IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Chukwu Amaechi James
01-12-2023

Outline

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

Executive Summary

- **Summary of methodologies**
  - Understanding The Business and The Problem
  - Choosing a Classification Model Analytic Approach
  - Collecting the Data via REST API and Web scraping
  - Data Wrangling (Exploratory Data Analysis and Determining Training Labels)
  - Completing The EDA with SQL
  - EDA with Visualization(Exploring and Preparing Data)
  - Interactive Visual Analytics with Folium
  - Building Dashboard Application with Plotly Dash
  - Predictive Analysis (Machine Learning Prediction)

Executive Summary

- **Summary of all results**
  - Results of EDA
  - Results of Interactive Maps and Dashboards
  - Results of Predictive Analysis (Machine Learning Prediction Results)

Introduction

- Project background and context

  - 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 we can determine if the first stage will land, we 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

  - Which features (variables) of the collected data (like Flight Number, Orbit, Launch Site, Payload Mass) has the most influence on the success of the first stage. What other features are influencing this stage?

  - What algorithm or machine learning model will work best for the determination of the success of the first stage?

  - Are their progress in the rate of successful landings with time(Yearly)?

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

- Via SpaceX REST API
- Via Web Scraping from Wikipedia

Perform data wrangling

- Filter the data to drop columns that are not needed
- Attend to missing values in the data frame
- Converting the input to a binary classifier via 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

- How to build, tune, evaluate classification models

Data Collection

How data sets were collected.

A combination of API queries from the SpaceX REST API and web scraping data from a table in SpaceX's Wikipedia entry were used in the data collection procedure.

In order to gather all the information necessary for a more thorough study, we had to employ both of these data collection techniques for the launches.

Data columns that was retrieved via the SpaceX REST API:

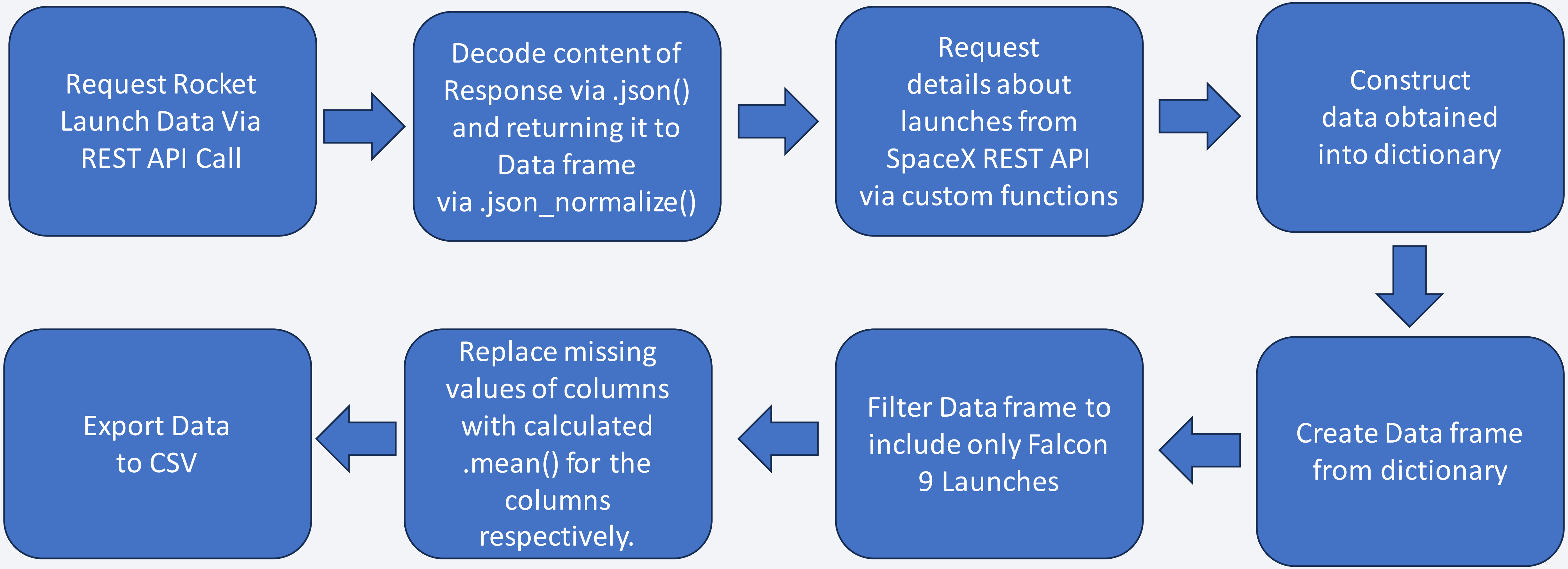
Flight Number, Date, Booster Version, Payload Mass, Orbit, Launch Site, Outcome, Flights, Grid Fins, Reused, Legs, Landing Pad, Block, Reused Count, Serial, Longitude and Latitude

Data columns that was retrieved via Web Scrapping Wikipedia:

Flight Number, Launch Site, Payload, Payload Mass, Orbit, Customer, Launch

Outcome, Version Booster, Booster Landing, Date and Time

Data Collection – SpaceX API



SpaceX REST API URL: Click Here for the link

Github URL for Data Collection via REST API

Data Collection - Scraping

Request Falcon 9 Launch Data from Wikipedia

Create Beautiful Soup object from HTML Response

Extract Column names from HTML table header

Collect Data by parsing HTML tables

Construct Data obtained into dictionary format

Create Data frame from the dictionary

Export Data to CSV

View URL for SpaceX Wikipedia here

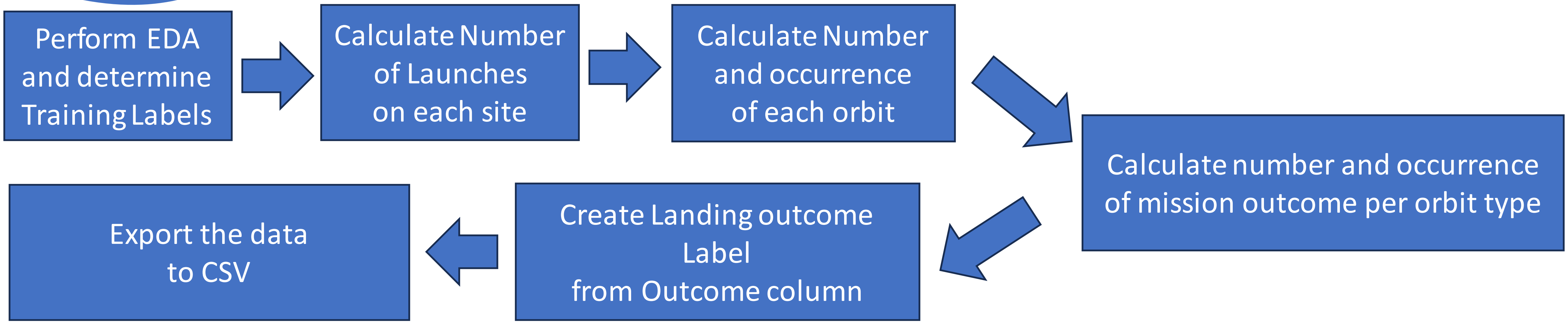
Github URL for Data Collection via Web Scrapping

Data Wrangling

There are multiple instances in the data set where the booster failed to land properly. There are instances where a landing attempt was made but was unsuccessful due to an accident.

- True Ocean, True RTLS, True ASDS means the mission has been successful.
- False Ocean, False RTLS, False ASDS means the mission was a failure.

Mostly, we translate those results into Training Labels, where "0" denotes an unsuccessful booster landing and "1" indicates a successful booster landing.

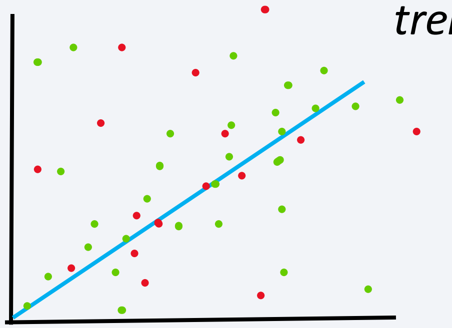


GitHub URL for Data Wrangling Code

EDA with Data Visualization

- Charts/Graphs plotted:
  - Scatter Plots:

- Flight Number vs. Payload Mass
- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Orbit vs. Flight Number
- Payload vs. Orbit Type
- Orbit vs. Payload Mass

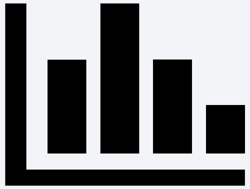


Scatter plots for viewing the relationship between features/variables. The relationship is called correlation

- Line Charts:
  - Success Rate vs. Year

Line Charts for visualizing data features/variables and their trends. It shows global behavior and can predict hidden insights.

- Bar Graphs:
  - Success Rate vs. Orbit



Bar graphs for visualizing the relationship between numeric and categoric features/variables.

[GitHub URL for EDA with Data Visualization Code]

EDA with SQL

➢ Used SQL Queries to:

- Display the names of the unique launch sites in the space mission
- Display five(5) records where launch sites begin with the string ‘CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster versions which have carried the maximum payload mass
- List the failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015
- Ranking the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20 in descending order

GitHub URL for EDA with SQL Code

Build an Interactive Map with Folium

- Used Markers for the Launch Sites:
  - Using the latitude and longitude of the NASA Johnson Space Center as a starting point, a marker was added featuring a circle, pop-up label, and text label.
  - In order to display each launch site's geographic location and proximity to the coast and equator, markers with circles, pop-up labels, and text labels have been added.
- Used colored Markers to differentiate the launch outcome for the Launch Sites respectively:
  - In order to determine which launch sites have comparatively high success rates, colored markers of successful (Green) and unsuccessful (Red) launches were added using Marker Cluster.
- Distances between a Launch Site to its proximities:
  - Colored lines have been added to indicate the distances between the launch site, KSC LC-39A, and its surroundings, such as the closest city, railway, highway, and coastline.

GitHub URL for Interactive Visual Analytics with Folium Code

Build a Dashboard with Plotly Dash

- Plots/Graphs and interactions added to the Dashboard:

  - Dropdown List: To enable selection of the different launch sites.

  - Pie Chart: To display the overall number of successful launches for all sites as well as the success vs. failure counts for a selected site.

  - Slider: To help select Payload mass range for a selected launch site.

  - Scatter Plot: To show the correlation between Payload Mass and Launch Success Rate for the different Booster Versions.

GitHub URL for Interactive Visual Analytics with Folium Code

Predictive Analysis (Classification)

- Imported Libraries and defined Auxiliary functions
- Loaded the dataframe
- Created a numpy array from the Class-column in the data
- Standardized the data in the Class-column
- Split the data into training and testing data
- Test the accuracy of all the models(Logistic Regression, SVM, Decision Tree classifier and K-nearest Neighbours)
- It turned out they all had same accuracy level hence any of them can be used for the Predictive Analysis

Import Libraries and Define Auxiliary Functions

```
In [50]: import piplite
         await piplite.install(['numpy'])
         await piplite.install(['pandas'])
         await piplite.install(['seaborn'])
```

Load the dataframe

Load the data

```
In [54]: data.head()
```

Out[54]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	Landing
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	

```
In [57]: Y=data['Class'].to_numpy()
         Y

Out[57]: array([0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1,
               1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
               1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,
               1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
               1, 1], dtype=int64)
```

Standardize the data in X then reassign it to the variable X using the transform provided below.

```
In [61]: # students get this

         transform = preprocessing.StandardScaler()
         X = transform.fit_transform(X)
         X

Out[61]: array([[-1.71291154e+00, -1.94814463e-16, -6.53912840e-01, ...,
                 -8.35531692e-01,  1.93309133e+00, -1.93309133e+00],
                [-1.67441914e+00, -1.19523159e+00, -6.53912840e-01, ...,
                 -8.35531692e-01,  1.93309133e+00, -1.93309133e+00],
                [-1.63592675e+00, -1.16267307e+00, -6.53912840e-01, ...,
                 -8.35531692e-01,  1.93309133e+00, -1.93309133e+00],
                ...,
                [ 1.63592675e+00,  1.99100483e+00,  3.49060516e+00, ...,
                  1.19684269e+00, -5.17306132e-01,  5.17306132e-01],
                [ 1.67441914e+00,  1.99100483e+00,  1.00389436e+00, ...,
                  1.19684269e+00, -5.17306132e-01,  5.17306132e-01],
                [ 1.71291154e+00, -5.19213966e-01, -6.53912840e-01, ...,
                 -8.35531692e-01, -5.17306132e-01,  5.17306132e-01]])
```

```
X_train, X_test, Y_train, Y_test

In [62]: X_train, X_test, Y_train, Y_test = train_test_split( X, Y, test_size=0.2, random_state=2)
```

we can see we only have 18 test samples.

```
In [63]: Y_test.shape
Out[63]: (18,)
```

Find the method performs best:

```
Analysis = pd.DataFrame({'Method' : ['Test Data Accuracy']})

knn_accuracy=knn_cv.score(X_test, Y_test)
Decision_tree_accuracy=tree_cv.score(X_test, Y_test)
SVM_accuracy=svm_cv.score(X_test, Y_test)
Logistic_Regression=logreg_cv.score(X_test, Y_test)

Analysis['Logistic_Reg'] = [Logistic_Regression]
Analysis['SVM'] = [SVM_accuracy]
Analysis['Decision Tree'] = [Decision_tree_accuracy]
Analysis['KNN'] = [knn_accuracy]

Analysis.transpose()
```

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

Finding

All the models have the same accuarncy and any of them can be applied to p

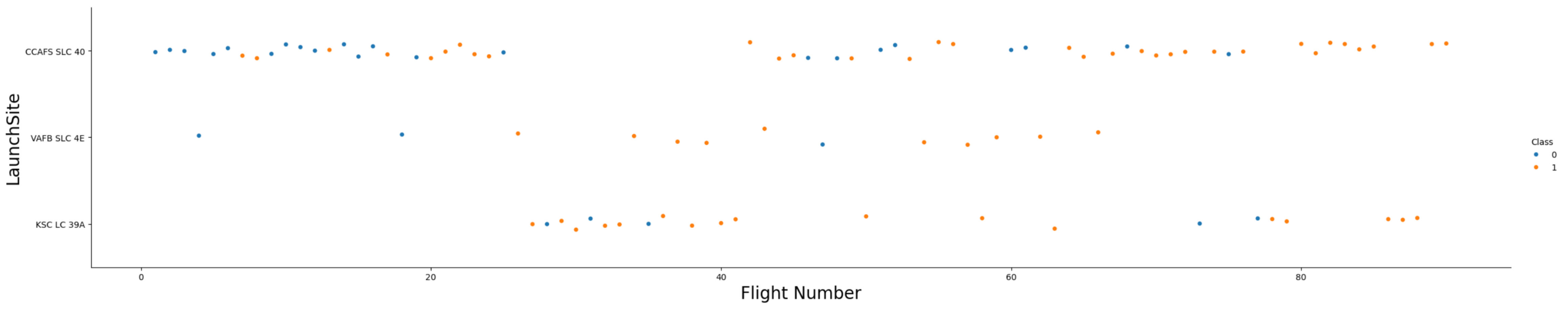
GitHub URL for Machine Learning Predictive Analysis Code

Results

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

Section 2

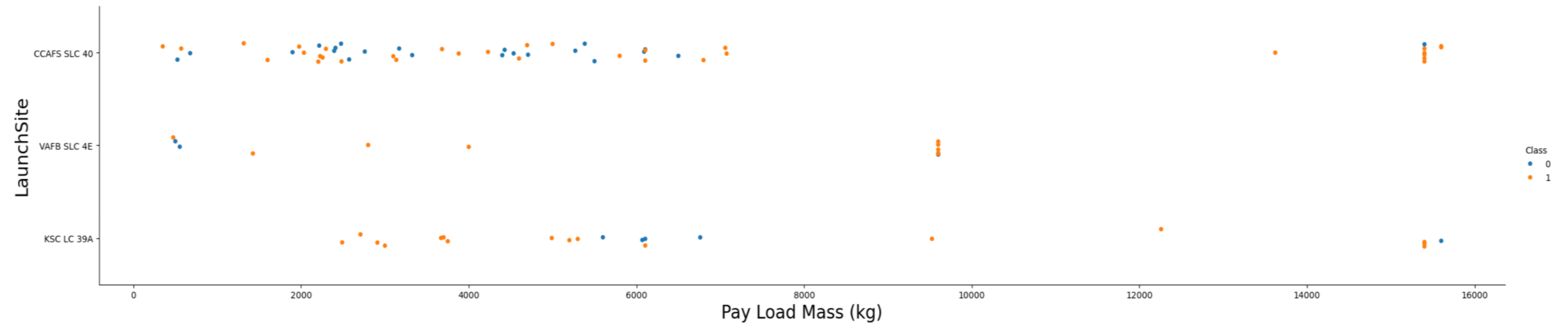
Insights drawn from EDA



Flight Number vs. Launch Site

- All of the most recent flights were successful, while none of the earlier ones were.
- Approximately 50% of all launches occur at the CCAFS SLC 40 launch site.
- The success rates are higher for KSC LC 39A and VAFB SLC 4E.
- One can presume that the likelihood of success increases with each new launch.

Payload vs. Launch Site

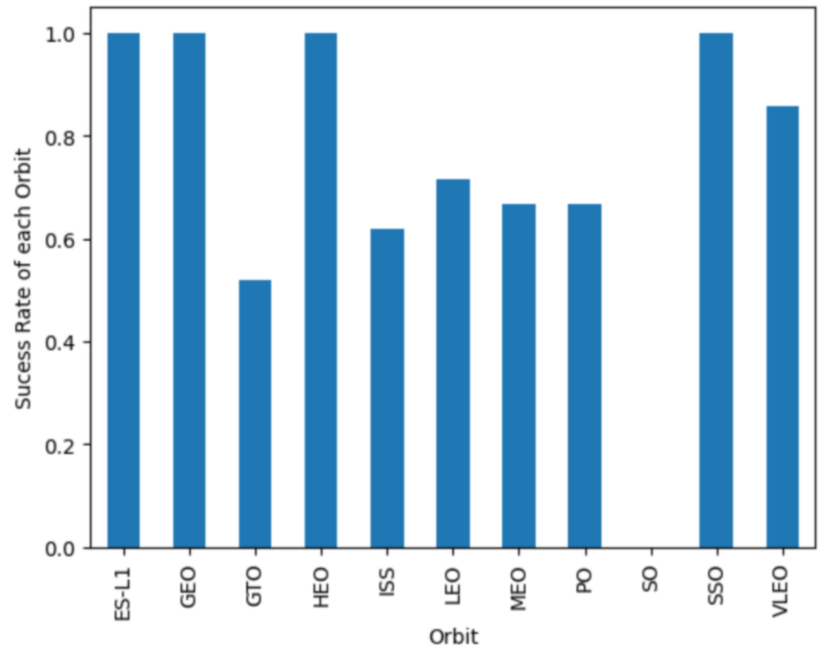


Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

- The majority of launches with payload masses exceeding 7000 kg were successful. The higher the payload mass, the higher the success rate for each launch site.
- Under 5500 kg, KSC LC 39A also has a 100% success rate for payload mass.

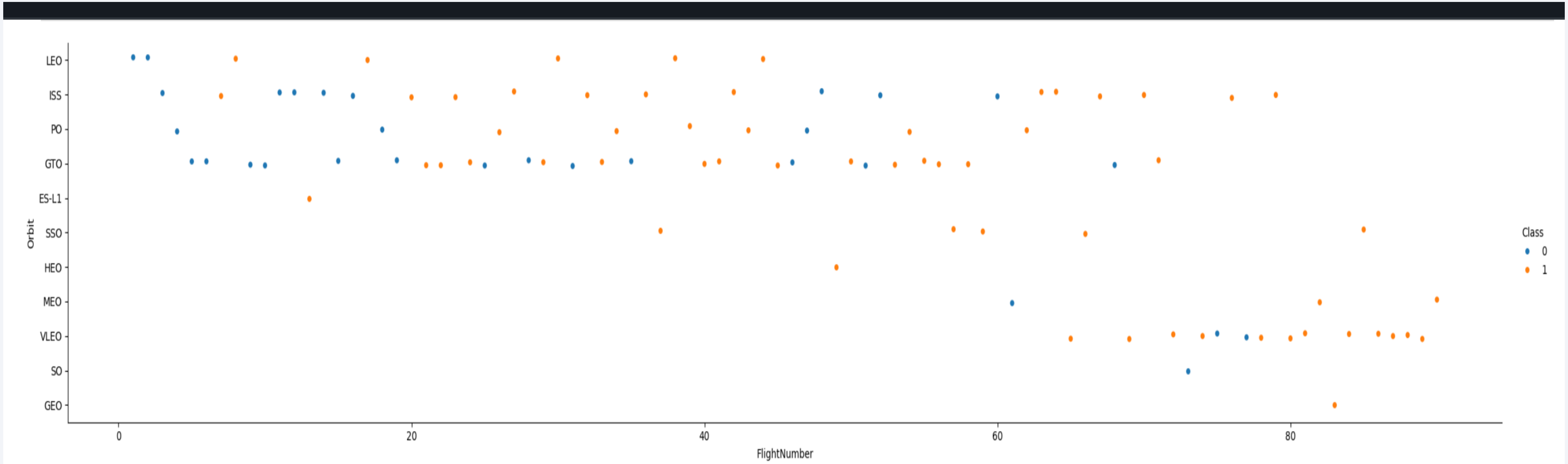
Success Rate vs. Orbit Type

- 100% success rate orbits:
- GEO, HEO, SSO, and ES-L1
- • Orbits that never succeed:
- - SO
- • Orbits with a 50% to 85% success rate:
- - PO, MEO, ISS, GTO, and LEO



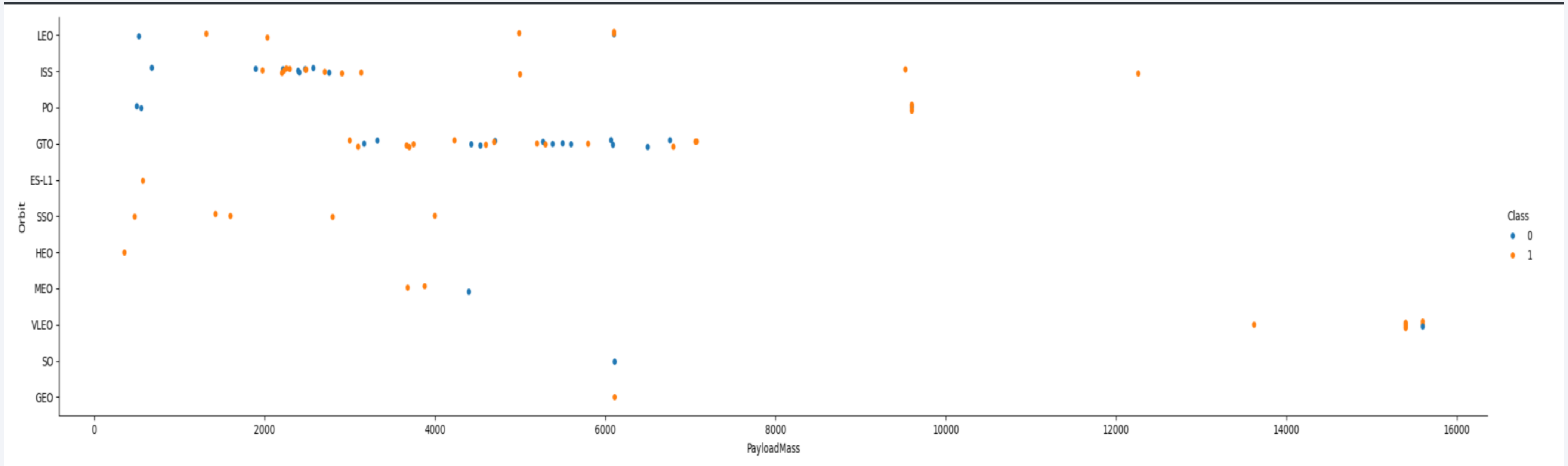
Analyze the ploted bar chart try to find which orbits have high sucess rate.

Flight Number vs. Orbit Type



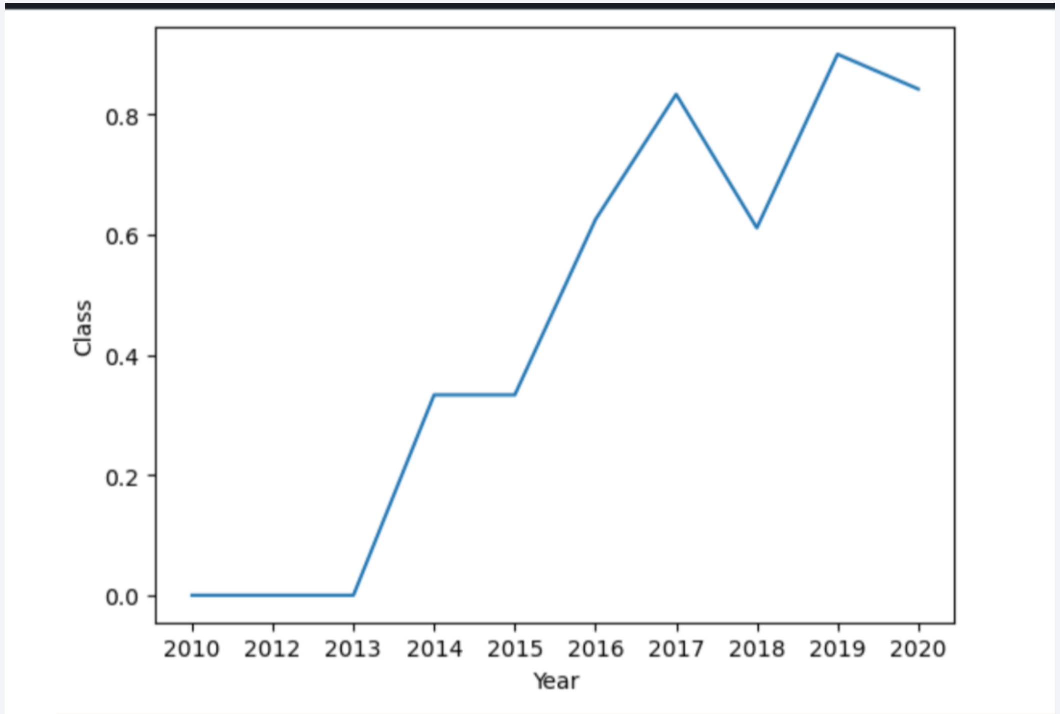
While in GTO orbit, there doesn't seem to be any correlation between the number of flights and success; however, in LEO orbit, it does appear to be related.

Payload vs. Orbit Type



In some orbits, the success rate of the launches can be significantly impacted by the payload weight. For instance, the success rate for the Low Earth orbit is increased with heavier payloads. Reduced payload weight for a GTO orbit has also been found to increase launch success.

Launch Success Yearly Trend



Since 2013, the success rate has been rising until 2020.

All Launch Site Names

```
In [9]: %sql select DISTINCT Launch_Site from SPACEXTABLE;

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

Out[9]:

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

Showing/displaying the names of the unique launch sites.
The application of "DISTINCT" in the query allows the removal of duplicate LAUNCH SITES in the column.

Launch Site Names Begin with 'CCA'

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Cus
6/4/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	S
12/8/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	(
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	(
10/8/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	
3/1/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	

<u>Display of five (5) records where launch sites begin with the string 'CCA'.</u>

Total Payload Mass

Task 3

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

19]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTABLE' WHERE Customer = 'NASA (CRS)';
```

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

19]:

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

Using DISTINCT in the query, displays all the unique values for the Launch_Site column from SPACEX table

Average Payload Mass by 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 LIK
```

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

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

The average payload mass carried by booster version F9 v1.1 was 2534.666 kg.

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

.5]:
```
%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing_Outcome" = "Success (ground pad)";
```

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

.5]:

MIN(DATE)
2015-12-22

First Successful Ground Landing Date is the 22nd of December, 2015.

Successful Drone Ship Landing with Payload between 4000 and 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

[16]:
```
%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTABLE WHERE "Landing_Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__
```

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

[16]:

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

The following drone ships laded successfully:
F9 FT B1022, F9 FT B1026, F9 FT B1021.2, and F9 FT B1031.2.
Their payload mass is larger than 4000 but less than 6000.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTABLE 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

Total number of successful mission outcomes is 100
Total number of successful mission failure is 1

Boosters Carried Maximum Payload

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS_
```

```
 * 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

These 12 booster types carried the largest payload mass.

2015 Launch Records

```
%sql SELECT substr(Date,0,5), substr(Date, 6,2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Outcome",
```

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

substr(Date,0,5)	substr(Date, 6,2)	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)

- List of the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.
- The results show two unsuccessful landing attempts at the same launch site (CCAFS LC-40), one in January and one in April.

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

```
%sql SELECT * FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE '%Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2017') ORDER BY
```

```
 * 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	4:45:00	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)
17/12/2019	0:10:00	F9 B5 B1056.3	CCAFS [illegible]	JCSat-18 / Kacific 1, [illegible]	6956	GTO	Sky Perfect JSAT, [illegible]	Success	Success

d (additional servers needed) Python | I... Mem: 447.68 / 6144.00 ... Mode: E... Ln 1, Col ... English (United Sta... jupyter-labs-eda-sql-coursera_sqllite.ip... 1

Near record 10:58 PM 12/6/2023

- Ranking for the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order:
- 33 Rank Landing Results From June 4, 2010, to March 20, 2017
- From 2010-06-04 to 2017-03-20, there were thirty-one launches.

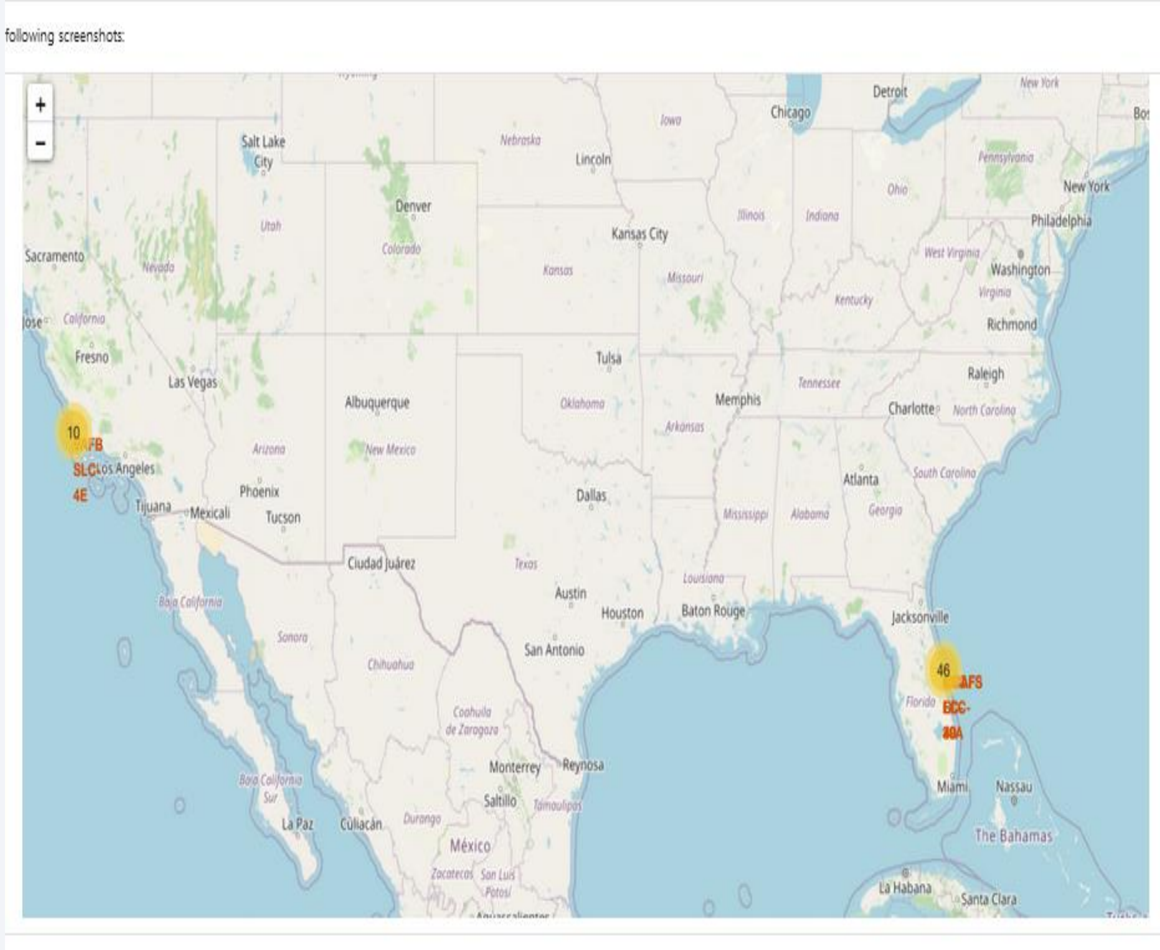
Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

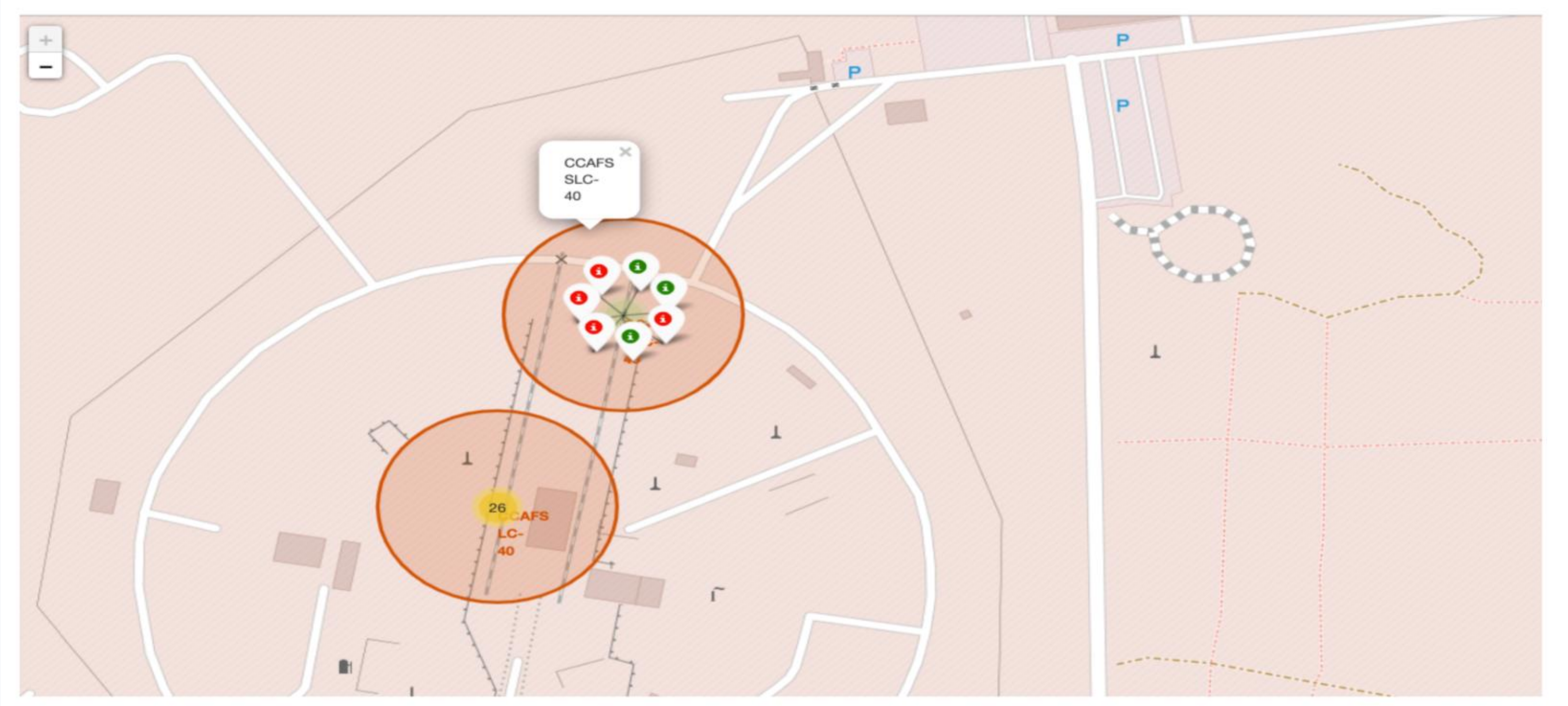
- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

Stamen Terrain Folium Map Display of all the Launch Sites



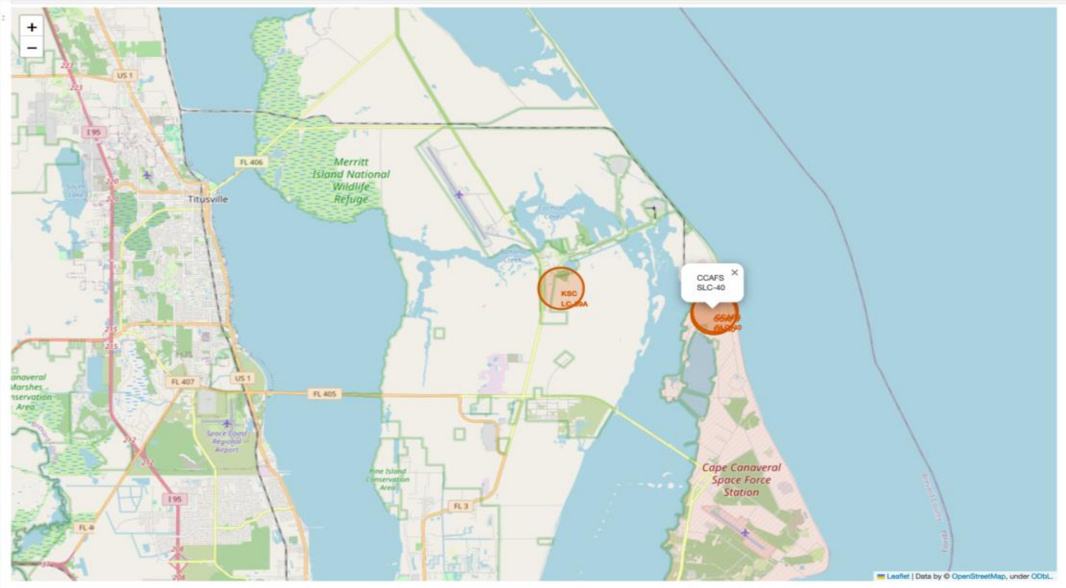
- **Launch sites are located close to the coast most likely due to needs for combustion and atmosphere.**
- **All launch sites are in proximity to the Equator line, considering that anything close to the planet's equator is already traveling at 1670 km/h. A ship launched from the equator launches into space at the same speed as it did before it orbits the planet. Inertia is the cause of this. The spacecraft will be able to maintain its orbital velocity at this speed.**

Display of Success Rates of Launch Sites with Color-Labeled Markers in Marker Clusters



Green Marker indicates successful launch of Class=1
Red Marker indicates unsuccessful launch of Class=0

Launch Sites Proximities to Railways, High ways, Coastlines and Cities



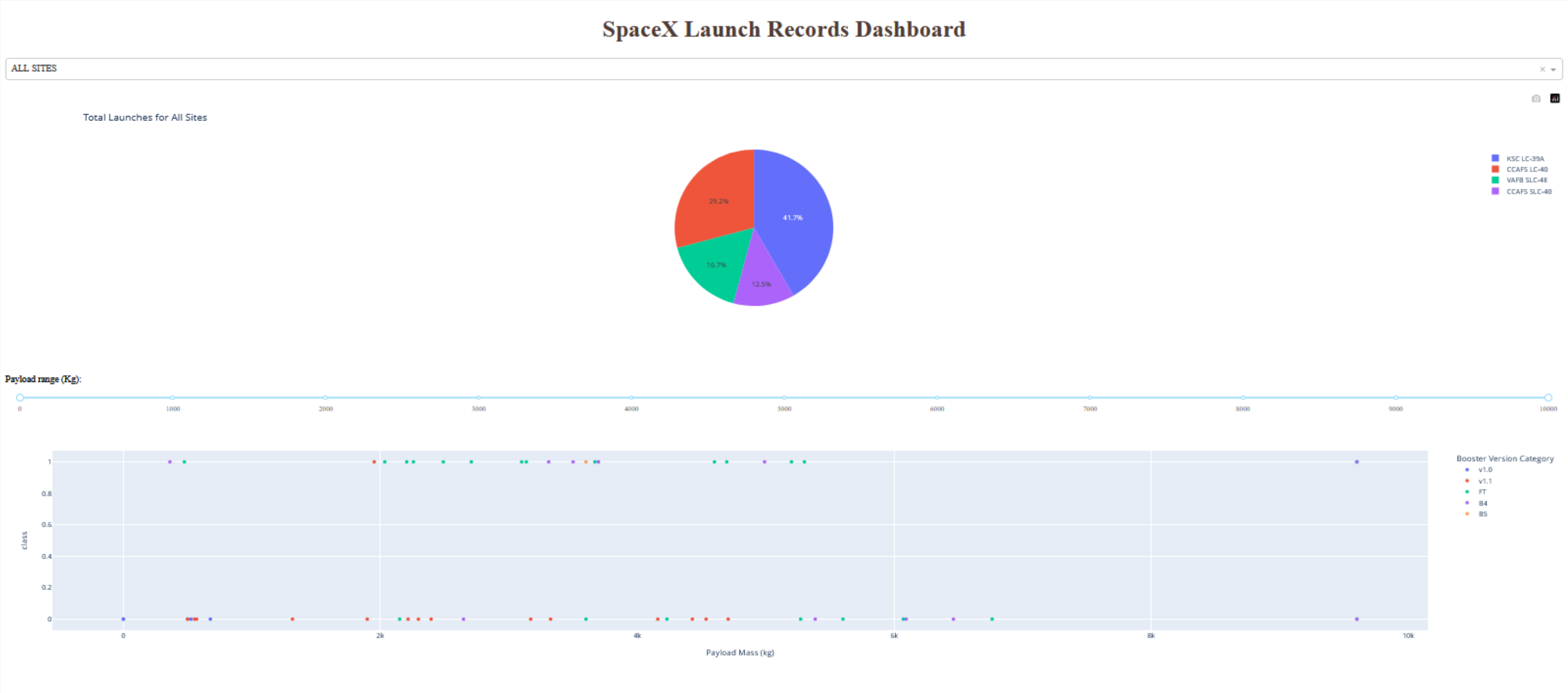
My Findings

- As stated earlier, launch sites are in close proximity to the equator so as to minimize fuel consumption by using Earth's ~ 30km/sec eastward spin to help spaceships get into orbit. it can take optimum advantage of the Earth's substantial rotational speed. Sitting on the launch pad near the equator, it is already moving at a speed of over 1650 km per hour relative to Earth's center.
- Launch sites are in close proximity to coastlines and they fly over the ocean during launch. This has a safety advantage for the crew to choose to abort launch and attempt water landing in case of failure. This also minimizes the danger of the risk of explosion debris falling on people and properties.
- Launch sites are in close proximity to highways, which allows people and properties to be easily transported.
- Launch sites are in close proximity to railways, which allows transportation of heavy cargoes.
- Launch sites are not in close proximity to cities, which minimizes the danger of explosion debrises falling on population densed areas.

Section 4

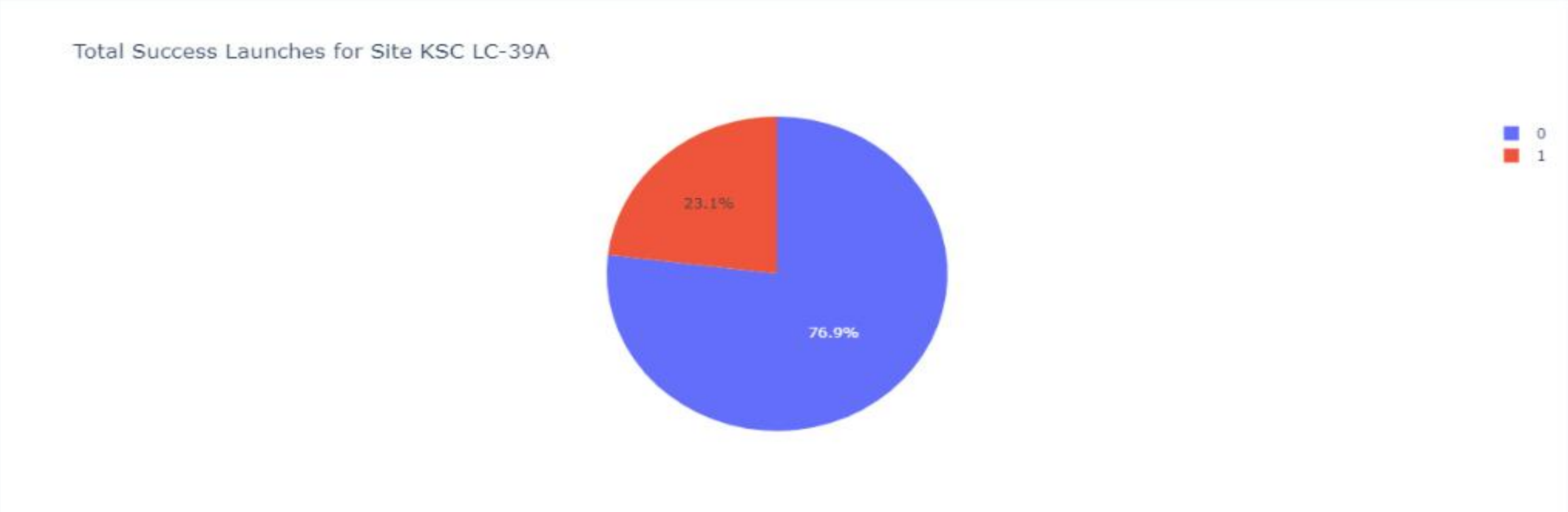
Build a Dashboard with Plotly Dash

Dashboard Display of Total Launches for all Sites



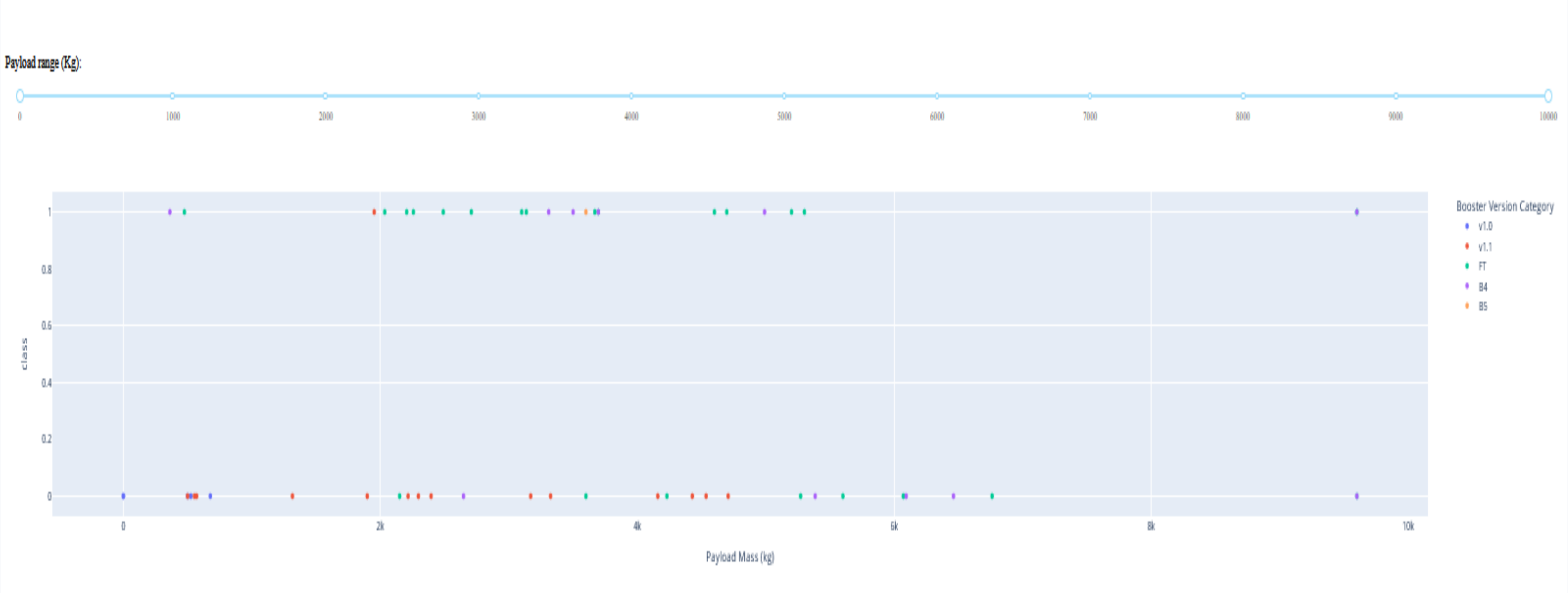
- CCAFS LC-40 has the lowest successful launches
- KSC LC-39A has the highest successful launches and good logistics.

Launch Site with Highest Launch Success Ratio



KSC LC-39A had the highest launch success rate of about 76.9% with aq total of 10 successful landings and 3failed landings recorded.

Payload Mass vs Launch Outcome for all Launch Sites



From the charts payloads between 2000 and 5500 kg have the highest success rate
Among the payload ranges, booster version FT has the highest success rate.

Section 5

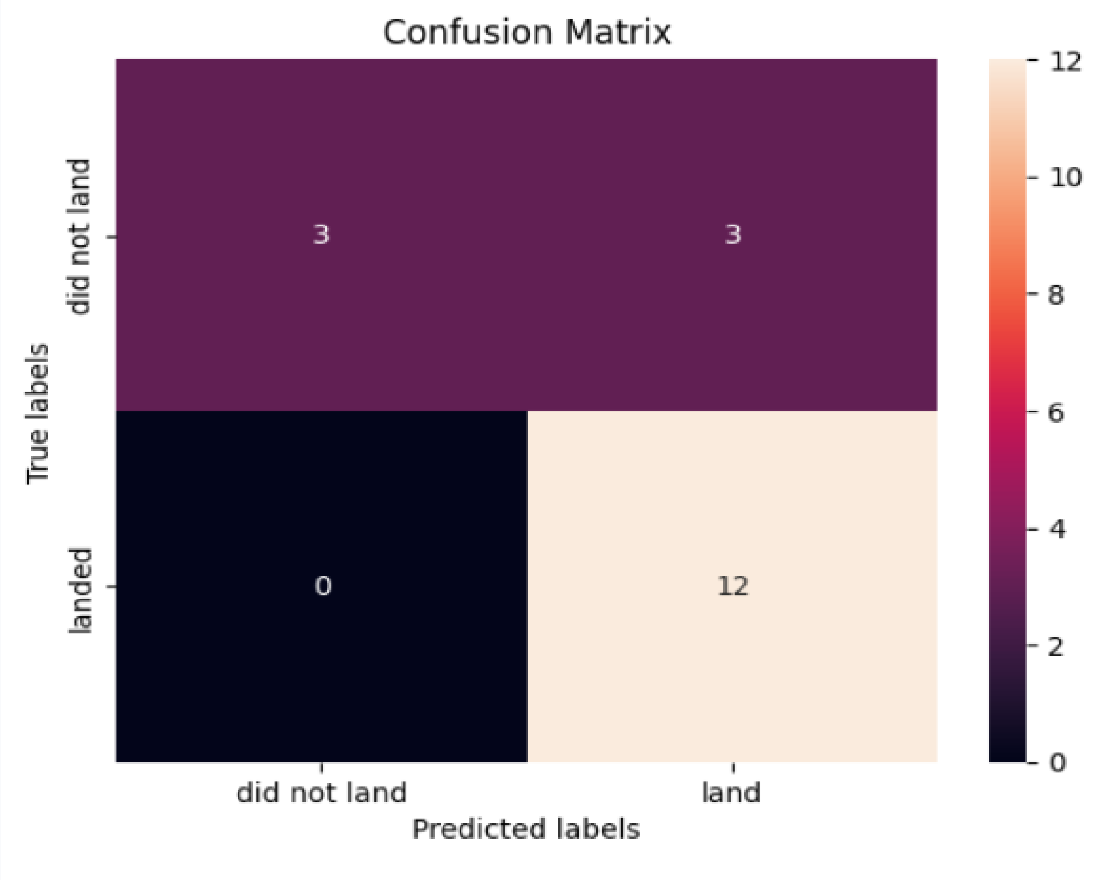
Predictive Analysis (Classification)

Classification Accuracy

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

All the models have the same accuracy and any of them can be applied

Confusion Matrix



The confusion matrix is the same for all methods.

Conclusions

- The accuracy results for all the machine learning models are the same, hence we cannot conclude which method (Logistics Regression, Support Vector Machine, Decision tree or KNN) performs the best prediction.
- Orbits ES-L1, GEO, HEO and SSO had a complete success rate of 100%.
- Launches that had lesser payload mass showed better results than launches that had higher payload mass.
- The success rate of all launches in all launch site increased over the years.
- Launch Sites are in proximity to the Equator line and the coast.
- Amongst all the Launch sites, KSC LC-39A had the highest success rate.
- The application of more data sources helped improve the efficiency and accuracy of the Machine Learning Algorithms.

Appendix

- https://api.spacexdata.com/v4/launchpads/
- https://api.spacexdata.com/v4/payloads/
- https://api.spacexdata.com/v4/cores/
- https://api.spacexdata.com/v4/launches/past
- https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
- https://github.com/magickalmix/Machine-Learning-with-Python/blob/main/Completing%20SpaceX%20eda%20with%20sqllite.ipynb
- https://github.com/magickalmix/Machine-Learning-with-Python/blob/main/Machine%20Learning%20with%20Python%20Reg-Simple-Linear-Regression-Co2.jupyterlite.ipynb
- https://github.com/magickalmix/Machine-Learning-with-Python/blob/main/Space%20X%20Falcon%209%20First%20Stage%20Landing%20Prediction_webscraping.ipynb
- https://github.com/magickalmix/Machine-Learning-with-Python/blob/main/SpaceX%20Falcon%209%20first%20stage-data-collection-api.ipynb
- https://github.com/magickalmix/Machine-Learning-with-Python/blob/main/SpaceX%20Falcon%209-Data%20wrangling.ipynb
- https://github.com/magickalmix/Machine-Learning-with-Python/blob/main/SpaceX%20Launch%20Records%20Dashboard
- https://github.com/magickalmix/Machine-Learning-with-Python/blob/main/SpaceX-Data%20wrangling.ipynb
- https://github.com/magickalmix/Machine-Learning-with-Python/blob/main/SpaceX-eda-sql-coursera_sqllite%20(2).ipynb
- https://github.com/magickalmix/Machine-Learning-with-Python/blob/main/SpaceX-webscraping.ipynb
- https://github.com/magickalmix/Machine-Learning-with-Python/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite(1).ipynb
- https://github.com/magickalmix/Machine-Learning-with-Python/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite(fixed%20codes).ipynb
- https://github.com/magickalmix/Machine-Learning-with-Python/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb
- "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv"

Thank you!