Data Science Capstone project

Outline

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

Executive Summary

- I collected data on Falcon 9 Launches from SpaceX and analyzed the data using python data analysis and visualization libraries to determine the success and cost of SpaceX Falcon 9 launches.

- The results show that the success or failure of the SpaceX Falcon 9 launch can be predicted using launch data.

Introduction

- SpaceX is and aerospace manufacturer that launches multiple rockets including the Falcon 9. The cost of a launch is 62 million dollars compared to 165 million for other providers. The savings are due the ability of SpaceX to land and reuse the first stage of the Falcon 9.

- Not all launches and landings are successful, which creates a large discrepancy between launch costs. I will be predicting whether or not the Falcon 9 first stage will land successfully, allowing SpaceX to provide substantial savings.

Methodology

- Data collection methodology:
  - Describe how data were collected
- Perform data wrangling
  - Describe how data were processed
- 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

Methodology

Data collection

- The data on Falcon 9 launches was collected using a SpaceX REST API as well as web scraping the SpaceX wikipedia page. The data was then cleaned to show only Falcon 9 launches.

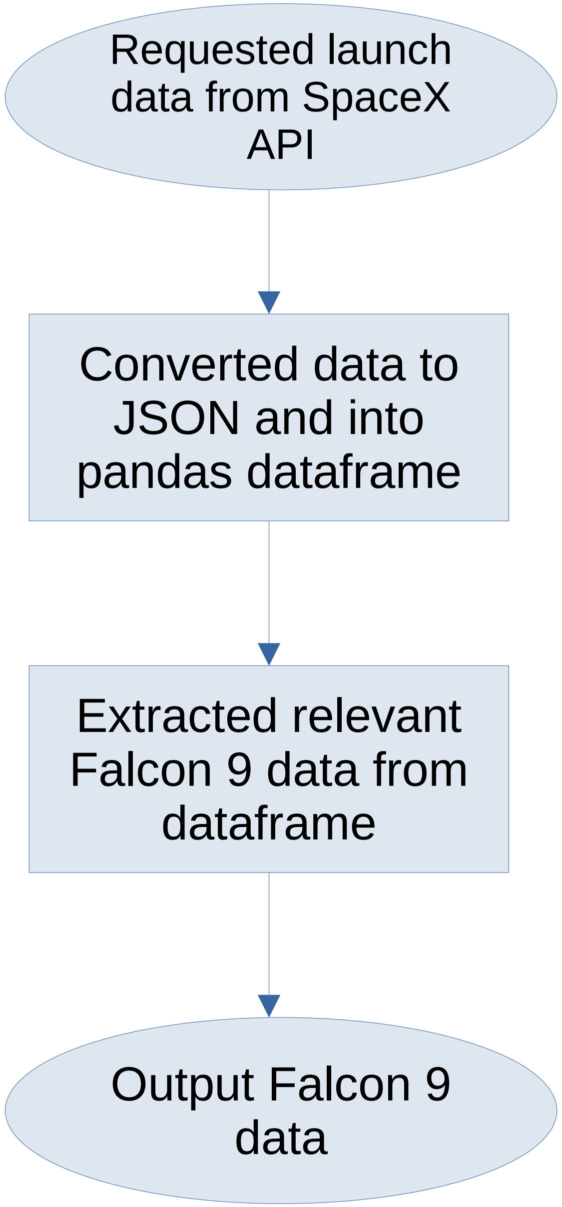
- A get request was made to the SpaceX REST API and the Beautiful Soup library was made to web scrape the SpaceX wikipedia page.

Data collection – SpaceX API

Requested rocket launch data from the SpaceX API using a get request and normalized to json into a Pandas dataframe called data.

Extracted relevant Falcon 9 data to be analyzed.

https://github.com/neutronrats/spacex-capstone/blob/cfb7f702c963a93fec82940d75e9b821ae391ca8/jupyter-labs-spacex-data-collection-api_MH.ipynb



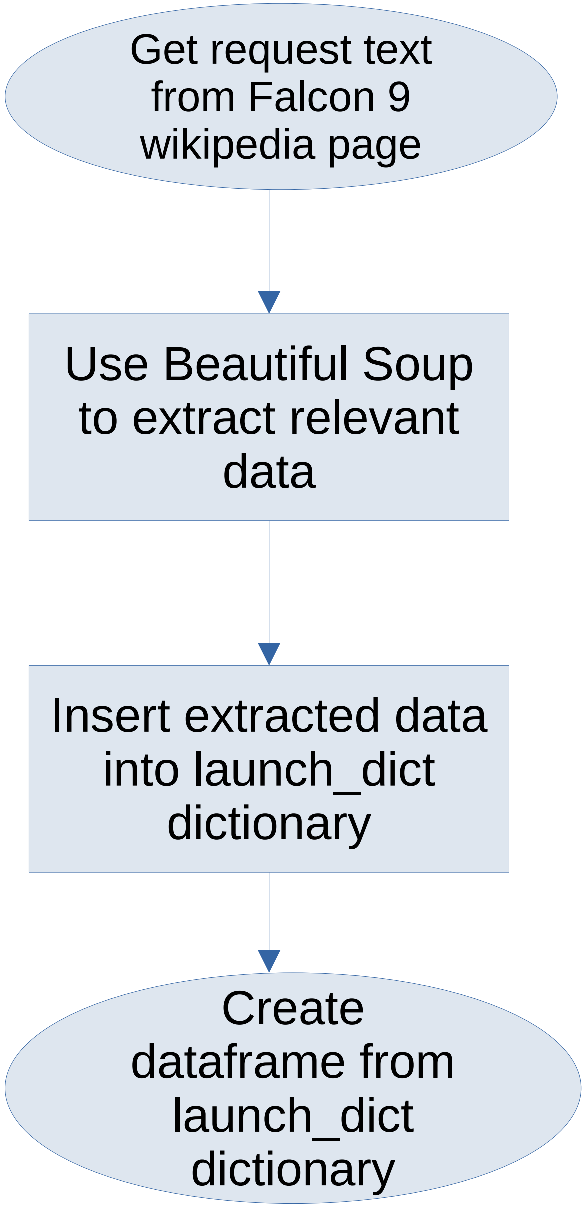
Data collection – Web scraping

Used a get request to download the text from the SpaceX Falcon 9 Launch wikipedia page. Then used the Beautiful Soup library to extract the relevant information.

The launch information was then added to a launch_dict dictionary.

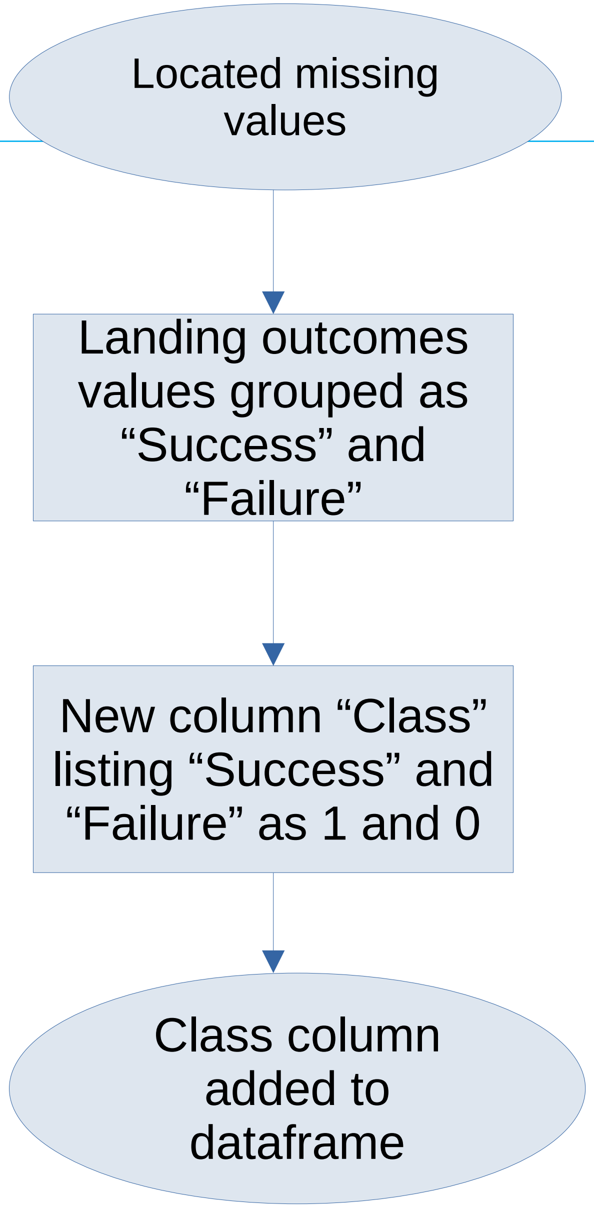
I then created a dataframe from the launch_dict dictionary.

https://github.com/neutronrats/spacex-capstone/blob/6ac2f8801d91f4fa25be8a5f995c351489414081/jupyter-labs-webscraping_MH.ipynb



Data wrangling

- Missing values were located and payload missing values were replaced with the payload mean value. Falcon 9 landing outcomes were separated by success and failures. A new column ‘Class’ was added, listing successes as 1 and failures as 0 values. The success rate was then calculated as 66% using the Class column.
- https://github.com/neutronrats/spacex-capstone/blob/39c79fb144f1e8e4d0a117a58ede15901e8646cd/labs-jupyter-spacex-Data%20wrangling_MH.ipynb



EDA with data visualization

- Charts plotted were the following:

Scatter plots:

- x="FlightNumber", y="PayloadMass"
- x="FlightNumber", y="LaunchSite"
- x="PayloadMass", y="LaunchSite"
- x="FlightNumber", y="Orbit"
- x="PayloadMass", y="Orbit"

Bar Plot:

- x="Orbit", y="Success Rate"

Line Plot

- x="Year", y="Success Rate”

https://github.com/neutronrats/spacex-capstone/blob/a2e756271fb8888123c575fbf984ac7efd9b07e9/jupyter-labs-eda-dataviz_MH.ipynb

EDA with SQL

```
1)select UNIQUE LAUNCH_SITE from SPACEXTBL;

2)select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5;

3)select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)';

4)select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1';

5)select min(DATE) from SPACEXTBL where LANDING__OUTCOME = 'Success (ground pad)';

6)select BOOSTER_VERSION from SPACEXTBL where LANDING__OUTCOME = 'Success (drone ship)'
  and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;

7)select MISSION_OUTCOME, count(MISSION_OUTCOME) from SPACEXTBL group by MISSION_OUTCOME;

8)select unique(BOOSTER_VERSION) from SPACEXTBL where PAYLOAD_MASS__KG_ = (select
  max(PAYLOAD_MASS__KG_) from SPACEXTBL);

9)select monthname(DATE), BOOSTER_VERSION, LAUNCH_SITE, LANDING__OUTCOME from SPACEXTBL
  where LANDING__OUTCOME = 'Failure (drone ship)' and year(DATE) = 2015;

10)select LANDING__OUTCOME, count(LANDING__OUTCOME) from SPACEXTBL where DATE between
  '2010-06-04' and '2017-03-20' and LANDING__OUTCOME like 'Success%' group by
  LANDING__OUTCOME;
```

https://github.com/neutronrats/spacex-capstone/blob/17e5af00506bf910839d80998fd3615f4486b4d7/jupyter-labs-eda-sql-coursera_MH.ipynb

Build an interactive map with Folium

- The following map objects were created and added to the map: markers, circles and lines.
- The markers and circles were added to show the different Falcon 9 launch sites as well as to identify close by railways and coast lines. Lines were used to show the distance between coast lines and railways to the launch sites.
- https://github.com/neutronrats/spacex-capstone/blob/5fe51dc2a0ed1b35056a2aa9cd541dff63def6f3/lab_jupyter_launch_site_location_MH.ipynb

Build a Dashboard with Plotly Dash

- I added a pie chart displaying the success rates for every launch site. There is a drop down menu to select all or a specific launch site.
- I also added a scatter plot showing success rates by payload mass (kg). There is also a slider to select a range of payload mass (0 – 10000 kg)
- There plots and interactions are important to show how launch sites and payload mass can affect success rates.

https://github.com/neutronrats/spacex-capstone/blob/1d60a423229216528089eac8520587a0f17e568d/spacex_dash_app.py

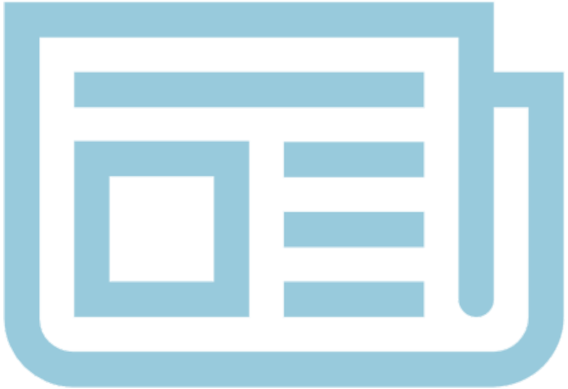
Predictive analysis (Classification)

- First I standardized the data using StandardScaler(). I then split the data into training and test data using train_test_split. I then used GridSearchCV to find the best parameters for: logistic regression, support vector machine, decision tree classifier and k nearest neighbours models. I used those parameters to train the different models on my train data. I then tested the models on my test data to check for accuracy.

- All the prediction methods performed similarly since the data set was very small.

https://github.com/neutronrats/spacex-capstone/blob/38b522b78f683b5fcc9e46f07059639dcb2c43ad/SpaceX_Machine%20Learning%20Prediction_Part_5_MH.ipynb

Results

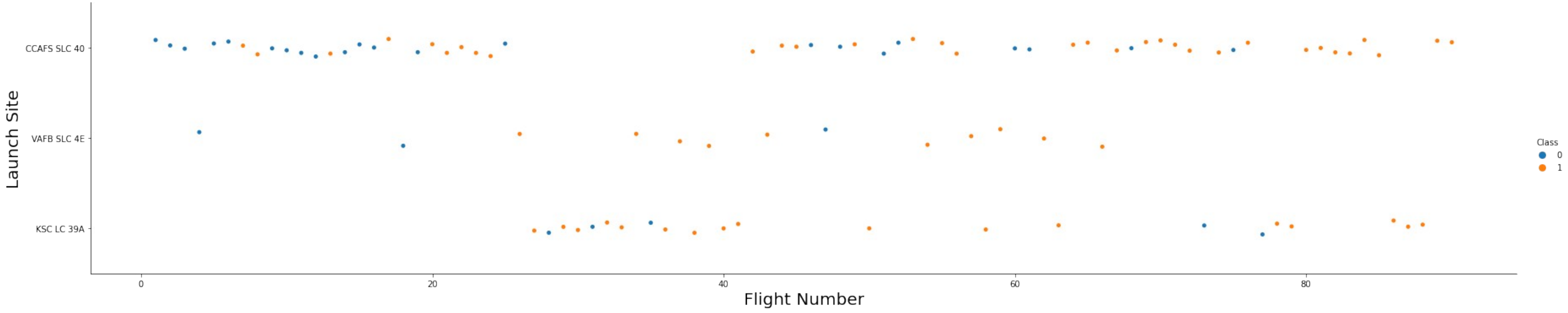


- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

EDA with Visualization

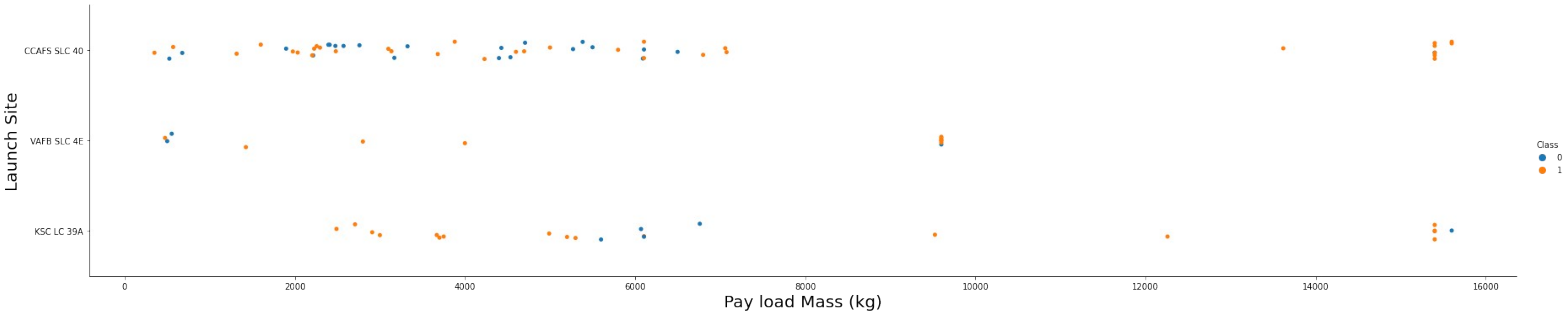
Flight Number vs. Launch Site

Show a scatter plot of Flight Number vs. Launch Site



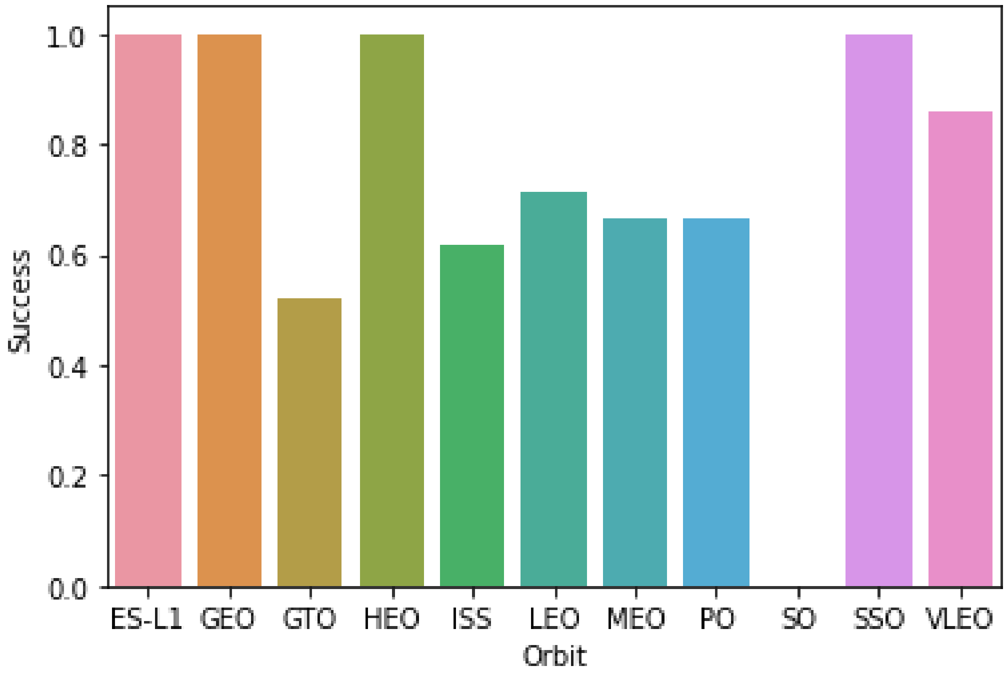
Payload vs. Launch Site

Show a scatter plot of Payload vs. Launch Site



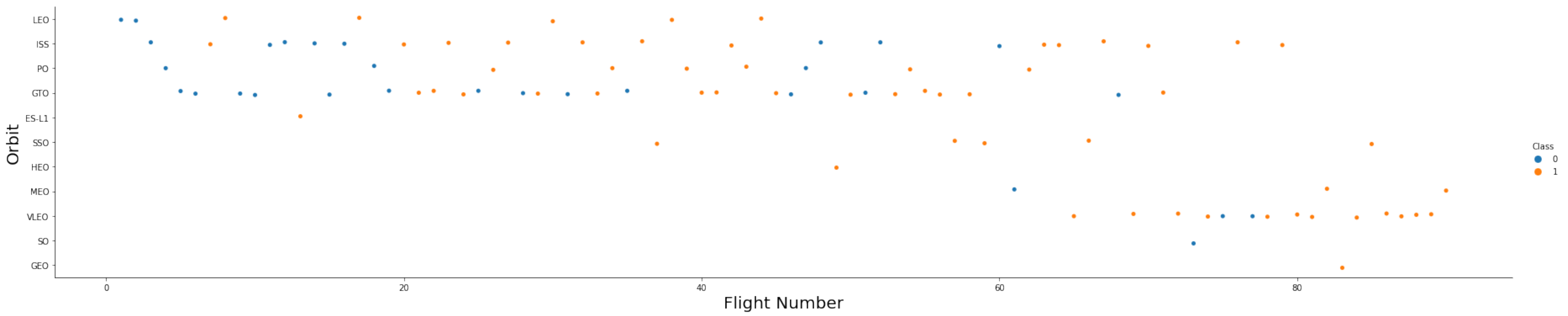
Success rate vs. Orbit type

Show a barchart for the success rate of each orbit type



Flight Number vs. Orbit type

Show a scatter point of Flight number vs. Orbit type

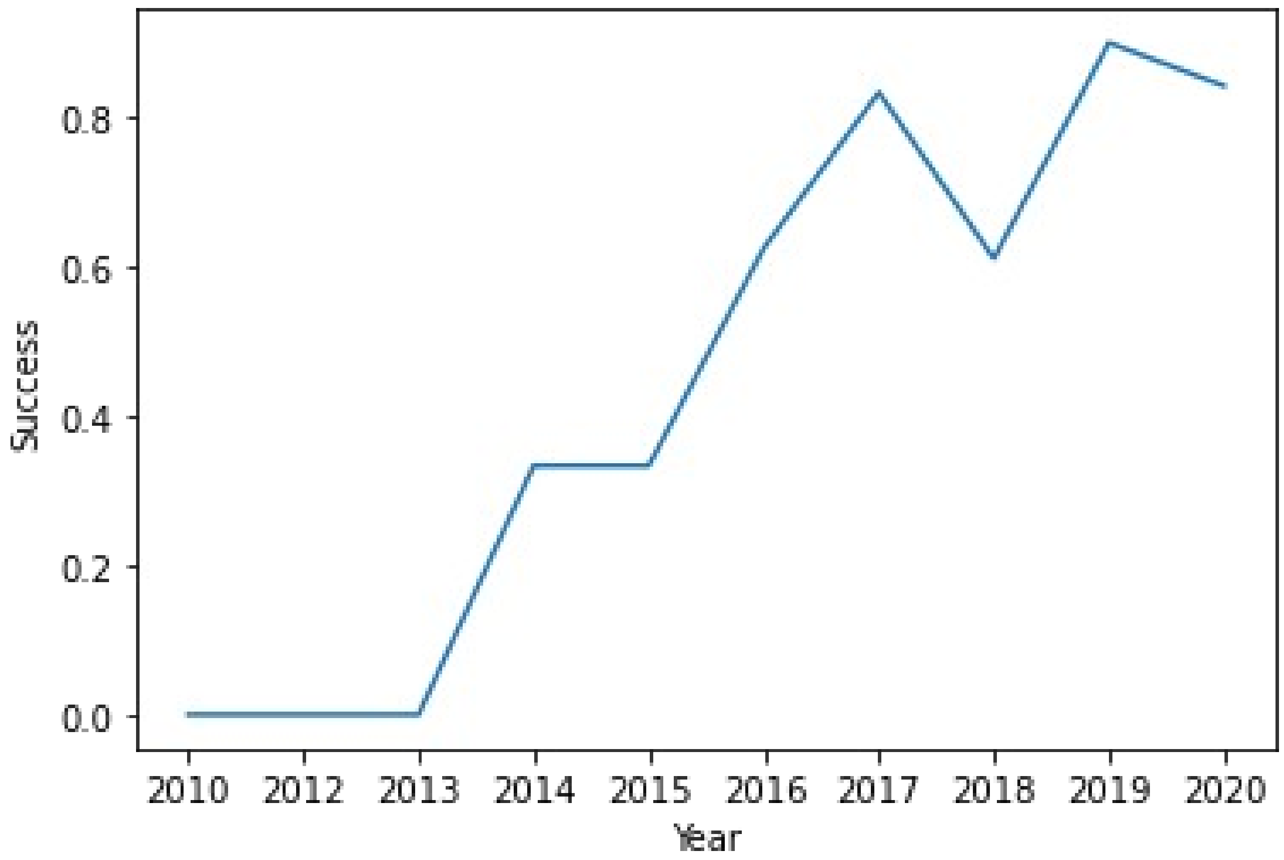


Payload vs. Orbit type

Show a scatter point of payload vs. orbit type

Launch success yearly trend

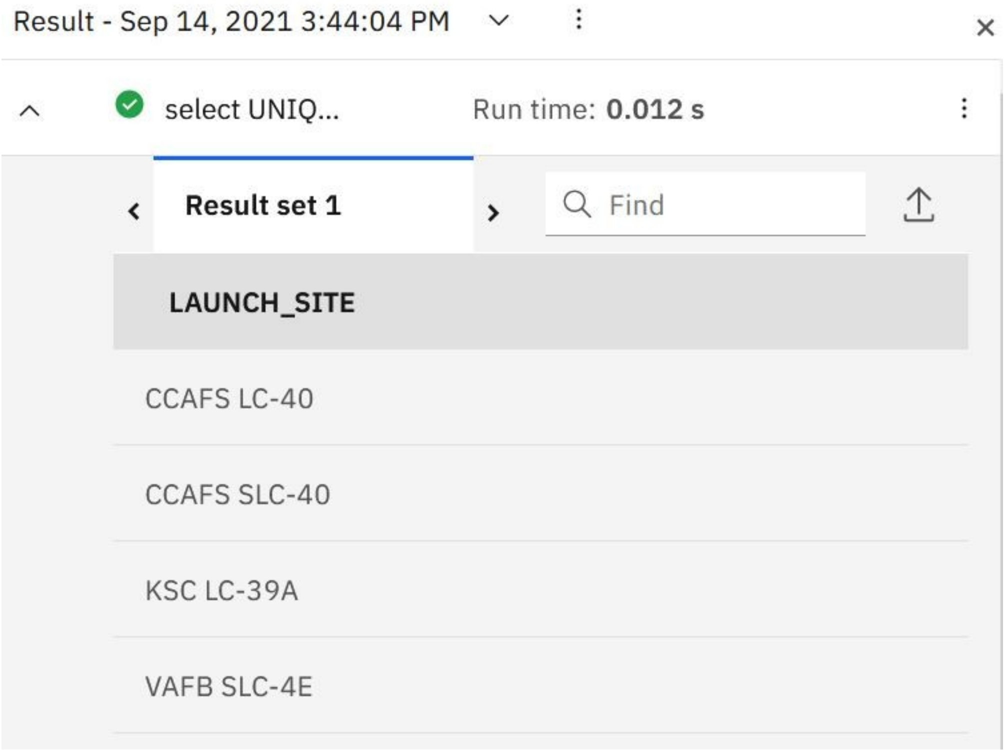
Show a line chart of yearly average success rate



EDA with SQL

All launch site names

```
select UNIQUE LAUNCH_SITE from SPACEXTBL;
```



Launch site names begin with `CCA`

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

DATE	TIME__UTC_	BOOSTER_VERSION ↑↓	LAUNCH_SITE	PAYLOAD	PAYLOAD_MAS
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677

Total payload mass

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

Result set 1

1
45596

Average payload mass by F9 v1.1

```
select avg(PAYLOAD_MASS__KG_) from SPACEXTBL
where BOOSTER_VERSION = 'F9 v1.1';
```

Result set 1

1
2928

First successful ground landing date

```
select min(DATE) from SPACEXTBL where
LANDING__OUTCOME = 'Success (ground pad)';
```

Result set 1

1
2015-12-22

Successful drone ship landing with payload between 4000 and 6000

```
select BOOSTER_VERSION from SPACEXTBL where
LANDING__OUTCOME = 'Success (drone ship)' and
PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;
```

Result set 1

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

Total number of successful and failure mission outcomes

```
select MISSION_OUTCOME, count(MISSION_OUTCOME)
from SPACEXTBL group by MISSION_OUTCOME;
```

Result set 1

MISSION_OUTCOME	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters carried maximum payload

```
select unique(BOOSTER_VERSION) from SPACEXTBL
where PAYLOAD_MASS__KG_ = (select
max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

Result set 1

BOOSTER_VERSION
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2

2015 launch records

```
select monthname(DATE), BOOSTER_VERSION,
LAUNCH_SITE, LANDING__OUTCOME from SPACEXTBL
where LANDING__OUTCOME = 'Failure (drone ship)'
and year(DATE) = 2015;
```

Result set 1

Find

1	BOOSTER_VERSION	LAUNCH_SITE	LANDING__OUTCOME
January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Rank success count between 2010-06-04 and 2017-03-20

```
select LANDING__OUTCOME, count(LANDING__OUTCOME)
from SPACEXTBL where DATE between '2010-06-04'
and '2017-03-20' and LANDING__OUTCOME like
'Success%' group by LANDING__OUTCOME;
```

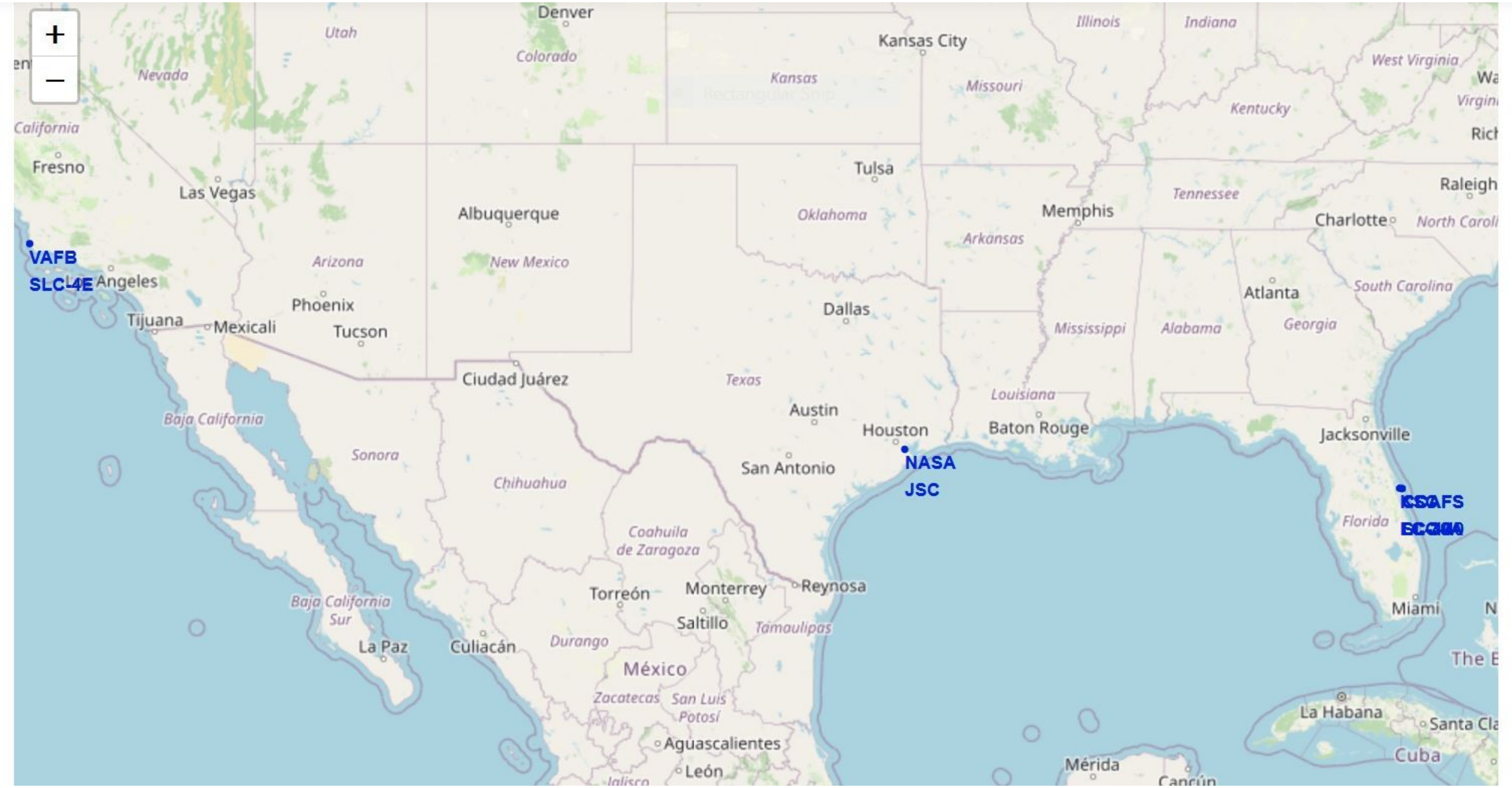
Result set 1

Find

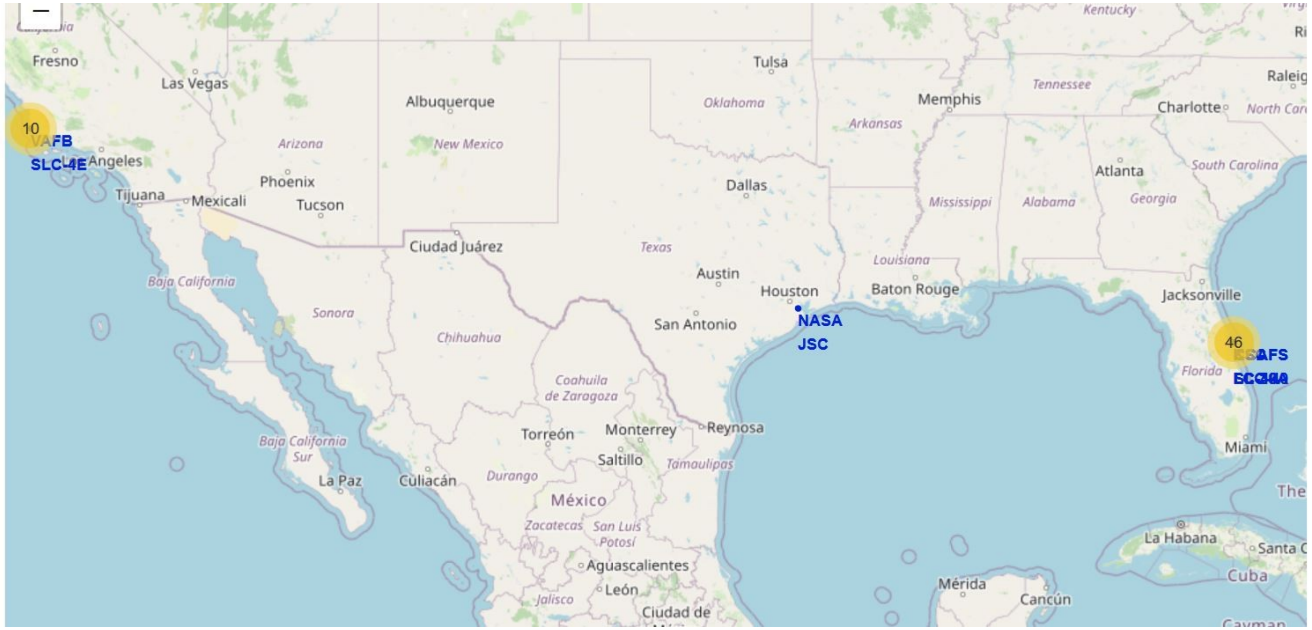
LANDING__OUTCOME	2
Success (drone ship)	5
Success (ground pad)	3

Interactive map with Folium

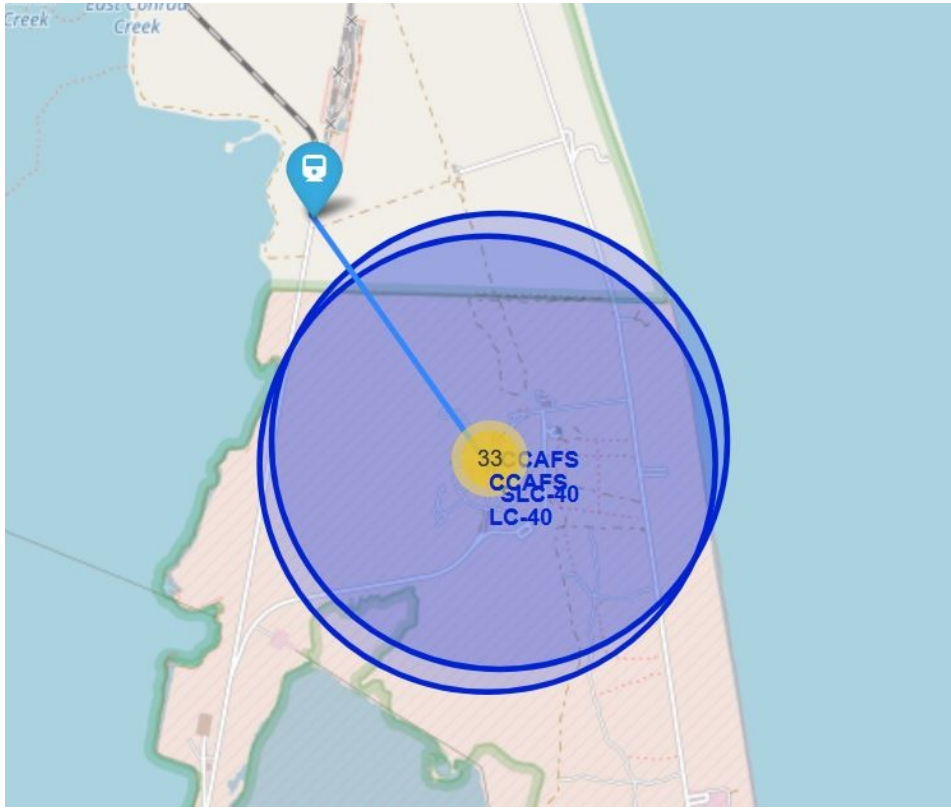
Launch Site Map

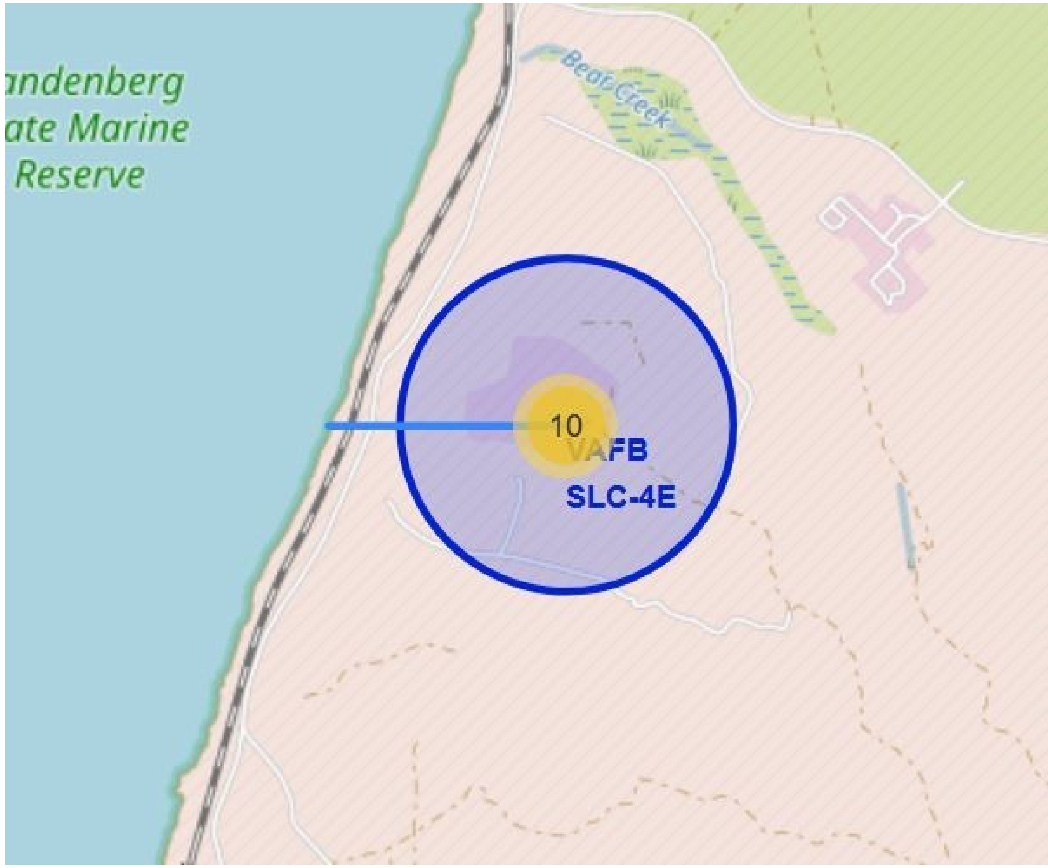


Launch Records



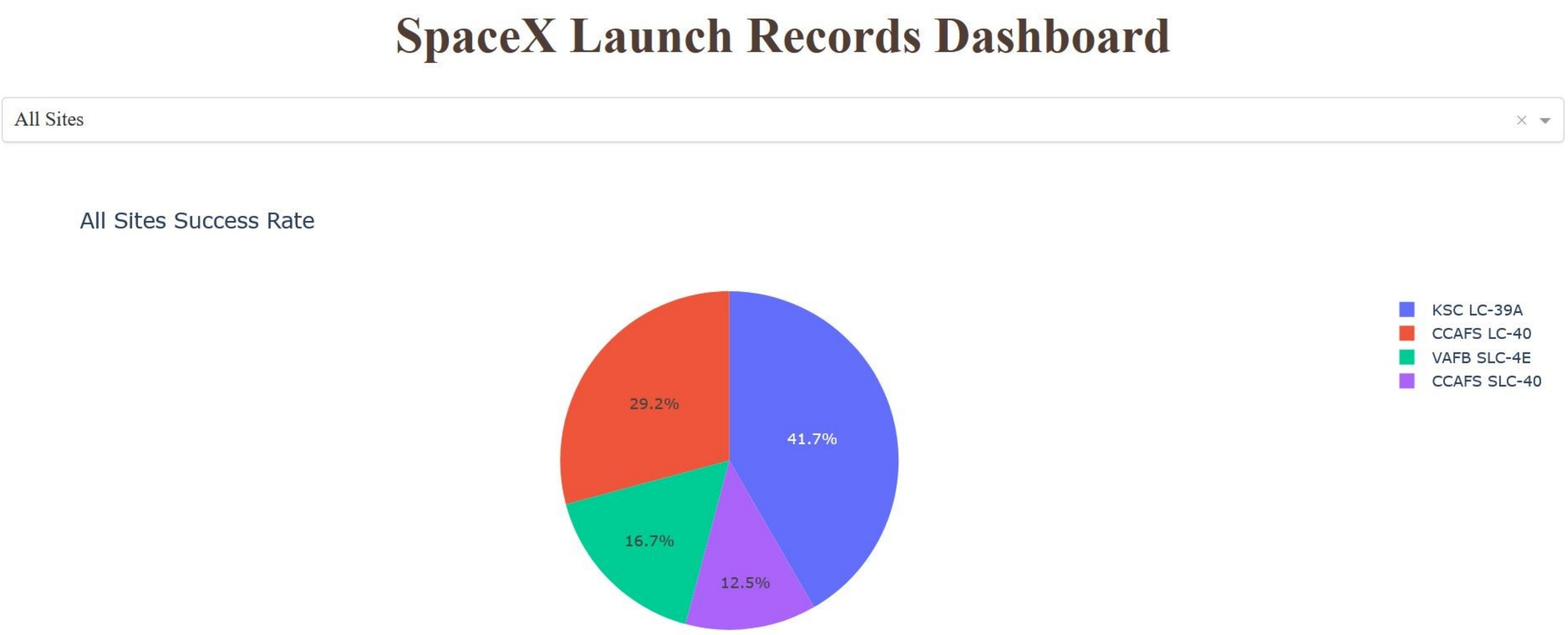
Proximity to Launch sites



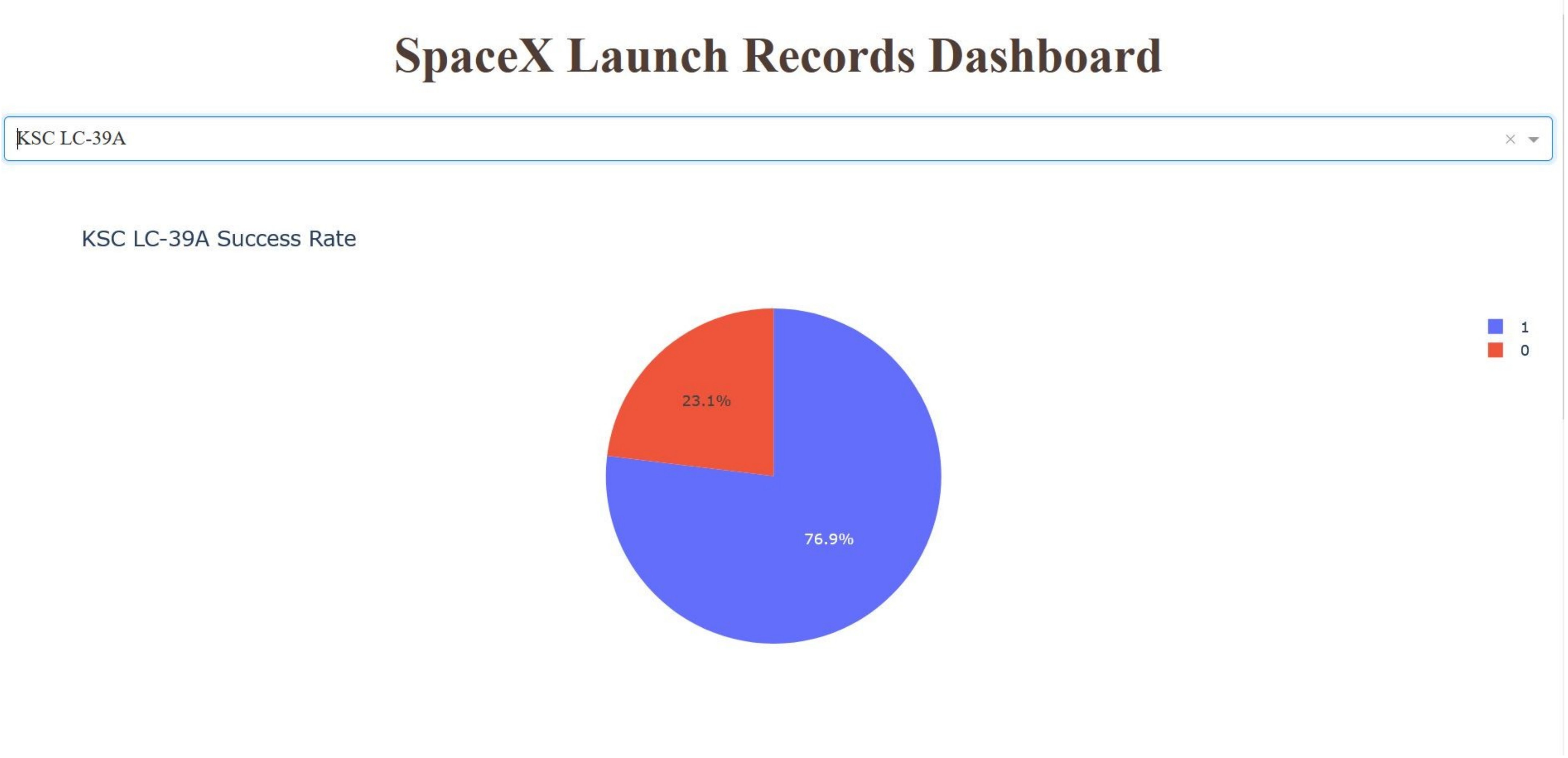


Build a Dashboard with Plotly Dash

Pie Chart for All Sites Success



Pie Chart for Highest Success

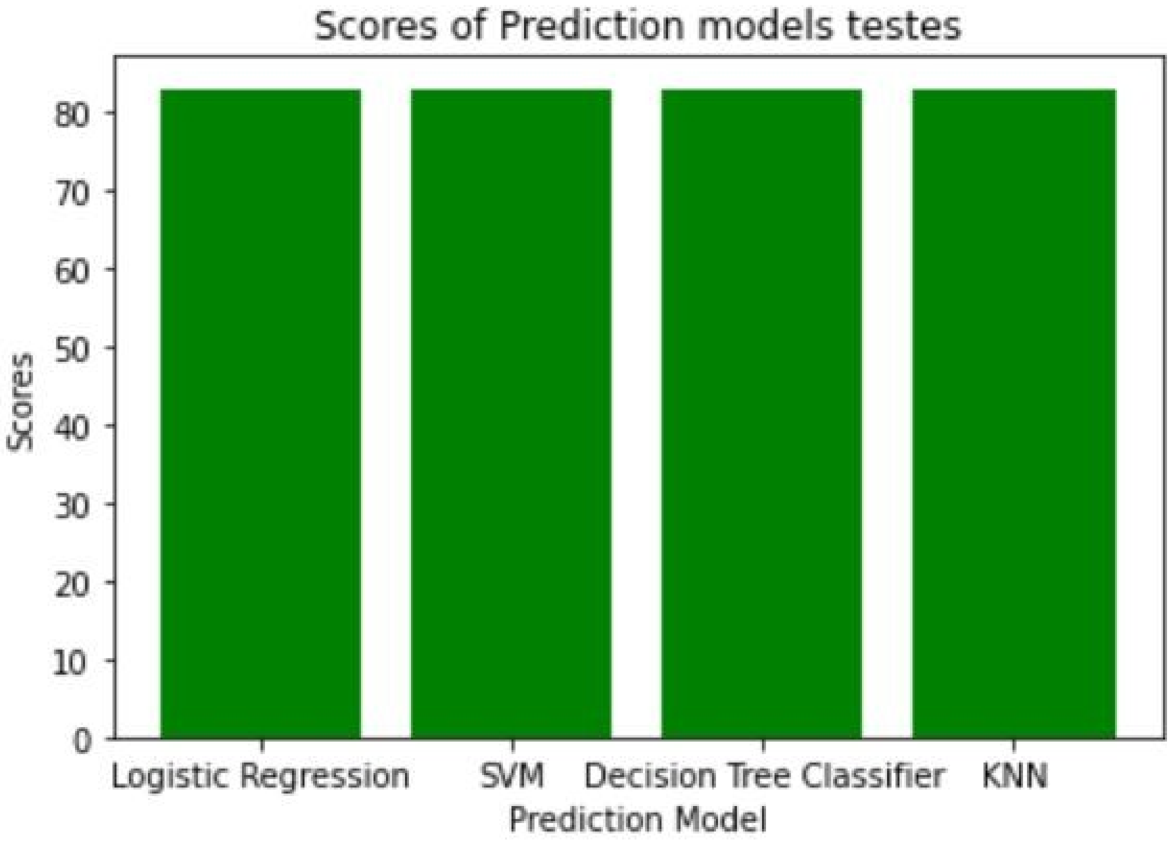


Scatter Plot for All Site Outcome

Predictive analysis (Classification)

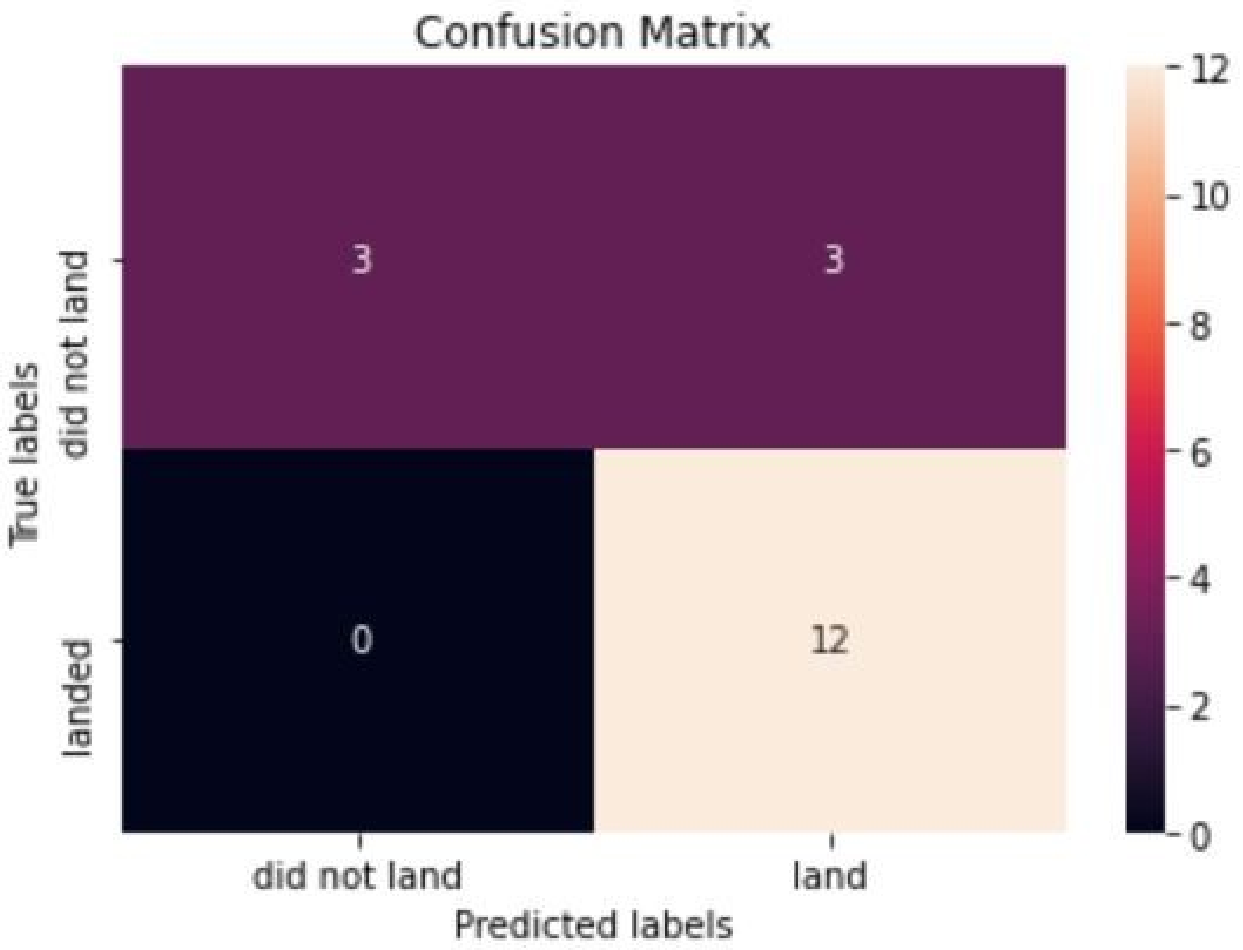
Classification Accuracy

All of the tested models have the same accuracy scores. This is due to the data set being so small.



Confusion Matrix

The confusion matrices for all models were the same since the data set was very small.



CONCLUSION

- There are many ways and many useful tools to acquire, extract and clean data
- To have an effective prediction model we need a large dataset
- Success rates of Falcon 9 landings have been increasing over time