IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Angel Norman Zapata Sumano
20 March 2023

Outline

Executive Summary

Introduction

Methodology

Results

Conclusion

Appendix

Executive Summary

This project is a comprehensive analysis of data related to SpaceX. The project utilizes various data science techniques such as API use, web scraping, EDA, data visualization, SQL, and machine learning algorithms to gain insights into SpaceX's missions and launches.

The analysis was conducted using Python, including the Pandas, NumPy, and Matplotlib libraries, among others. The code used to conduct the analyses is included in this repository.

Introduction

The space industry is growing and becoming more accessible, with several companies leading the way, including Virgin Galactic, Rocket Lab, Blue Origin, and SpaceX. SpaceX has achieved many milestones, including manned missions and satellite launches, in part due to its ability to reuse the first stage of its Falcon 9 rocket, which makes launches more affordable. However, the success of the first stage landing is critical in determining the cost of each launch.

Problems

Determine	The price for each launch.
Predict	Whether SpaceX will reuse the first stage using machine learning models.
Determine	The factors that affect the first stage's successful landing and predict whether it will be reusable.

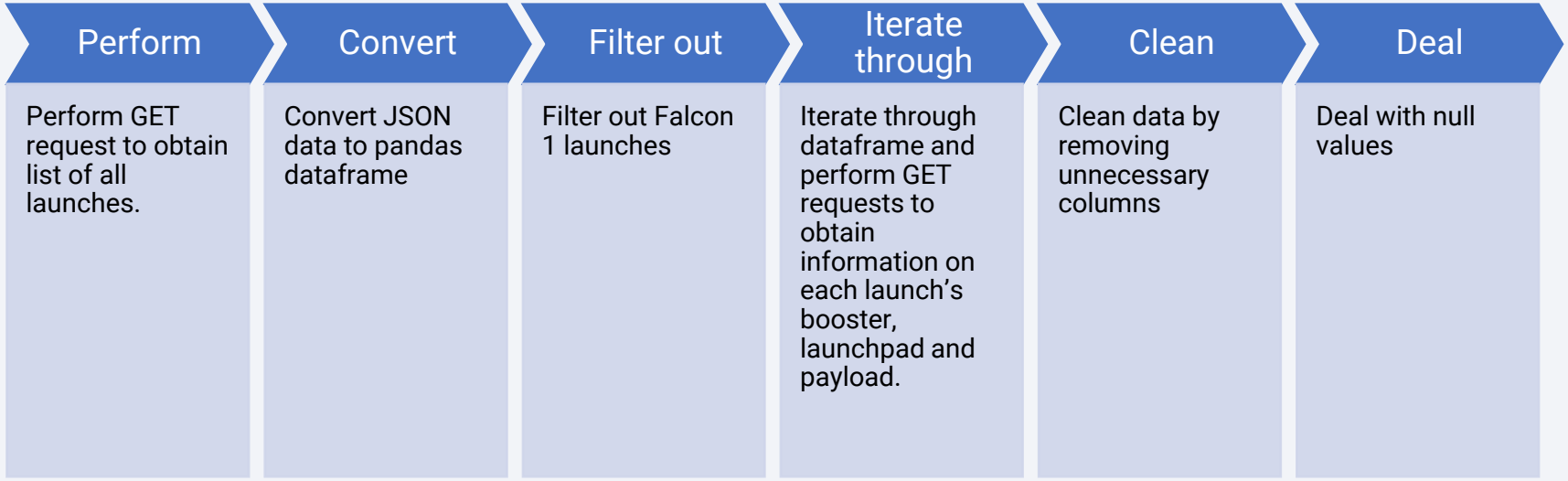
Section 1

Methodology

Methodology

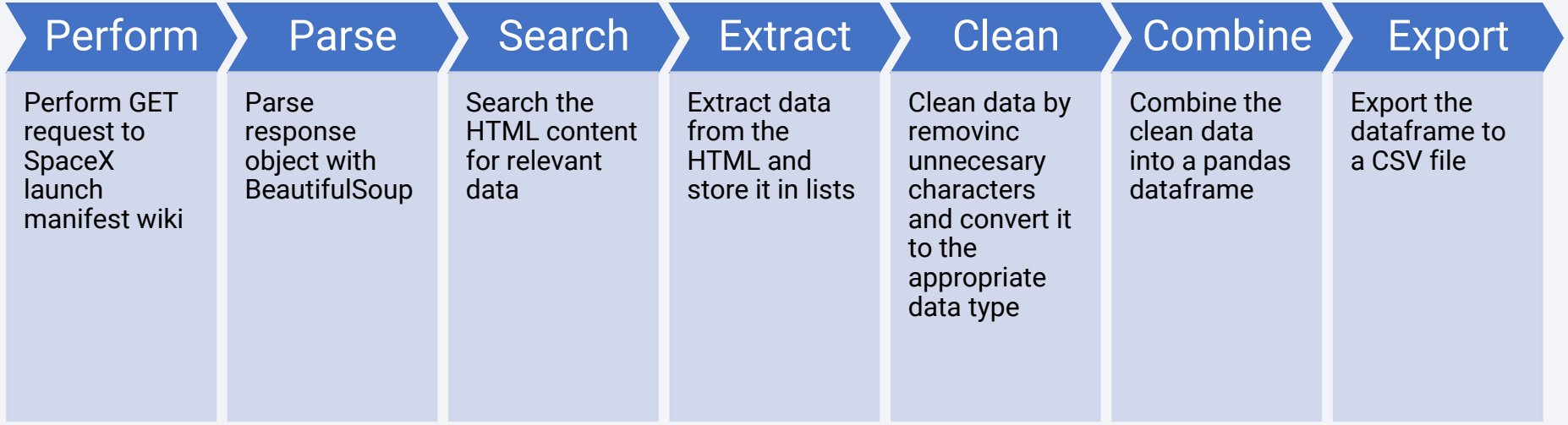
Data collection methodology:	• Launch data was gathered from the SpaceX REST API and web scraping from related Wiki pages
Perform data wrangling:	• The data wrangling process involved cleaning and transforming the raw dataset obtained from the SpaceX API and web scraping.
Perform exploratory data analysis (EDA) using visualization and SQL	
Perform interactive visual analytics using Folium and Plotly Dash	
Perform predictive analysis using classification models	• How to build, tune, and evaluate classification models

Data Collection – SpaceX API



Data Collection - SpaceX REST API Notebook

Data Collection - Scraping



Data Collection - Web Scraping Notebook

Data Wrangling

LOAD DATA INTO A PANDAS DATAFRAME

USE *INFO* AND *DESCRIBE* METHODS TO CHECK THE NUMBER OF ROWS AND COLUMNS, DATA TYPES, AND SUMMARY STATISTICS OF EACH COLUMN

CREATE A COLUMN CALLED “CLASS” REPRESENTING THE OUTCOME FOR EACH LAUNCH (0 FOR UNSUCCESSFUL AND 1 FOR SUCCESSFUL)

MAP THE “OUTCOME” COLUMN TO A BINARY VALUE

Data Wrangling Notebook

EDA with Data Visualization

- EDA Data Visualization Notebook

- The data were explored through various visualizations, such as line charts, bar charts, and scatterplots to understand the patterns and trends in the data.

- The plotted charts were chosen to provide insights and explore relationships between launch success and other parameters.

EDA with SQL

- EDA with SQL Notebook
- Query 1: Select the unique launch sites in SpaceX mission.
- Query 2: Select the first 5 records where launch sites begin with “CCA”.
- Query 3: Select the total payload mass carried by boosters launched by NASA (CRS).
- Query 4: Select the average payload mass carried by booster version F9 v1.1
- Query 5: Select the first successful landing in ground pad date.

EDA with SQL

- Query 6: Select the boosters that are successful in drone ship and have a payload mass between 4000 and 6000 kg.
- Query 7: Select the total of successful and failed outcomes.
- Query 8: Select the boosters that have carried maximum payload mass.
- Query 9: Select the months and booster versions that failed landing outcomes in drone ships during 2015.
- Query 10: Rank the successful outcomes between 04-06-2010 and 20-03-2017.

Build an Interactive Map with Folium

- Interactive Map Notebook
- In order to build an interactive map, we used circles, markers, and polylines.
  - Circles were used to display the launch sites on the map.
  - Markers were used to display:
    - Successful/failed launches
    - The highway, coastline, railway, and city closest to the launch site.
  - Polylines were used to display the distance between the launch site and relevant markers.

Build a Dashboard with Plotly Dash

- SpaceX Plotly Dashboard File
- The dashboard has a dropdown menu to select the launch site that we want to analyze in the charts, it also includes a range slider to select payload a payload mass range for the “Correlation between Payload and Success” chart.
- The dashboard comprises two charts; the former is a pie chart that displays the successful vs failed launches for the selected launch site, the latter is a scatterplot showing the correlation between Payload Mass and Successful launches.

Predictive Analysis (Classification)

Predictive Analysis Notebook

Data collection and exploration.	Gathered data from multiple sources, cleaned and preprocessed it, and performed exploratory data analysis.
Feature engineering	Created new features and engineered existing ones to improve the model's performance.
Model selection	Chose multiple classification models based on their suitability for the problem and compared their performance.
Model evaluation	Evaluated the models using performance metrics and cross-validation techniques to ensure that the selected model has the best performance.
Model tuning	Tuned the hyperparameters of the model using grid search and randomized search techniques to optimize its performance.

Results

EXPLORATORY DATA ANALYSIS RESULTS

INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS

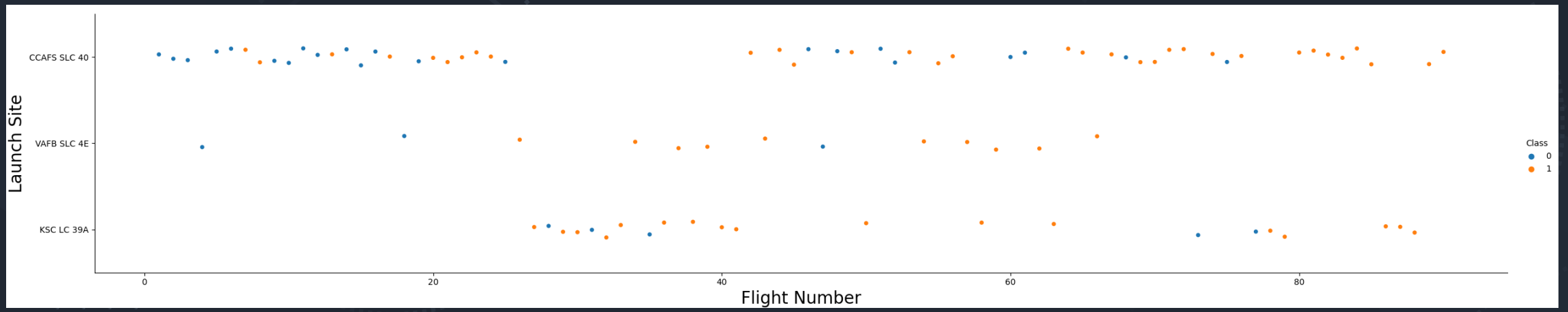
PREDICTIVE ANALYSIS RESULTS

Section 2

Insights drawn from EDA

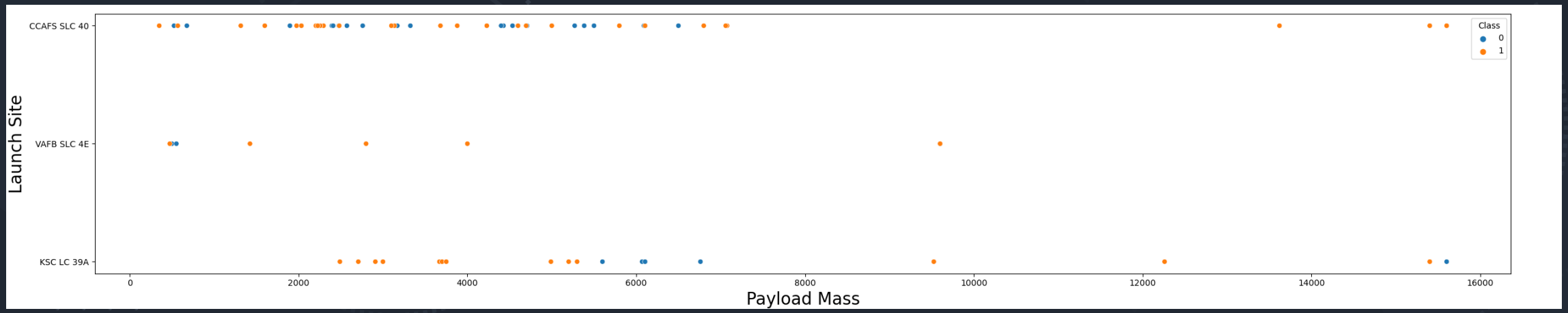
Flight Number vs. Launch Site

- This scatterplot suggests that certain launch sites are preferred for specific missions, also there is an increase in the frequency of space missions indicating a growing interest in space exploration.



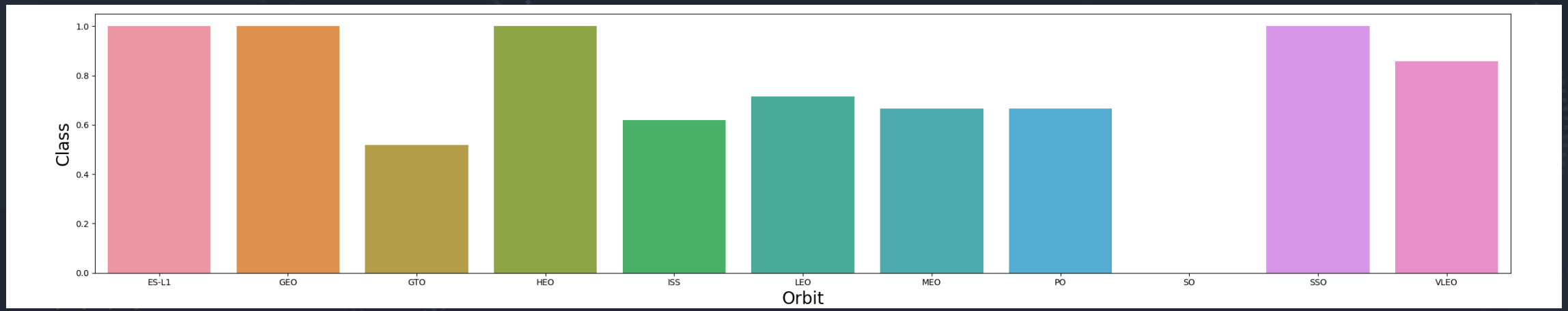
Payload vs. Launch Site

- This scatterplot shows no strong correlation between the variables, we should consider that other factors such as the type of booster, mission requirements, and destination can also impact the maximum payload that can be launched.



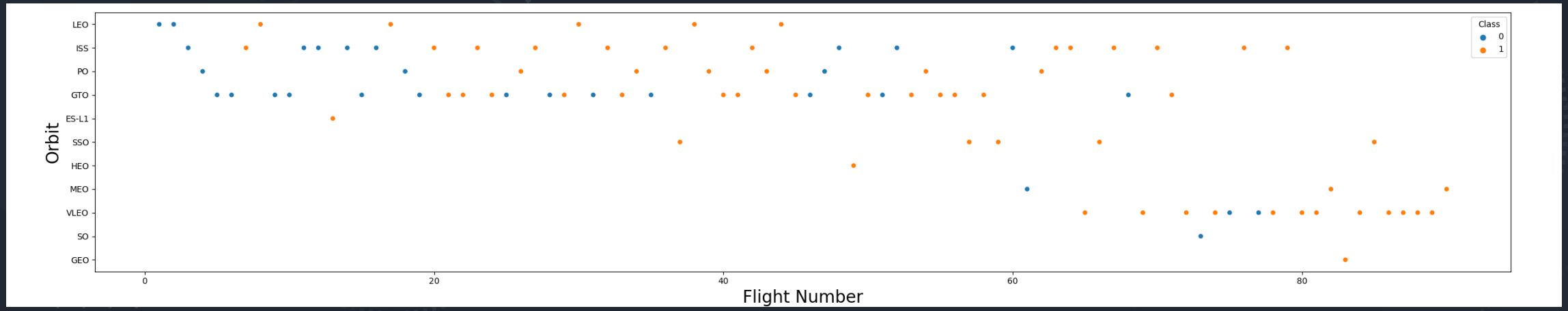
Success Rate vs. Orbit Type

- This chart shows that certain orbit types have a greater success rate than others, such as ES-L1, GEO, HEO, and SSO; while orbits like SO, GTO, and ISS have the least success rate.



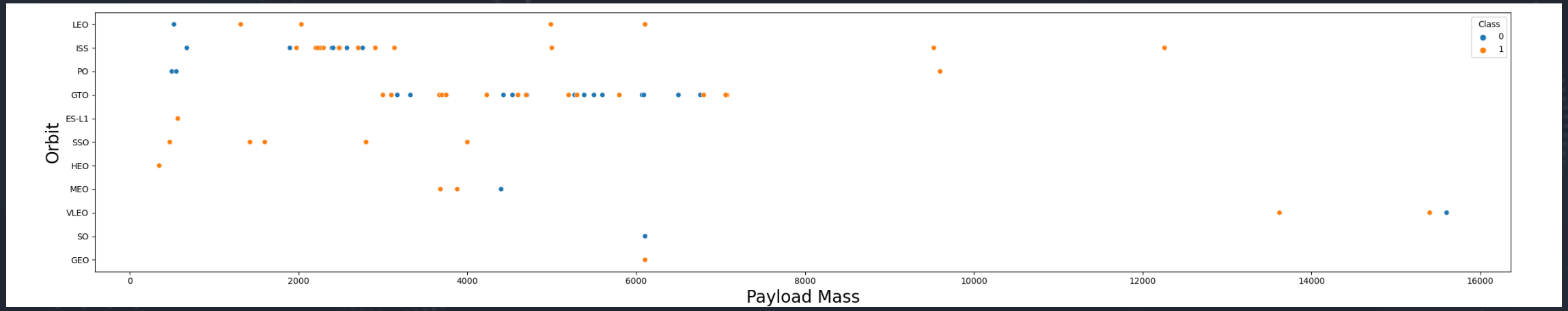
Flight Number vs. Orbit Type

- The scatter plot shows that some orbit types' success rate increases proportionally to the number of flights, such as the LEO orbit.



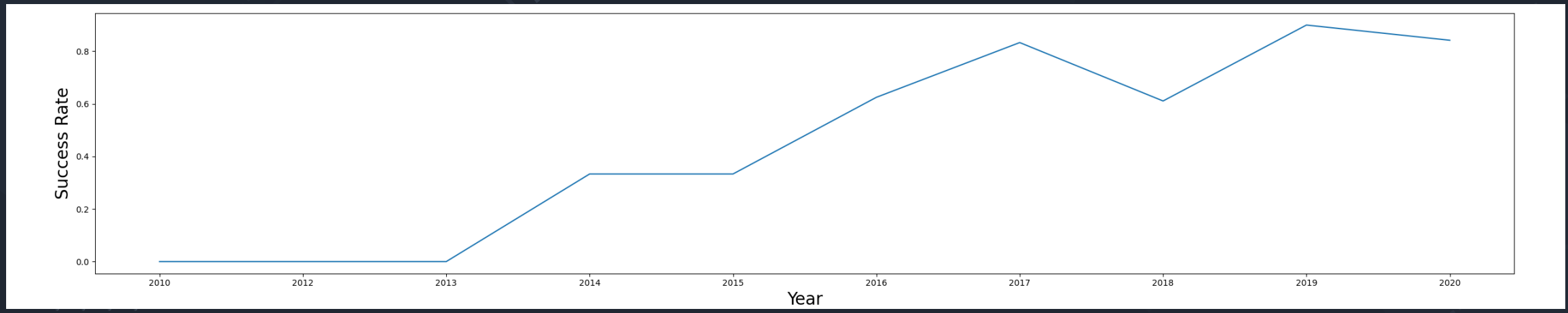
Payload vs. Orbit Type

- The scatter plot shows that some orbits' success rate increases with a heavier payload, for example, Polar, LEO, and ISS.



Launch Success Yearly Trend

- We can observe that the success rate trend has been increasing since 2013.



All Launch Site Names

```
%%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

Query 1: Select the unique launch sites in the SpaceX mission.

Launch Site Names Begin with 'CCA'

```
%%sql
SELECT *
FROM SPACEXTBL
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
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Query 2: Select the first 5 records where launch sites begin with “CCA”.

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

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

Average Payload Mass by F9 v1.1

```
%%sql
select avg(payload_mass__kg_) as average_payload_mass
from spacextbl
where "Booster_Version" = 'F9 v1.1';
```

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

average_payload_mass
2928.4

Calculate the average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date

```
%%sql
select min("Date")
from spacextbl
where "Landing _Outcome" = 'Success (ground pad)';
```

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

min("Date")
01-05-2017

Display the date of the first successful landing outcome on a ground pad.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
select distinct("Booster_Version")
from spacextbl
WHERE "Landing _Outcome" = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ > 4000
AND PAYLOAD_MASS__KG_ < 6000;
```

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

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

List the names of boosters that have successfully landed on a drone ship and had a payload mass greater than 4000 but less than 6000.

Total Number of Successful and Failure Mission Outcomes

```
%%sql
select "Mission_Outcome", count("Mission_Outcome")
from spacextbl
group by "Mission_Outcome"
```

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

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

Calculate the total of successful and failed outcomes.

Boosters Carried Maximum Payload

```
%%sql
SELECT "Booster_Version"
FROM spacextbl
WHERE PAYLOAD_MASS__KG_ = (
  SELECT MAX(PAYLOAD_MASS__KG_)
  FROM spacextbl
);
```

```
 * 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
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

Return the names of the boosters that have carried the maximum payload mass

2015 Launch Records

```
%%sql
SELECT SUBSTR("Date", 4, 2) AS month,
       "Landing _Outcome",
       "Booster_Version",
       "Launch_Site"
FROM spacextbl
WHERE SUBSTR("Date", 7, 4) = '2015'
  AND "Landing _Outcome" LIKE '%drone ship%';
```

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

month	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
06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

Return the failed landing outcomes in drone ships, their booster versions, and launch site names for the year 2015.

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

```
%%sql
SELECT "Landing _Outcome", COUNT(*) AS successful_landings
FROM spacextbl
WHERE "Date" BETWEEN '04-06-2010' AND '20-03-2017'
  AND "Landing _Outcome" LIKE '%Success%'
GROUP BY "Landing _Outcome"
ORDER BY successful_landings DESC;
```

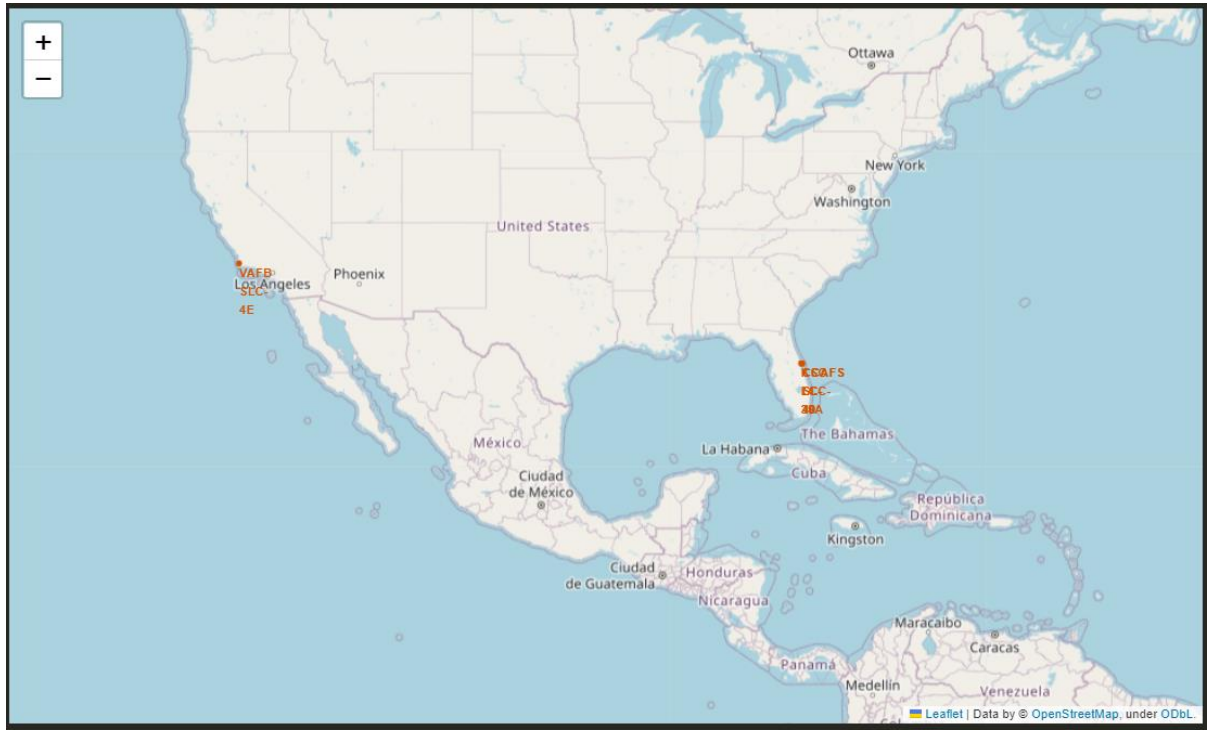
```
 * sqlite:///my_data1.db
Done.
```

Landing _Outcome	successful_landings
Success	20
Success (drone ship)	8
Success (ground pad)	6

Rank the count of landing outcomes between the dates 2010-06-04 and 2017-03-20, in descending order

Section 3

Launch Sites Proximities Analysis

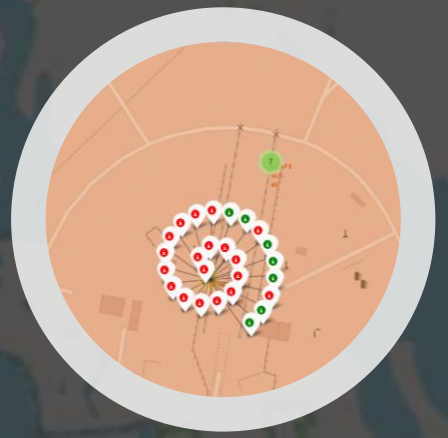


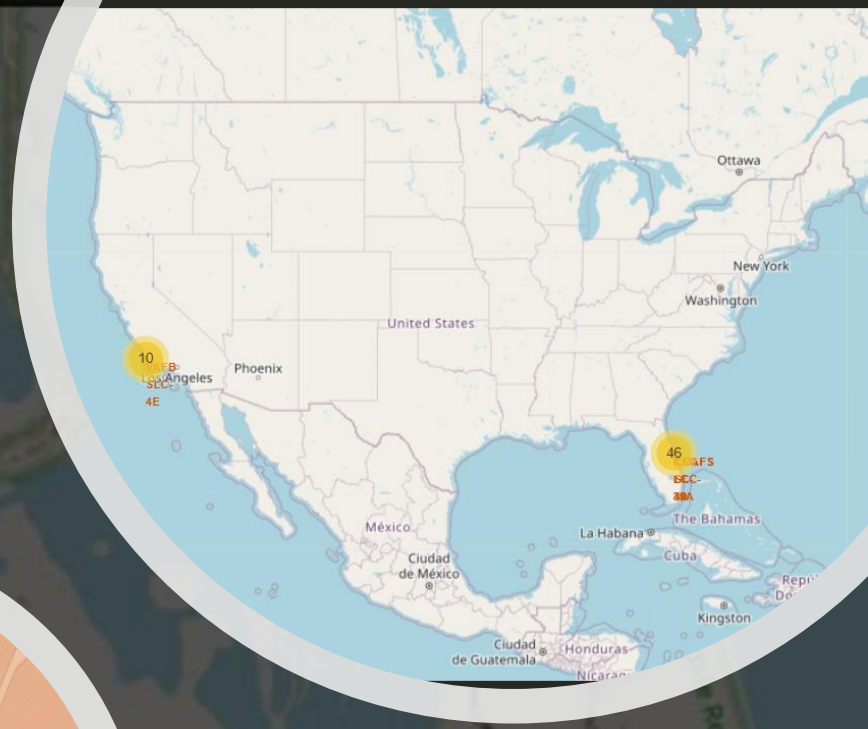
Launch Sites Map

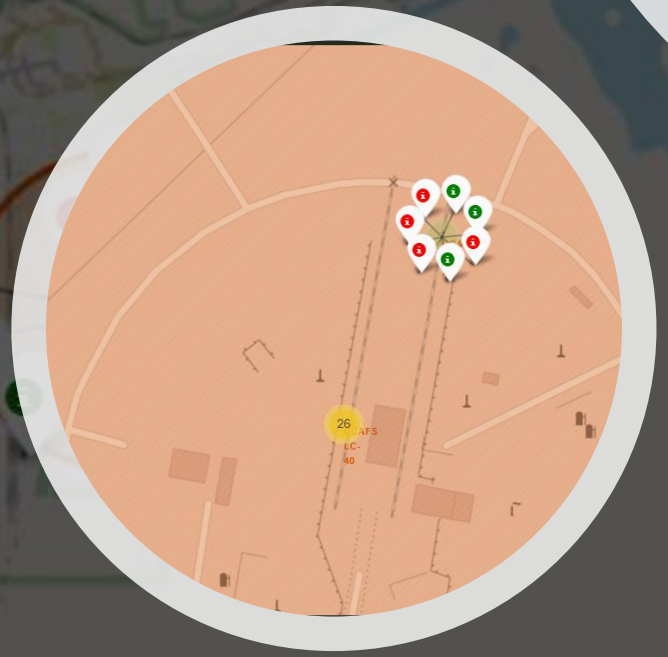
- This map displays markers for the different launch sites, we can observe that the launch sites are located in proximity to the coast.

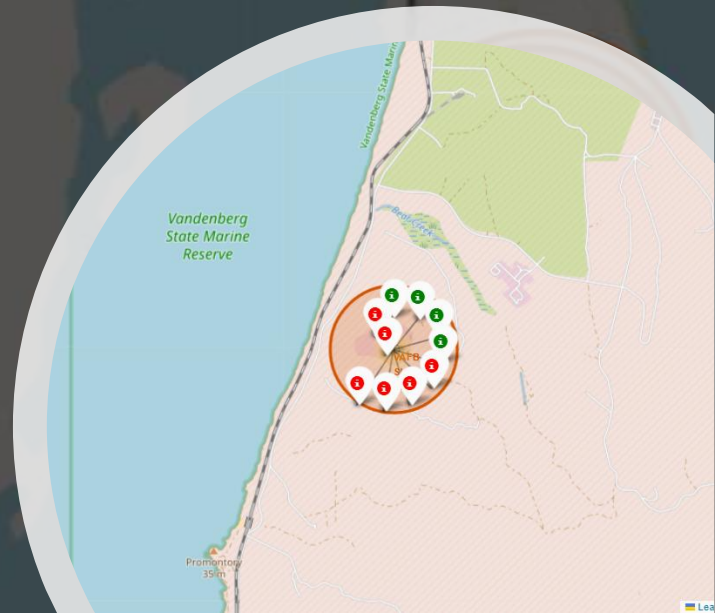
Successful/Failed Launches for Launch Site

- The markers display the successful/failed outcomes, the markers make it easier to identify the success rate of each launch site.



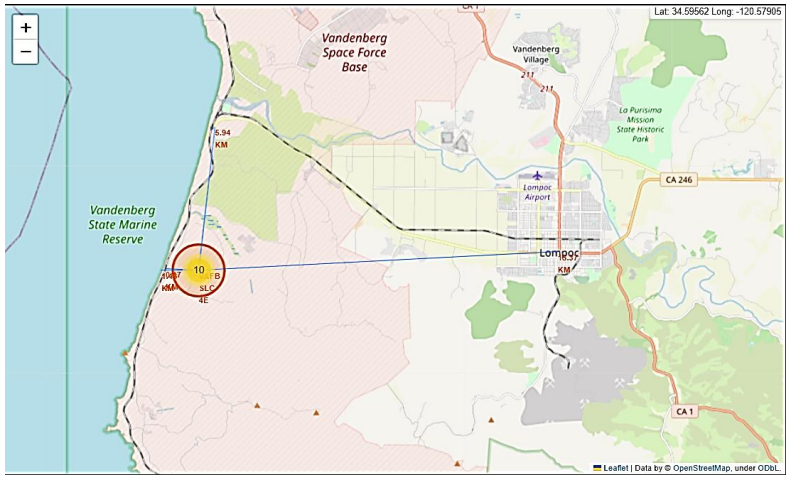






Proximities Map

- The map displays the launch sites' distance to highways, railways, cities, and coastlines. We can notice that the launch sites are located distant to cities, close to the coast and railways.

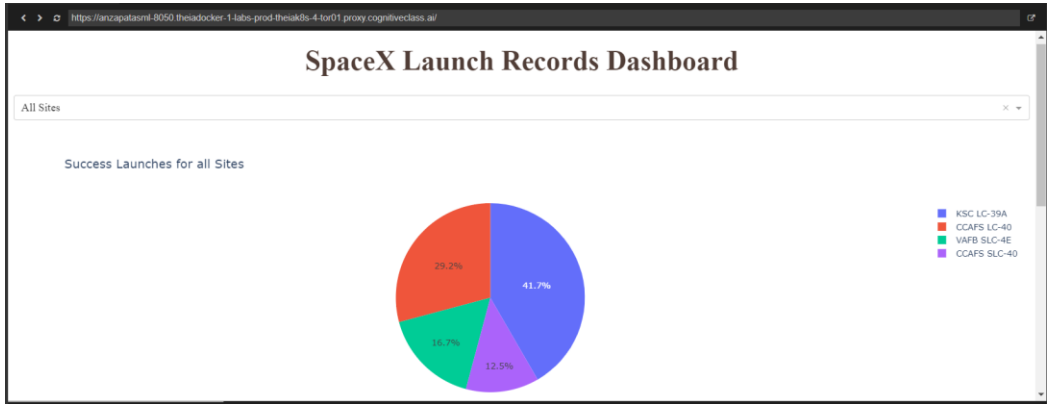


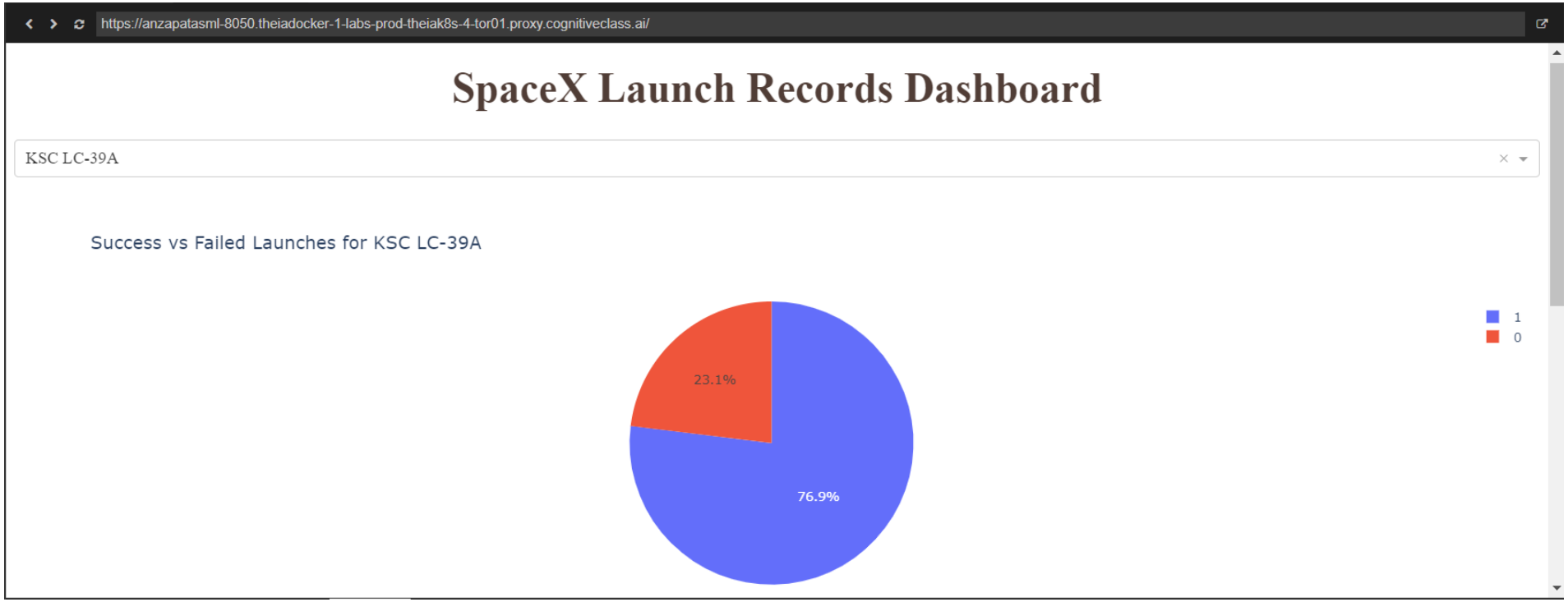
Section 4

Build a Dashboard with Plotly Dash

Successful Launches for all Sites Plot

- The pie chart works with a dropdown menu that selects the launch sites, it displays the successful launches. We can notice that KSC-LC-39A has the greatest success rate.

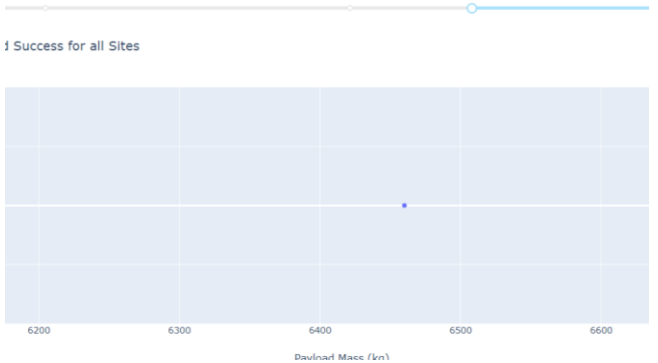




Successful vs Failed Launches for KSC LC-39A

- This pie chart displays the success/fail ratio for KSC LC-39A, the launch site with the best success/fail ratio.

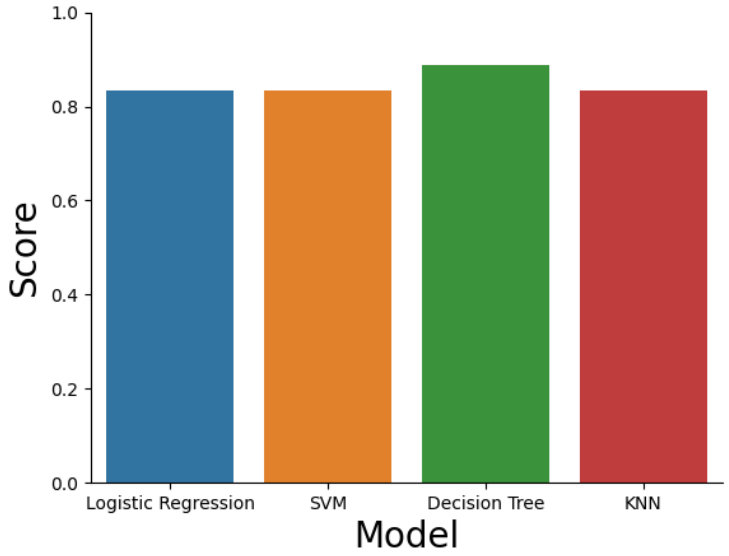
Correlation between Payload and Success plot



- These scatter plots show the correlation between payload and successful launch.
- We can see that the payload range with a greater success rate is between 2000 and 4000 kg.

Section 5

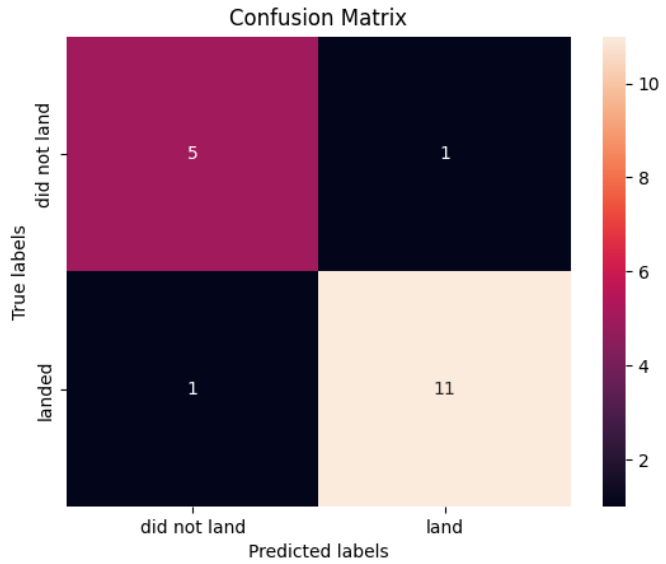
Predictive Analysis (Classification)



Classification Accuracy

- The model with the highest classification accuracy is the Decision Tree (0.89), as shown in the graph.

Confusion Matrix



- The confusion matrix of the Decision Tree shows that it only had 2 wrong predictions out of 18.

Conclusions

Launch success rate has an increasing trendline as more technologies are developed ensuring the success of the missions.

Launches from KSC LC-39A have the greatest success rate out of all the launch sites, but it is strongly related to the Orbit Type of the missions launched from that site.

The Decision Tree Classifier is the suggested machine learning algorithm for this task.

Appendix

```
parameters = {'criterion': ['gini', 'entropy'],
              'splitter': ['best', 'random'],
              'max_depth': [None, 2, 4, 6, 8, 10, 12, 14, 16, 18],
              'min_samples_split': [2, 5, 10],
              'min_samples_leaf': [1, 2, 4],
              'max_features': ['auto', 'sqrt', 'log2', None]}
tree = DecisionTreeClassifier()
```

```
parameters = {'C': [0.01, 0.1, 1, 10, 100],
              'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
              'degree': [2, 3, 4, 5],
              'gamma': ['scale', 'auto', 0.1, 1, 10],
              'shrinking': [True, False],
              'tol': [1e-3, 1e-4, 1e-5]}
svm = SVC()
```

```
parameters = {'n_neighbors': [n for n in range(1, 12)],
              'weights': ['uniform', 'distance'],
              'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
              'p': [1,2,3]}

KNN = KNeighborsClassifier()
```

- For this project we used different parameter grids than the ones provided for us in the notebooks, the parameters will be added to this appendix.

Thank you!