IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rakeshsarma Karra

Mar 04, 2025

Outline

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

Executive Summary

Performed data analysis and modeling on space X falcon 9 data to get an understanding about the success and failure rates of flight landings.

Summary of methodologies

- Data collection
- Data wrangling
- Exploratory Data Analysis using visualization and SQL
- Interactive visual analytics using Folium and Plotly Dash Correlations
- Predictive analysis using classification models

Summary of all results

Model	Training Accuracy	Testing Accuracy
Logistic Regression	84.64%	83.33%
Support Vector Machines	84.82%	83.33%
Decision Tree Classifier	87.32%	83.33%
K-Nearest Neighbors	84.82%	83.33%

Introduction

Project background and context

In this capstone, I worked with a problem statement about the success and failure(ratio's) of Falcon 9 first stage landings. 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 I can determine if the first stage will land, I 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.

Problems you want to find answers

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 launch success rate?

Section 1

Methodology

Data Collection

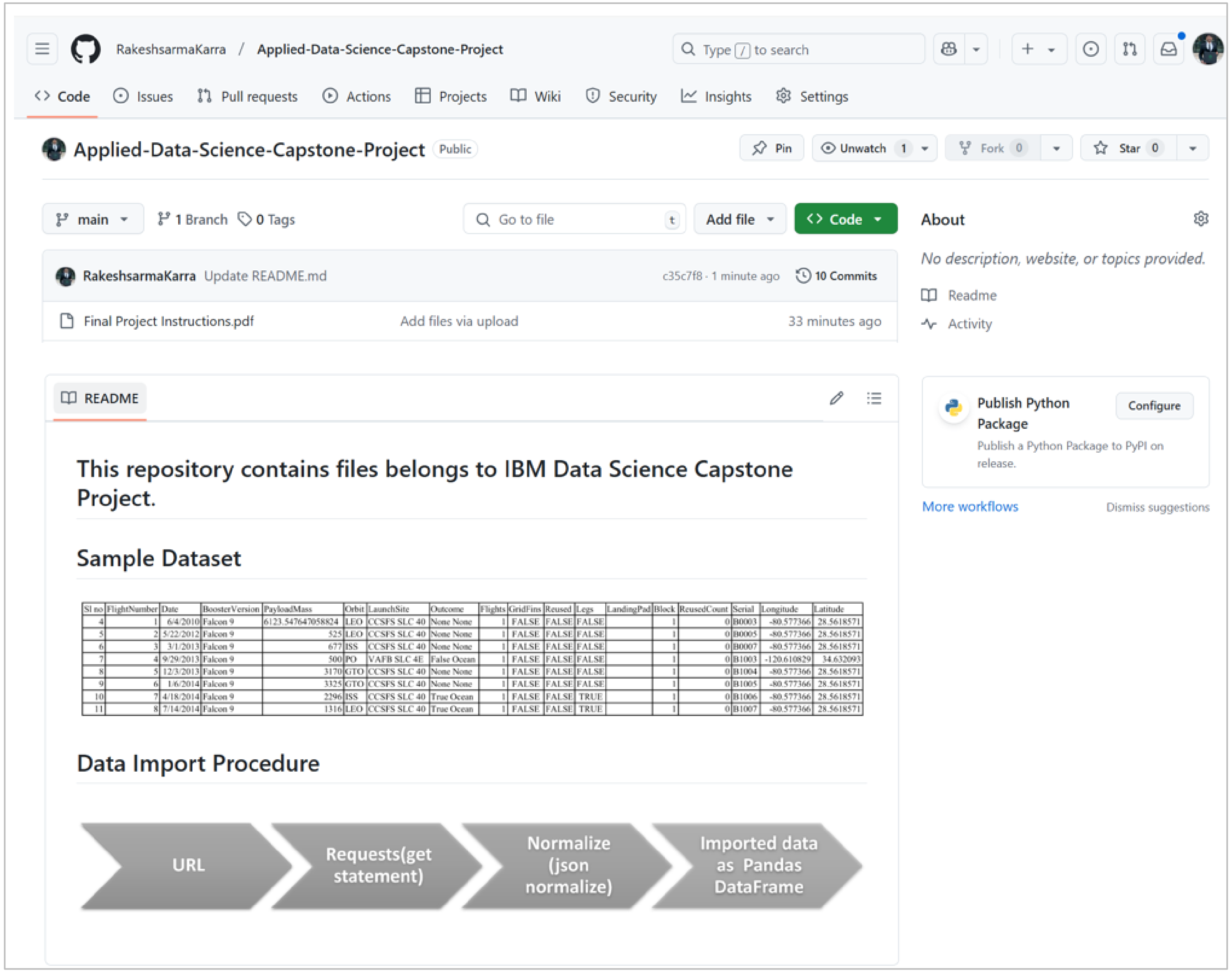
Total 5 datasets are collected for the project from the internet(Web page: Space X).

1. **Rockets**
2. **Launchpads**
3. **Payloads**
4. **Cores**

Major work of the project is based on the dataset names **API_call_spacex_api.json**.
“It’s actually a json file. I normalized the data using **json_normalize** function to convert it into a structured data.

Data collection process use key phrases and flowcharts
URL = ‘https://api.spacexdata.com/v4/datafilename.ext’
request = requests.get(URL)
Data = pd.DataFrame(request[3]) #Place the table number

GitHub Repository



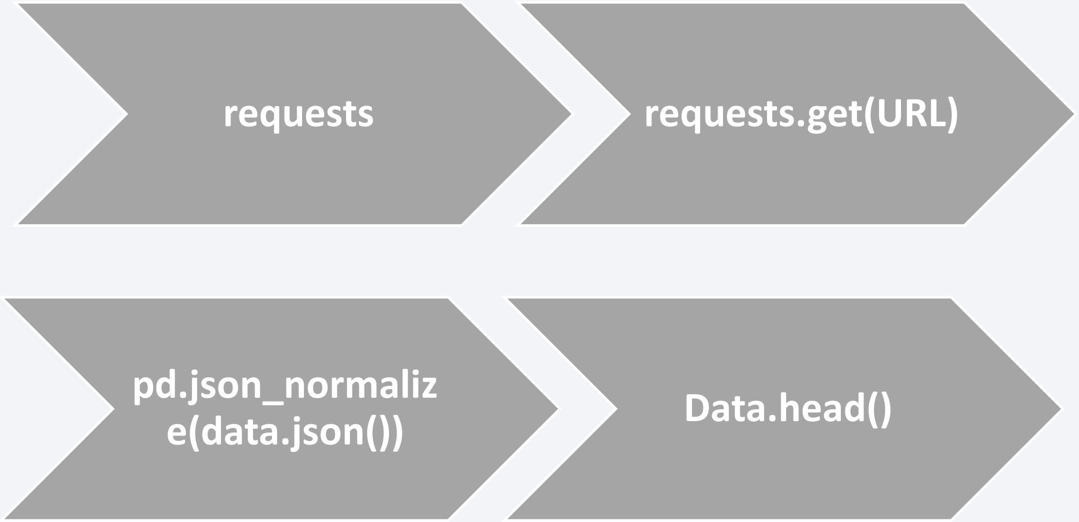
Please, click on the image to view the repository.

Data Collection – SpaceX API

- Imported libraries requests, pandas, numpy to perform web scrapping and data transformations.
- Data is collected form the space X web page.

GitHub URL

- https://github.com/RakeshsarmaKarra/Applied-Data-Science-Capstone-Project/tree/main



Data Collection - Scraping

- Initially to access the data from web, sent the request in form of json.
- After getting the access (i.e status code as 200), converted the json file into pandas dataframe using json_normalize function.

GitHub URL
https://github.com/RakeshsarmaKarra/Applied-Data-Science-Capstone-Project/blob/main/jupyter-labs-webscraping.ipynb

```
[ ]  # use requests.get() method with the provided static_url
     # assign the response to a object
     data = requests.get(static_url)
     data = data.text
```

Create a `BeautifulSoup` object from the HTML `response`

```
[ ]  # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
     Soup = BeautifulSoup(data, "html.parser")

```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
[ ]  # Use soup.title attribute
     Soup.title.text
```

```
'List of Falcon 9 and Falcon Heavy launches - Wikipedia'
```

Data Wrangling

- Initially, I focused on the accuracy of the dataset.
- Next, generated visualizations on all columns to understand the distribution of the data points.

Steps followed:

Data Cleaning

Null values

NA values

Data Understandings

Value counts for the columns LaunchSite, Orbit, Outcome

Data Preprocessing

Creating target column landing_class based on good and bad outcomes.

GitHub Link:
https://github.com/RakeshsarmaKarra/Applied-Data-Science-Capstone-Project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Catplot:

- To find the payload mass(kg) variations for each flight number.
- To find which launch site launches each flight number.
- To find which launch site launches according to the pay load mass(kg).

Bar chart:

- To find class variations on each orbit.
- To find payload mass(kg) variations on each orbit.

Scatter plot:

- To find flight numbers with each orbit.

Line plot:

- To find success rates according to the date of launch.

GitHub Link:
https://github.com/RakeshsarmaKarra/Applied-Data-Science-Capstone-Project/blob/main/edadataviz.ipynb

EDA with SQL

- Using **create table** statement, created a table SPACEXTABLE excluding date column with null values.
- Using **distinct** statement, extracted unique Launch_Site values.
- Using **like** operator extracted CCA launch site values.
- Using **sum** function, calculated payload mass(kg) for NASA (CRS) customers.
- Using **min** function, extracted the first launch date for the successful flight landing launch outcome.
- Using **sub queries**, extracted booster_versions for maximum payload mass(kg).
- Using **substr**, extracted month and year from dates, to find landing outcome, launch site, booster versions in the year 2015 for failure (drone ship) landing outcomes.

Github Link:

- https://github.com/RakeshsarmaKarra/Applied-Data-Science-Capstone-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- With the help of folium library, I have created an interactive map visualizations.
- Initially, I subset the data to achieve latitude & longitude values for each launch site.
- Using circle function, created a circle(radius of 1000miles) on each launch site.
- Using marker function, created a marker with the customizations like icon_size, icon_anchor for these co-ordinates.
- To get lines for each distance, I calculated distance between launch site and selected place.
- With the help of folium.map and folium.marker functions, I created lines to attach these two places.

GitHub Link:

- https://github.com/RakeshsarmaKarra/Applied-Data-Science-Capstone-Project/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- With the help of plotly and dash libraries, created an interactive dashboard.
- Using Dropdown option from dcc library, created options like ALL Sites, site1.
- Using callback function, pass the input variables and extracted the required figures, tables etc. by keeping them in output statement.
- Using RangerSlider function from dcc library, created a slider to select pay load mass(kg) value.
- Questions I tried to solve:
    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 launch success rate?

GitHub Link:
https://github.com/RakeshsarmaKarra/Applied-Data-Science-Capstone-Project/blob/main/Final%20Project.py

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- Before modeling, I performed data preprocessing to convert categorical columns into numerical.
- Predicted variable(Y): Class(data points: 0, 1)
- Predictor variable(X): Remaining columns
- Using sklearn imported preprocessing library and called StandardScalar() function to standardize the required columns.
- Performed train test split for training and testing the model performance.
- From sklearn imported module model_selection to call GridSearchCV to find the best hyperparameters.
- Practiced 4 classification models:

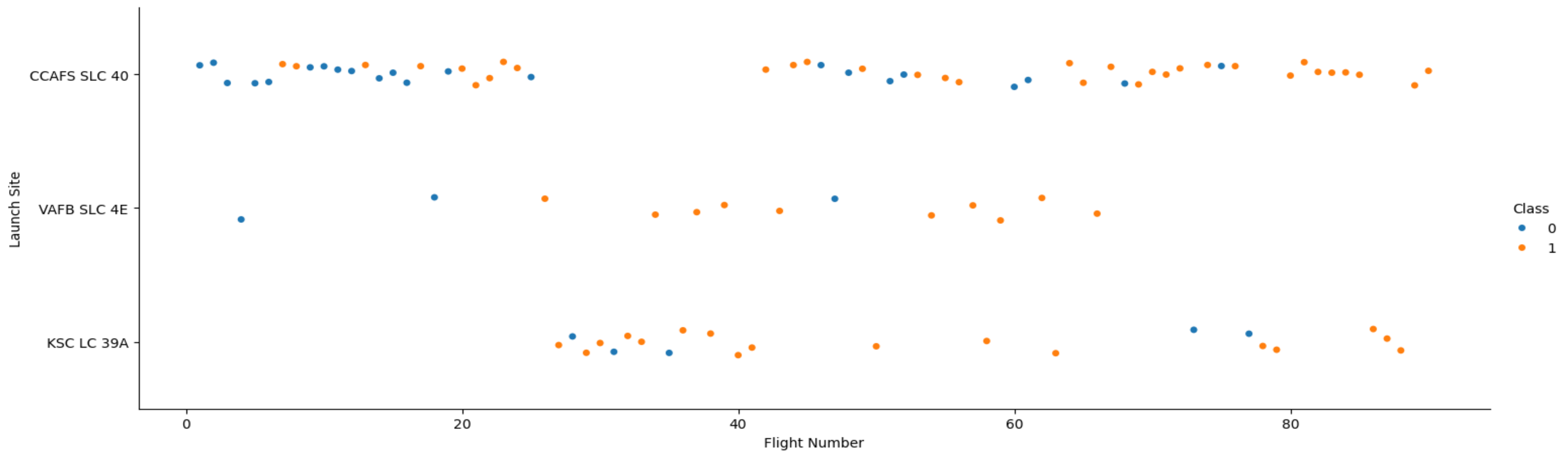
Model Name	Parameters
Logistic Regression	{"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}
Support Vector Machines	{'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'), 'C': np.logspace(-3, 3, 5), 'gamma':np.logspace(-3, 3, 5)}
Decision Tree Classifier	{'criterion': ['gini', 'entropy'], 'splitter': ['best', 'random'], 'max_depth': [2*n for n in range(1,10)], 'max_features': ['auto', 'sqrt'], 'min_samples_leaf': [1, 2, 4], 'min_samples_split': [2, 5, 10]}
K-Nearest Neighbors	{'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'], 'p': [1,2]}

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

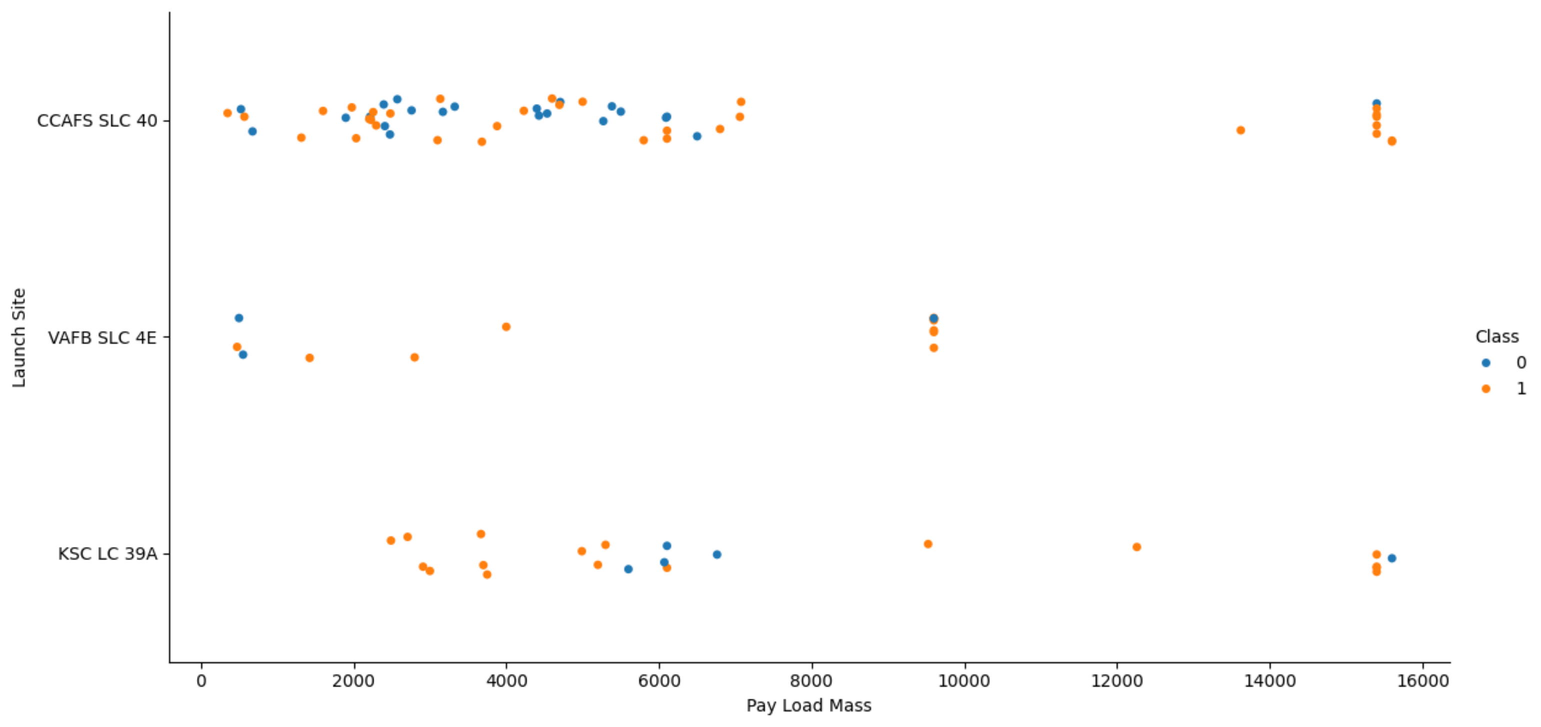
```python
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
plt.figure(figsize=(8,12))
sns.catplot(x = 'FlightNumber', y='LaunchSite', hue='Class', data=df, aspect=3)
plt.xlabel('Flight Number')
plt.ylabel('Launch Site')
plt.show()
```



More number of flights are launched from the site **CCAF SLC 40**. Flightnumbers from 25 to 41 approximately are launched from the site **KSC LC 39A**

Payload vs. Launch Site

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y='LaunchSite', x='PayloadMass', data=df, hue='Class', aspect=2, height=6, width=4)
plt.ylabel('Launch Site')
plt.xlabel('Pay Load Mass')
plt.show()
```

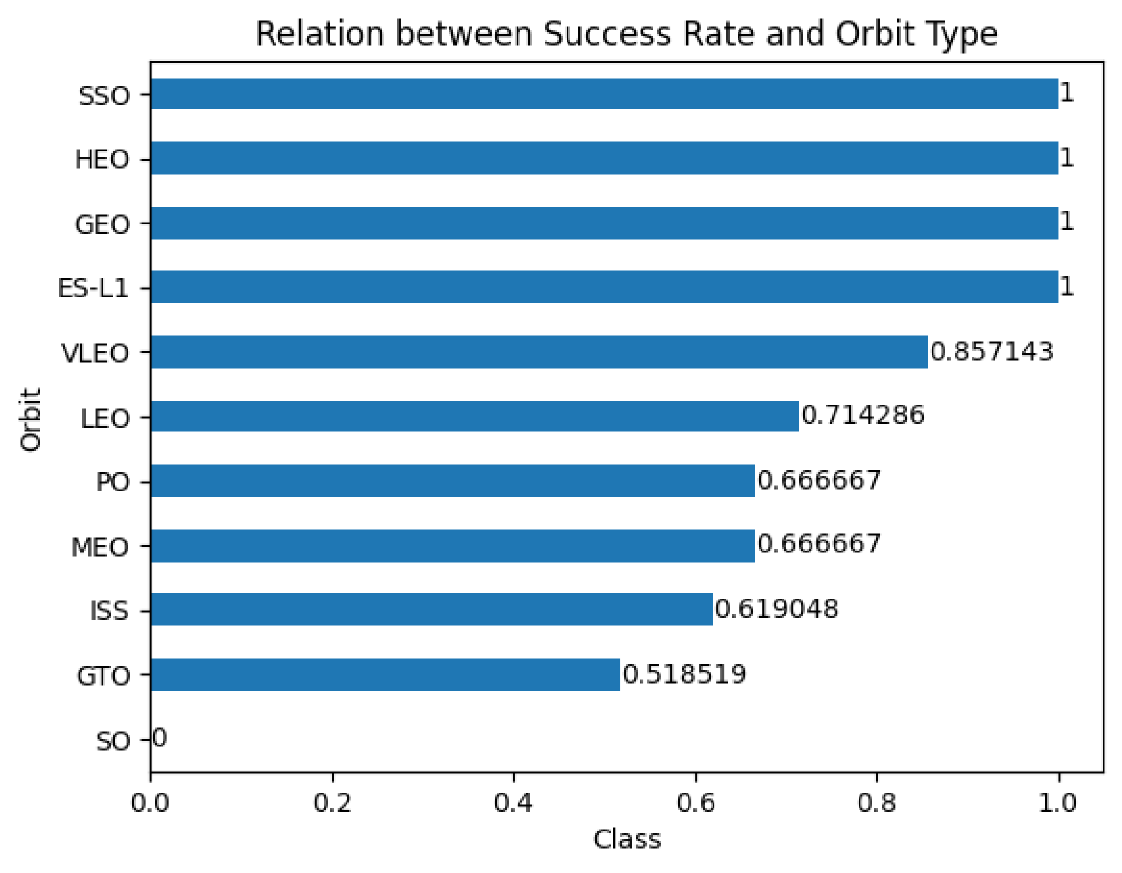


Now if you observe Payload Mass Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type

```
success_rate = pd.DataFrame(df.groupby(['Orbit'])['Class'].mean())
success_rate = success_rate.sort_values(by = 'Class', ascending=True)
success_rate
```

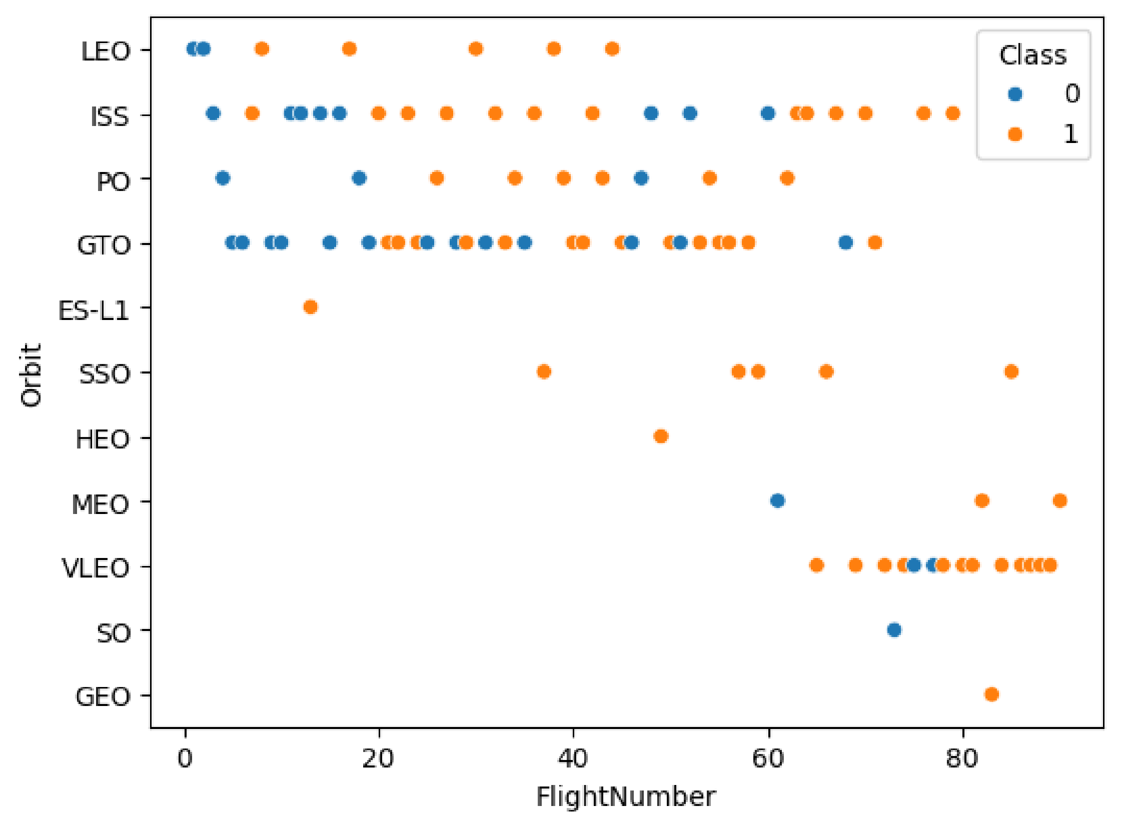
```
ax = success_rate.plot(kind='barh',
                            y = 'Class')
ax.set_ylabel('Orbit')
ax.set_xlabel('Class')
ax.set_title('Relation between Success Rate and Orbit Type')
ax.bar_label(ax.containers[0])
ax.legend_.remove()
plt.show()
```



Orbits SSO, HEO, GEO, ES-L1 has 100% success ratio where SO has 0%.

Flight Number vs. Orbit Type

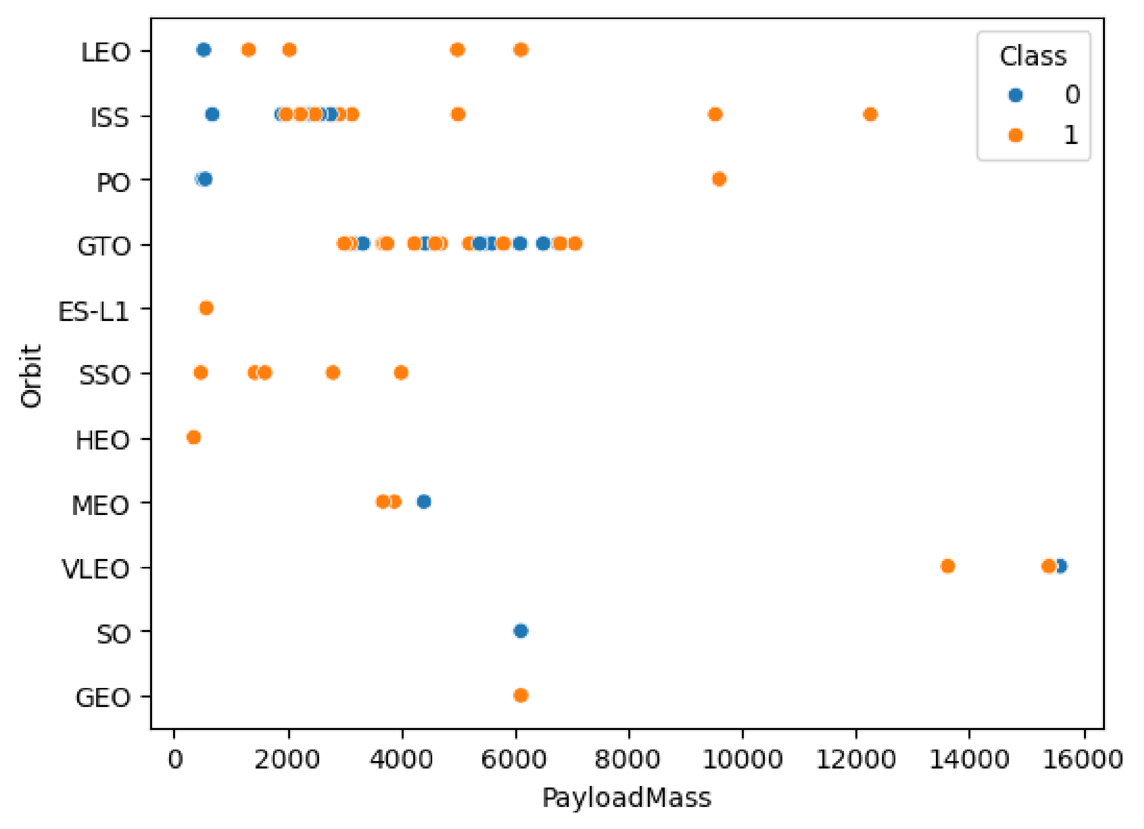
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.scatterplot(x='FlightNumber', y='Orbit', data=df, hue='Class')
plt.xlabel('FlightNumber')
plt.ylabel('Orbit')
plt.show()
```



You can 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.

Payload vs. Orbit Type

```python
# Plot a scatter point chart with x axis to be Payload Mass and y axis to be the Orbit, and hue to be the class value
sns.scatterplot(x='PayloadMass', y='Orbit', data=df, hue='Class')
plt.xlabel('PayloadMass')
plt.ylabel('Orbit')
plt.show()
```

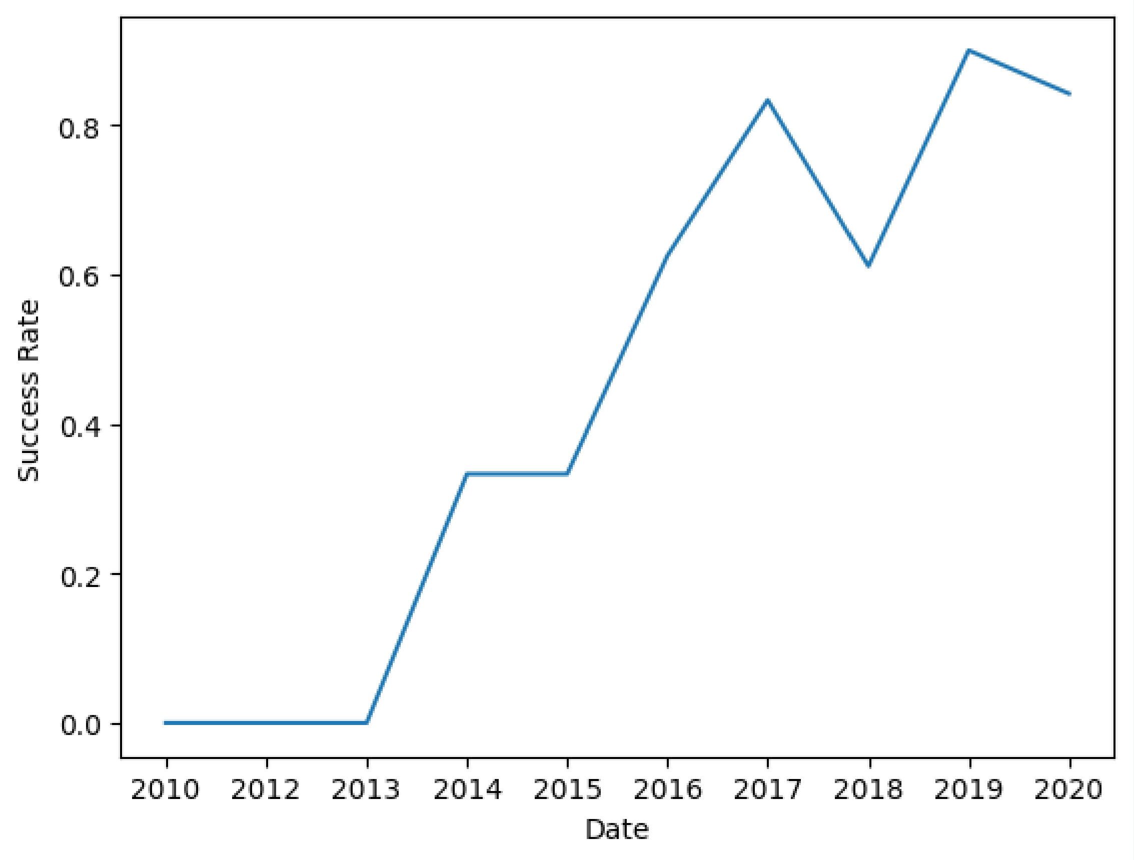


With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
sns.lineplot(x='Date',y='Class',data=su_mean)
plt.xlabel('Date')
plt.ylabel('Success Rate')
plt.show()
```



Success rate started increasing from 2013.

All Launch Site Names

- There are 3 launch sites in the data.
- Majority of the flight launches are from CCAFS SLC 40 site.

```
df.shape
df['LaunchSite'].value_counts()
```

	count
LaunchSite	
CCAFS SLC 40	55
KSC LC 39A	22
VAFB SLC 4E	13

dtype: int64

Launch Site Names Begin with 'CCA'

- `%sql select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5;`

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

- These are the 5 sample records.
- Majority of the flights launched from this site are from Orbit.

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) as Total_Payload_Mass from SPACEXTBL \
    where Customer like 'NASA (CRS)';
```

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

Total_Payload_Mass
45596

- Total payload mass(kg) is 45596.

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) as AVG_PAYLOAD_MASS__KG from SPACEXTBL \
    where Booster_Version like 'F9 v1.1%';


```

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

AVG_PAYLOAD_MASS__KG
2534.6666666666665

- Average payload mass(kg) is 2534.67

First Successful Ground Landing Date

```
1 %sql select min(Date) as First_Date from SPACEXTBL where Landing_Outcome = 'Success';
```

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

First_Date
2018-07-22

- First successful landing date is 2018-07-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select distinct Booster_Version from SPACEXTBL where \
    (PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 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

- These are the booster versions whose payloads mass are in between 4000 and 6000 and landing outcome is Success (drone ship).

Total Number of Successful and Failure Mission Outcomes

```
%sql select distinct Mission_Outcome, count(Mission_Outcome) as No_of_Outcomes from SPACEXTBL \
    group by Mission_Outcome;
```

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

Mission_Outcome	No_of_Outcomes
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- Majority of the missions are success.

Boosters Carried Maximum Payload

```
%sql select distinct Booster_Version as Names_of_BoosterVersions from SPACEXTBL where \
    PAYLOAD_MASS__KG_ in (select max(PAYLOAD_MASS__KG_) from SPACEXTBL group by PAYLOAD_MASS__KG_);

```

Names_of_BoosterVersions
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
F9 v1.1 B1015
F9 v1.1 B1016
F9 v1.1 B1018
F9 FT B1019
F9 v1.1 B1017
F9 FT B1020
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1024
F9 FT B1025.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1031.1
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 FT B1034
F9 FT B1035.1
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1037
F9 B4 B1039.1
F9 FT B1038.1
F9 B4 B1040.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 FT B1035.2
F9 FT B1036.2
F9 B4 B1043.1
F9 FT B1032.2
F9 FT B1038.2
F9 B4 B1044
F9 B4 B1041.2
F9 B4 B1039.2
F9 B4 B1045.1
F9 B5 B1046.1
F9 B4 B1043.2
F9 B4 B1040.2
F9 B4 B1045.2
F9 B5B1047.1
F9 B5B1048.1
F9 B5 B1046.2
F9 B5B1049.1
F9 B5 B1048.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5B1050
F9 B5B1054
F9 B5 B1049.2
F9 B5 B1048.3
F9 B5B1051.1
F9 B5B1056.1
F9 B5 B1049.3
F9 B5 B1051.2
F9 B5 B1056.2
F9 B5 B1047.3
F9 B5 B1048.4
F9 B5B1059.1
F9 B5 B1056.3
F9 B5 B1049.4
F9 B5 B1046.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1059.2
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5B1058.1
F9 B5 B1049.5
F9 B5 B1059.3
F9 B5B1060.1
F9 B5 B1058.2
F9 B5 B1051.5
F9 B5 B1049.6
F9 B5 B1059.4
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5B1062.1
F9 B5B1061.1
F9 B5B1063.1
F9 B5 B1049.7
F9 B5 B1058.4

2015 Launch Records

```
%sql select distinct substr(Date, 6,2) as Month_no, Landing_Outcome, Booster_Version, Launch_Site \
    from SPACEXTBL where Landing_Outcome = 'Failure (drone ship)' and substr(Date, 0,5) = '2015' ;
```

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

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

- Drone ship landing missions was failed in January and April months in the year 2015

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

```
%sql select distinct Landing_Outcome, count(Landing_Outcome) as No_of_outcomes from SPACEXTBL \
    where Date between '2010-06-04' and '2017-03-20' group by Landing_Outcome order by No_of_outcomes desc;
```

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

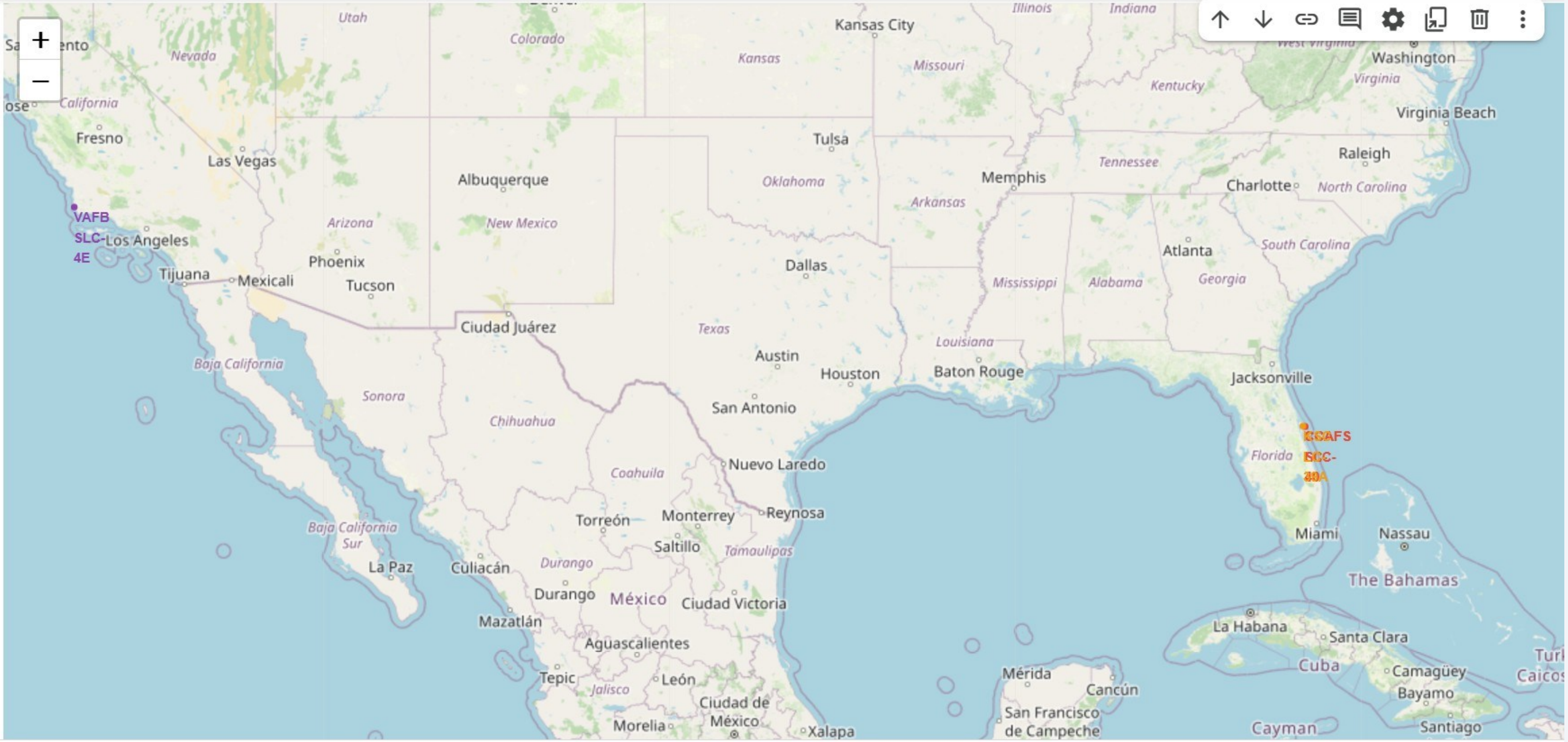
Landing_Outcome	No_of_outcomes
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- Rank of failure (drone ship) is 2 (No of outcomes is 5) and success (ground pad) is 2 (No of outcomes is 3).

Section 3

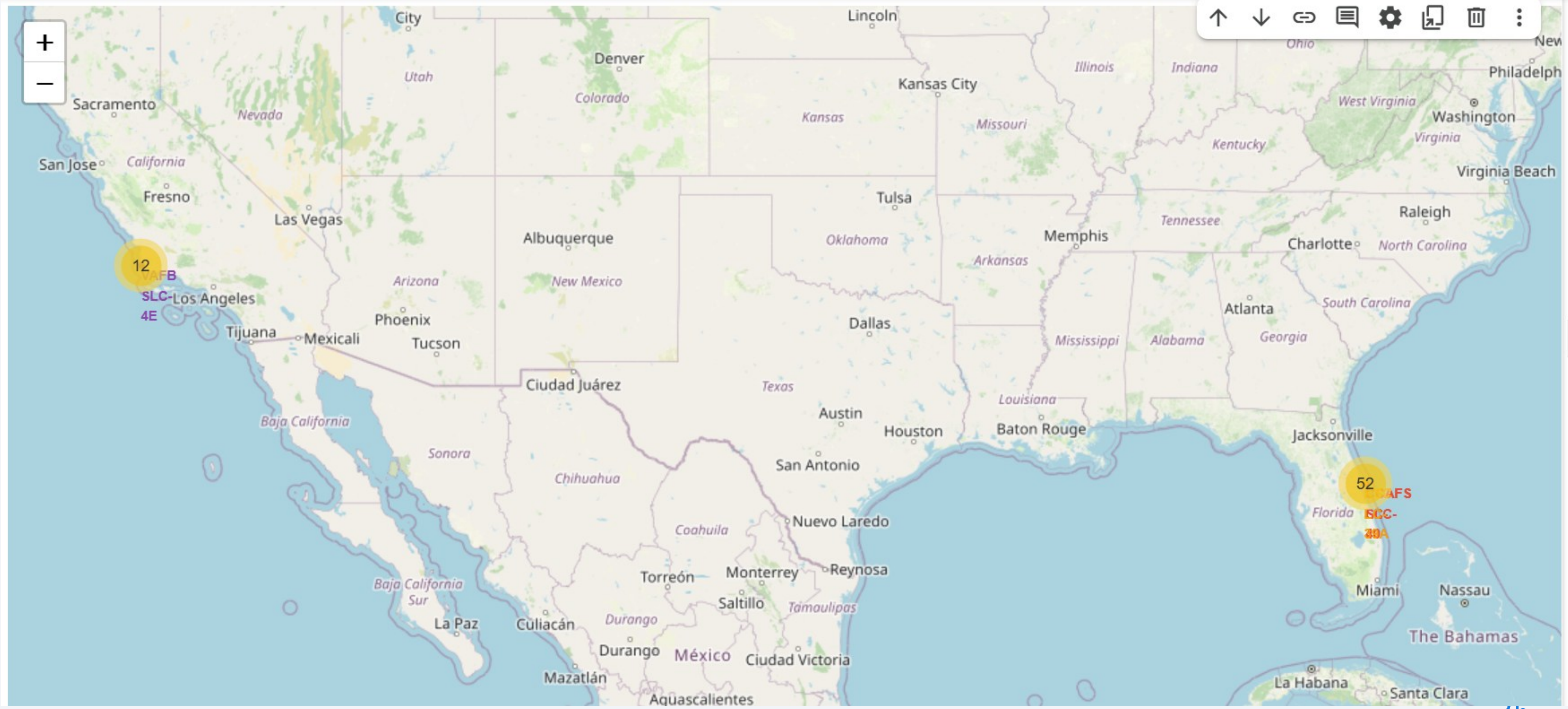
Launch Sites Proximities Analysis

Launch Sites Map Visualizations



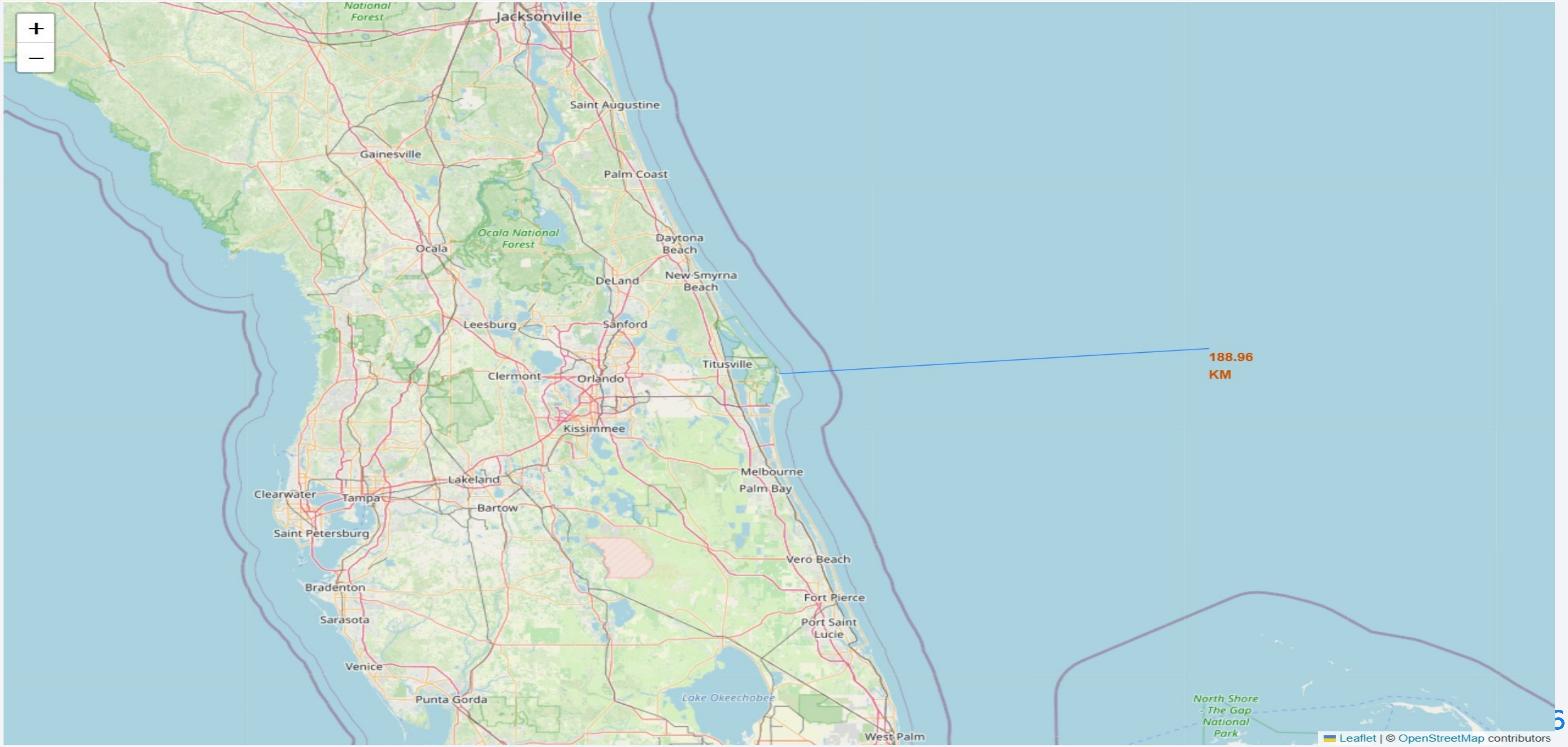
- Latitude's and longitude's are not near to 0. So, these coordinates are not in proximity to the equator line.
- If you view the map, you will understand that these launch sites are near the ocean.

Launch Sites with Flight Launch Details



- Majority of the launches are from south side of America (52 launches).

Launch Sites with it's Proximities

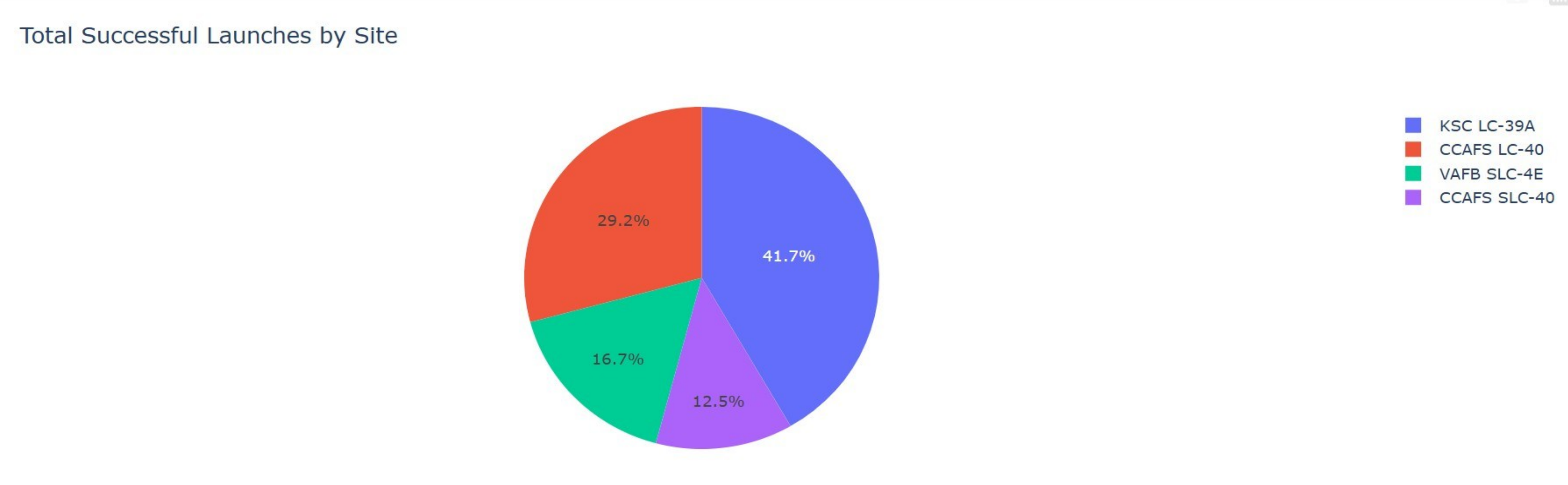


- Generated line is the distance between CCAFS LC-40 launch site and my selected coastal line.

Section 4

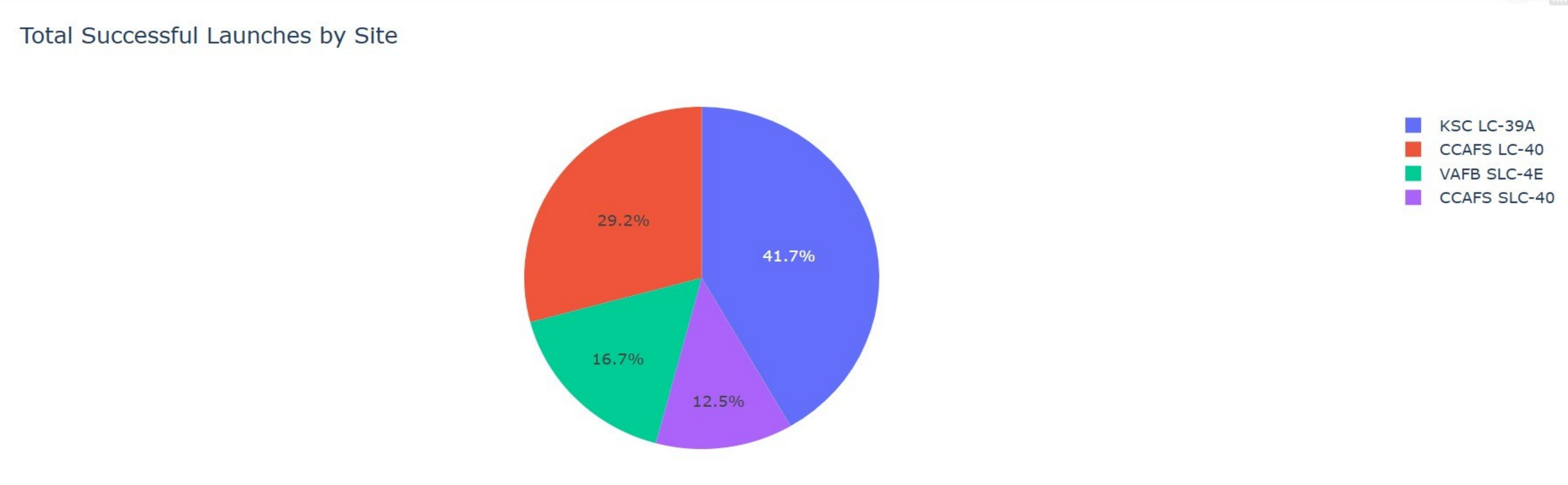
Build a Dashboard with Plotly Dash

Launch Success Counts for All Sites



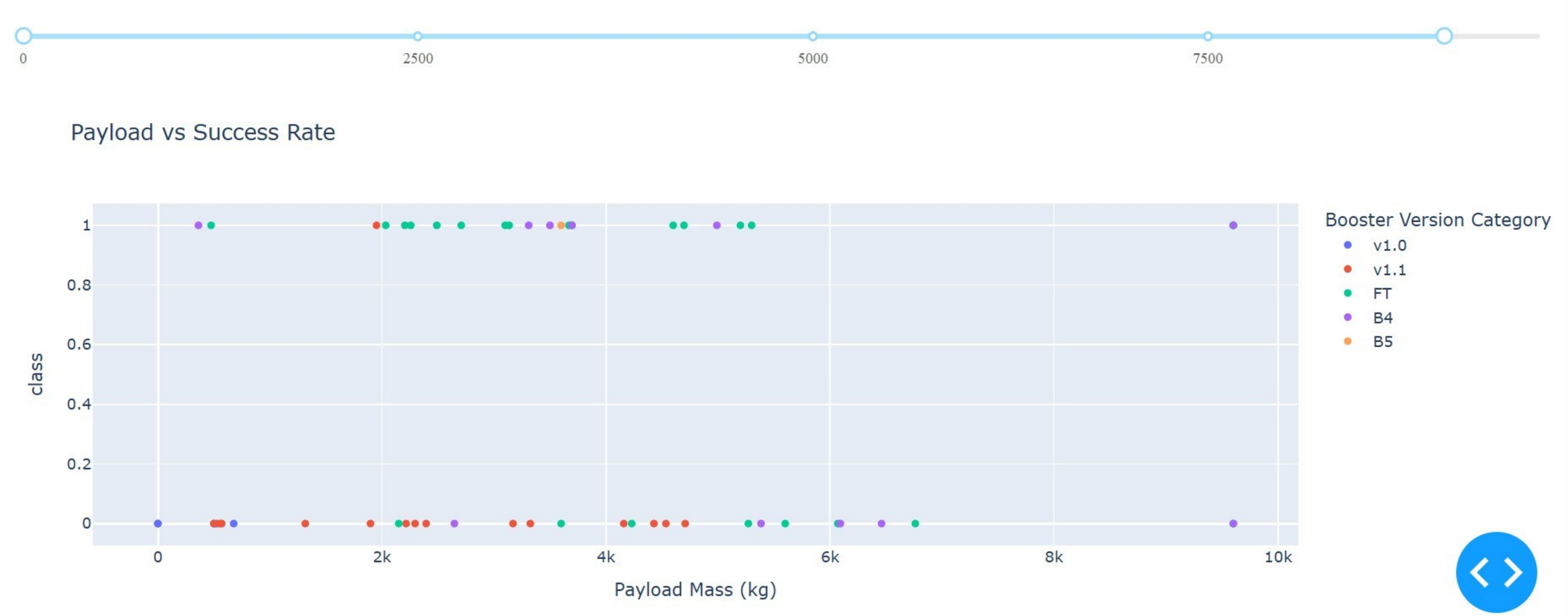
- KSC LC - 39A launch site has highest launch success count(10 launches).
- CCAFS SLC - 40 launch site has lowest launch success count(3 launches).

Pie Chart – Highest Launch Success Ration



- KSC LC - 39A launch site has highest launch success ratio.
- CCAFS SLC - 40 launch site has lowest launch success ratio.

Payload Vs Success Ratio

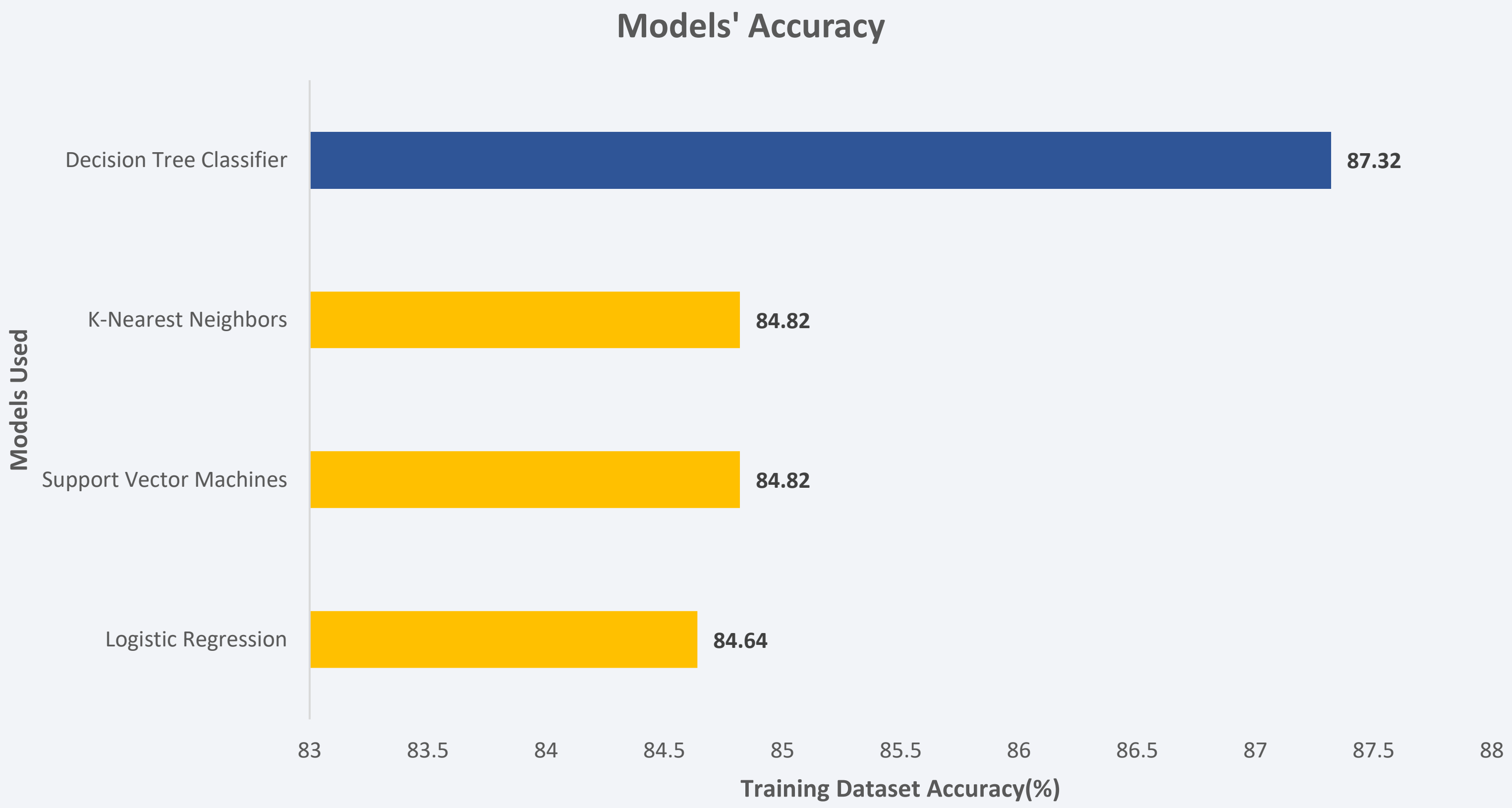


- I didn't see any strong relationship between pay load mass and success ratio.
- FT booster version category contains highest success rates at payload mass (kg) 7000 which is selected from the range slider.

Section 5

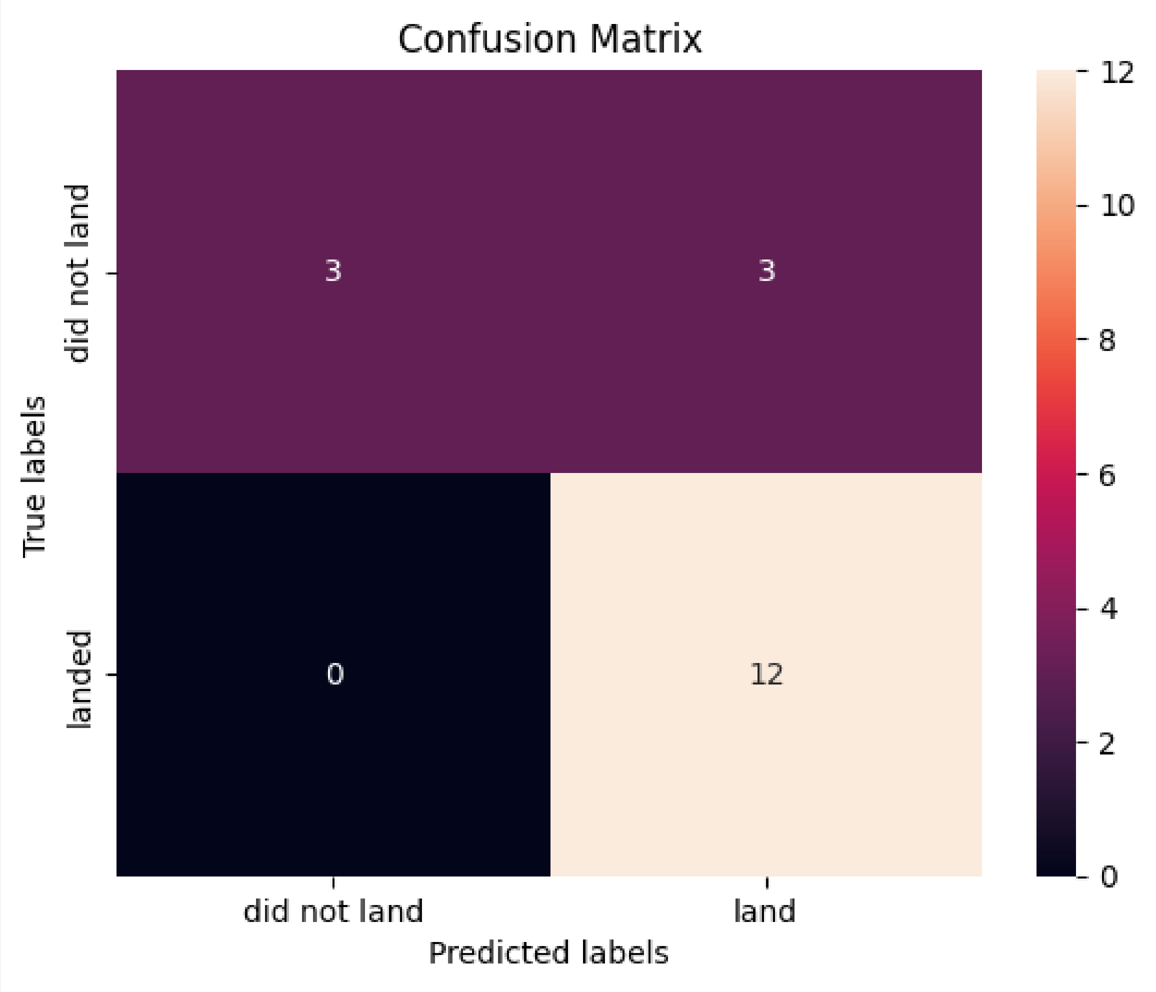
Predictive Analysis (Classification)

Classification Accuracy



- Out of all the 4 models, decision tree classifier performed well.

Confusion Matrix



- Decision Tree Classifier model predicted unsuccessful landing outcomes correctly, but 3 outcomes are wrongly predicted as unsuccessful landings.

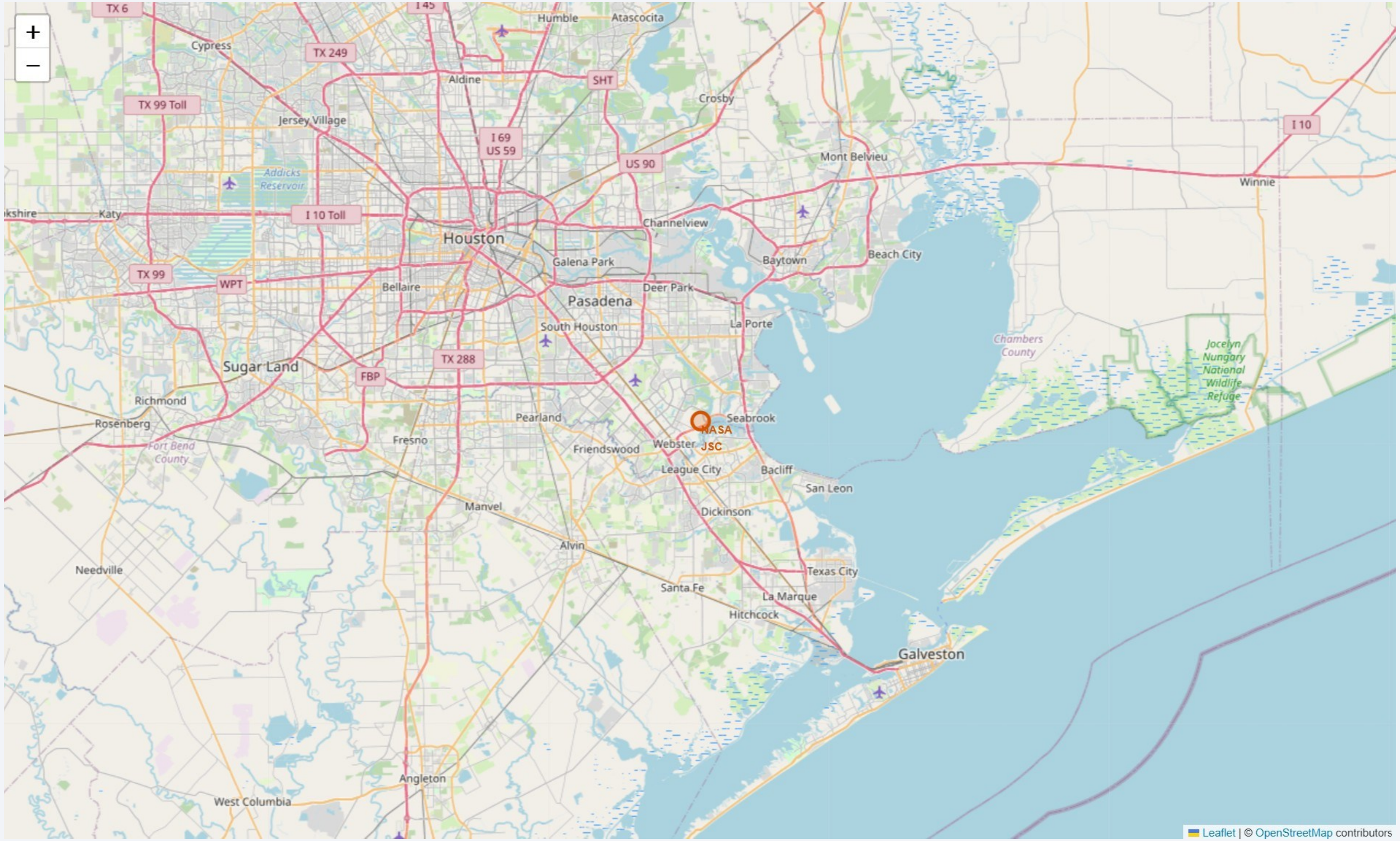
Conclusions

- Predictive Analytics is successful completed whith out underfitting or overfitting the data.
- Out of all the models we have practiced, Decision Tree Classifier performed well.
- Highest launch success ratio is from the launch site **CCAFS SLC - 40**
- Successfully landed flights after their launch is high in south of the America.

Future Work

- These is data weightage issue in the dataset.
- More number of data points are from launchsite **CCAFS SLC – 40**.
- I would like to add more data points on remaining launch sites and make it into a consistent dataset.

Appendix



Houston
Pasadena
Sugar Land
Galveston
NASA JSC
Seabrook
Webster
League City
Friendswood
Pearland
Baytown
La Porte
Deer Park
Galena Park
Channelview
South Houston
Bellaire
Katy
Cypress
Humble
Atascocita
Aldine
Crosby
Jersey Village
Mont Belvieu
Beach City
Winnie
Richmond
Rosenberg
Fresno
Manvel
Alvin
Dickinson
Bacliff
San Leon
Texas City
La Marque
Hitchcock
Santa Fe
Needville
Angleton
West Columbia
Chambers County
Fort Bend County
Jocelyn Nungary National Wildlife Refuge
Addicks Reservoir
TX 6
TX 249
TX 99 Toll
TX 99
TX 288
I 10 Toll
I 10
I 69 US 59
US 90
SHT
WPT
FBP
Leaflet | © OpenStreetMap contributors

Thank you!