IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Shahzada Umar Abbas Babar

2nd April, 2023

Outline

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

Executive Summary

- Summary of methodologies (SpaceX Data Collection)
  - ✓ Using SpaceX API
  - ✓ Through Web Scraping
  - ✓ SpaceX Data Wrangling
  - ✓ SpaceX Exploratory Data Analysis (SQL)
  - ✓ SpaceX EDA DataViz using Python (Pandas & Matplotlib)
  - ✓ SpaceX Launch Sites Analysis (Folium Interactive Visual Analytics & Ploty Dash)
  - ✓ SpaceX Machine Learning (Landing Prediction)
- **Summary of all results**
  - EDA Results
  - Interactive Visual Analytics & Dashboards
  - Predictive Analysis (Classification)

Introduction

- Project background and context
  - SpaceX Falcon 9 Rocket Launches cost $62 million
  - Competitors’ cost $165 + million
  - SpaceX reuses First Stage resulting in Cost Reductions
  - Predicting if first stage shall land helps determining cost of a launch
  - Information may be used by competitors
- Problems you want to find answers
  - We want to Predict if Falcon 9 (1st Stage) shall land successfully from the data shared on SpaceX website

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was 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

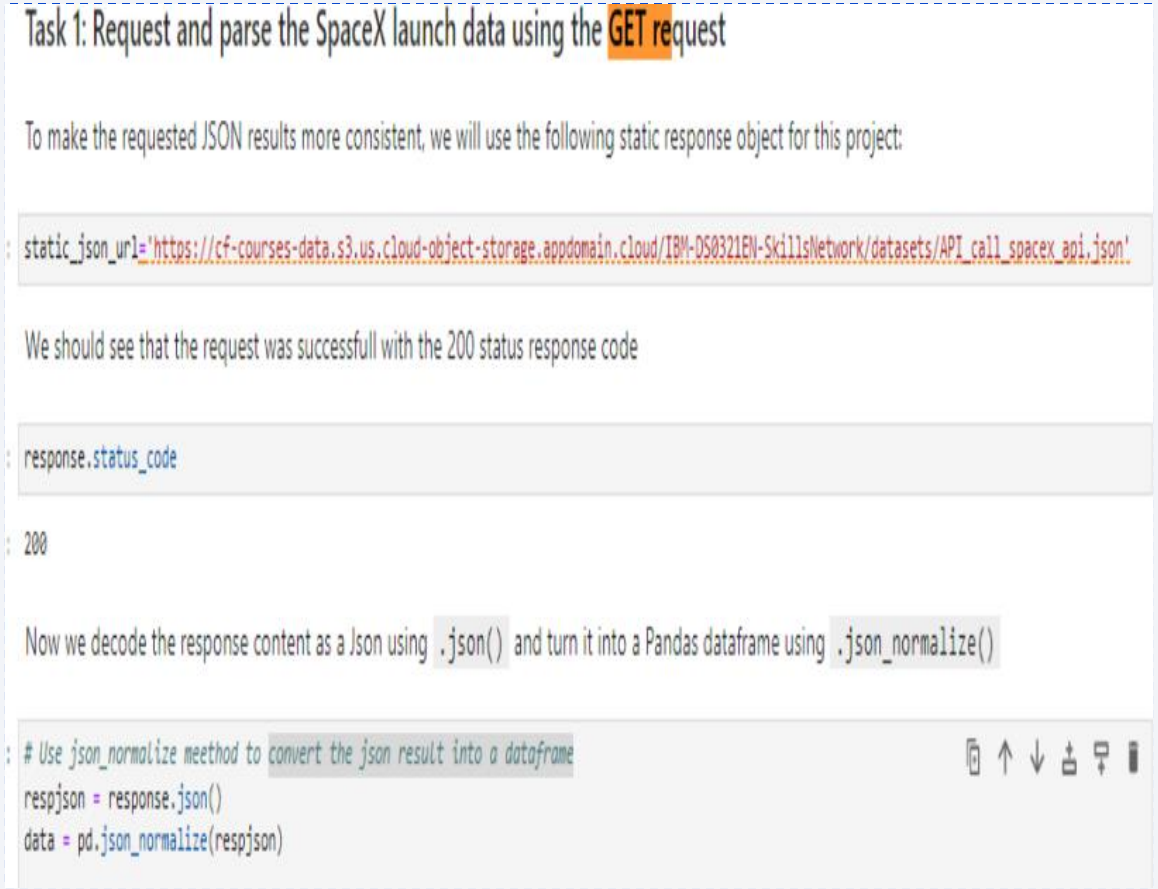
Data Collection

- SpaceX Falcon 9 Data was Collection (Methodologies)
  - Initially, information was gathered via the use of SpaceX API (an API based on REST architecture) by sending a get request to the API.
    - The process involved the creation of several utility functions that facilitated the utilization of the API for retrieving launch details using unique identification numbers, followed by retrieving rocket launch data from the SpaceX API's URL.
  - Ultimately, to ensure uniformity in the JSON results that were required, the SpaceX launch data was retrieved and analyzed via the GET request method. The response content was then decoded as a JSON outcome, which was further transformed into a Pandas data frame.
  - Furthermore, web scraping was conducted to accumulate past launch data of Falcon 9 from a Wikipedia page named "List of Falcon 9 and Falcon Heavy launches." The launch records were stored in an HTML format. Employing the BeautifulSoup and request libraries, the Falcon 9 launch data was extracted from the relevant HTML table on the Wikipedia page. Subsequently, the table was parsed and transformed into a Pandas data frame.

Data Collection – SpaceX API

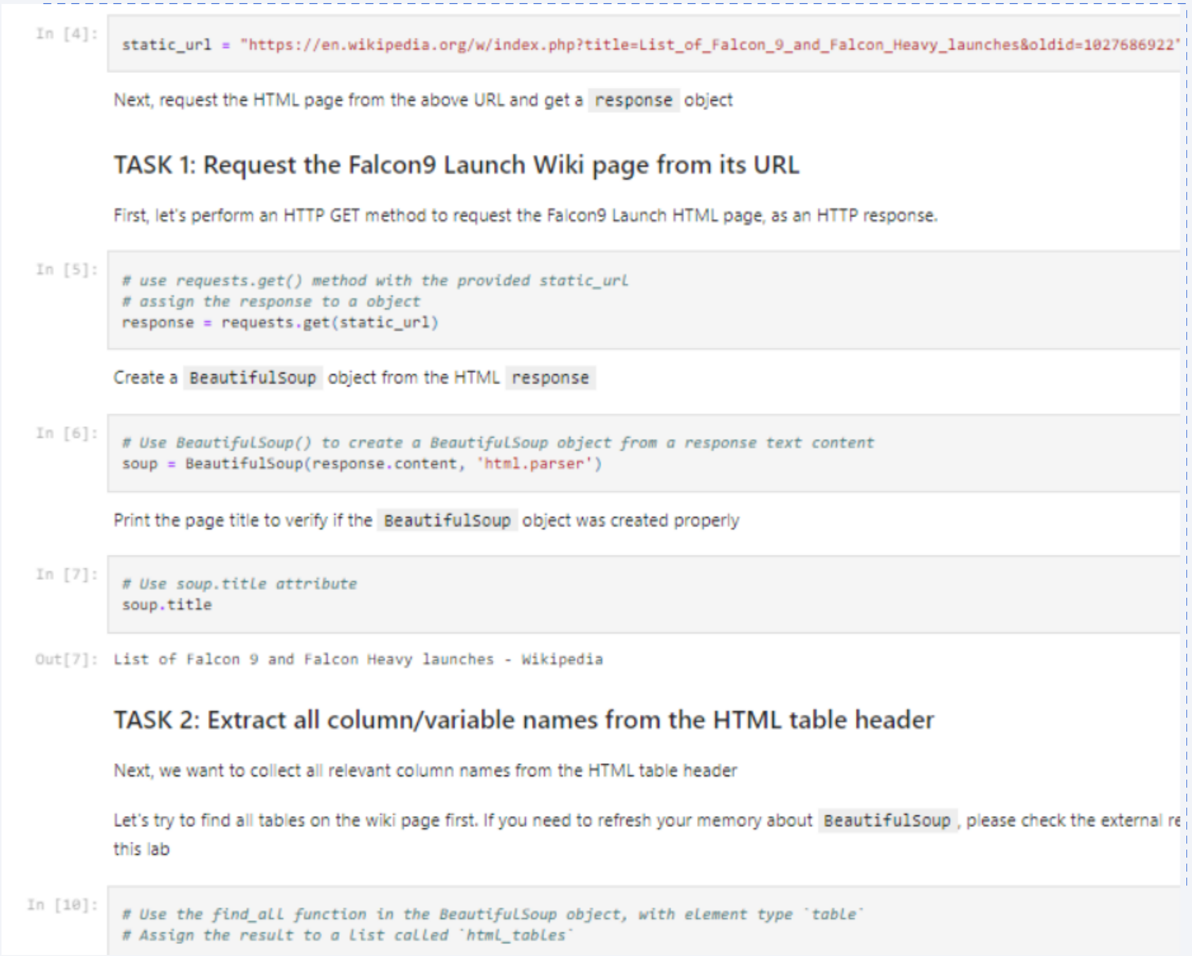
- Information was gathered by sending a GET request to the SpaceX API, which is a RESTful API. Following this, the SpaceX launch data was requested and analyzed using the GET request method. The response content was decoded as a JSON outcome, which was then transformed into a Pandas data frame.

- https://github.com/umarabbas7973/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/d5913cd9f46e3543e005ff2e8be28eaa8668e487/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- I utilized BeautifulSoup and request to scrape data from a Wikipedia page and retrieve the historical launch records of Falcon 9. The process involved extracting the Falcon 9 launch records from an HTML table on the Wikipedia page and then parsing the launch HTML to create a data frame.
- https://github.com/umarabbas7973/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/d5913cd9f46e3543e005ff2e8be28eaa8668e487/jupyter-labs-webscraping.ipynb



Data Wrangling

- Once I had collected the data and converted it into a Pandas data frame, I filtered it by the BoosterVersion column to include only Falcon 9 launches. I then addressed missing values in the LandingPad and PayloadMass columns by replacing the missing PayloadMass values with the mean value of the column. I also conducted Exploratory Data Analysis (EDA) to uncover patterns in the data and identify a suitable label for training supervised models.

- https://github.com/umarabbas7973/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/d5913cd9f46e3543e005ff2e8be28eaa8668e487/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is
variable `landing_class`:

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
df['Class'] = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1)
df['Class'].value_counts()
```

```
1    60
0    30
Name: Class, dtype: int64
```

This variable will represent the classification variable that represents the outcome of each launch. If t
first stage landed Successfully

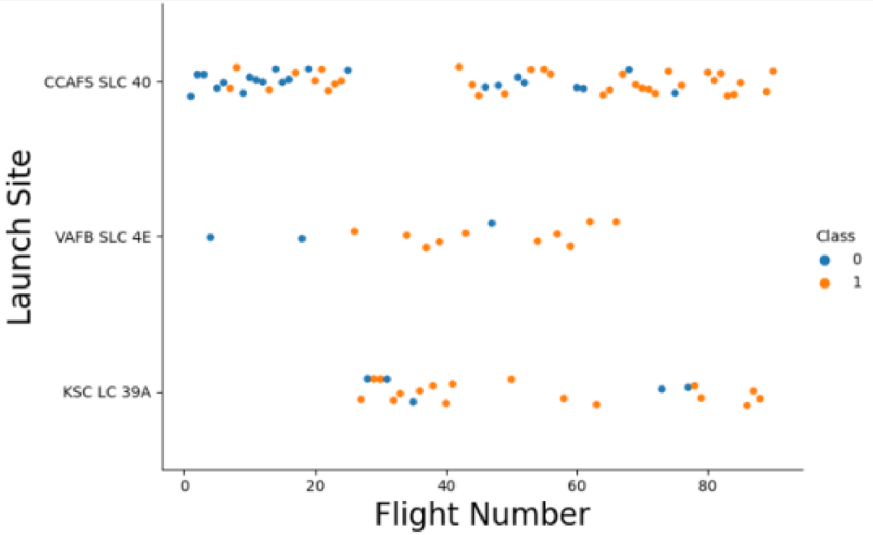
```
landing_class=df['Class']
df[['Class']].head(8)
```

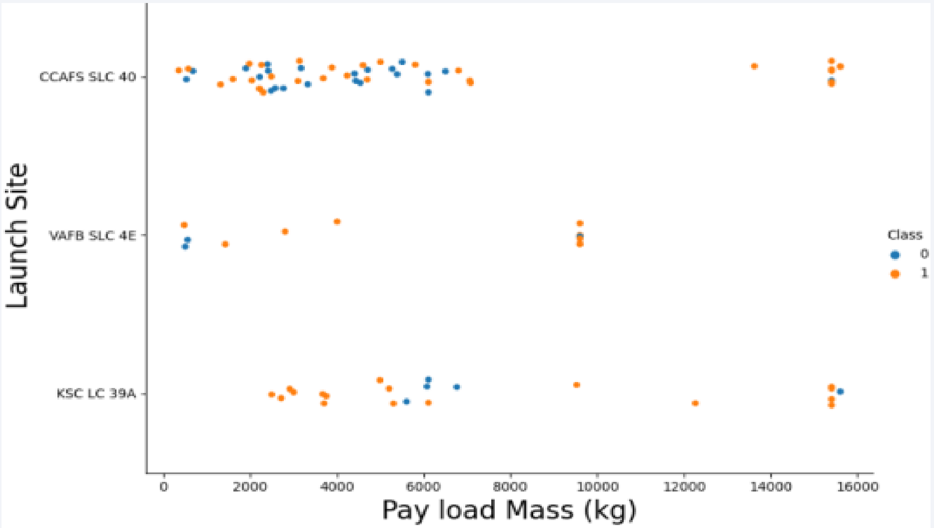
	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

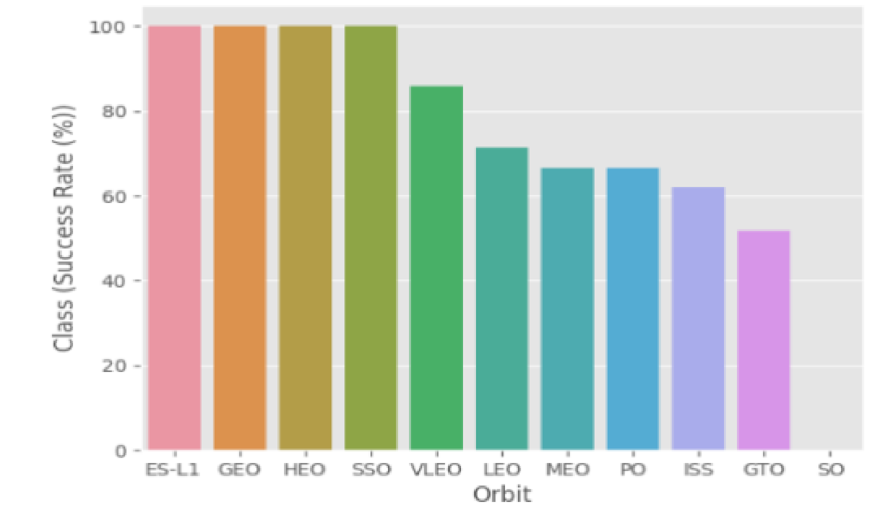
EDA with Data Visualization

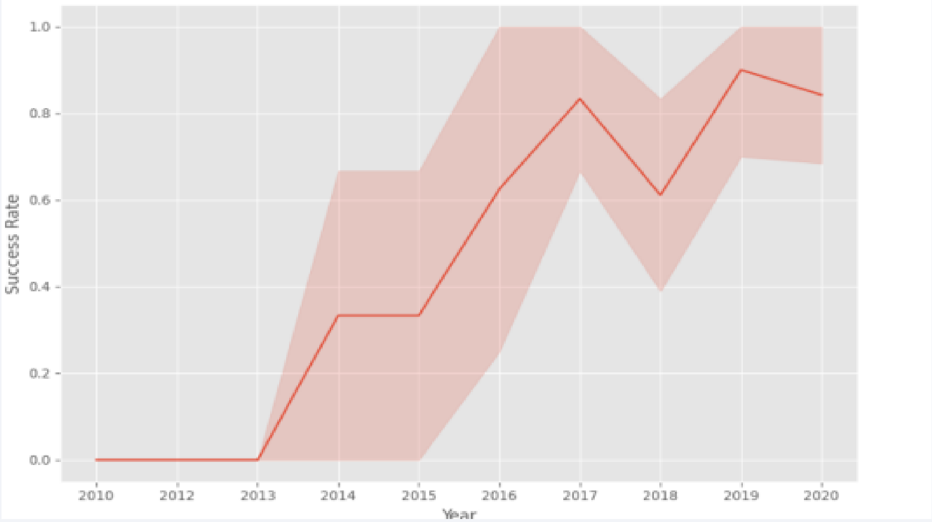
- I conducted data analysis and feature engineering using Pandas and Matplotlib, including exploratory data analysis and data preparation through feature engineering.
- To visualize relationships within the data, I used scatter plots to examine the connections between Flight Number and Launch Site, Payload and Launch Site, Flight Number and Orbit type, and Payload and Orbit type.
- I also used bar charts to visualize the success rate of each orbit type and a line plot to examine the yearly trend of launch success.
- https://github.com/umarabbas7973/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/d5913cd9f46e3543e005ff2e8be28eaa8668e487/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with Data Visualization Plots









EDA with SQL

- The following SQL Queries were performed for EDA
  - Display the names of unique launch sites in the SpaceX mission

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

  - Display 5 records with launch sites beginning with string 'CCA'

```
%sql SELECT * FROM 'SPACEXTBL' WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

  - Display the total Payload Mass carried by Boosters launched by NASA (CRS)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';
```

  - Display the Average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';
```

  - List the date when the first successful landing outcome in Ground Pad was achieved

```
%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";
```

EDA with SQL

- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000 (%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
- List the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

https://github.com/umarabbas7973/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/d5913cd9f46e3543e005ff2e8be28eaa8668e487/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

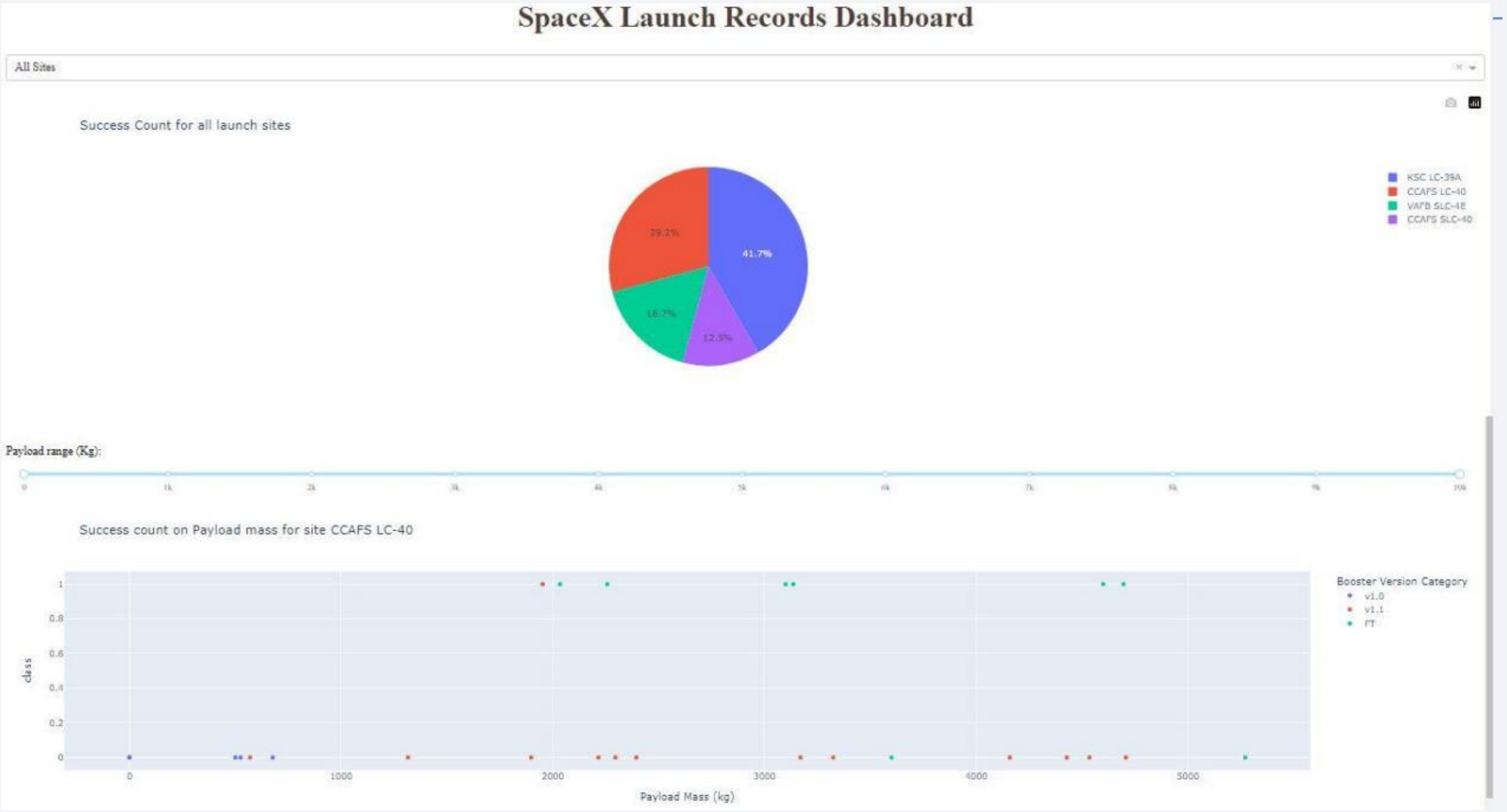
- Folium map was created to indicate the locations of all launch sites, and used map objects like markers, circles, and lines to represent the success or failure of launches at each site.
- Additionally, a launch set was generated that classifies outcomes as either a failure (represented by 0) or success (represented by 1).

- https://github.com/umarabbas7973/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/d5913cd9f46e3543e005ff2e8be28eaa8668e487/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- I developed an interactive dashboard application using Plotly Dash, which involved adding several components. First, I added a Launch Site Drop-down Input Component to the dashboard. Then, I created a callback function that would render a success-pie-chart based on the selected site dropdown. Additionally, I added a Range Slider that allows users to select the payload, and created a callback function to render a success-payload-scatter-chart scatter plot.
- https://github.com/umarabbas7973/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/d5913cd9f46e3543e005ff2e8be28eaa8668e487/Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash%20-%20spacex_dash_app.py

Build a Dashboard with Plotly Dash



Predictive Analysis (Classification)

- To build, evaluate, improve, and identify the best performing classification model, I followed several steps. First, I loaded the data into a Pandas DataFrame and conducted exploratory data analysis to determine the training labels. This involved creating a NumPy array from the "Class" column using the "to_numpy()" method, which I assigned to the variable "Y" as the outcome variable. Next, I standardized the feature dataset (x) using the "preprocessing.StandardScaler()" function from Sklearn. Finally, I split the data into training and testing sets using the "train_test_split" function from sklearn.model_selection, with a test_size of 0.2 and a random_state of 2.
- To determine the best ML model/method for optimal performance on the test data, I followed several steps. First, I created an object for each of the algorithms being considered, including SVM, Classification Trees, k-nearest neighbors, and Logistic Regression. Next, I created a GridSearchCV object and assigned a set of parameters for each model.
- For each of the models under evaluation, I created a GridsearchCV object with cv=10, and then fit the training data into the GridSearch object for each to find the best hyperparameters. After fitting the training set, I outputted the GridSearchCV object for each of the models, and displayed the best parameters using the data attribute "best_params_" and the accuracy on the validation data using the data attribute "best_score_".

- Finally, I used the method "score" to calculate the accuracy on the test data for each model and plotted a confusion matrix for each using the test and predicted

Predictive Analysis (Classification)

- Displayed in the table below are the accuracy scores for the test data, which compares the performance of SVM, Classification Trees, k-nearest neighbors, and Logistic Regression to determine which method performed the best.

https://github.com/umarabbas7973/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/d5913cd9f46e3543e005ff2e8be28eaa8668e487/SpaceX_Machine%20Learning%20Prediction_Part_5%20(1).ipynb

Out[68]:

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Results

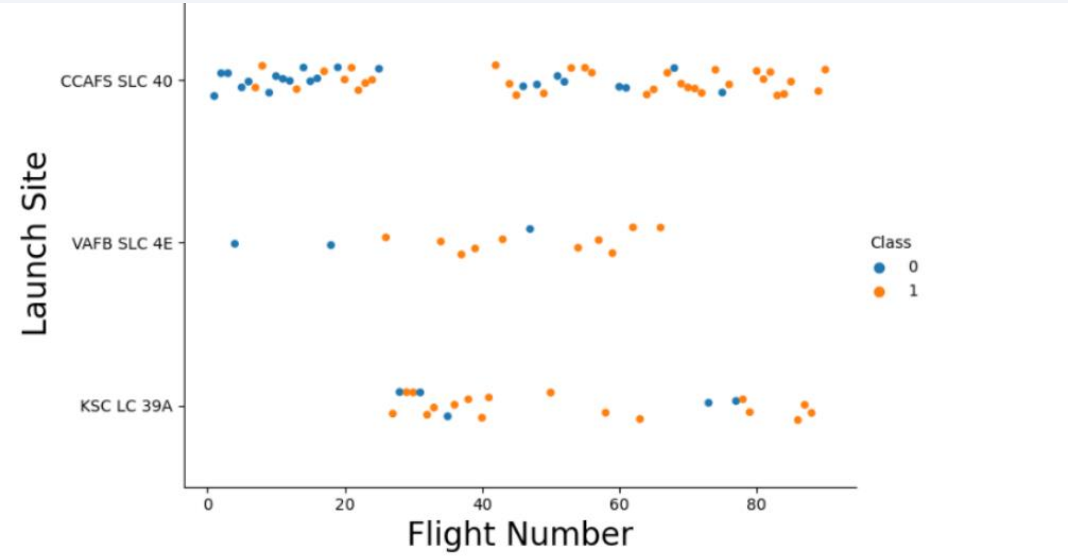
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

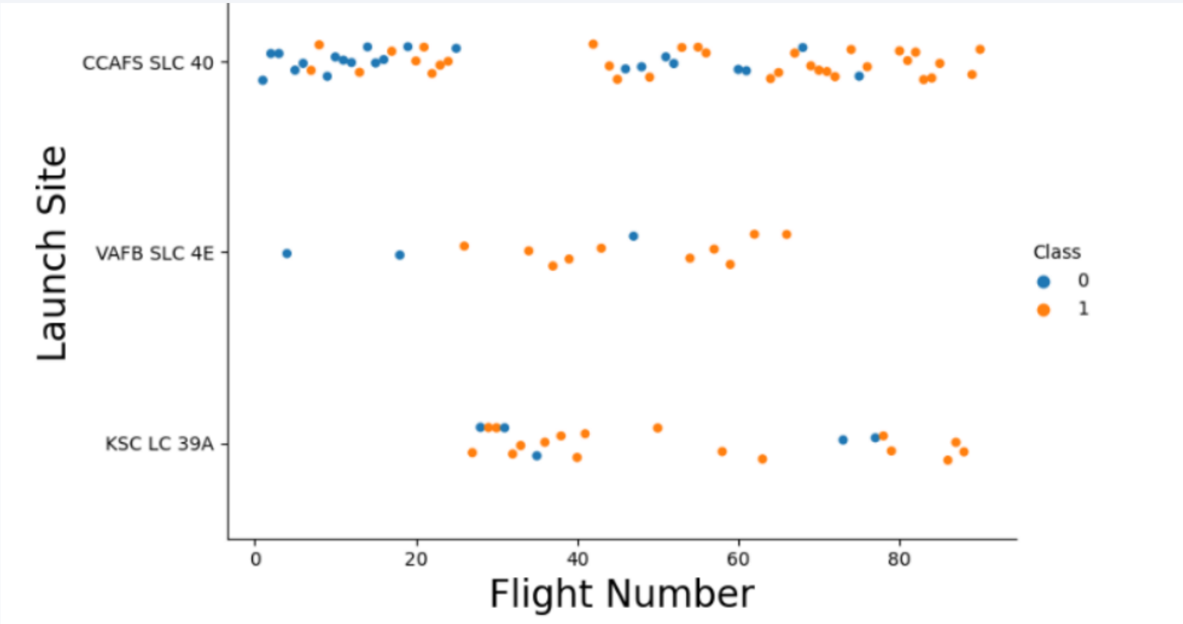
Insights drawn from EDA

Flight Number vs. Launch Site

A Scatter Plot of Flight Number vs. Launch Site



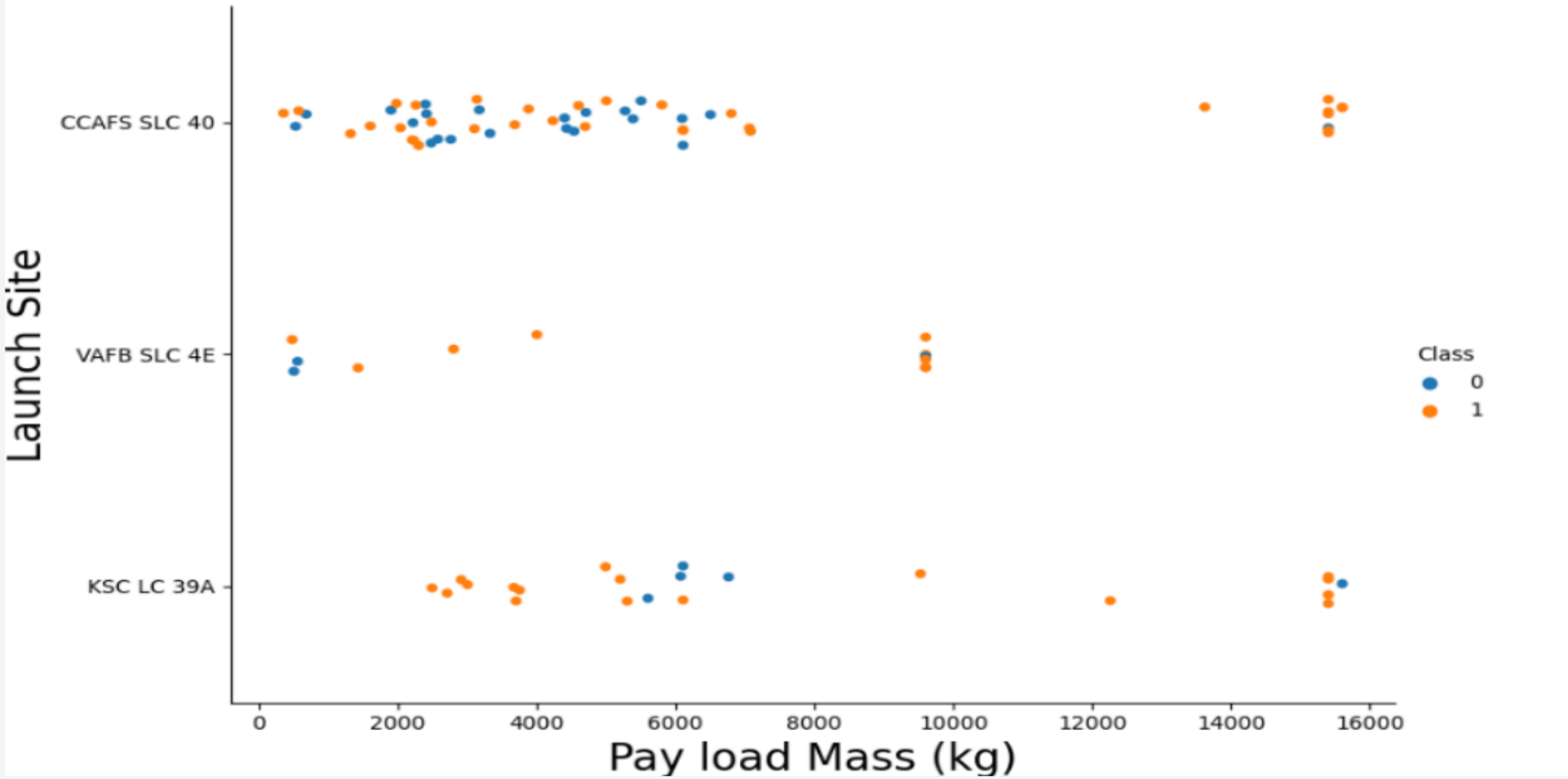
Scatter Plot with Explanations Flight Number vs Launch Site



It can be inferred that as the flight number increases at each of the three Launching Sites, the success rate also increases. After the 50th Flight Number, the VAFB SLC 4E Launch Site had a 100% success rate. Similarly, both KSC LC 39A and CCAFS SLC 40 had a 100% success rate after the 80th flight.

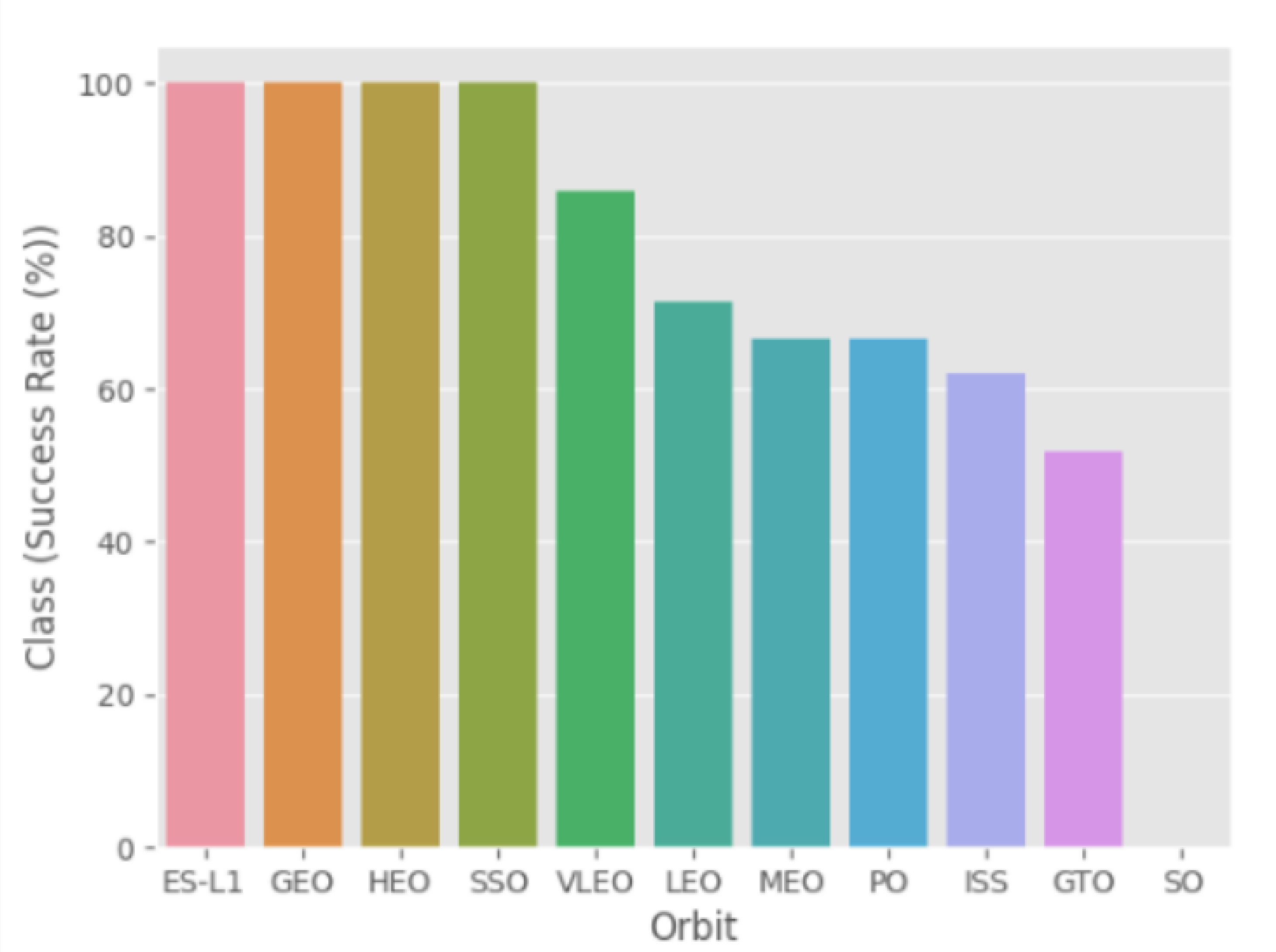
Payload vs. Launch Site

Scatter Plot of Payload vs Launch Site

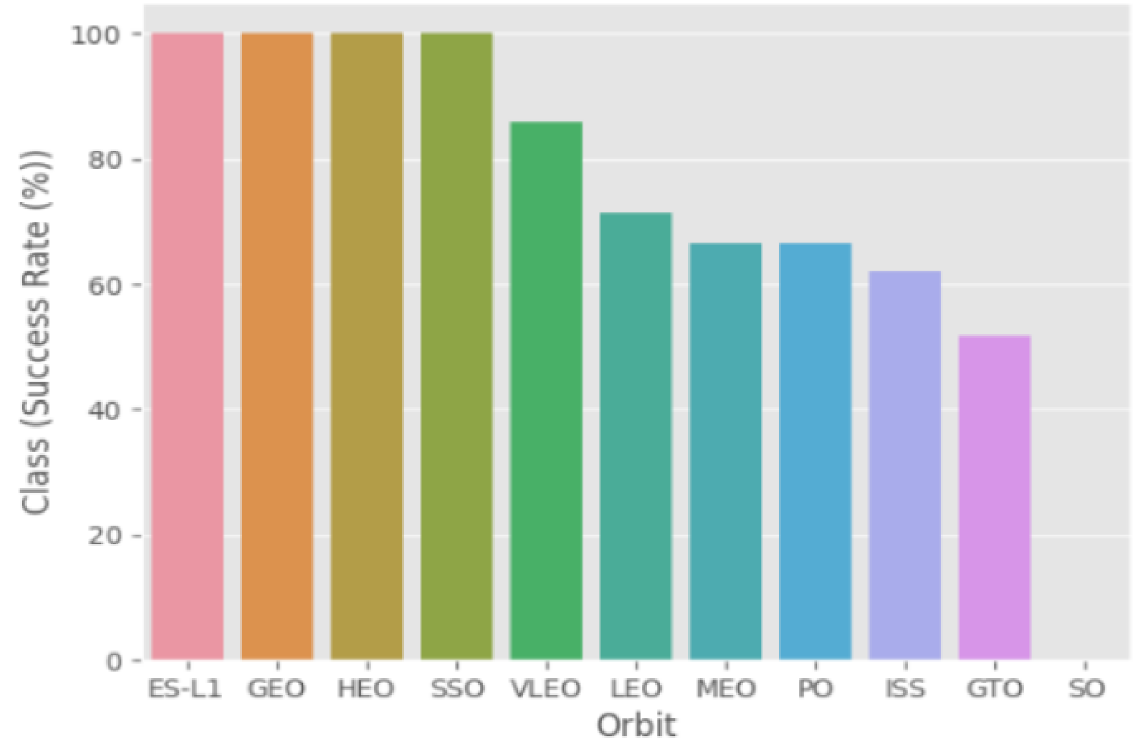


If you take a look at the scatter plot of Payload vs Launch Site, you'll notice that there have been no rockets launched at the VAFB-SLC Launch Sites for heavy payload masses (greater than 10,000).

Success Rate vs. Orbit Type



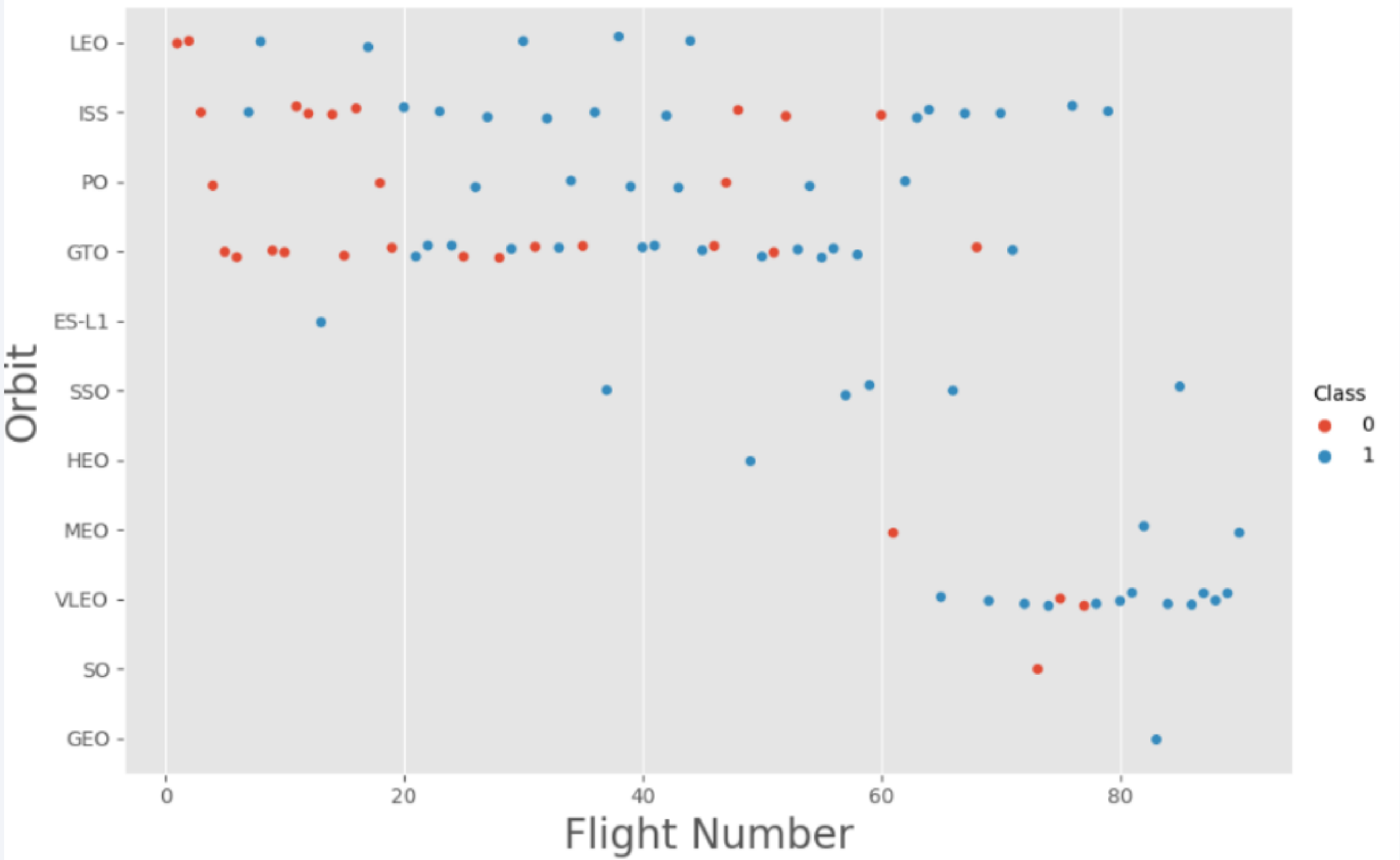
Success Rate vs. Orbit Type



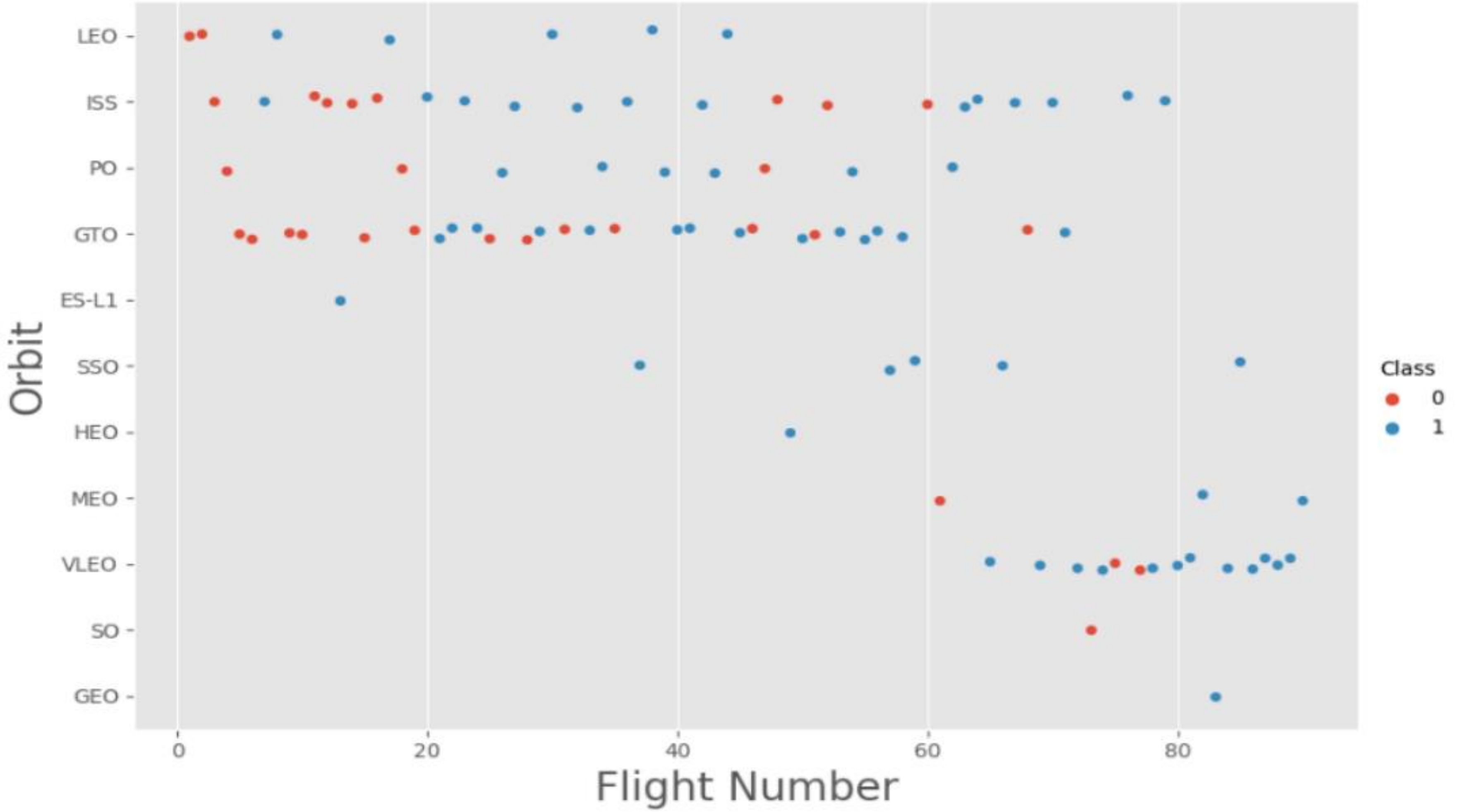
Analyze the ploted bar chart try to find which orbits have high sucess rate.

Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%, with SO orbit having the lowest success rate at ~50%. Orbit SO has 0% success rate.

Flight Number vs. Orbit Type



Flight Number vs. Orbit Type



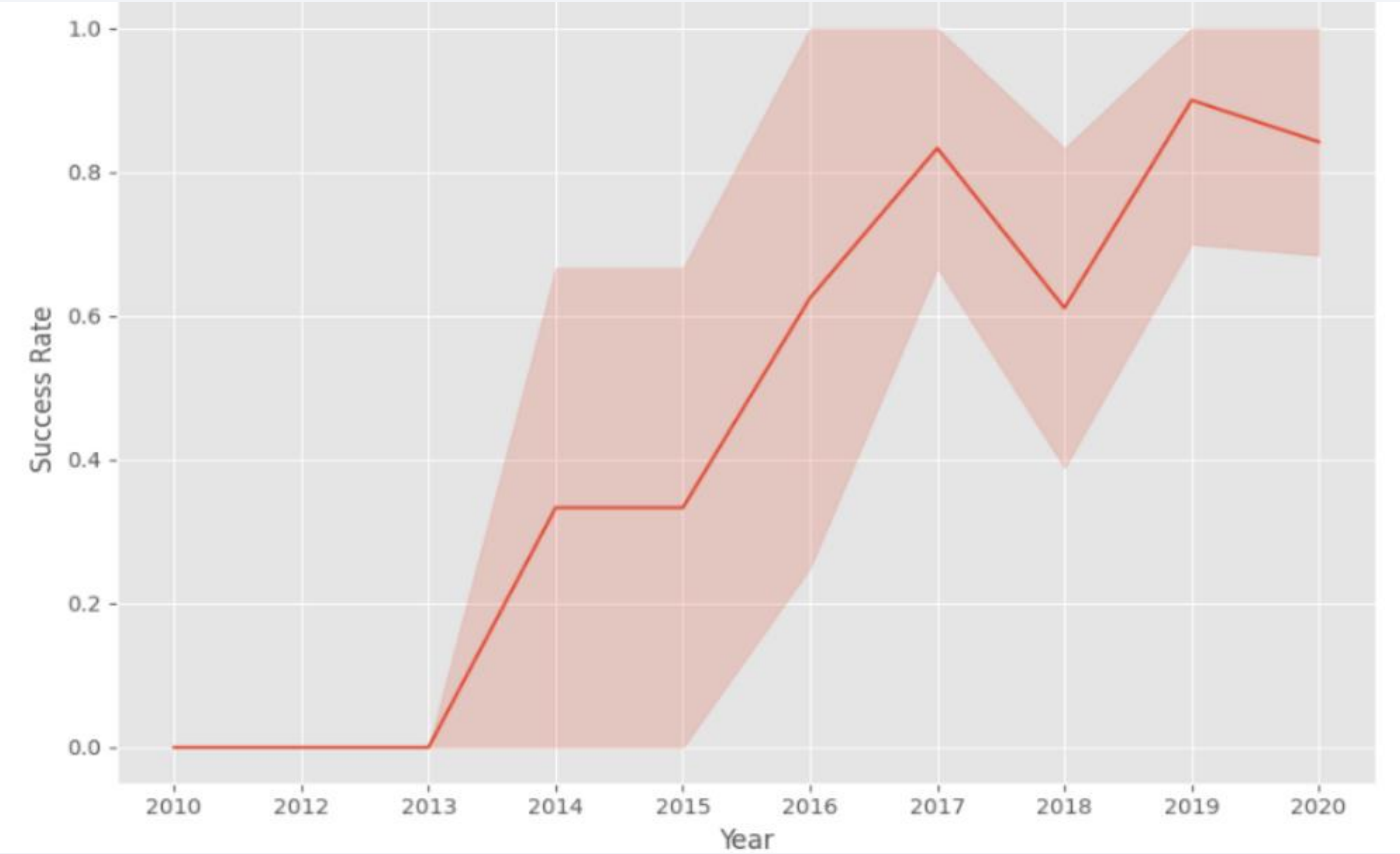
You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type

Payload vs. Orbit Type

- • When it comes to heavy payloads, the success rate for landing or positive landing rate is higher for Polar, LEO, and ISS. •
- However, for GTO, we cannot distinguish this well as both the positive landing rate and negative landing (unsuccessful mission) have nearly equal chances.

Launch Success Yearly Trend



Since 2013 the Success Rate is improving

All Launch Site Names

Find the names of the unique launch sites

Task 1

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

In [31]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

Out[31]:

Launch_Sites
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Used ‘SELECT DISTINCT’ statement to return only the unique launch sites from the ‘LAUNCH_SITE’ column of the SPACEXTBL table

Launch Site Names Begin with 'CCA'

Find 5 records where launch sites begin with `CCA`

Task 2

Display 5 records where launch sites begin with the string 'CCA'

In [72]:
```
%sql SELECT * FROM 'SPACEXTBL' WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

Out[72]:

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

Used 'LIKE' command with '%' wildcard in 'WHERE' clause to select and display a table of all records where launch sites begin with the string 'CCA'

Total Payload Mass

Calculate the total payload carried by boosters from NASA

Task 3

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

In [17]:

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';
```

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

Out[17]:

Total Payload Mass(Kgs)	Customer
45596	NASA (CRS)

Used the ‘SUM()’ function to return and display the total sum of ‘PAYLOAD_MASS_KG’ column for Customer ‘NASA(CRS

Average Payload Mass by F9 v1.1

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

Task 4

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';
```

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

Payload Mass Kgs	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

Used the ‘AVG()’ function to return and display the average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date

Find the dates of the first successful landing outcome on ground pad

Task 5

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

Hint:Use min function

```
%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";
```

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

MIN(DATE)
01-05-2017

Used the 'MIN()' function to return and display the first (oldest) date when first successful landing outcome on ground pad 'Success (ground pad)'happened.

Successful Drone Ship Landing with Payload between 4000 and 6000

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

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
# %sql SELECT * FROM 'SPACEXTBL'
```

```
%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOA
```

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

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

Used ‘Select Distinct’ statement to return and list the ‘unique’ names of boosters with operators >4000 and

Total Number of Successful and Failure Mission Outcomes

Calculate the total number of successful and failure mission outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") 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

Used the ‘COUNT()’ together with the ‘GROUP BY’ statement to return total number of missions outcomes

Boosters Carried Maximum Payload

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

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

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

Using a Subquerry to return and pass the Max payload and used it list all the boosters that have carried the Max payload of 15600kgs

2015 Launch Records

Task 9

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

```
%sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Outcome", "Landing _Outcome"
```

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

substr(Date,7,4)	substr(Date, 4, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	Landing _Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Failure (drone ship)

Used the ‘subsrt()’ in the select statement to get the month and year from the date column where substr(Date,7,4)='2015' for year and Landing_outcome was ‘Failure (drone ship’) and return the records nmatching the filter.

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

Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

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

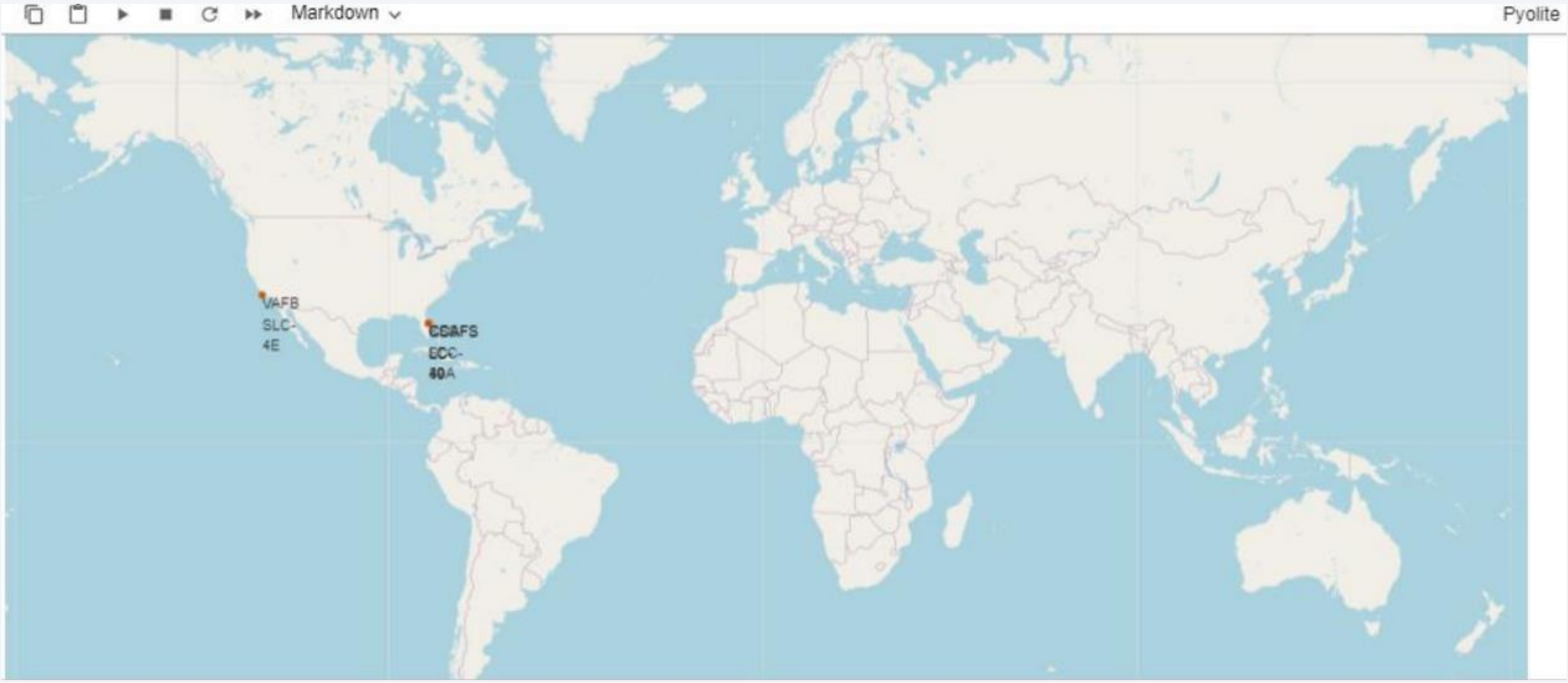
```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-10-2020	12:25:57	F9 B5 B1051.6	KSC LC-39A	Starlink 13 v1.0, Starlink 14 v1.0	15600	LEO	SpaceX	Success	Success
18-08-2020	14:31:00	F9 B5 B1049.6	CCAFS SLC-40	Starlink 10 v1.0, SkySat-19, -20, -21, SAOCOM 1B	15440	LEO	SpaceX, Planet Labs, PlanetIQ	Success	Success
18-07-2016	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-04-2018	22:51:00	F9 B4 B1045.1	CCAFS SLC-40	Transiting Exoplanet Survey Satellite (TESS)	362	HEO	NASA (LSP)	Success	Success (drone ship)

Section 3

Launch Sites Proximities Analysis

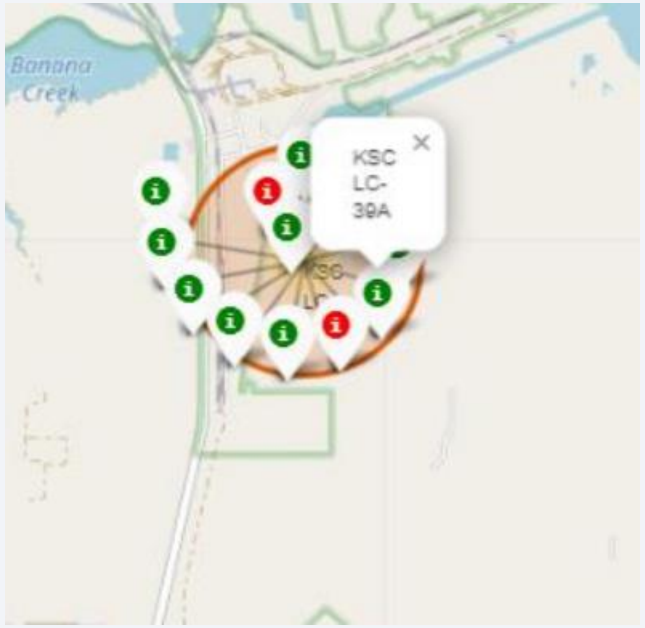
Markers of all launch sites on global map



All launch sites are in proximity to the Equator, (located southwards of the US map). Also all the Launch Sites are in very close proximity to the coast.

Launch outcomes for each site on the map With Color Markers

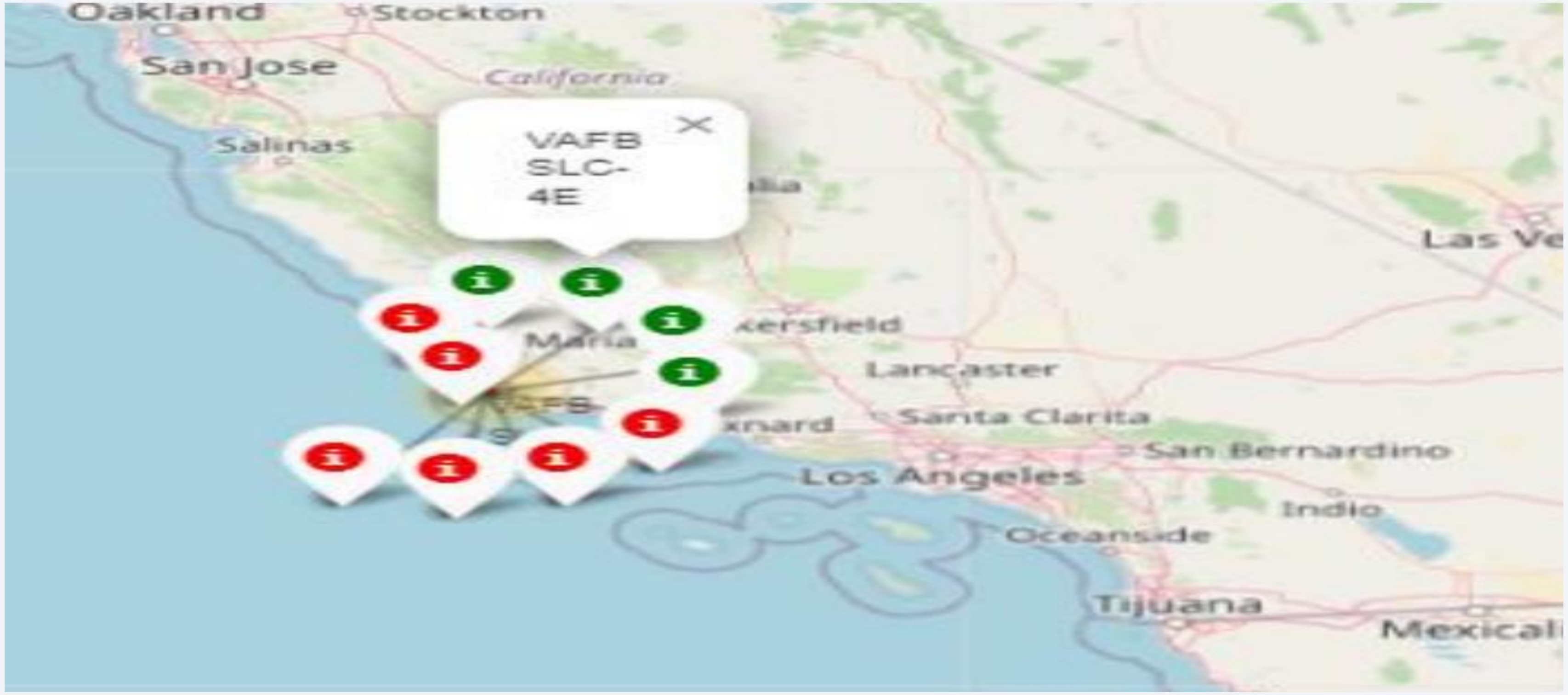
Florida Sites



In the Eastern coast (Florida) Launch site KSC LC-39A has relatively high success rates compared to CCAFS SLC-40 & CCAFS LC-40.

Launch outcomes for each site on the map With Color Markers

West Coast/California

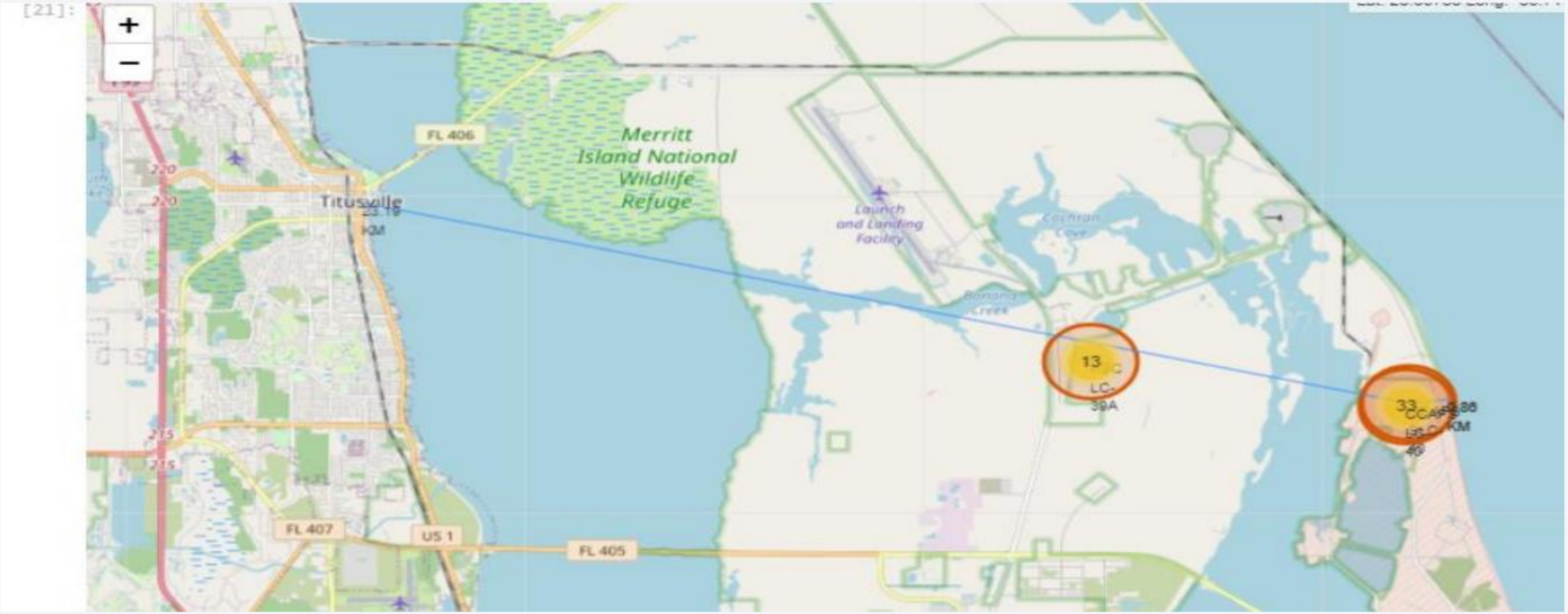


In the West Coast (California) Launch site VAFB SLC-4E has relatively lower success rates 4/10 compared to KSC LC39A launch site in the Eastern Coast of Florida.

Distances between a launch site to its proximities

Launch site CCAFS SLC-40 proximity to coastline is 0.86km

Distances between a launch site to its proximities

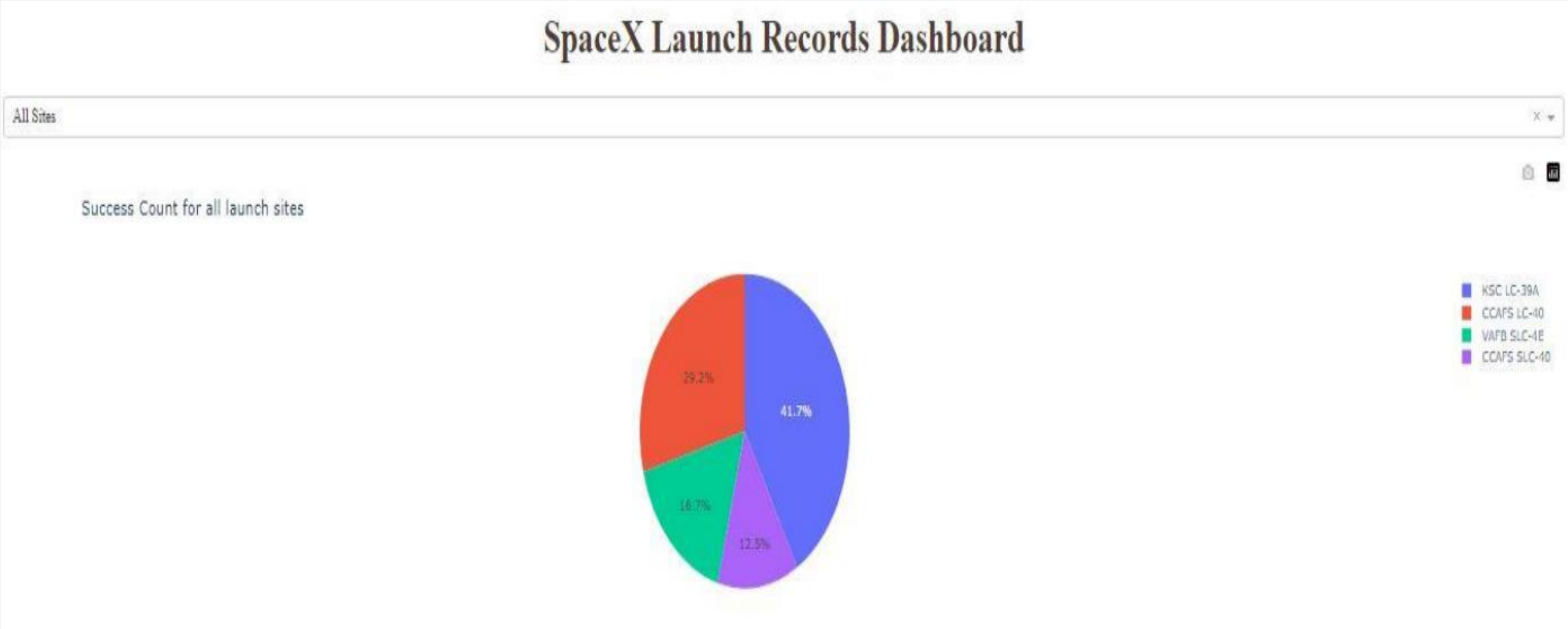


Launch site CCAFS SLC-40 closest to highway (Washington Avenue) is 23.19km

Section 4

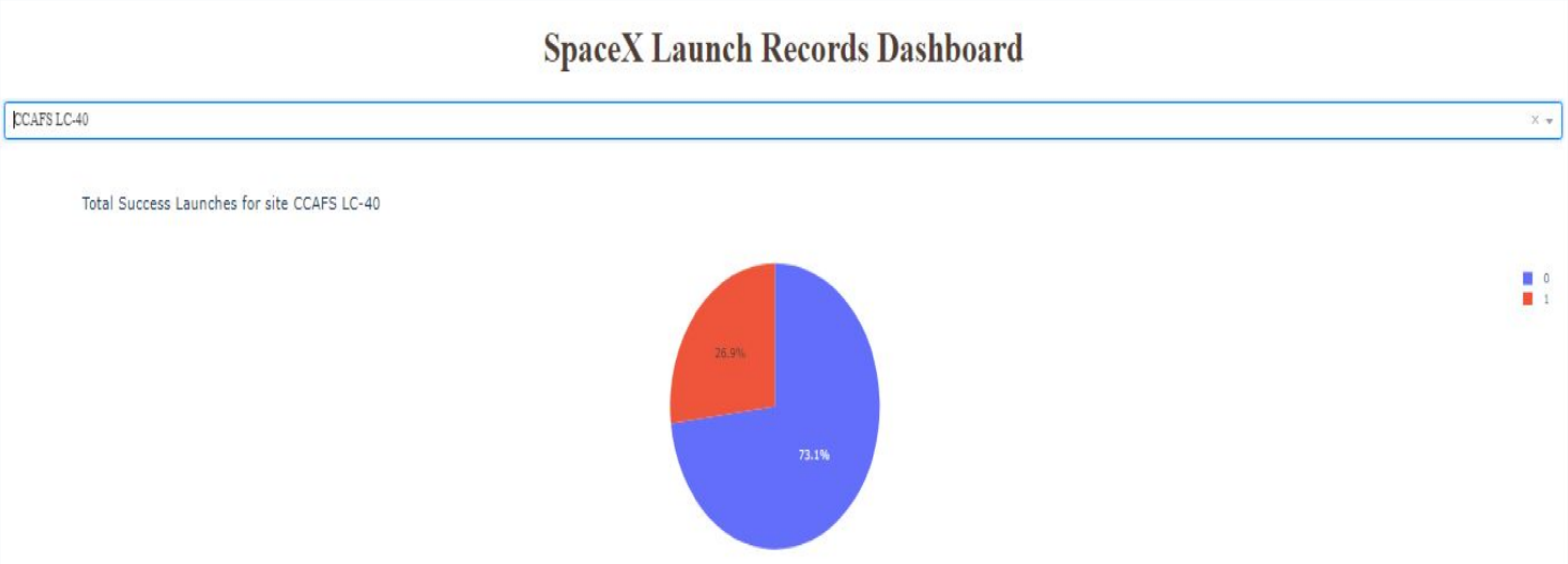
Build a Dashboard with Plotly Dash

Pie-Chart for launch success count for all sites



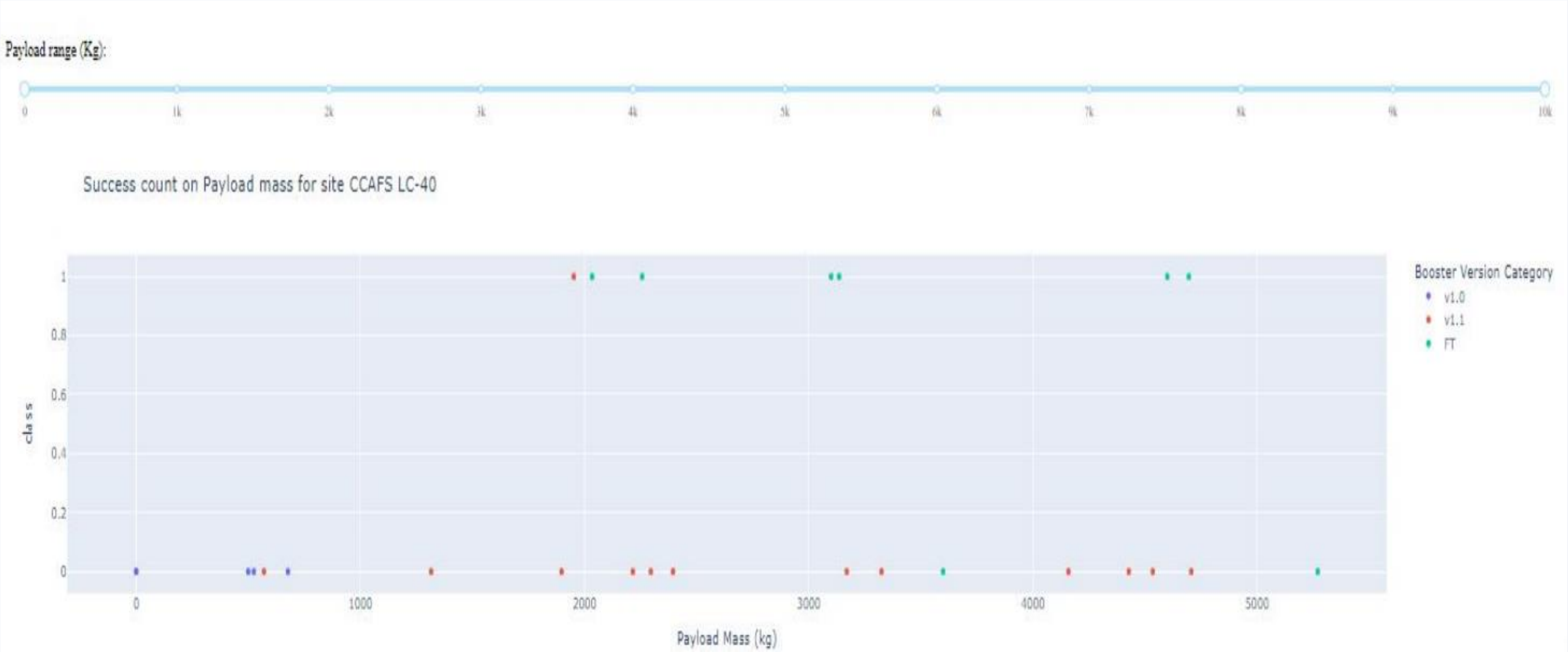
Launch site KSC LC-39A has the highest launch success rate at 42% followed by CCAFS LC-40 at 29%, VAFB SLC-4E at 17% and lastly launch site CCAFS SLC-40 with a success rate of 13%

Pie chart for the launch site with 2 nd highest launch success ratio



Launch site CCAFS LC-40 had the 2nd highest success ratio of 73% success against 27% failed launches

Payload vs. Launch Outcome scatter plot for all sites



For Launch site CCAFS LC-40 the booster version FT has the largest success rate from a payload mass of >2000kg

Section 5

Predictive Analysis (Classification)

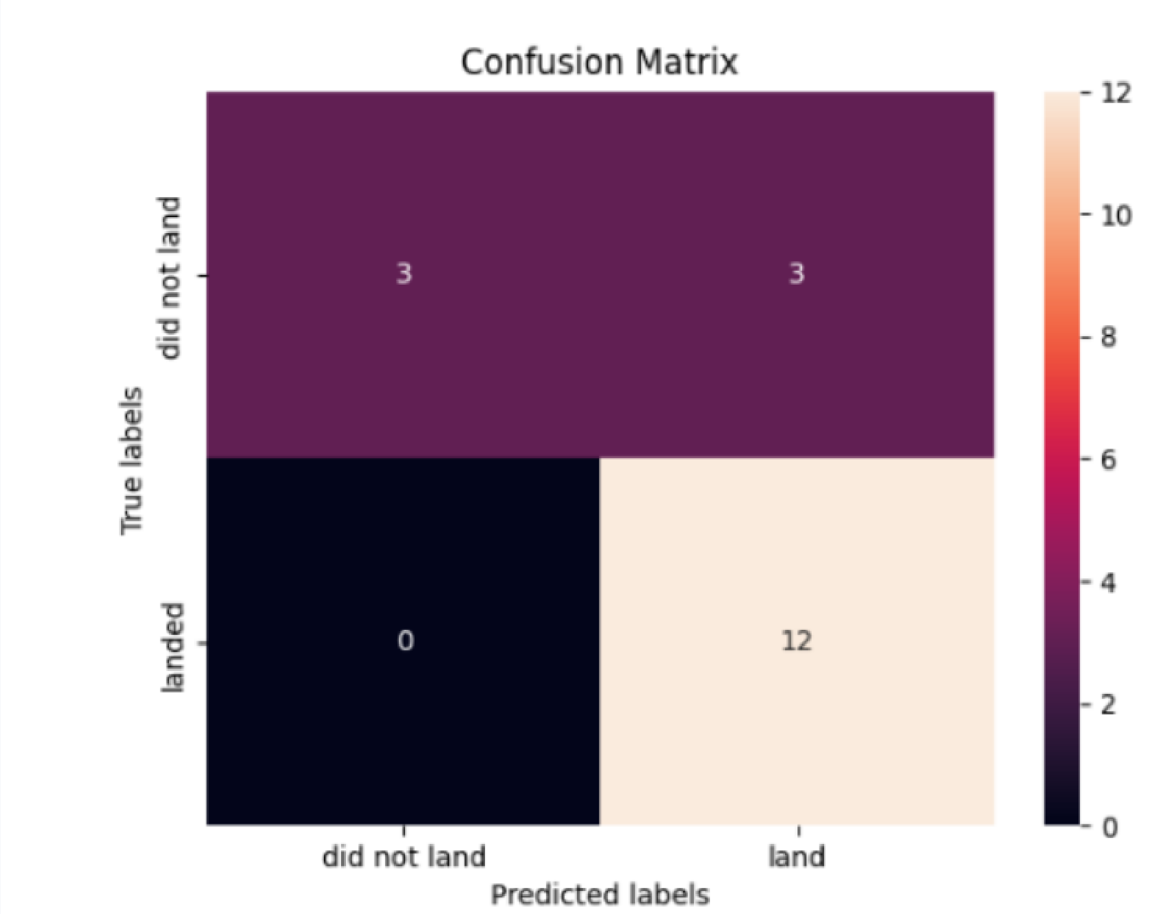
Classification Accuracy

Out[68]:

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

All the methods perform equally on the test data: i.e. They all have the same accuracy of 0.833333 on the test Data

Confusion Matrix



All the 4 classification models had the same confusion matrixes and were able equally distinguish between the different classes. The major problem is false positives for all the models

Conclusions

- LAUNCH SITES report different Success Rates ranging with CCAFS LC-40 with a 60% Success Rate to KSC LC-39A & VAFB SLC 4E having Success Rates of 77%
- The Success Rates increases with the Increase in No. of Flights for the 3 Launch Sites
- NO Rockets were launched with Heavy PayLoads (>10,000) from the VAFB-SLC Launch Site
- SO Orbit has the Lowest Success Rate at around 50%
- The Success Rate in LEO Orbit appears to be related to the No. of Flights
- There is no CLEAR relationship b/w No. of Flights and Success Rate for GTO Orbit
- Polar, LEO and ISS Orbits have higher Success Rates for Heavy PayLoads
- The Success Rates have been increasing consistently since 2013 up-to 2020

Appendix

Please find in the data used for whole analysis in the .rar file attached

Data Used.rar

Thank you!