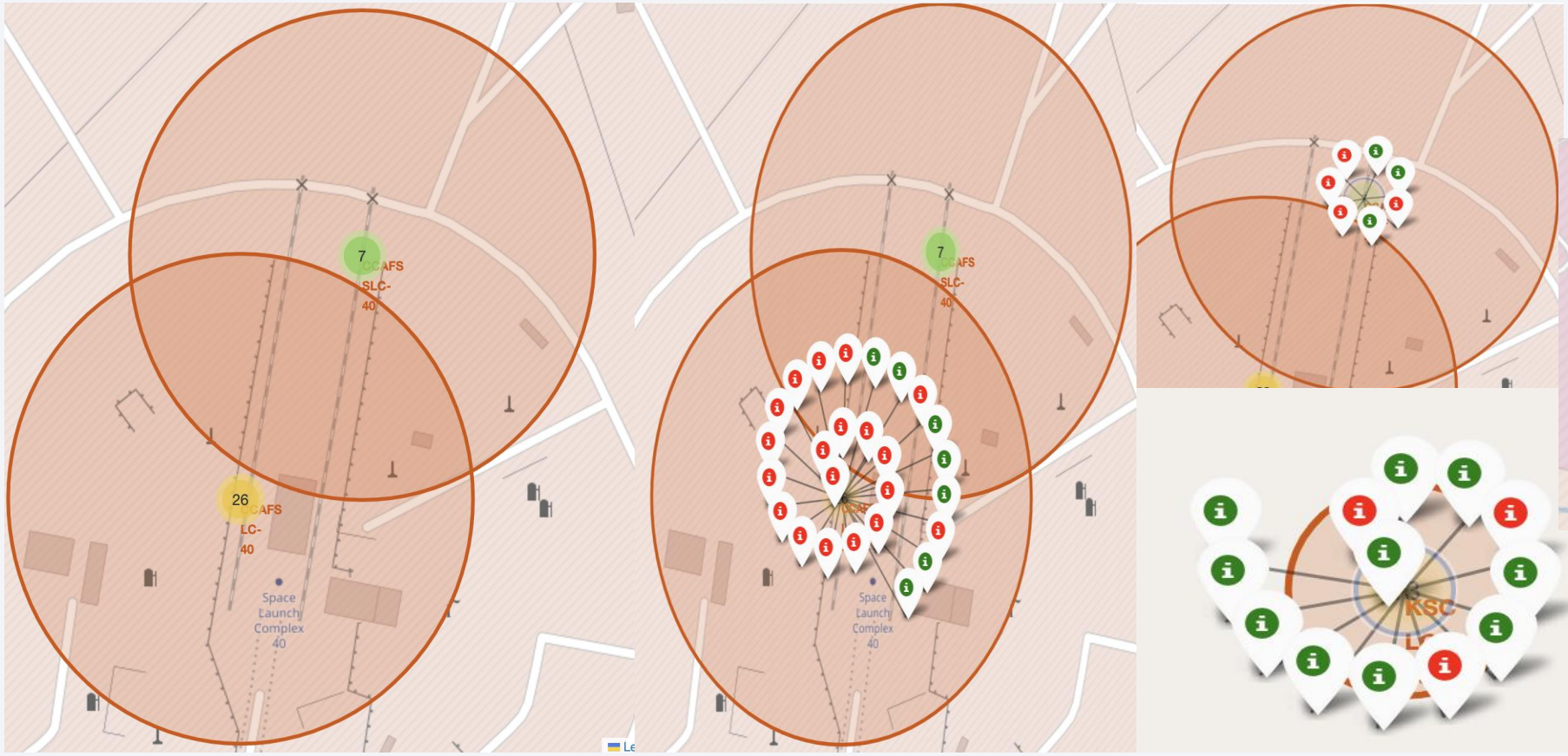
SpaceX all launch sites marked around the globe, and we could clearly see, they are all in USA. (cost of Florida and Califronia).



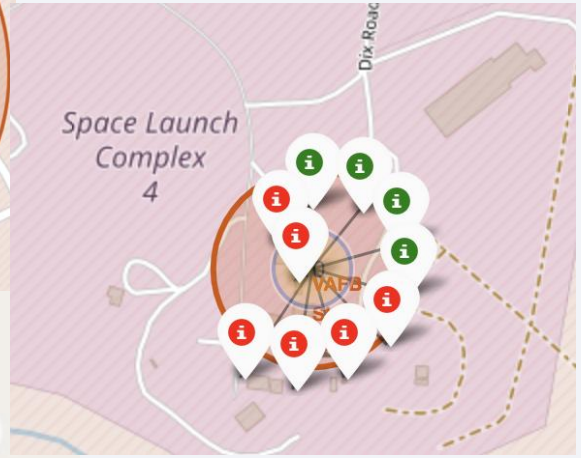
Launch sites at Cost of Florida and Califronia

Color labels showing Zoom out and zoom in Launch sites for SpaceX

Cost of Florida Launch sites.

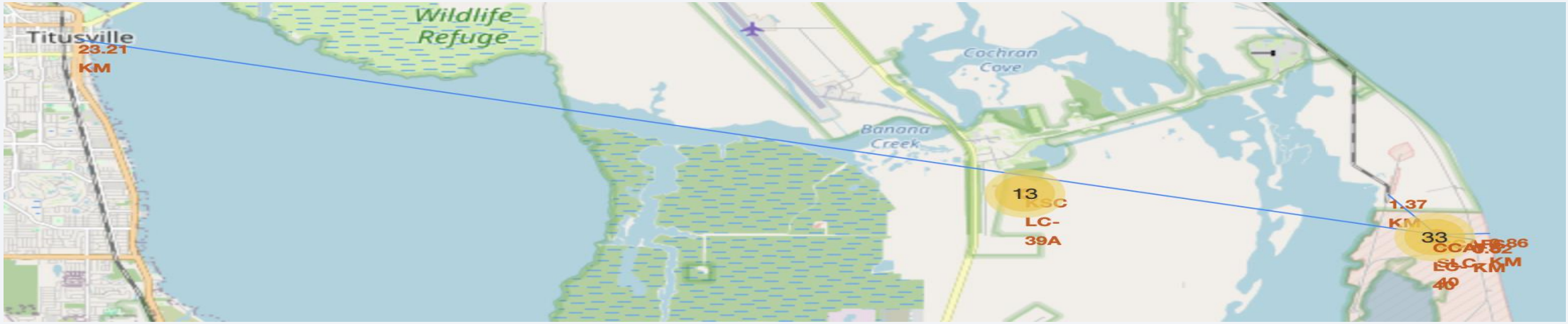


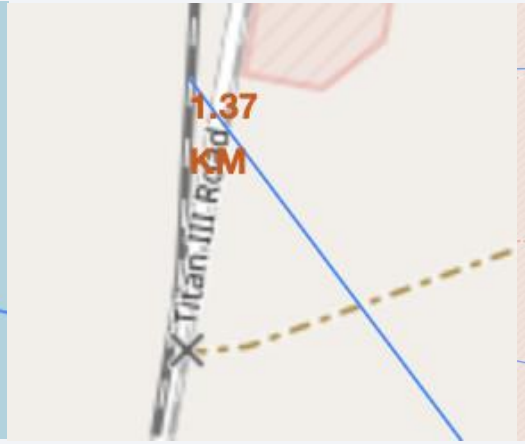
Cost of Califronia

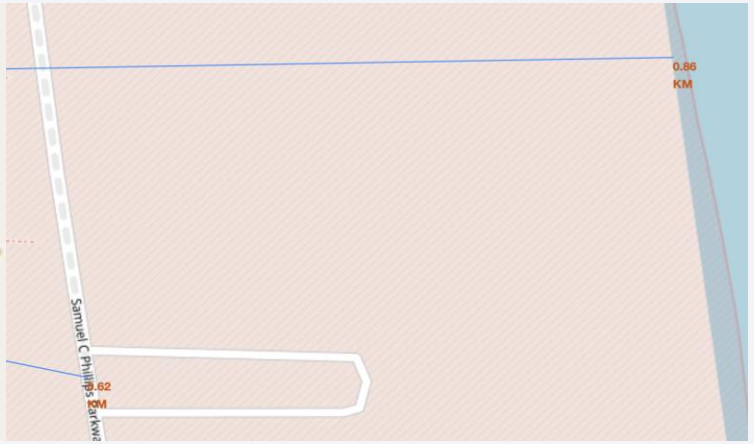


Launch site distance to close by landmarks

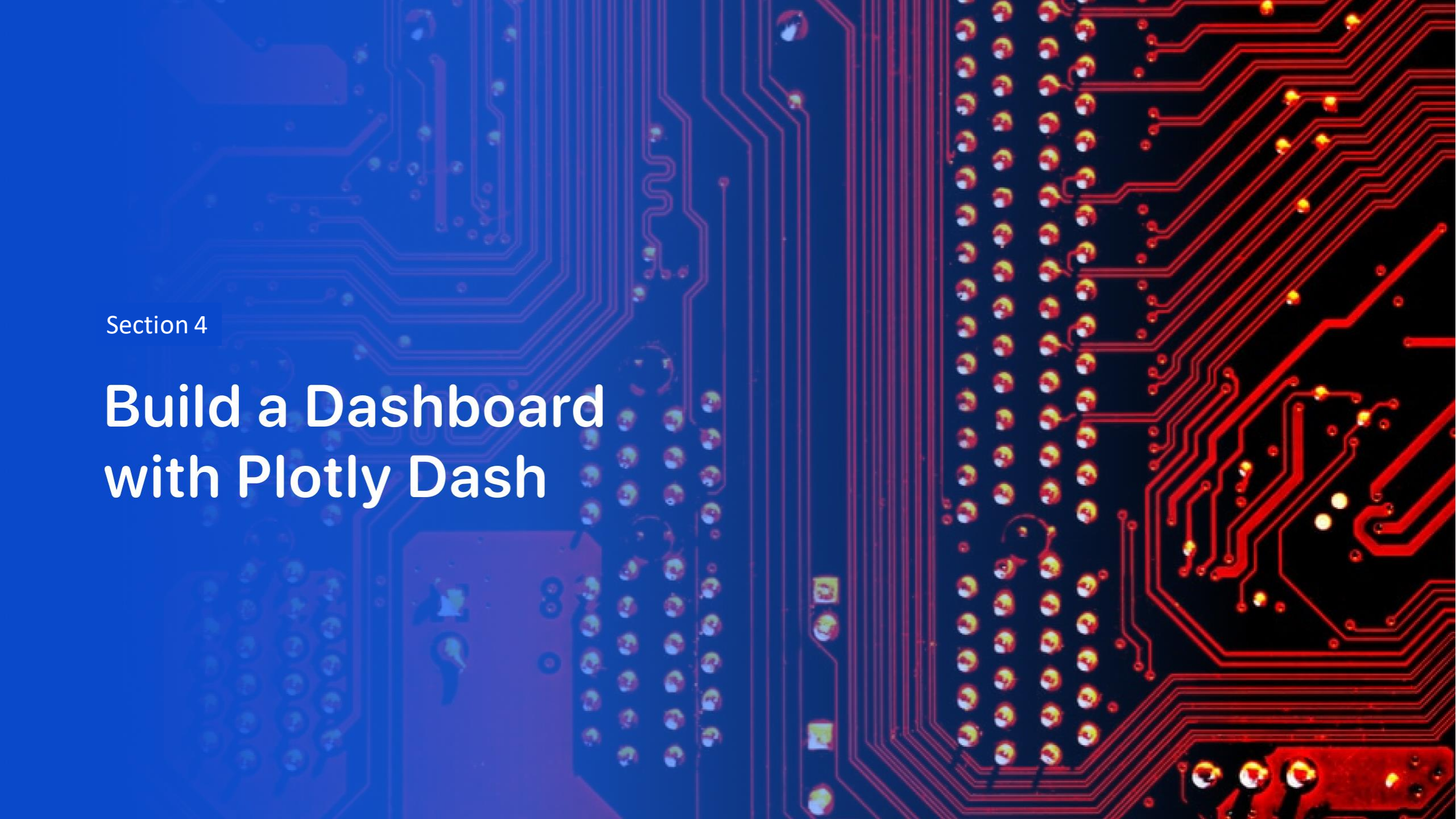
- The distance to close by landmarks is important because they shouldn't be close to any landmarks. Based on our reaserch, they are clearly not as the closest city is 23 Km away.





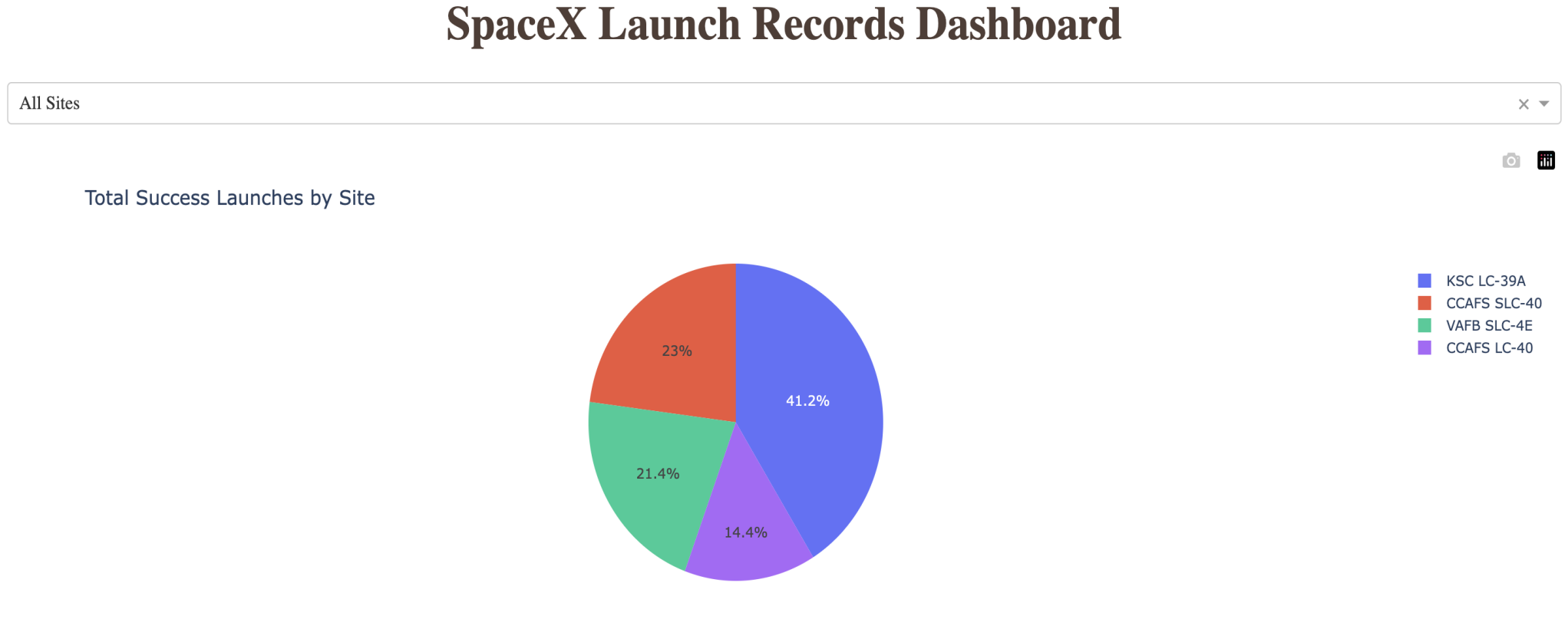


Distances
City: 23.2 Km
Highway: 0.6 KM
Railroad: 1.37Km


Section 4
Build a Dashboard with Plotly Dash

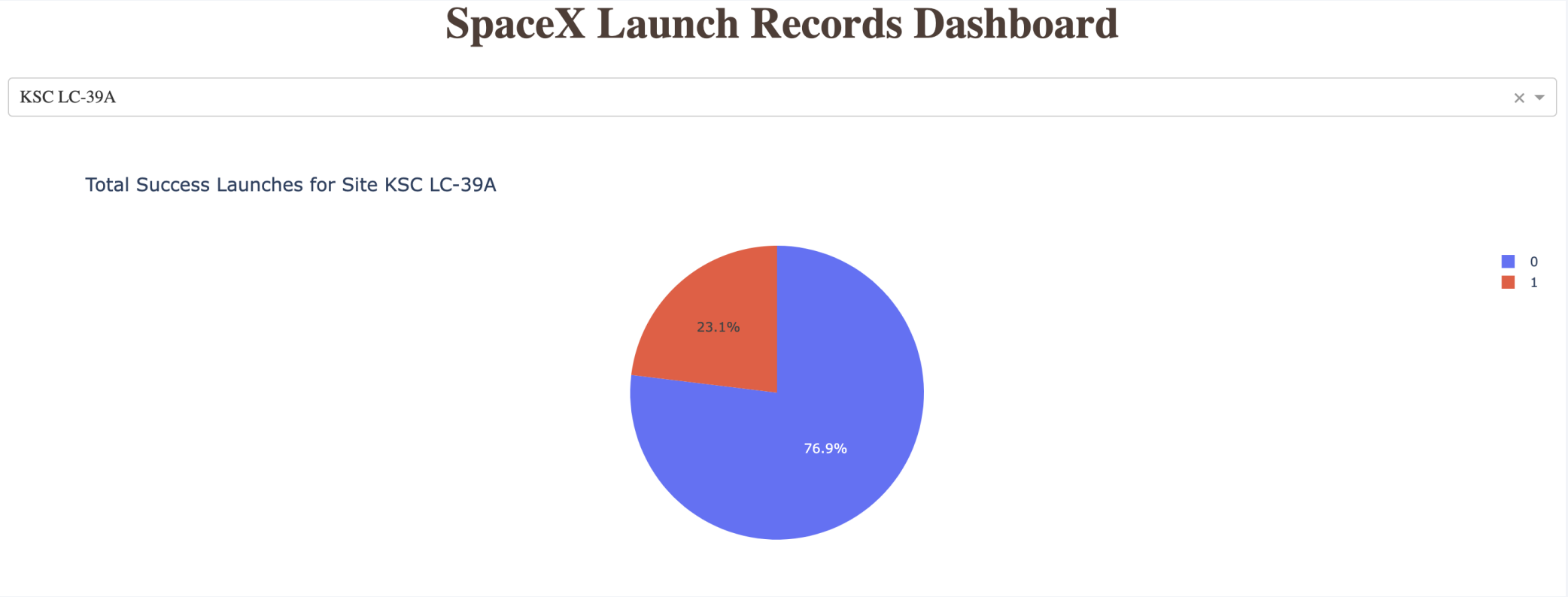
Piechar of Total Success Launches by Sites

We could see that KSC LC-39A had the most success launches with 41.2%.



The greatest success launch ratio piechart

- KSC LC-39A had the greatest success launch ratio of 76.9%



Correlation between Payload and Success rate for all launch sites scatter plot

Payload upto 3000

Payload between 4000 and 10000.

We could clearly see that the success rate for Payload up to 4000 is a lot higher than payload between 4000-10000. To be spcific, Booster Versions FT had the most success launch rate.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Add all the best_score results for accuracy for each model in a dataframe and visualized using sns.barplot and the **Tree** model had the highest accuracy classification of 87%.

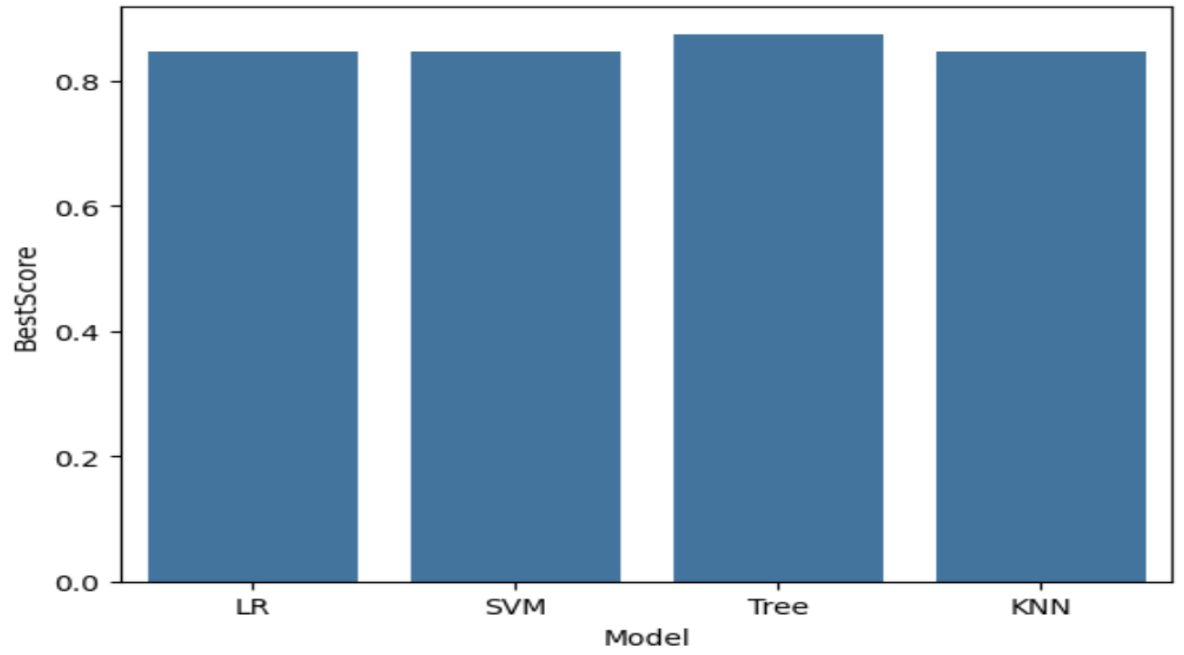
[16]:
```
results = {'Model':['LR','SVM','Tree','KNN'], 'BestScore':[0.846,0.848,0.875,0.848]}
df_r = pd.DataFrame.from_dict(results)
df_r
```

[16]:

	Model	BestScore
0	LR	0.846
1	SVM	0.848
2	Tree	0.875
3	KNN	0.848

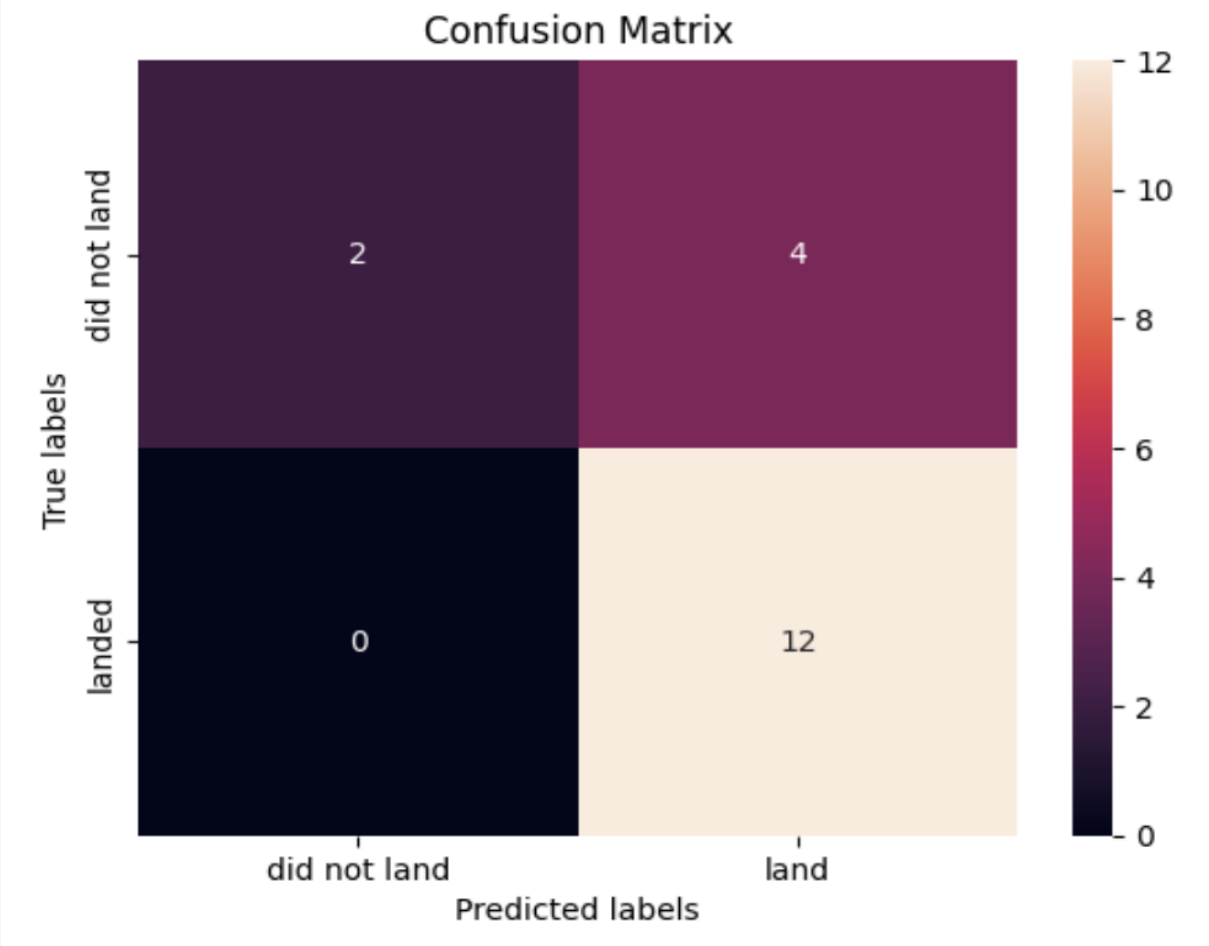
[18]:
```
sns.barplot(df_r, x="Model", y="BestScore")
```

[18]:
```
<AxesSubplot:xlabel='Model', ylabel='BestScore'>
```



Confusion Matrix of the best performing model

- Confusion matrix of the Tree model: it shows that the classifier is doing a great job in distinguishing between classes. The only concerning part is the false positive of 2 but compared to the other models, it's the best performing model.



Conclusions

- A direct coralation between flightnumber and success rate. (higher fligh == higher success rate for launch sites).
- Yearly average success rate shows the sucess rate since 2013 kept increasing till 2020.
- direct coralation between PayloadMass and success rate for orbits LEO, ISS, SSO, and PO.
- Orbit ES-L1, GEO, HEO, and SSO had super high success rate, While GTO with the lowest success rate.
- Decision Tree had the highest accuracy among machine learning predicaiton models.

Appendix

- Python code snippets for task 7 SQL queries: Created sub 7a and 7b to get better understanding of the outcome of the query as some of the entries were not entered as failure but instead failed or something similar.

Task 7

List the total number of successful and failure mission outcomes

```
print('98 is supposed to be sucess count')
%sql SELECT COUNT(Mission_Outcome) AS succes_count FROM SPACEXTBL GROUP BY Mission_Outcome;
```

```
98 is supposed to be sucess count
 * sqlite:///my_data1.db
Done.
```

succes_count
1
98
1
1

Task 7 but redone (7a*)

List the total number of successful outcomes

```
%sql SELECT COUNT(Mission_Outcome) AS succes_count FROM SPACEXTBL WHERE Mission_Outcome == 'Success';
```

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

succes_count
98

Task 7 but redone (7b*)

List the total number of failure mission outcomes

```
%sql SELECT COUNT(Mission_Outcome) AS fail_count FROM SPACEXTBL WHERE Mission_Outcome != 'Success';
```

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

fail_count
3

Thank you!