SPACE Y PROJECT

IBM CAPSTONE PROJECT

By: Chuluun-Erdene BATSAIKHAN

Executive Summary

- Introduction
- data collection and data wrangling methodology
- EDA and interactive visual analytics methodology
- Completed the required predictive analysis methodology
- EDA with visualization results slides
- EDA with SQL results
- interactive map with Folium results
- Plotly Dash dashboard results slides
- predictive analysis (classification)
- Conclusion

Introduction

The project aims to predict the success of Falcon 9 rocket's first stage landing. The cost-effective approach of SpaceX reusing the first stage makes it important to determine if it will land successfully.

The project aims to answer questions such as the likelihood of successful landing, factors influencing the outcome, and optimal launch sites for achieving successful landings.

It aims to provide valuable insights for companies bidding against SpaceX for rocket launches.

Data collection

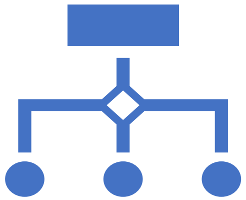
THE PURPOSE OF THIS PROJECT WAS TO EXTRACT LAUNCH DATA FROM THE SPACEX API AND NORMALIZE IT INTO A FLAT .CSV FILE.

TO ACHIEVE THIS, A SERIES OF HELPER FUNCTIONS WERE DEFINED TO EXTRACT INFORMATION USING IDENTIFICATION NUMBERS IN THE LAUNCH DATA

THE DATASET INCLUDED INFORMATION ABOUT THE ROCKET, LAUNCHPAD, PAYLOAD, AND OTHER CORE INFORMATION SUCH AS LANDING OUTCOME AND OTHER LAUNCH AND LANDING SPECIFICATIONS.

THE DATA WAS PROVIDED IN THE FORM OF A .JSON FILE.

Data collection



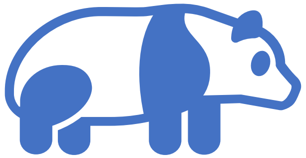
API Extraction Process

To extract data from the API, the first step was to request a response from the SpaceX API URL.

This response was then converted to a .JSON file and normalized.

Custom API functions were created to specify the data lists to retrieve.

A dictionary was also created for the data fields.



Data Processing

Using the dictionary, a pandas dataframe was created and the data was matched to the dictionary field names.

The data was then filtered to only include Falcon 9 launches.

Finally, the data was exported to a flat .csv file.

Data collection – Web Scrapping

To extract the Falcon 9 launch records from the Wikipedia page, a request was made to the page URL.

The HTML content of the page was then scraped using the BeautifulSoup Python package to create a BeautifulSoup object.

The code iterated through the HTML content to find the Falcon 9 launch records table.

A dictionary was created to define the data fields for the launch records.

The HTML table was parsed into a list of records and then converted into a Pandas dataframe.

The data was then matched to the dictionary field names in the dataframe.

Data wrangling

In this step, determined number of landing outcomes for Falcon 9 launches

- Assigned the outcomes to variable "landing_outcomes"
  - Created a list based on "Outcome" data, where bad outcomes set to zero and others set to one
    - Assigned resulting list to variable "landing_class"
    - "landing_class" represents classification variable indicating successful landing (value of one) or unsuccessful landing (value of zero)

EDA and interactive visual analytics methodology

The aim of this step was to conduct an exploratory data analysis on six variables, including Flight Number, Payload Mass, Launch Site, Orbit, Class, and Year, to investigate their relationships.

To visualize these relationships, several charts were created, such as scatter charts for

Flight Number vs. Payload Mass,

Flight Number vs. Launch Site,

Payload vs. Launch Site,

Orbit vs. Flight Number,

Payload vs. Orbit Type,

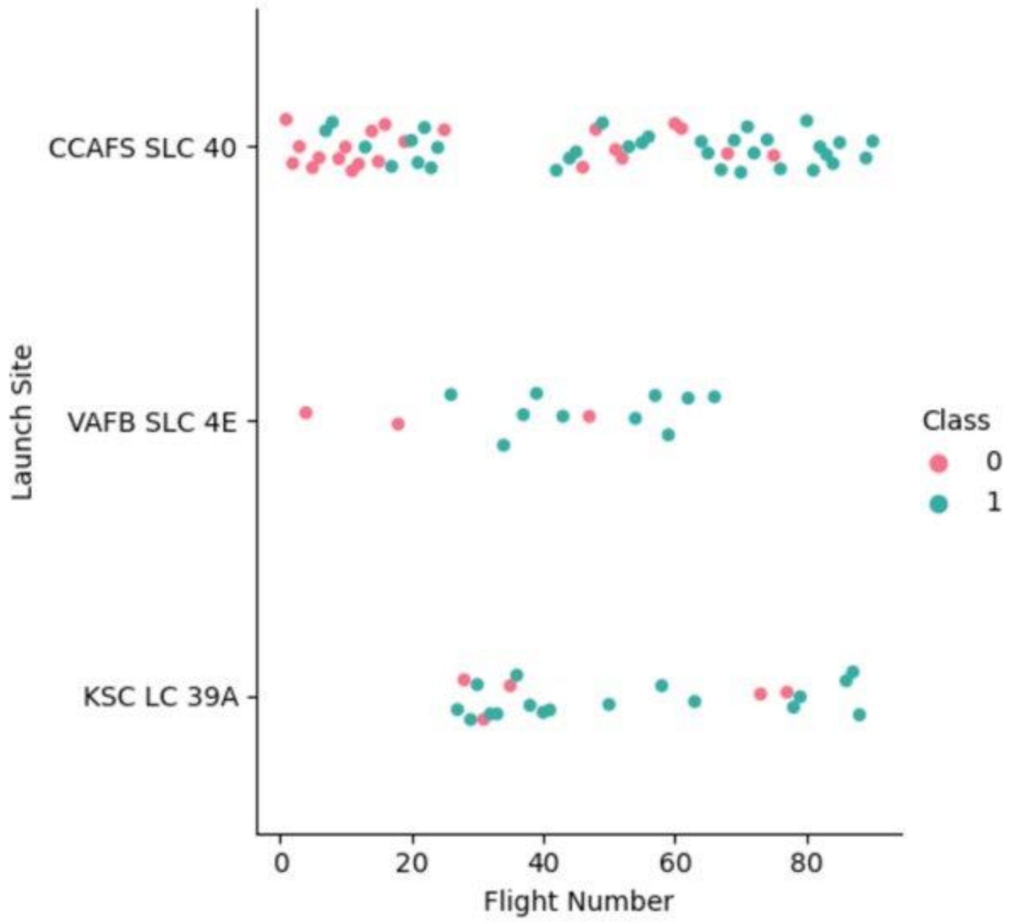
and Orbit vs. Payload Mass.

Additionally, bar charts were created to compare the mean of Orbit, and line charts were plotted to visualize the success rate vs. year.

Flight Number vs. Launch Site

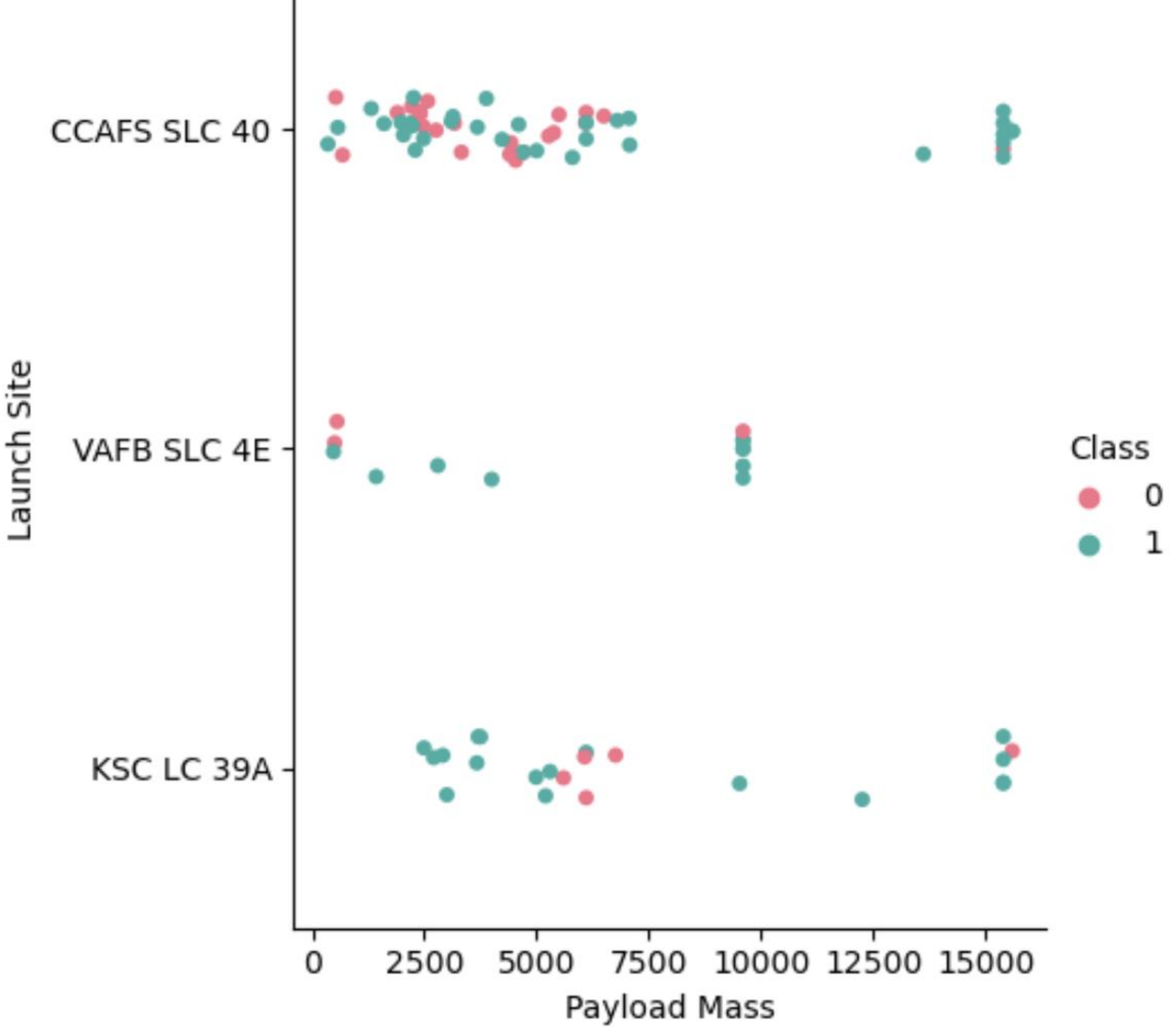
- Overall, the number of successful launches increased as the number of flight numbers increased.

- CCAFS LC-40 Launch site had the most flight number in both successful and unsuccessful launches
  - As the number of flight number increased its success also increased



Note: Successful launches shown as green

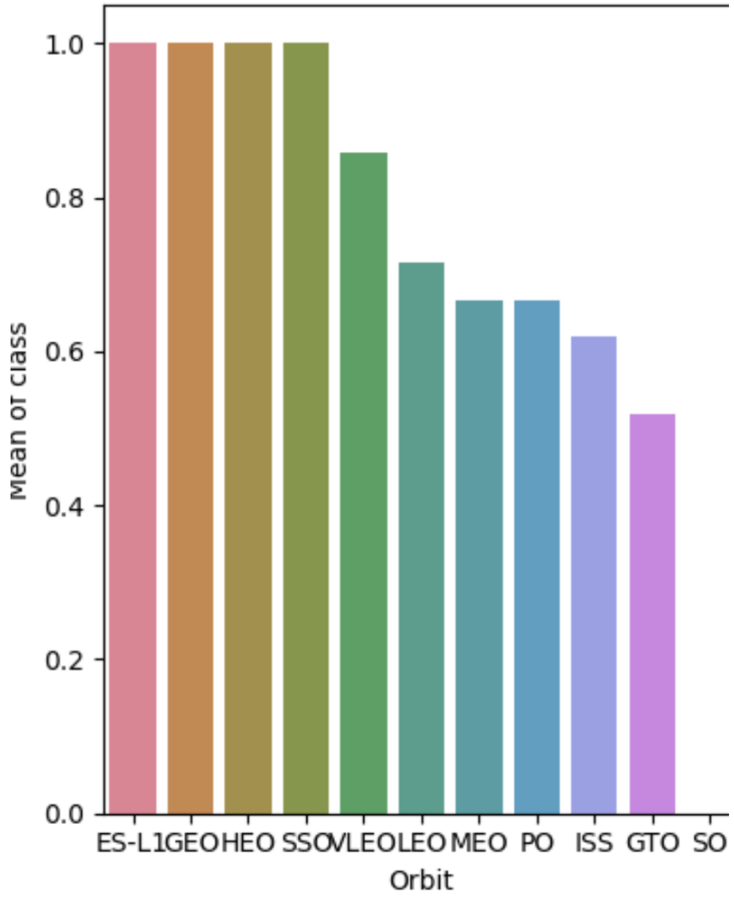
Payload Mass vs Launch Site



Note: Successful launches shown as green

- Success of the launches could not be determined by the payload mass.
- CCAFS LC-40 launch site mostly launched mid-lower payload mass flights.
  - It also successfully launched several higher payload mass flights without failure.

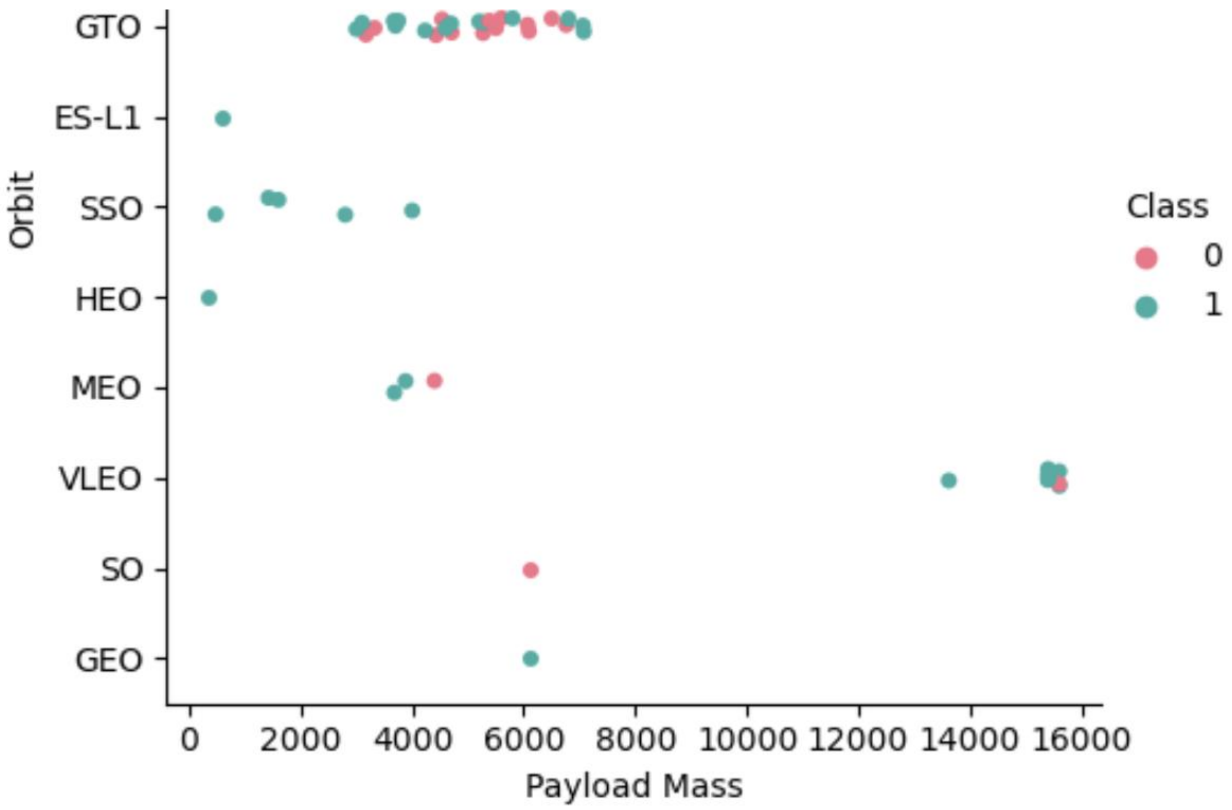
Flight number and Orbit type



Note: Successful launches shown as green

- Overall, as the number of flight increased the successful launch also improved.
- GTO, ISS, VLEO, PO orbits had the most flight number
    - Of them the mean of class and orbit graph displayed VLEO orbit had the most successful launches.

Payload and Orbit type

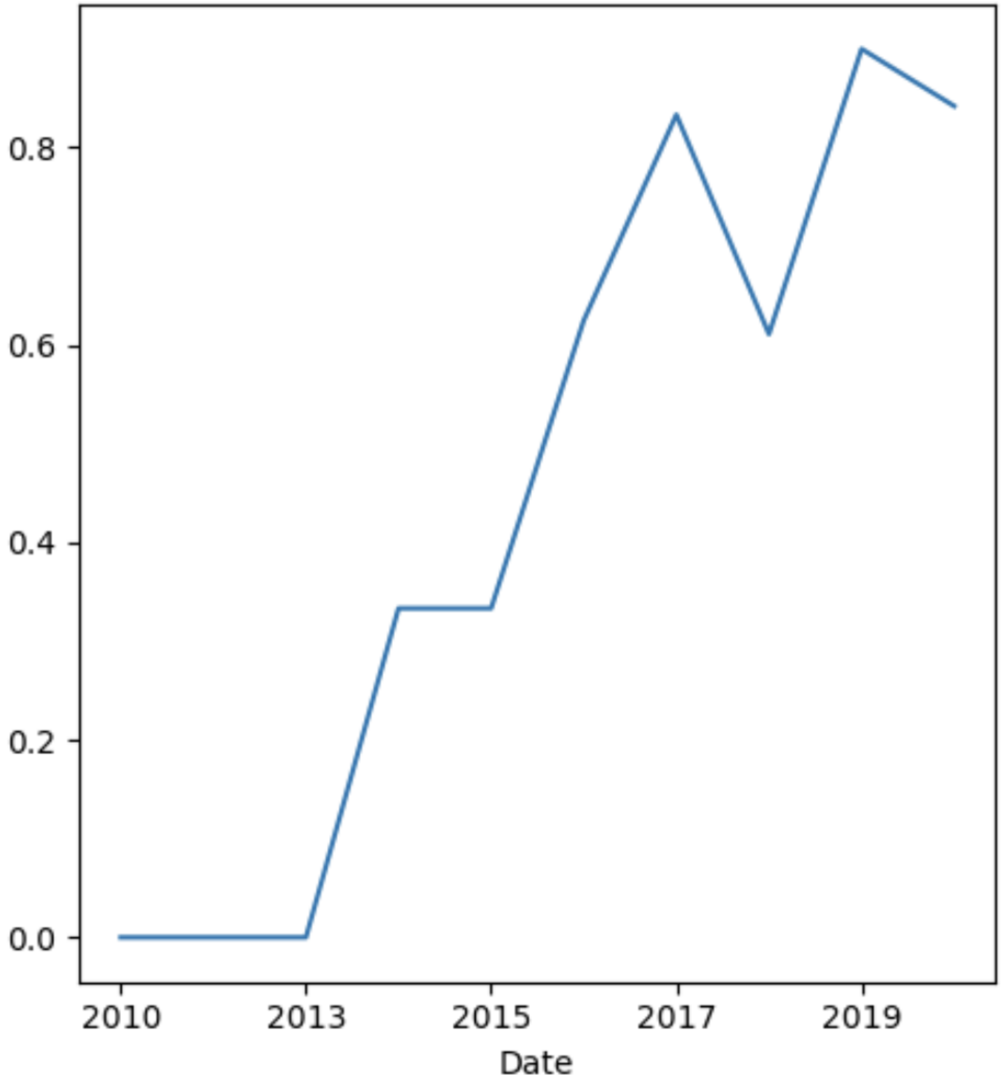


Note: Successful launches shown as green

- The successful launches could not be determined by the correlation between orbit type and payload mass.

Success rates of launches over the years

- In general, success rates for launches increased trend over the years of 2010 and 2020, but shown slight decline between 2017 and 2019.



EDA with SQL results

This step is to gain a better understanding of the dataset by loading it into an IBM DB2 database and querying it using SQL magic in Python.

Various queries were performed, including displaying

- unique launch sites,
- total payload mass carried by NASA,
- average payload mass of a booster version,
- and listing the date of successful drone ship landings.

The project also involved ranking the count of successful landing outcomes in a specific time frame and listing the records based on month names, successful landing outcomes, booster versions, and launch site for the year 2015.

EDA with SQL results

Query results for unique launch sites, and launch sites begin with the string 'CCA'.

```
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

```
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	-
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	(p
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	(p
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

EDA with SQL results

Total payload mass carried by boosters launched by NASA (CRS)

```
sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_payload FROM SPACEXTBL WHERE Customer = "NASA (CRS)"
```

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

Total_payload
45596

Average payload mass carried by booster version F9 v1.1

```
sql SELECT AVG(PAYLOAD_MASS__KG_) as average_payload FROM SPACEXTBL WHERE Booster_Version = "F9 v1.1"
```

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

average_payload
2928.4

EDA with SQL results

The first successful landing outcome in ground pad

```
sql SELECT MIN(Date) FROM SPACEXTBL WHERE "Landing _Outcome" LIKE "%success%"
```

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

MIN(Date)
01-05-2017

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 FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND "Landin
```

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

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

EDA with SQL results

The total number of successful and failure mission outcomes

```
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

EDA with SQL results

The names of the booster_versions which have carried the maximum payload mass.

```
sql SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_)
```

```
* 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

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

```
%sql SELECT substr("Date", 4, 2) AS MONTH, "Booster_Version", "Landing _Outcome", "Launch_Site" \
FROM SPACEXTBL WHERE "Landing _Outcome" = "Failure (drone ship)" AND substr("Date",7,4)="2015"
```

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

MONTH	Booster_Version	Landing _Outcome	Launch_Site
01	F9 v1.1 B1012	Failure (drone ship)	CCAFS LC-40
04	F9 v1.1 B1015	Failure (drone ship)	CCAFS LC-40

EDA with SQL results

The count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017

```
%sql SELECT "Landing _Outcome", COUNT("Landing _Outcome") \
FROM SPACEXTBL WHERE "Date" >= "04-06-2010" and "Date" <= "20-03-2017" and "Landing _Outcome" LIKE "%success
GROUP BY "Landing _Outcome"\
ORDER BY "Landing _Outcome" DESC
```

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

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

Predictive analysis methodology

Data preprocessing step.

- Used the function get_dummies and features dataframe to apply OneHotEncoder to the column Orbits, LaunchSite, LandingPad, and Serial. Assigned the value to the variable features_one_hot.

Data Preparation:

- The independent variable 'Class' was split from the dataset, and the data was standardized using StandardScaler() to ensure uniformity and comparability of the features.

Data Splitting:

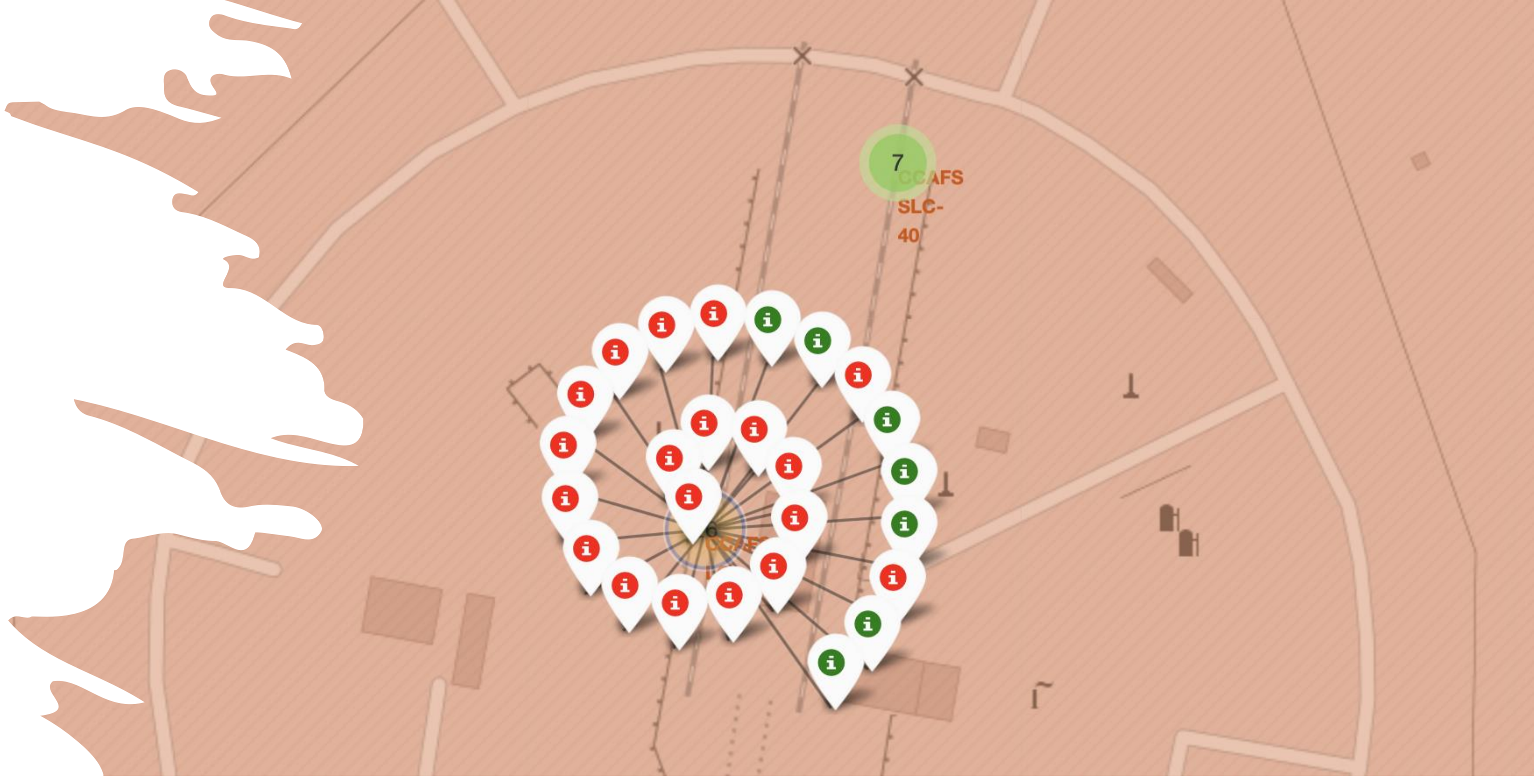
- The data was divided into training and test sets to train and evaluate the machine learning models.

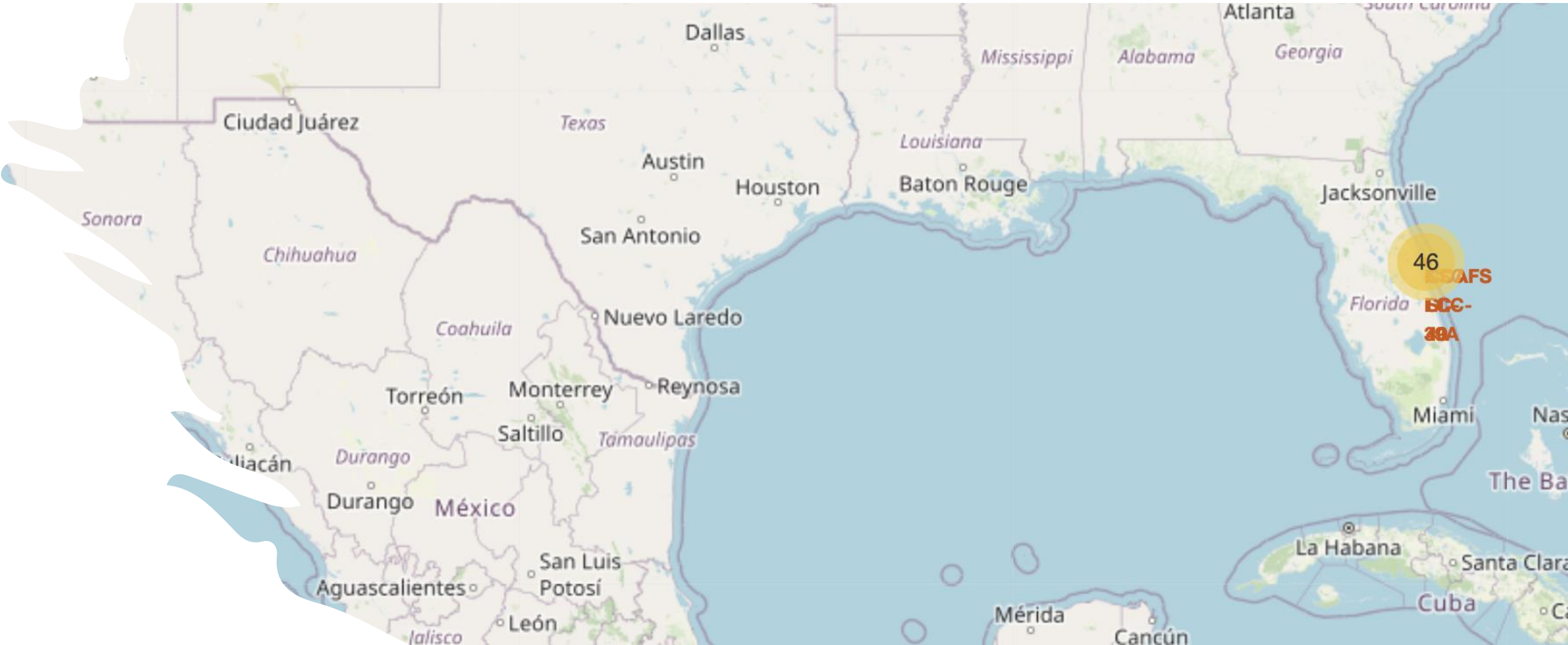
Model Training:

- Four different models, including logistic regression, SVM, decision tree, and KNN, were fitted on the training data to learn patterns and relationships between features and the target variable.

Interactive map with Folium

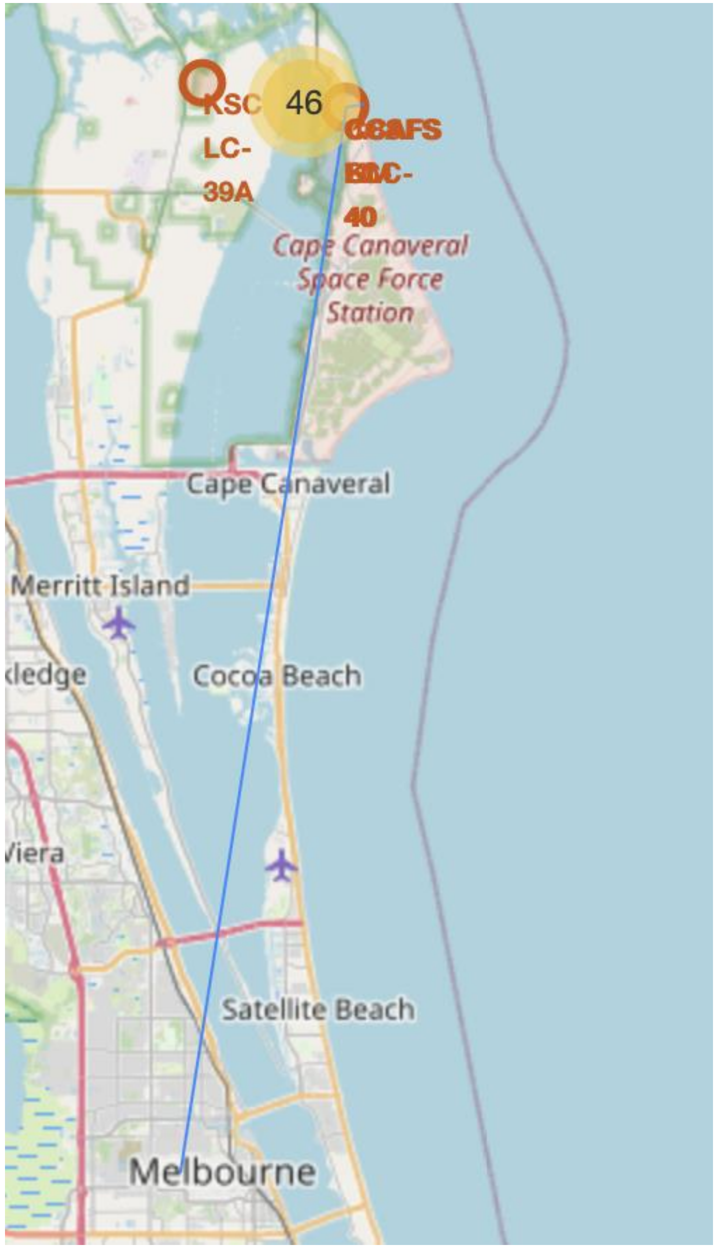
- Folium maps were used to visualize launch data on an interactive map. Latitude and longitude coordinates were used to add labeled circle markers at each launch site.
- MarkerCluster() was used to indicate successful and unsuccessful outcomes with green and red markers, respectively.





Interactive map with Folium

The distance to key locations on the map, such as the nearest city and the coastline, was calculated and marked with a line to visualize it.

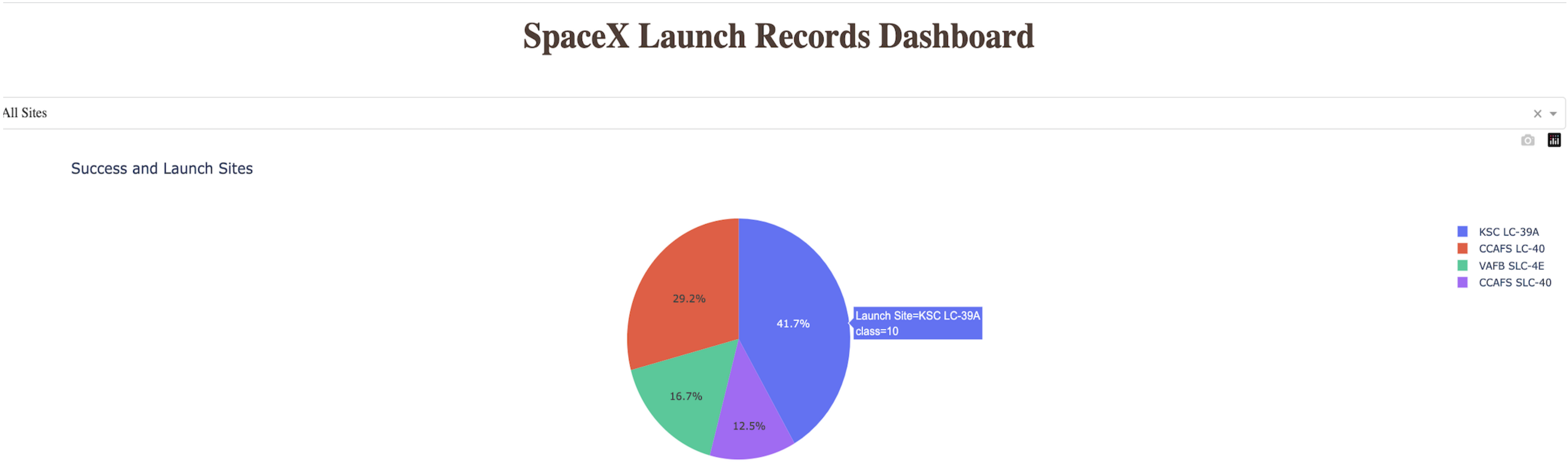


Plotly Dash dashboard

The dashboard consists of a pie chart and a scatter plot.

- **An interactive pie chart** is used to visualize the success rate of the launch sites, showing the distribution of successful landings across all launch sites or the distribution of successful landings for a specific individual launch site.
- **A scatter plot** is used to visualize how success varies depending on the payload mass and booster version category.

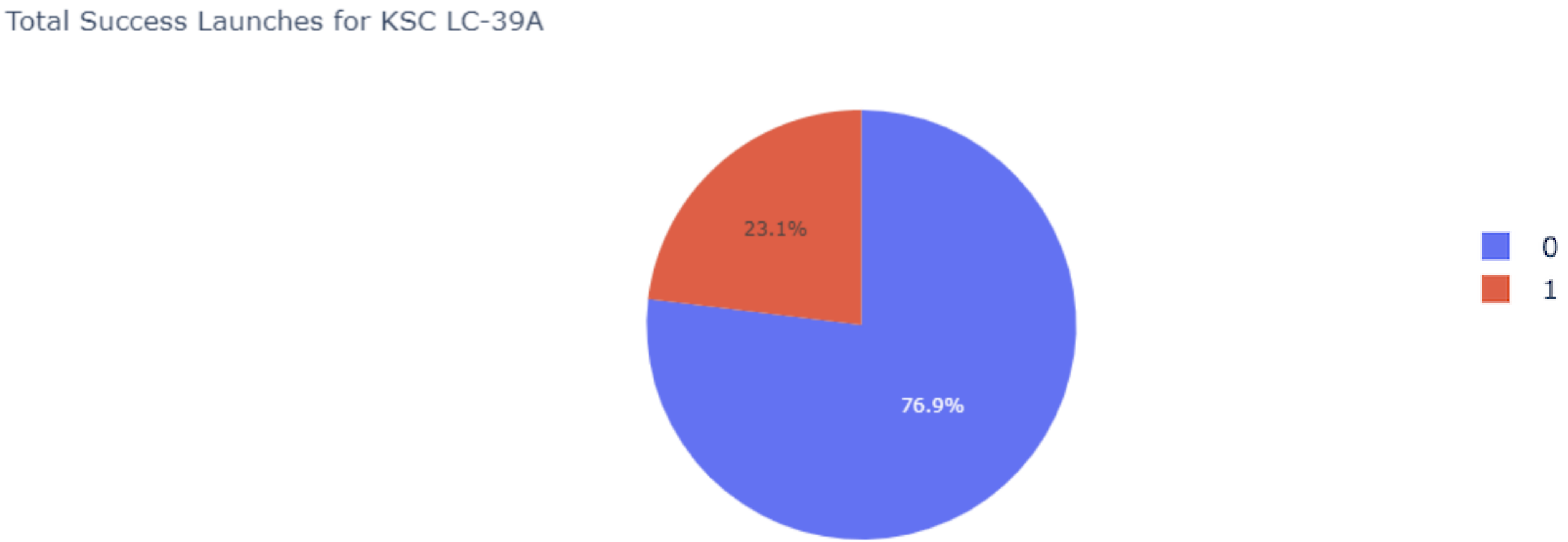
Plotly Dash dashboard



Overall, launch site KSC LC-39A had the most successful launches. The second most successful is launch site CCAFS LC-40. While the combined success of other two launch sites only accounted around 30 percent.

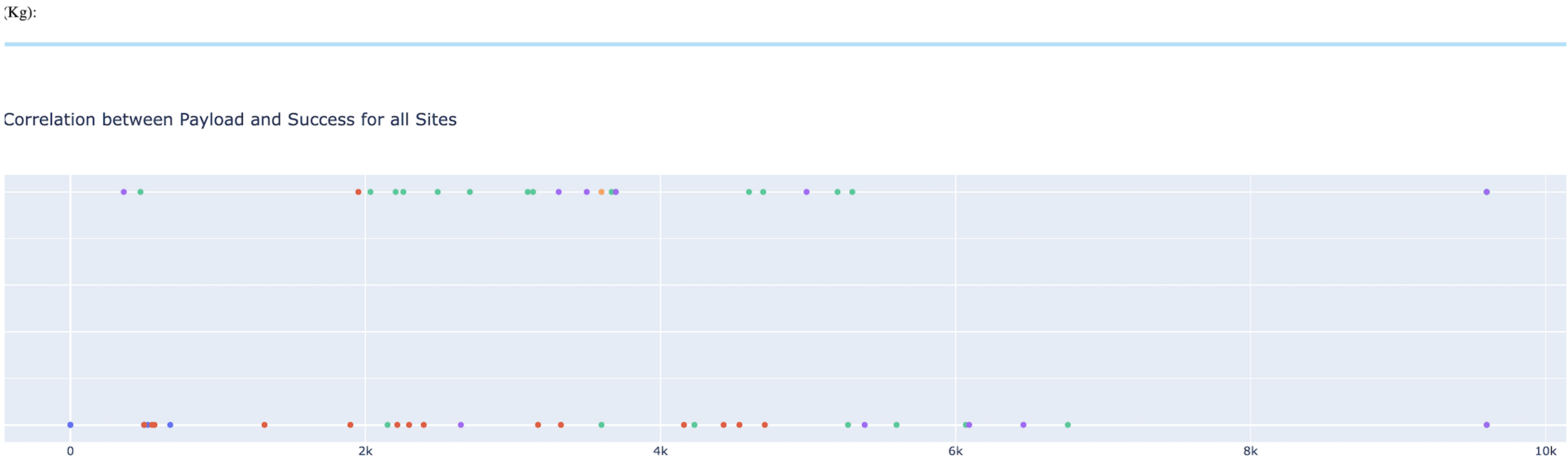
Plotly Dash dashboard

- The success rate for launch KSC LC-39A is 76,9%.



Plotly Dash dashboard

This graph shows correlation between payload and success by booster version categories in all launch sites. According to it, FT booster version had the most successful launches and few unsuccessful ones compared to others by different ranges of payload.



Predictive analysis (classification) results

The predictive analysis was done by splitting the independent variable 'Class' from the dataset and using StandardScaler() to fit and transform the data.

The data was then split into training and test sets. Several models such as

- logistic regression,
- SVM,
- decision tree,
- and KNN were fit on the training data,
- and GridSearchCV was used to tune optimal hyperparameters for each model.

The accuracy of each model was determined using the test dataset, and a confusion matrix was plotted for each model.

❿ Logistic regression

❿ SVM

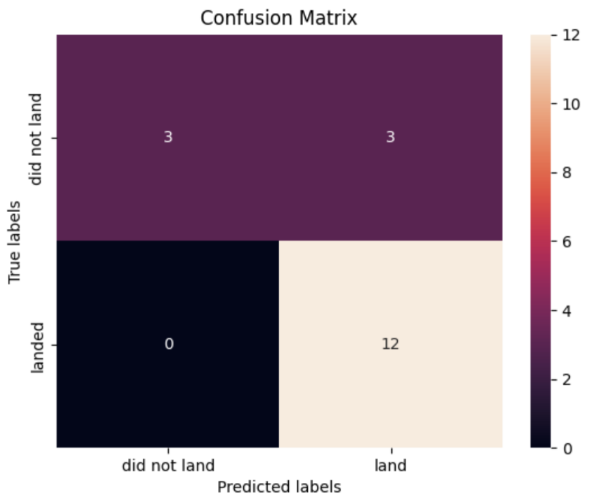
- In general, all the models produced same accuracy of ~83,33% with the same confusion matrix.
- This may be due to the limited data.
- The confusion matrix of these models have the tendency to give false positives or over predict the success rate.

```
svm_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```

We can plot the confusion matrix

```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

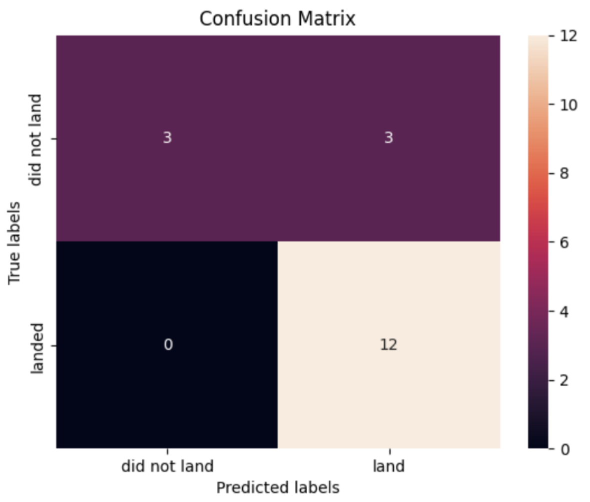



```
logreg_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```

Lets look at the confusion matrix:

```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



➓ Decision tree

➓ KNN

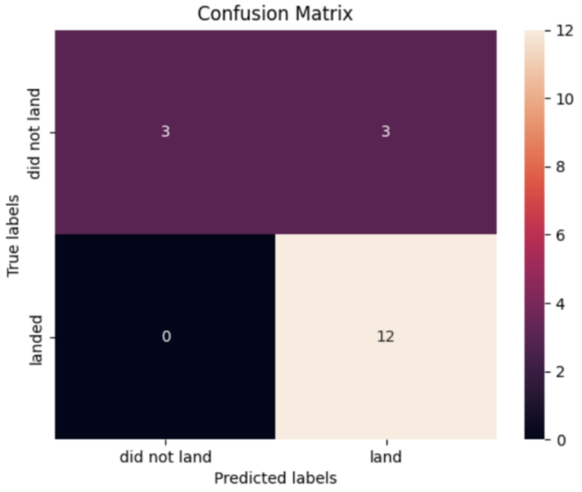
- In general, all the models produced same accuracy of ~83,33% with the same confusion matrix.
- This may be due to the limited data.
- The confusion matrix of these models have the tendency to give false positives or over predict the success rate.

```
knn_cv.score(X_test, Y_test)
```

0.8333333333333334

We can plot the confusion matrix

```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

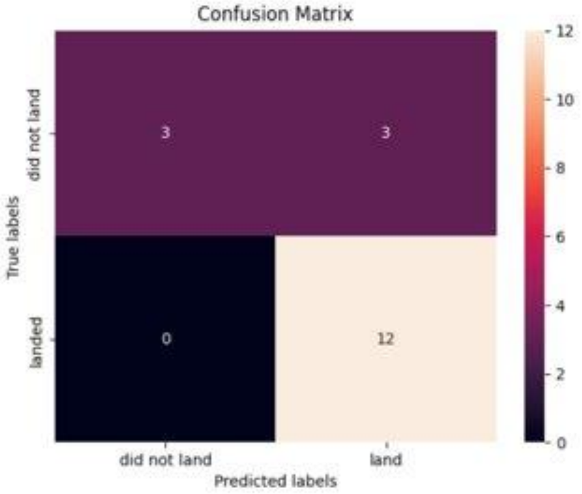



```
tree_cv.score(X_test, Y_test)
```

0.8333333333333334

We can plot the confusion matrix

```
yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusion

The goal was to develop a machine learning model to predict the success of stage 1 landings for rocket launches. Four machine learning models were developed, which showed an accuracy of approximately 83.33% in predicting successful landings using test data.

However, the models tend to over-predict successful landings and could benefit from more data for improved accuracy.

Additionally, the analysis revealed that the success rate of stage 1 landings has improved over time, with certain orbit types showing higher success rates.

Overall, further improvements and insights can be gained through continued analysis and data refinement.

Thank you for your attention