IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alexander G. P. M.
2025-02-08

Outline

- EXECUTIVE SUMMARY
- INTRODUCTION
- METHODOLOGY
- RESULTS
- CONCLUSION
- APPENDIX

Executive Summary

Summary of methodologies:

Data collection

Data wrangling

Exploratory Data Analysis with Data Visualization

Exploratory Data Analysis with SQL

Interactive Map with Folium

Dashboard with Plotly and Dash

Summary of all results:

Exploratory Data Analysis results

Interactive Analytics Demo in Screenshots

Predictive Analysis results

Introduction

- Project background and context:
  - SpaceX is a company that is working on the next generation of fully reusable launch vehicles, the company is growing every year and as such it also keeps improving the models used in its launches.
  - One of the main problems that SpaceX has is the launching phase, where it can sometimes fail, as such if we can determine if a given stage is successful then we can determine other given metrics as well.
  - This research is based of public information and is using data analysis coupled with machine learning models to predict the success of a launch.
- Problems you want to find answers:
  - How certain variables like payload mass, launch site, number of flights and orbits are affecting the success of the first stage landing?
  - Does the rate of success increase year after year?

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

Collected data using SpaceX's REST API

Collected data using Web Scraping from SpaceX's Wikipedia page.

Perform data wrangling

Filtering the data so only necessary values are used.

Dealing with null or missing values.

Using an encoder to prepare data in this case One Hot Encoding.

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

Perform predictive analysis using classification models

Data Collection

This stage involved collecting data from SpaceX's REST API and also Web Scraping a table found in its Wikipedia page.

Both of these steps were done to collect enough data to make a better analysis and understand better the information available to us.

From each source we collected certain data:

Data obtained and used from SpaceX's REST API:

- Flight Number, Data, Booster Version, Payload Mass, Orbit, Launch Site, Outcome, Flights, Grid Fins, Reused, Legs, Landing Pad, Block, Reused Count, Serial, Longitude and Latitude.

Data obtained and used from SpaceX's Wikipedia page:

- Flight No., Launch site, Payload, Payload Mass, Orbit, Customer, Launch Outcome, Version Booster, Booster Landing, Date and Time.

Data Collection – SpaceX API

01

Requesting data from SpaceX's REST API

02

Receiving the response

03

Decoding the response to JSON

04

Turning the decoded response into a DataFrame normalizing the JSON

05

Processing the data in the DataFrame, selecting only necessary data

06

Filtering the DataFrame to only include Falcon 9 launches

07

Processing missing values found in Payload Mass

08

Exporting the DataFrame to a CSV file for further exploration.

[Github Link]

Data Collection – Web Scraping

01

Requesting the SpaceX's Wikipedia webpage

02

Parsing the HTML response

03

Extracting all tables found

04

Extracting data from the third table

05

Creating a DataFrame with the extracted data

06

Exporting the DataFrame to a CSV file for further exploration

[Github Link]

Data Wrangling

The dataset contained data about landing and types of landing as well.

We mainly analyzed the landing outcomes.

A column that marks the landing outcome with a value of 1 or 0 depending if the landing was successful was created.

Data Wrangling

01

Calculating the number of launches on each site

02

Calculating the occurrence of each orbit

03

Calculating the occurrence of type of landings

04

Creating a column called Outcome based on the landing info

05

Exporting the data to CSV

[Github Link]

EDA with Data Visualization

Scatter plots created:

Flight Number vs Payload

Flight Number vs Launch Site

Payload Mass vs Launch Site

Flight Number vs Orbit Group

Payload Mass vs Orbit Group

Bar plots created:

Orbit Group vs Class

Line plots created:

Year vs Outcome

[Github Link]

EDA with Data Visualization – Features Engineering

01

Selecting features that are going to be used in a prediction model

02

Creating dummy variables for categorical columns

03

Casting all numeric columns to float data type

04

Exporting the processed data to a CSV file for further use

Github Link

EDA with SQL

Displaying the unique names of each launch site in the space mission → Displaying the first 5 records where launch sites begin with the string 'CCA' → Displaying the total payload mass carried by boosters launched by NASA (CRS) → Displaying average payload mass carried by booster version F9 v1.1

→ Listing the date when the first successful landing outcome in ground pad was achieved → Listing the names of the booster which have success in drone ship and have payload mass greater that 4000 but less than 6000 → Listing the total number of successful and failure mission outcomes → Listing the manes of the booster versions which have carried the maximum payload mass using a subquery

→ Listingthe records which will display the month names, failure landing outcomes in drone ship, booster version, launch site for the months in year 2015 → Ranking the count of landing outcomes (such as failure (drone ship) or success (ground pad)) between the dates 2010-06-04 and 2017-03-20, in descending order

Github Link

Build an Interactive Map with Folium

Circle markers	These markers were used to indicate the position of each launch site
Popup labels	These markers were used to add information when the user hover over a certain part of the map
Text labels	These markers were used to indicate the name of each launch site as well as additional info such as distance to railways, highway, costalines, cities, etc.
Line markers	These markers were used to indicate the distance to different spots in the map

[Github Link]

Build a Dashboard with Plotly Dash

Dropdown list:

This was used so the user can select a specific Launch Site or all sites

Pie chart:

This was used to show the successful vs failed for a given site or all sites.

Slider:

This was used to select a Payload range.

Scatterplot:

This was used to show the correlation between Payload and Launch Success

Github Link

Predictive Analysis (Classification)

We first processed the Class column to make it the predicted value.

We then transformed the other data using a standard scaler.

We split the data into training and testing sets.

We used four algorithms to predict the data: logistic regression, support vector machine, decision tree classifier and k nearest neighbors.

While using the four algorithms we also used GridSearchCV to get the best hyperparameters.

We checked each algorithm success with a confusion matrix.

We also checked jaccard and f1 scores of each algorithm.

[Github Link]

Predictive Analysis (Classification)

Processing predictors columns → Processing predicted column → Applying standard scaler → Splitting the predictors and predicted columns into training and testing sets → Using logistic regression with grid search → Calculating accuracy in the logistic regression model

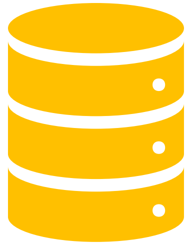
→ Showing confusion matrix of the logistic regression model → Using support vector machine with grid search → Calculating accuracy in the support vector machine model → Showing confusion matrix of the support vector machine model → Using decision tree classifier with grid search → Calculating accuracy in the decision tree classifier model

→ Showing confusion matrix of the decision tree classifier model → Using k nearest neighbors with grid search → Calculating accuracy in the k nearest neighbors model → Showing confusion matrix of the k nearest neighbors model → Comparing models using f1 and jaccard scores

Results - Exploratory data analysis results

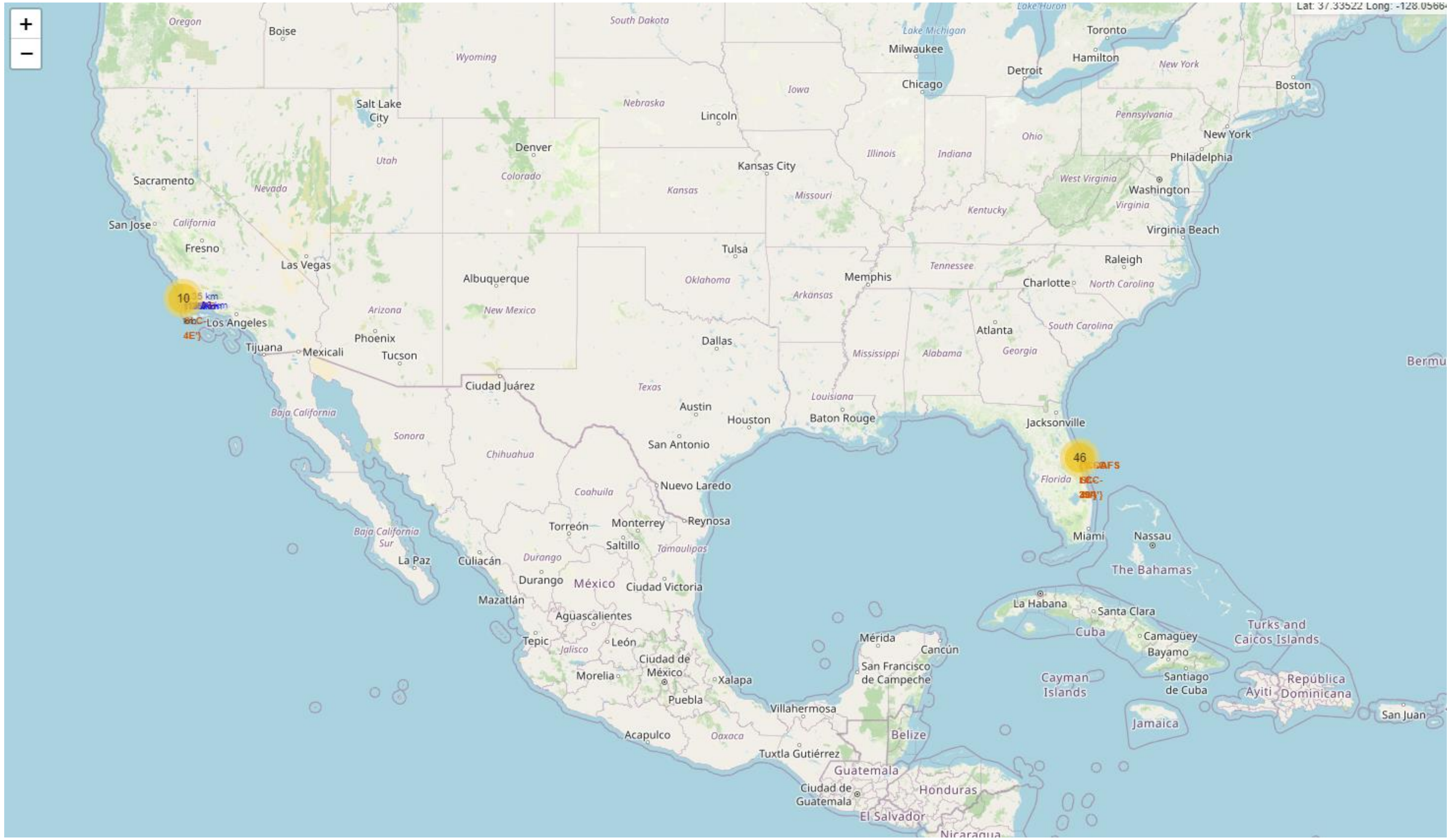
While analyzing visually the data, we saw that multiple relationships occur between different columns.

We also found how many successes and failures were there and in which category they fall.



While analyzing using SQL we found more information about payload mass and outcomes.

Results - Interactive analytics demo in screenshots



Results - Predictive analysis results

We can conclude that since the models created were trained and tested in the same data that all of them are almost equally valid to be used to predict success.

It should be noted that SVM shows a better F1 and Jaccard scores.

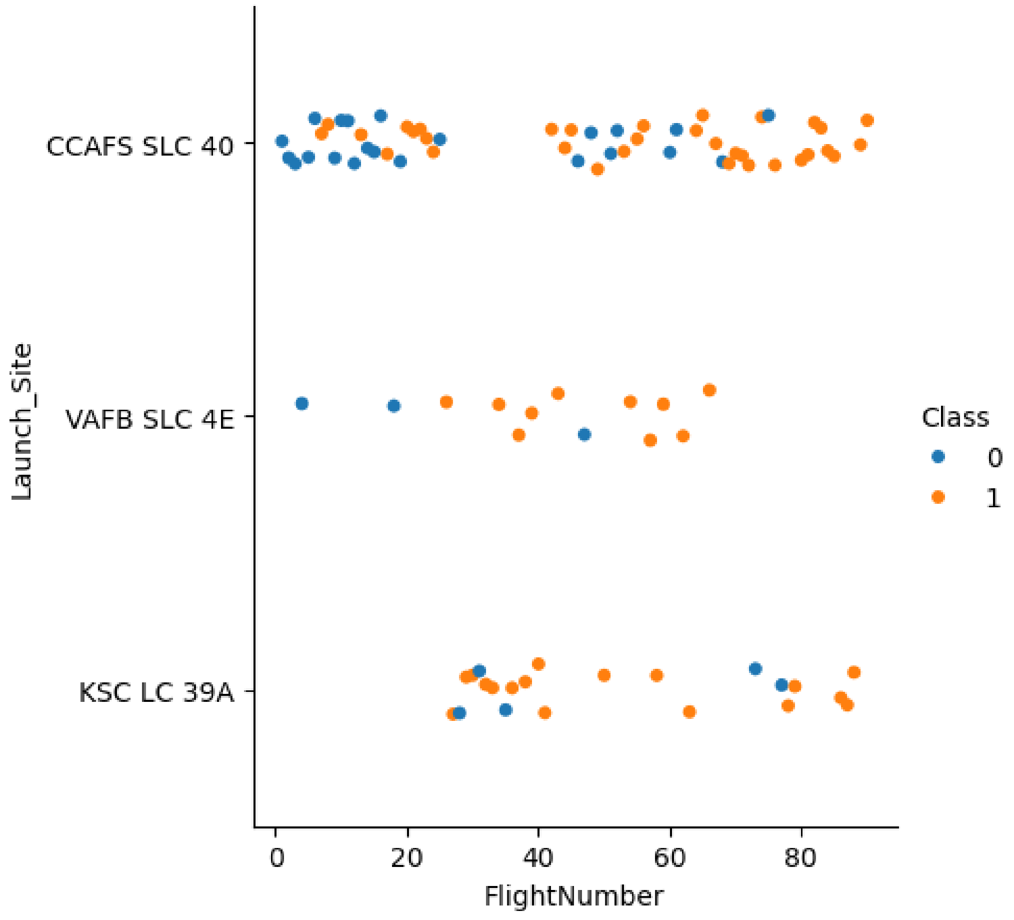
This results are assuming a parameter cv = 10 when using Grid Search so it is possible that better models exist.

Section 2

Insights drawn from EDA

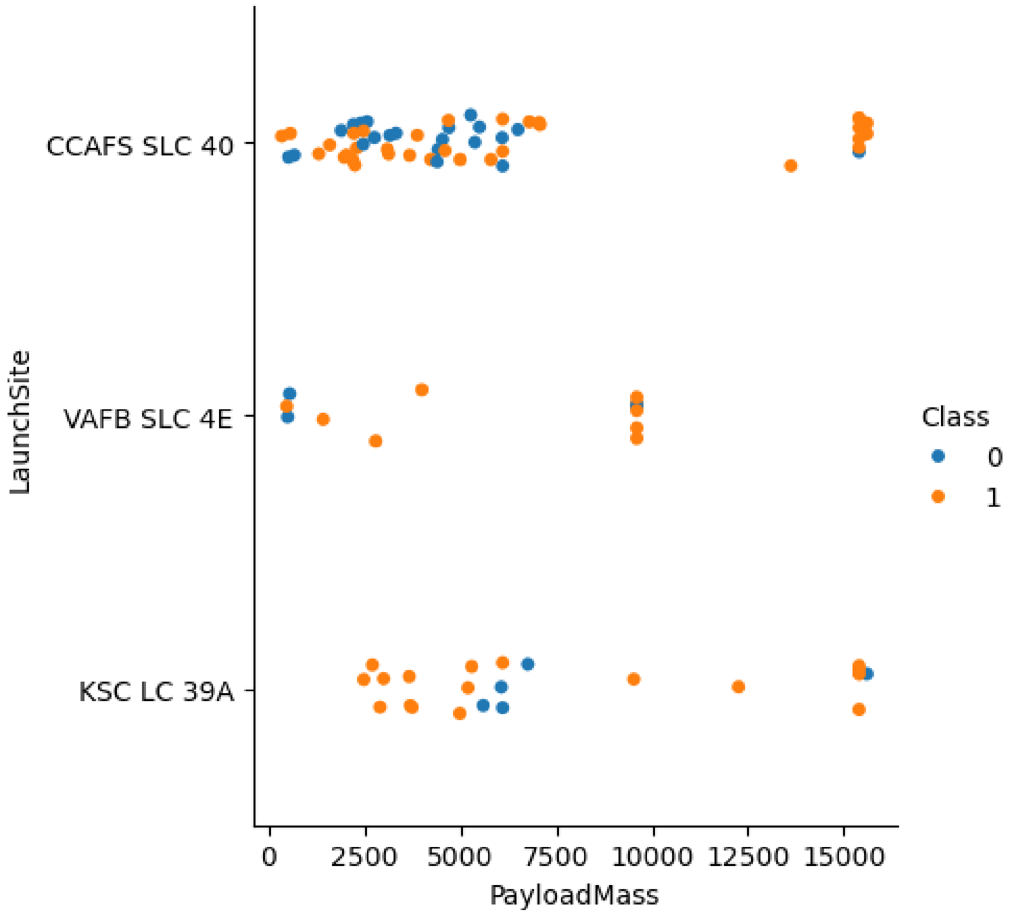
Flight Number vs. Launch Site

- The flights of launch site CCAFS SLC 40 have a higher rate and has a correlation with the number of flights.
- KSC LC 39A shows a better rate of success than the other sites.
- VAFB SLC 4E doesn't have as many launches as the other sites but it shows a nice success rate.



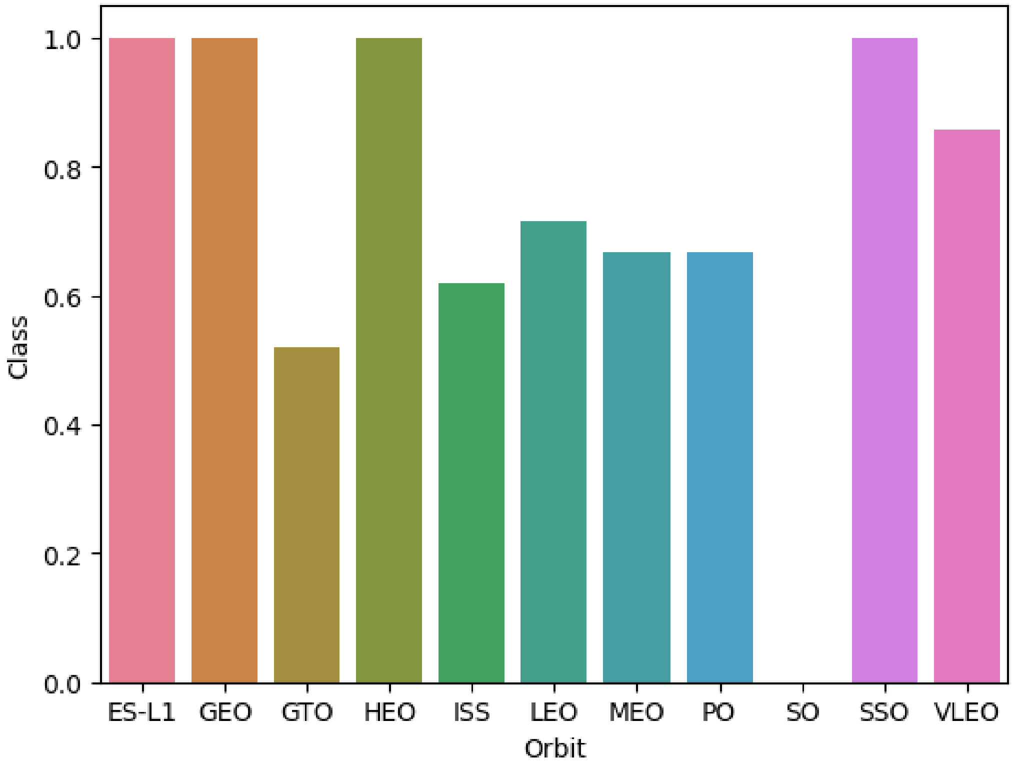
Payload vs. Launch Site

- It can be assumed that if the payload mass is higher the probability for success is higher.
- Most launches with payloads that were over 7500 kg were successful and as such we could see a trend.



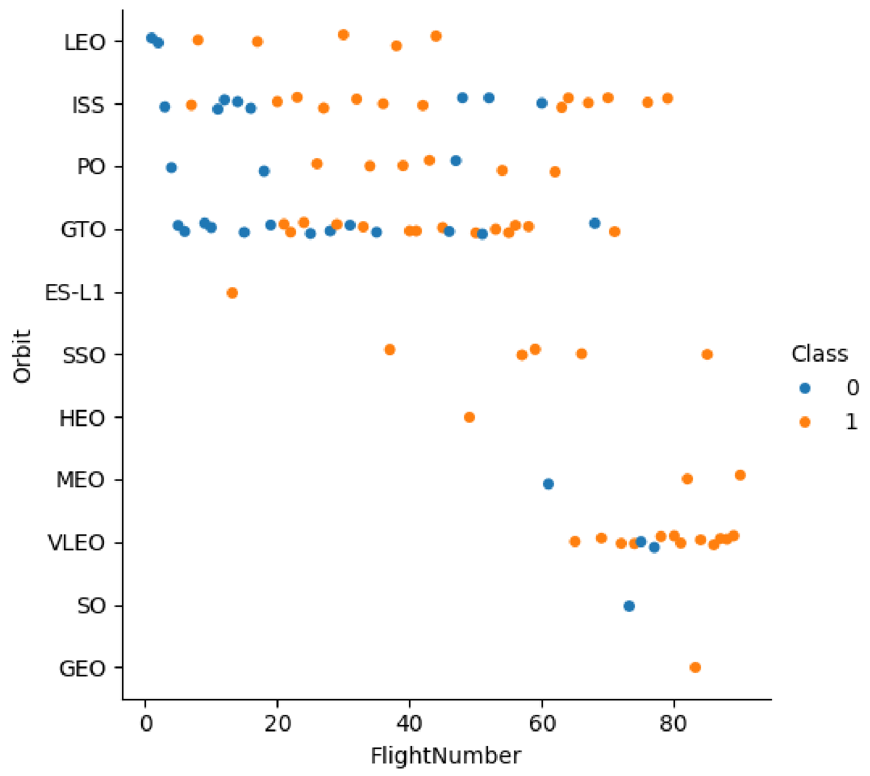
Success Rate vs. Orbit Type

- There are some orbits that with the data analyzed have a 100% success rate:
  - ES-L1
  - GEO
  - HEO
  - SSO
- Other orbits have a medium amount of success rate except for VLEO that has a rate superior to 80%.



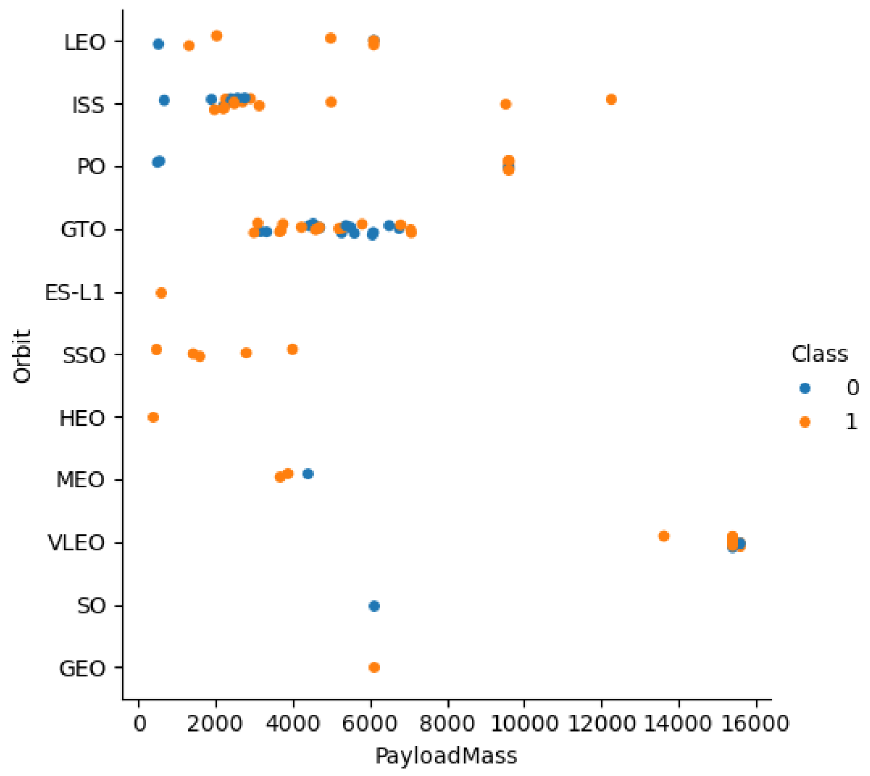
Flight Number vs. Orbit Type

- We can see that ES-L1, SSO, GEO and HEO have a 100% success rate but are also some of the least flighted orbits.
- Flights in the LEO orbit have a higher success rate which seem to improve with the number of flights.



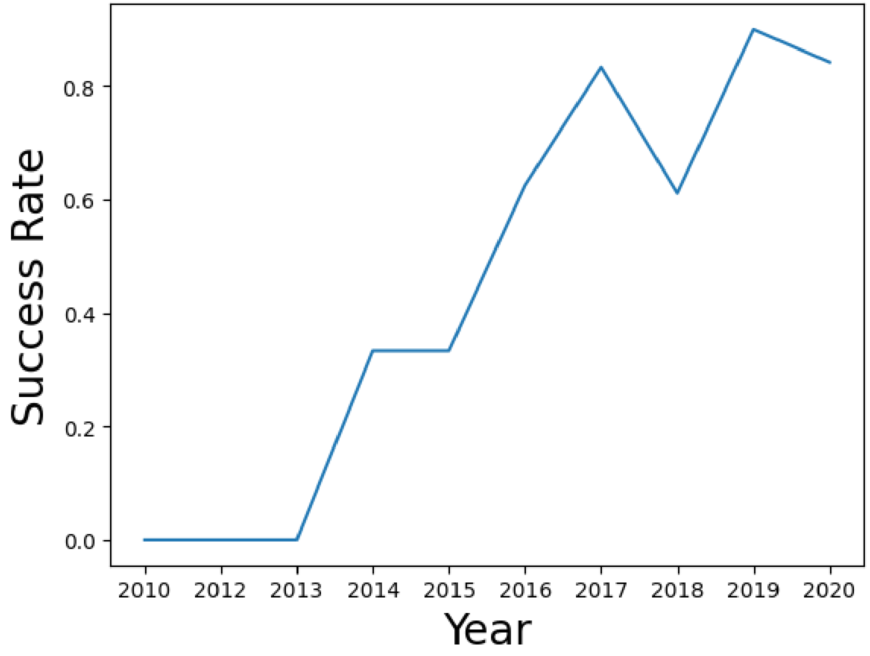
Payload vs. Orbit Type

- We can see that ES-L1, SSO, GEO and HEO which have a 100% success rate only have a payload that is lower to 6000 kg.
- Higher payloads masses in the LEO, ISS, PO and VLEO seem to indicate a better success if the payload is higher.



Launch Success Yearly Trend

- Success rates are growing since the year 2013 with three exception being in the years: 2015, 2018 and 2020.



All Launch Site Names

- We use select and distinct to get the unique launch sites present in the data.

```
%sql select distinct Landing_Outcome from SPACEXTBL
```

Python

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

Landing_Outcome
Failure (parachute)
No attempt
Uncontrolled (ocean)
Controlled (ocean)
Failure (drone ship)
Precluded (drone ship)
Success (ground pad)
Success (drone ship)
Success
Failure
No attempt

Launch Site Names Begin with 'CCA'

- We use where and like to apply a condition for the data we want to get, after that we limit the number of results.

```
%sql select * from SPACEXTBL WHERE Launch_Site Like 'CCA%' Limit 5
```

Python

```
 * 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

- We use the aggregation function sum to get the total payload mass carried where the customer was NASA (CRS).

```
%sql select sum(PAYLOAD_MASS__KG_) as 'Total Payload Mass' from SPACEXTBL where Customer = 'NASA (CRS)'
```

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

Total Payload Mass
45596

Average Payload Mass by F9 v1.1

- We use the aggregation function avg to get the average payload mass carried by the booster version F9 v1.1 filtered by where.

```
%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

First Successful Ground Landing Date

- We use the aggregation function min to get first successful landing outcome when the landing outcome was ground pad.

```
1  %sql select min(Date) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'
```

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

min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- We use where and between to get the successful drone ship launches with a payload between 4000 and 6000.

```
1  %sql select Booster_Version from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' and (PAYLOAD_MASS__KG_ between 4000 and 6000)
```

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

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

Total Number of Successful and Failure Mission Outcomes

- We use the aggregation function count to get the total number of successful and failed mission outcomes.

```
1  %sql select Mission_Outcome, count(*) as Total from SPACEXTBL group by Mission_Outcome
```

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

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

Boosters Carried Maximum Payload

- We use a subquery and the aggregation function max to get the list of the names of booster version which have carried the maximum payload mass.

```
%sql select Booster_Version from (select Booster_Version, max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

Booster_Version
F9 B5 B1048.4

2015 Launch Records

- We use a function substr to extract the date and then list the necessary data for the year 2015.

```
%sql select substr(Date, 6, 2) as month, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTBL where substr(Date, 0, 5) = '2015' and Landing_Outcome = 'Failure (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

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

- We rank the landing outcomes between 2010-06-04 and 2017-03-20.

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

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

Landing_Outcome	Total
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

Section 3

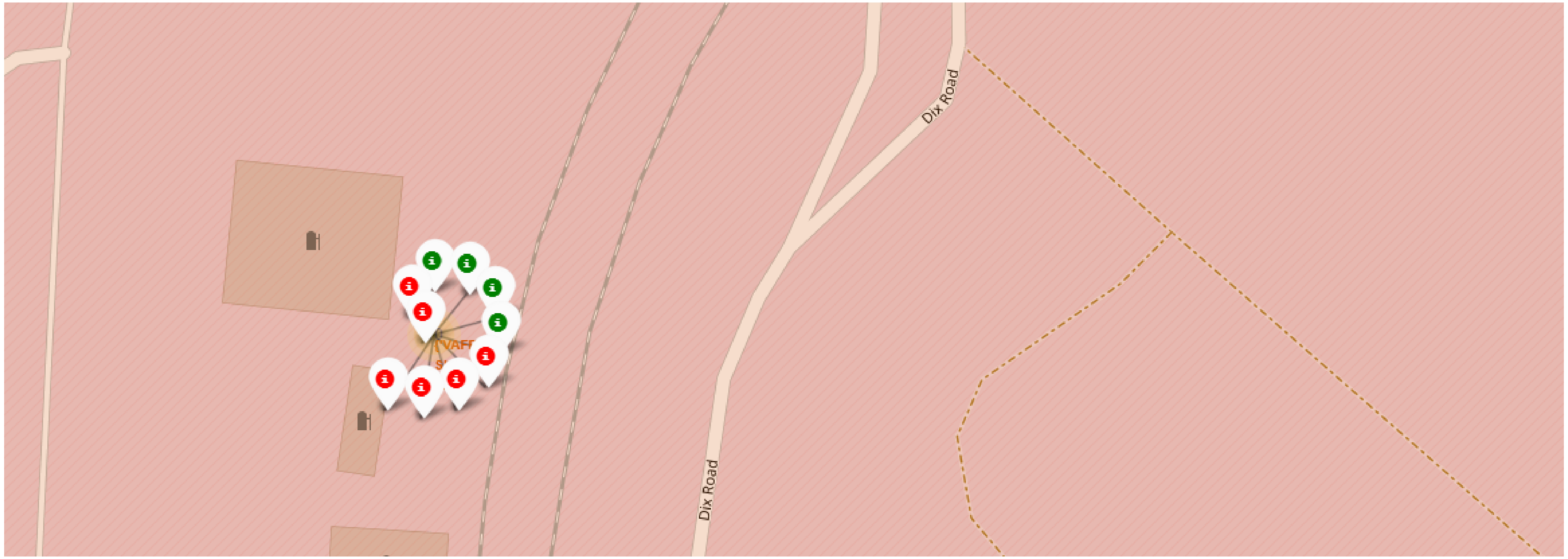
Launch Sites Proximities Analysis

Launch sites on a map

- Launch sites are closer to the coast since it helps to minimize the chances people getting injured due to debris or explosions.

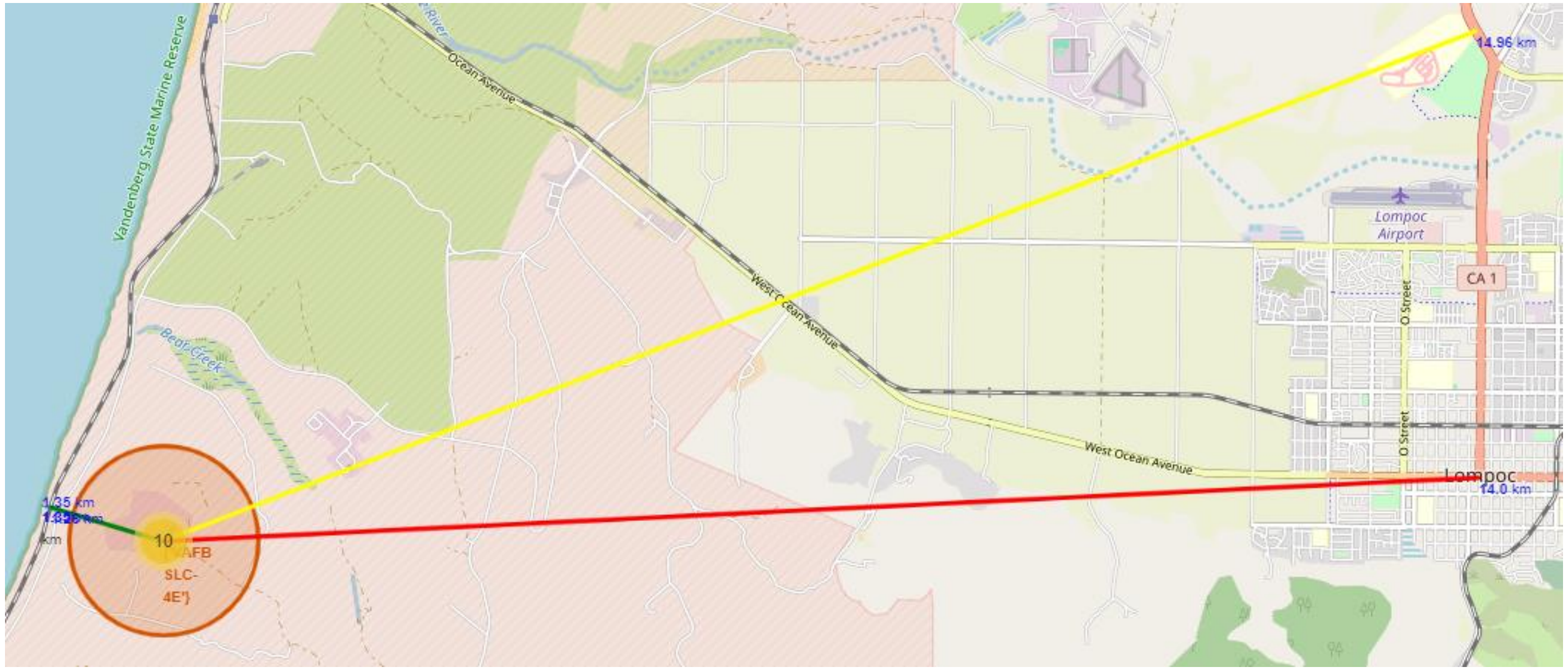
Success/failed launches for each site on the map

- We added colored markers to indicate the number of success/failed launches for each site, the image shows the markers for site VAFB SLC-4E.



Marker for distances to spots on the map

- We added line markers to indicate the distance to the coastline, railways, cities and highways, the image shows the markers for site VAFB SLC-4E.

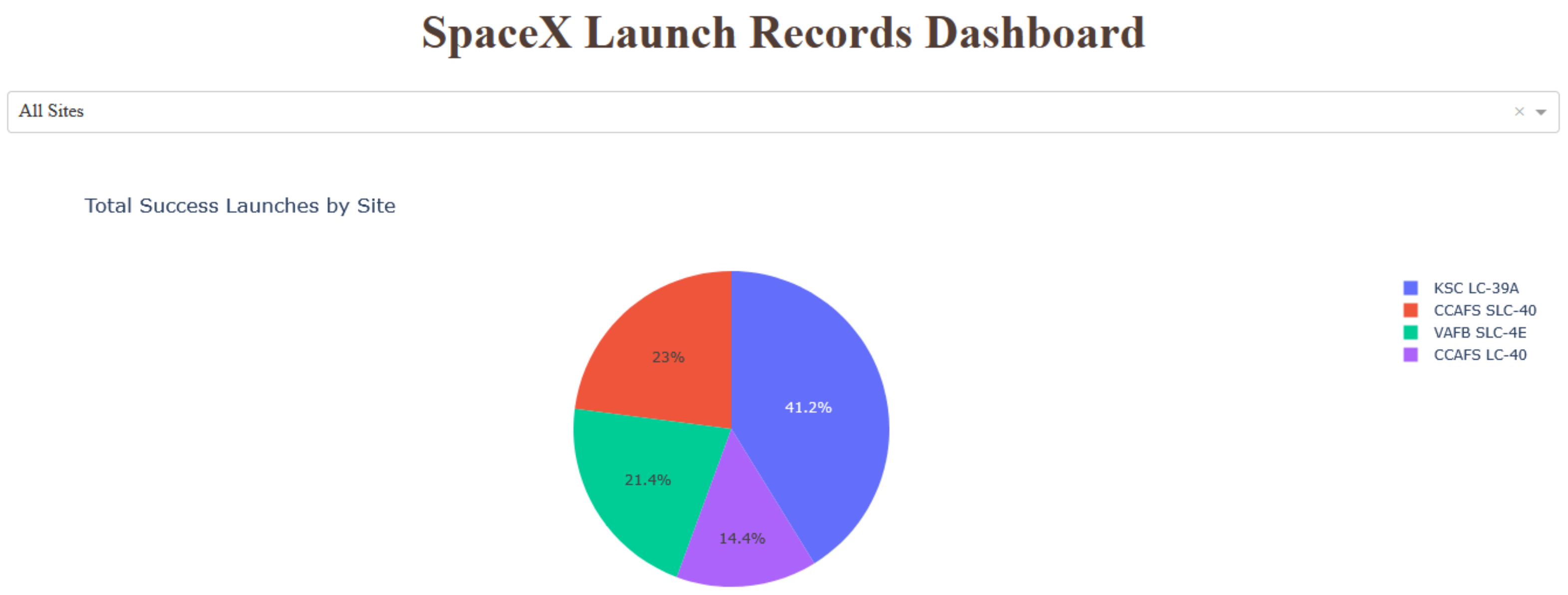


Section 4

Build a Dashboard with Plotly Dash

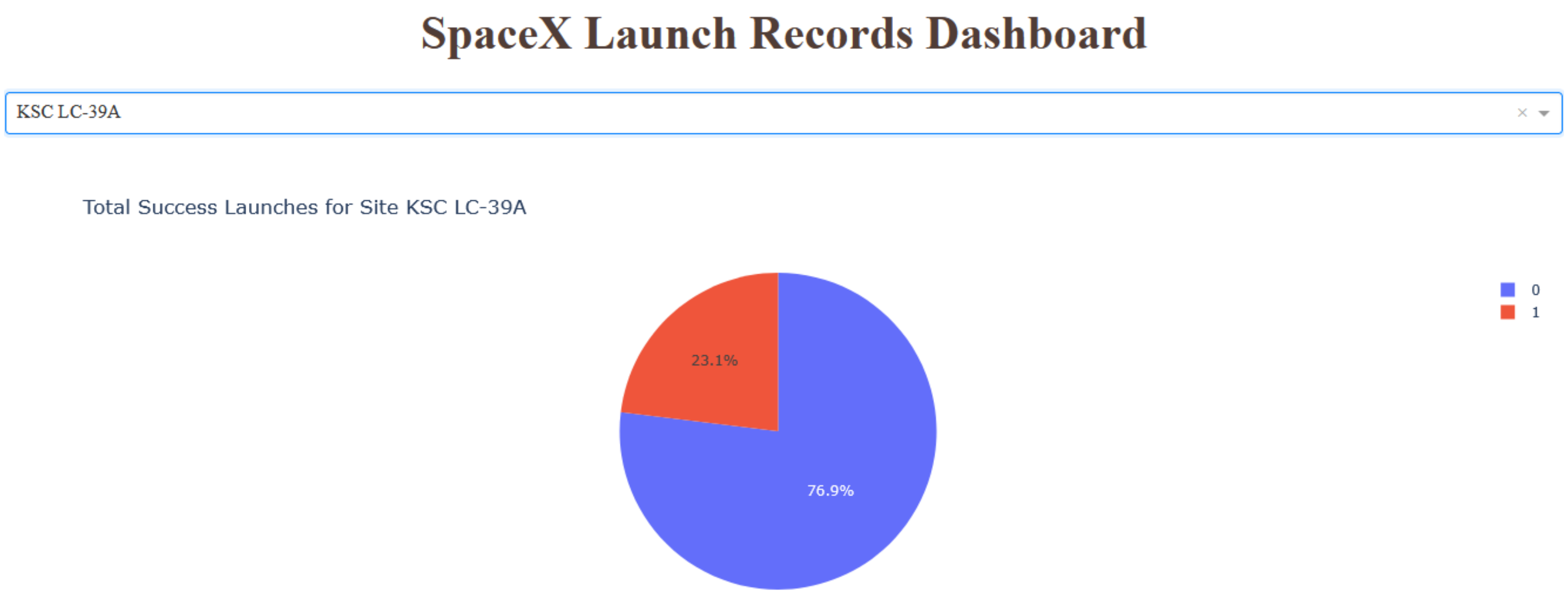
Launch Success Percentages for All Sites in a Pie Chart

- We can see that KSC LC-39A has the higher percentage of successful launches.



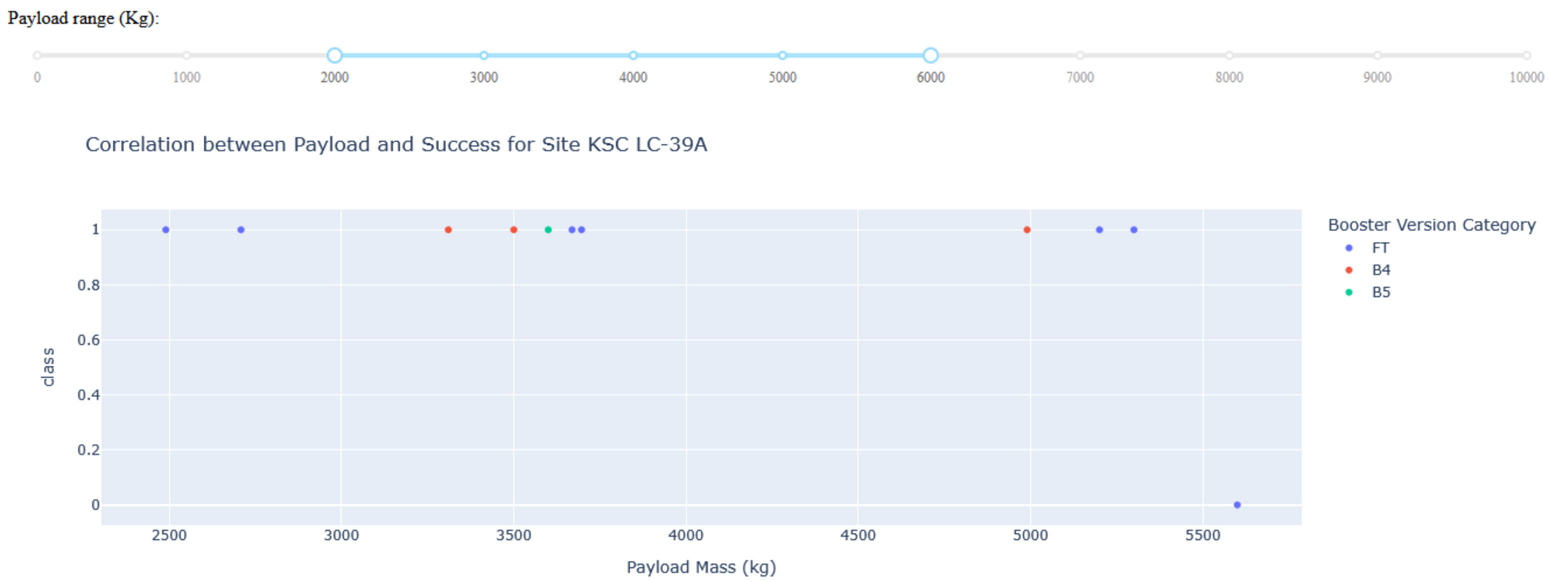
Pie Chart for Launch Site with the Highest Success Ratio

- We can see that KSC LC-39A has a 76.9% on successful launches and only a 23.1% for failed launches.



Payload vs. Launch Outcome Scatterplot for All Sites

- We can see that almost all boosters were successful when they had a payload between 2000 and 6000 kg, with only one exception that is from the FT booster.

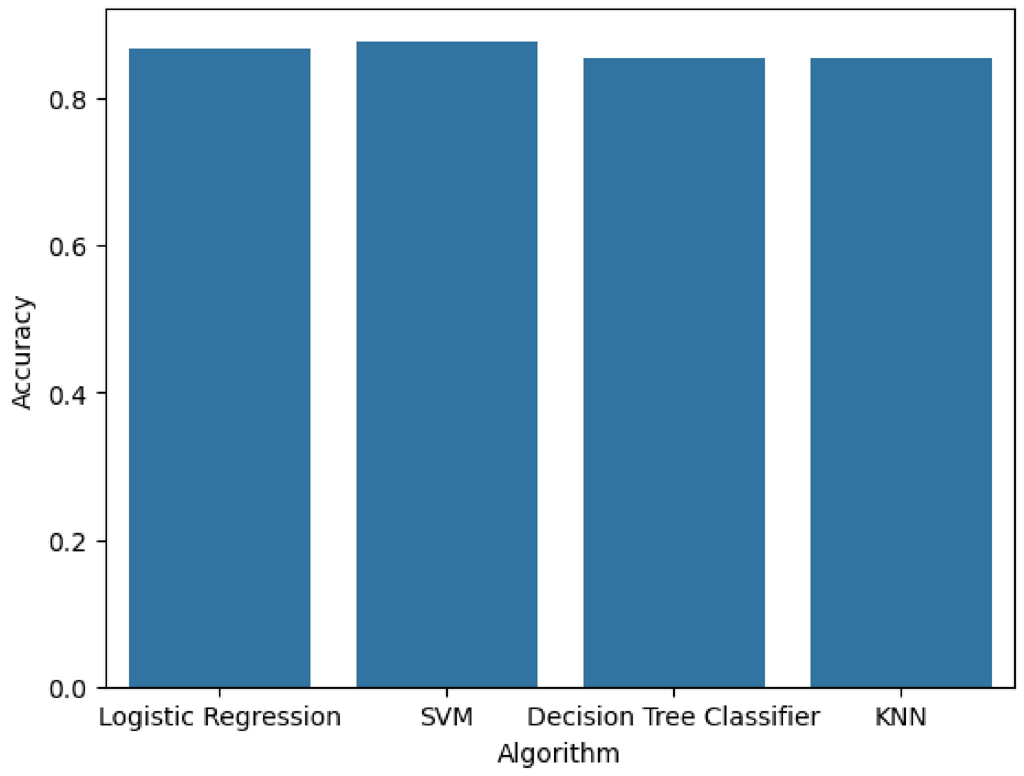


Section 5

Predictive Analysis (Classification)

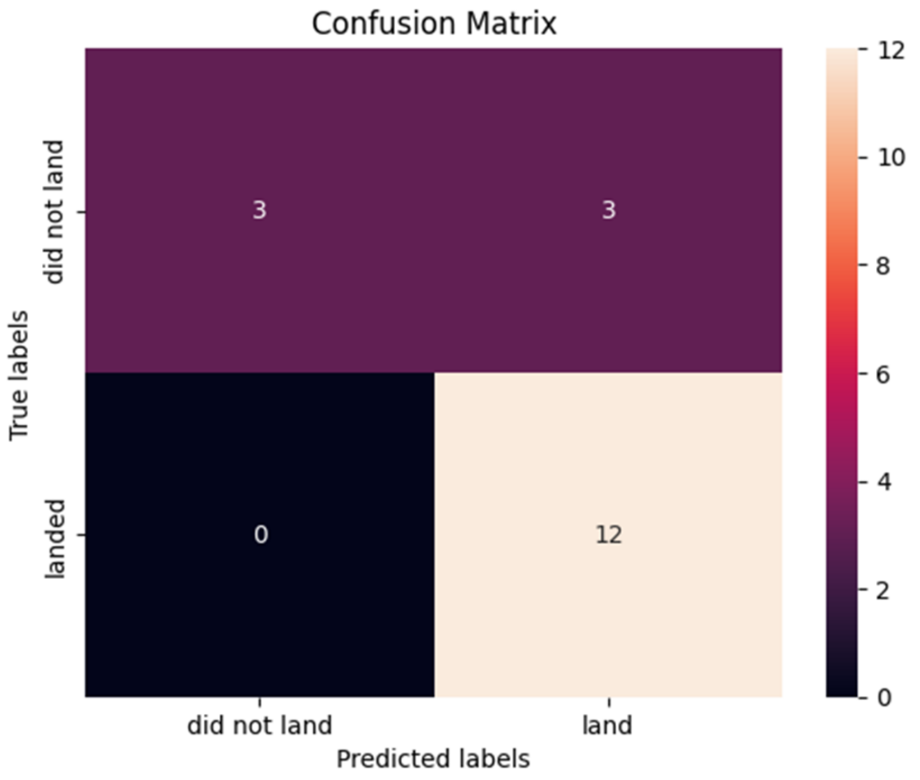
Classification Accuracy

- We can see that all models have a nice accuracy, but the best one is the SVM model.



Confusion Matrix

- We can see in the confusion matrix that only 3 were false positives in the SVM Confusion Matrix.



Conclusions

SVM is the best algorithm for this dataset.

Success in launches depend on the orbit used.

Launch sites are in proximity to the Equator line and all the sites are in very close proximity to the coast.

The success rate of launches usually increases over the years as there are further improvements on technology.

KSC LC-39A has the highest success rate of all the sites.

Orbits ES-L1, GEO, HEO and SSO have 100% success rate.

Appendix

- All data referred in this presentation can be found in the following Github repository: https://github.com/AlexanderPayanoMiranda/Applied-Data-Science-Capstone-Submission

Thank you!