Winning Space Race with Data Science

Julio Nakama

February 15th, 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis with SQL and plots
  - Interactive Visual Analytics with Folium
  - Predictive analysis with Machine Learning techniques
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive analytic results
  - Predictive analysis results

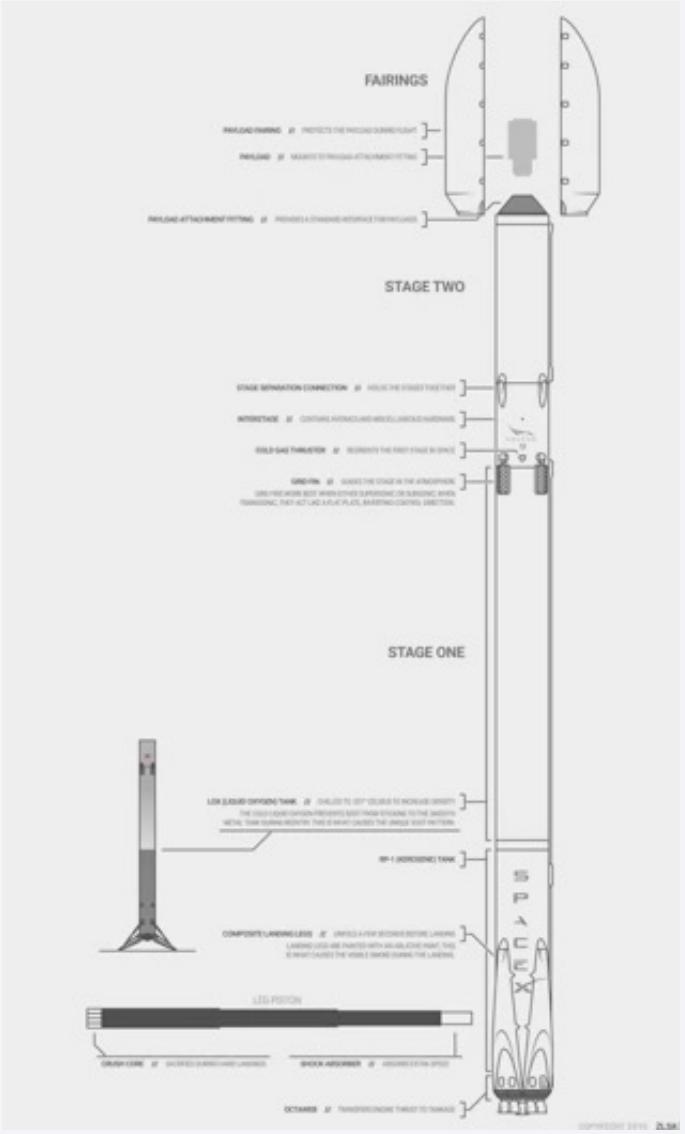
Introduction

- Project background and context

  Space X's rocket launches cost 62 million dollars, whereas other providers can cost up to165 million dollars. This difference is due to the reuse of the rocket's first stage by Space X.

  The study's objective is to use Space X launches' information to help Space Y to determine a launch's cost.

- Problems you want to find answers
  - What factors influence the success of the first stage landing?
  - Is it possible to build an accurate model to predict the landing success?



Section 1

Methodology

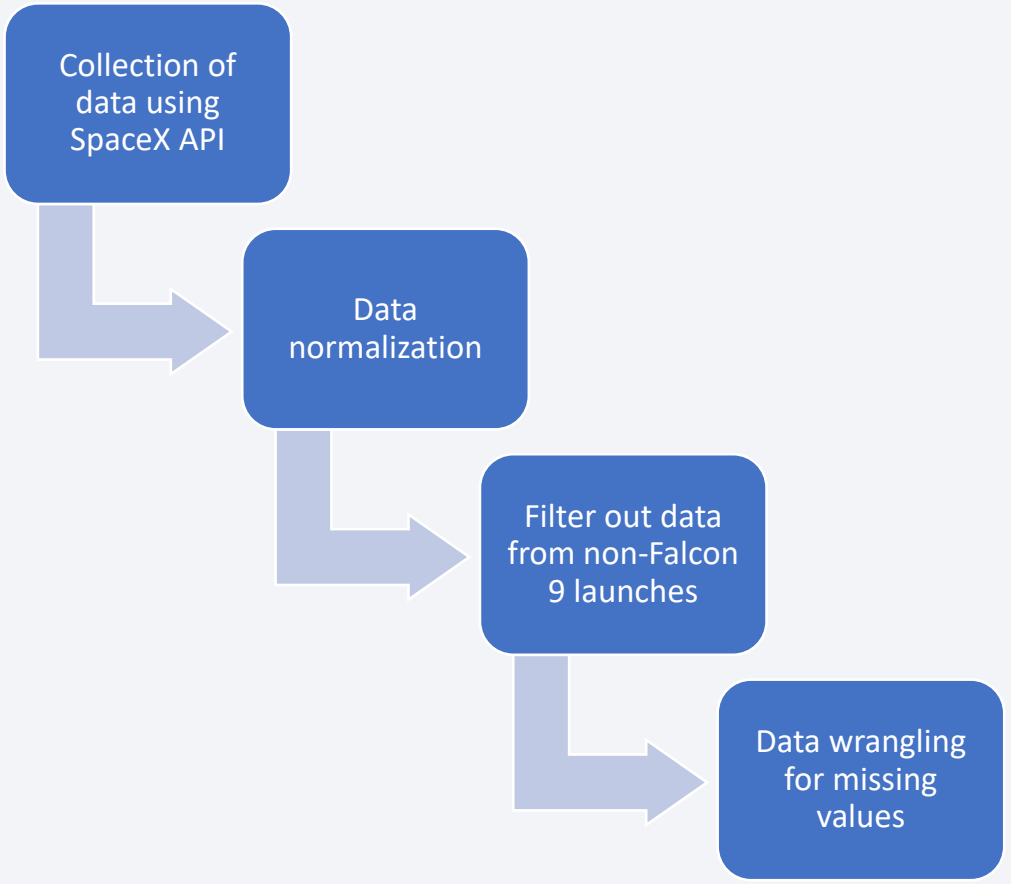
Methodology

Executive Summary

- Data collection methodology:
  - Data was retrieved using SpaceX API and Web Scrapping.
- Perform data wrangling
  - One hot encoding was applied to categorical values, and data cleaning of null values and non-relevant columns.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - ML models were built with standardized data and were evaluated to find the best predictive model.

Data Collection – SpaceX API

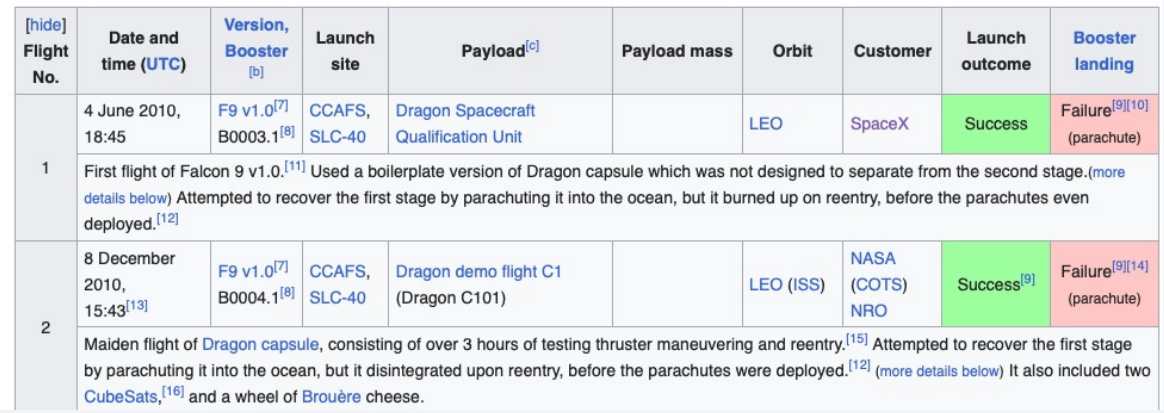
- SpaceX makes available to the public launching data: https://api.spacexdata.com/v4/launches/past
- We used REST calls to retrieve the data through the API.
- We normalized the data, filter it to only keep information from Falcon 9 launches, and cleaned and filled the missing values



Source code:
https://github.com/LeWare6/SpaceX_Launch_Analysis/blob/8e10ee2b0ff9d2d21cce64e5830bd01b359d0544/1_1_Data_Collection_with_API.ipynb

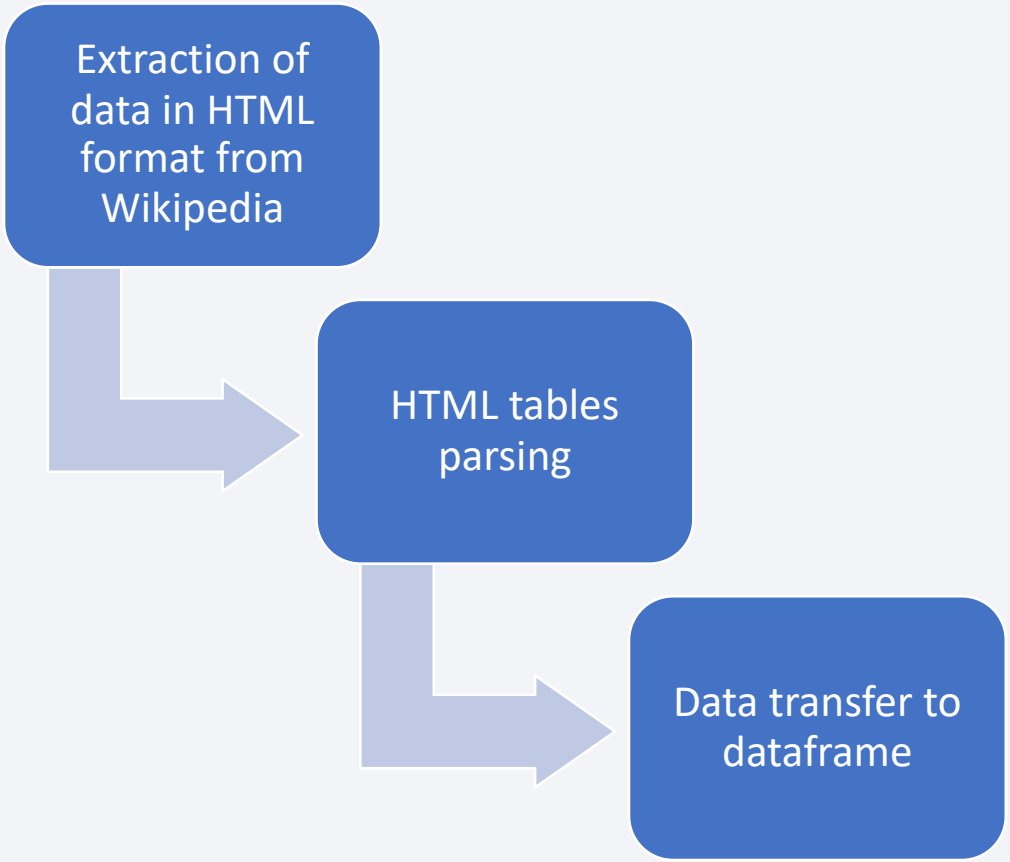
Data Collection - Scraping

- Using Web Scrapping, SpaceX launches data could also be retrieved from Wikipedia: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches



[hide] Flight No.	Date and time (UTC)	Version, Booster [b]	Launch site	Payload[c]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
1	4 June 2010, 18:45	F9 v1.0[7] B0003.1[8]	CCAFS, SLC-40	Dragon Spacecraft Qualification Unit		LEO	SpaceX	Success	Failure[9][10] (parachute)
	First flight of Falcon 9 v1.0.[11] Used a boilerplate version of Dragon capsule which was not designed to separate from the second stage.(more details below) Attempted to recover the first stage by parachuting it into the ocean, but it burned up on reentry, before the parachutes even deployed.[12]								
2	8 December 2010, 15:43[13]	F9 v1.0[7] B0004.1[8]	CCAFS, SLC-40	Dragon demo flight C1 (Dragon C101)		LEO (ISS)	NASA (COTS) NRO	Success[9]	Failure[9][14] (parachute)
	Maiden flight of Dragon capsule, consisting of over 3 hours of testing thruster maneuvering and reentry.[15] Attempted to recover the first stage by parachuting it into the ocean, but it disintegrated upon reentry, before the parachutes were deployed.[12] (more details below) It also included two CubeSats,[16] and a wheel of Brouère cheese.								

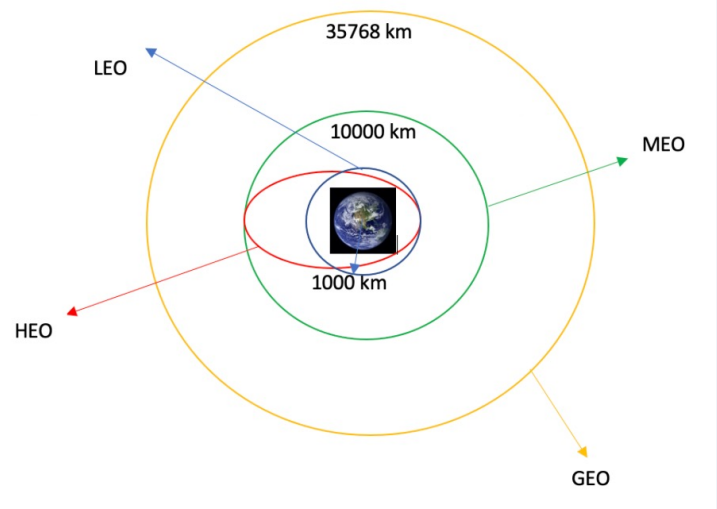
- We extract the tables in HTML format and convert them to dataframes



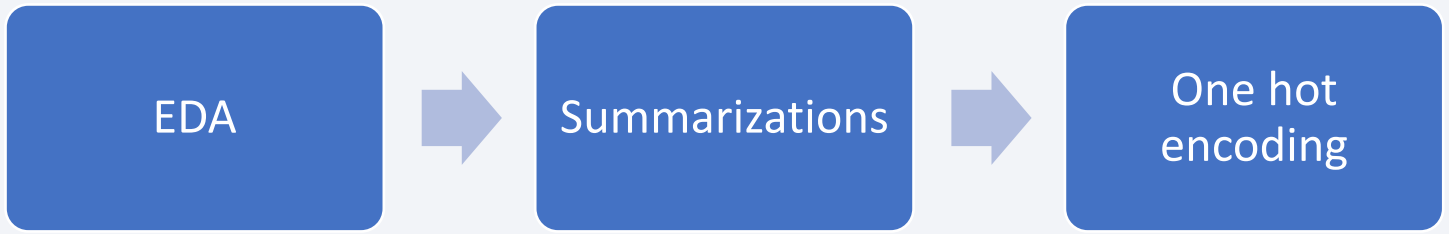
Source code:
https://github.com/LeWare6/SpaceX_Launch_Analysis/blob/8e10ee2b0ff9d2d21cce64e5830bd01b359d0544/1_2_Data_Collection_with_Web_Scraping.ipynb

Data Wrangling

- Exploratory Data Analysis (EDA) was performed on the dataset where we calculated the following:
  - Missing values in each attribute
  - Number of launches on each launch site
  - Occurrence of each rocket orbit
  - Occurrence of landing outcome per orbit type



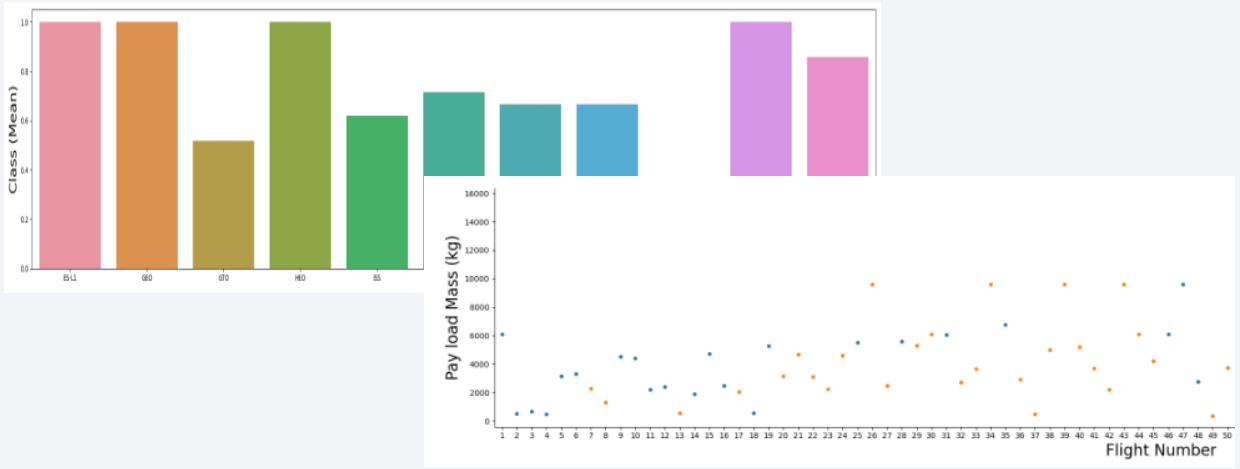
- One hot encoding was performed for one of the columns.



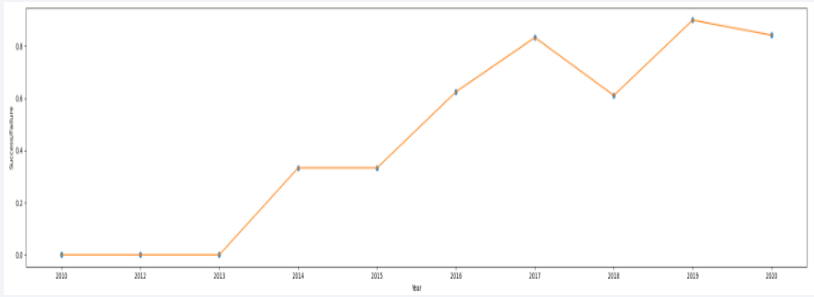
Source code:
https://github.com/LeWare6/SpaceX_Launch_Analysis/blob/8e10ee2b0ff9d2d21cce64e5830bd01b359d0544/2_Data_Wrangling.ipynb

EDA with Data Visualization

- Scatterplots and bar charts were used to explore the relationships between variables, such as:
  - Flight number vs payload mass
  - Flight number vs launch site
  - Payload mass vs launch site
  - Flight number vs orbit type
  - Payload mass vs orbit type



- A line chart were used to explore success trend



Source code:
https://github.com/LeWare6/SpaceX_Launch_Analysis/blob/8e10ee2b0ff9d2d21cce64e5830bd01b359d0544/3_2_EDA_with_Data_Visualization.ipynb

EDA with SQL

- The following are the SQL queries that were performed:
  - Names of unique launch sites in the space mission
  - Five oldest records where launch sites begin with CCA
  - Total payload mass carried by boosters launched by Nasa (CRS)
  - Average payload mass carried by booster version F9 v1.1
  - Date when the first successful landing outcome in a ground pad was achieved
  - Name of the boosters which have success in drone ships and have payload mass between 4000 and 6000 kg
  - Number of successful and failed mission outcomes
  - Names of the boosters which have carried the maximum payload mass.
  - Failure landing outcomes in drone ships in 2015.
  - Ranking of different successful landing outcomes between 04-06-2010 and 20-03-2017 in descending order.

Source code:
https://github.com/LeWare6/SpaceX_Launch_Analysis/blob/8e10ee2b0ff9d2d21cce64e5830bd01b359d0544/3_1_EDA_with_SQL.ipynb

Build an Interactive Map with Folium

- We created markers, in the form of circles, and labels in all launch sites on a folium map.
- We added launch records marks grouped by marker cluster.
- We added a Mouse Position to get coordinates for a mouse over a point on the map.
- We added distance markers, in the form of lines, and labels to indicate the distance to nearby places

Source code:
https://github.com/LeWare6/SpaceX_Launch_Analysis/blob/8e10ee2b0ff9d2d21cce64e5830bd01b359d0544/4_1_Interactive_Data_Analytics_with_Folium.ipynb

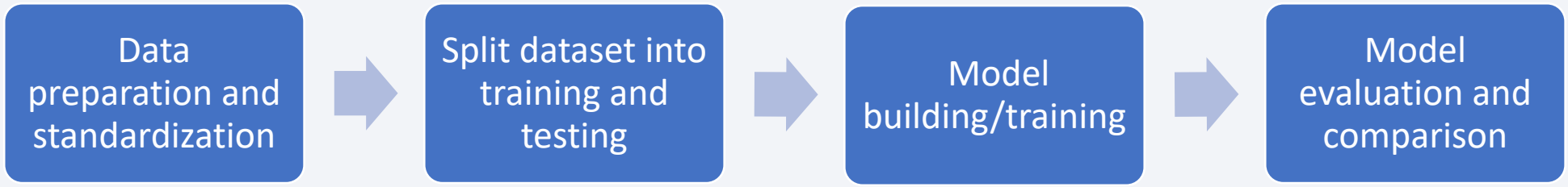
Build a Dashboard with Plotly Dash

- An interactive dashboard with Plotly Dash was built
- It contains a pie chart showing the number of launches from a given launch location.
- It also includes a scatter plot that shows the relationship between the launch outcome and the rocket's payload mass (kg).
- Both graphs are interactives according to a selected launch locations and payload mass range.

Source code:
https://github.com/LeWare6/SpaceX_Launch_Analysis/blob/8e10ee2b0ff9d2d21cce64e5830bd01b359d0544/4_2_Spacex_Dash_App.py

Predictive Analysis (Classification)

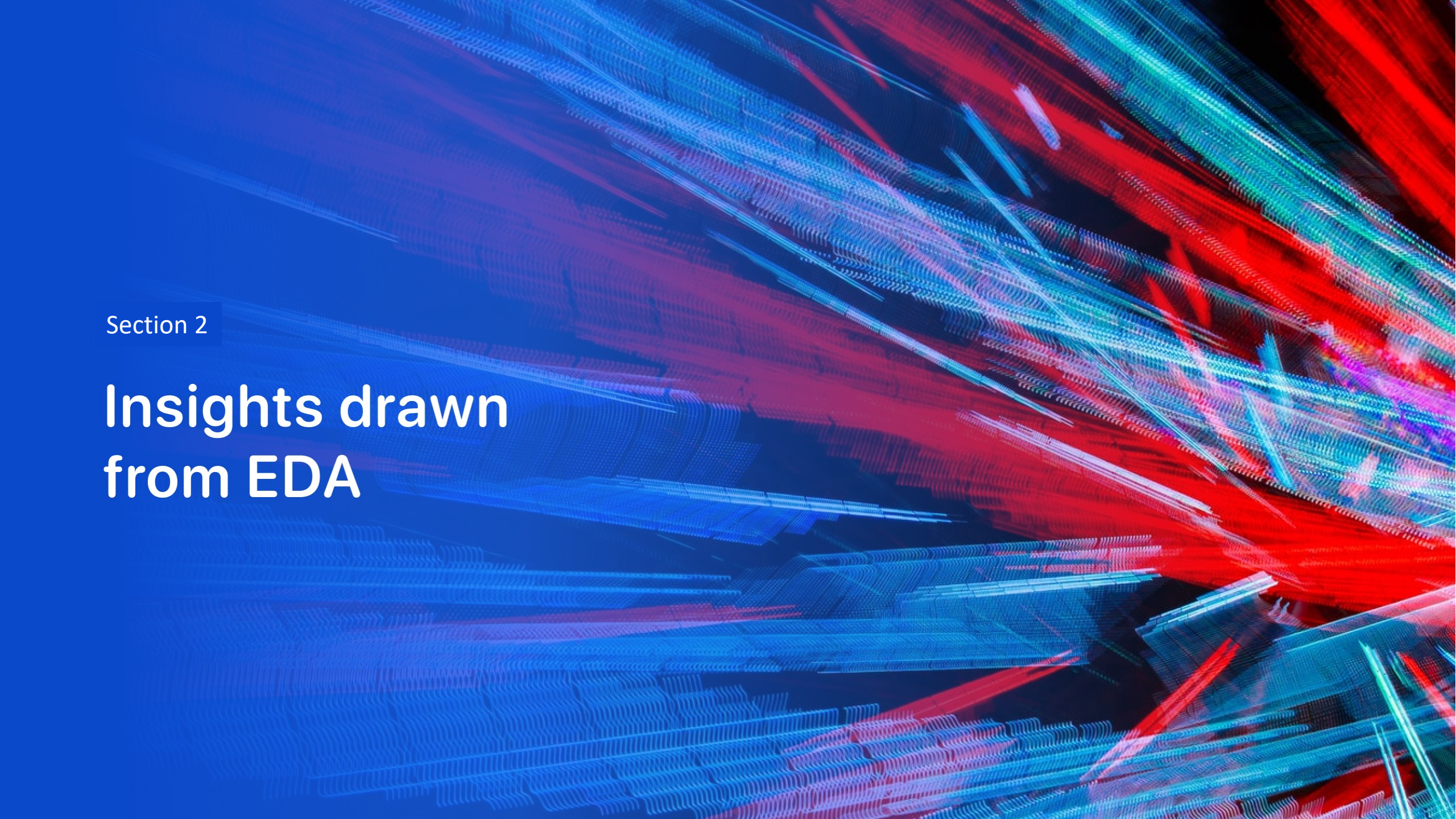
- The data was standardized and split into training and testing.
- Different machine learning models, such as Logistic Regression, Support Vector Machine (SVM), Decision Trees and K-nearest neighbors (KNN), were built using Grid Search, which includes finding the best hyperparameters for the models.
- The models' accuracies were calculated and compared to determine the best predictor.
- A confusion matrix was also plotted to evaluate the models' performance.



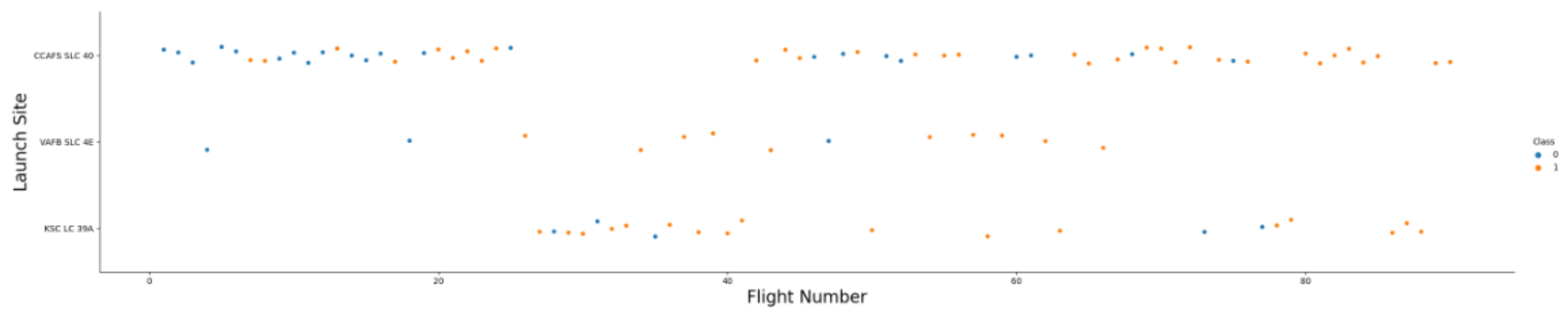
Source code:
https://github.com/LeWare6/SpaceX_Launch_Analysis/blob/8e10ee2b0ff9d2d21cce64e5830bd01b359d0544/5_ML_Predictive_Analysis.ipynb

Results

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

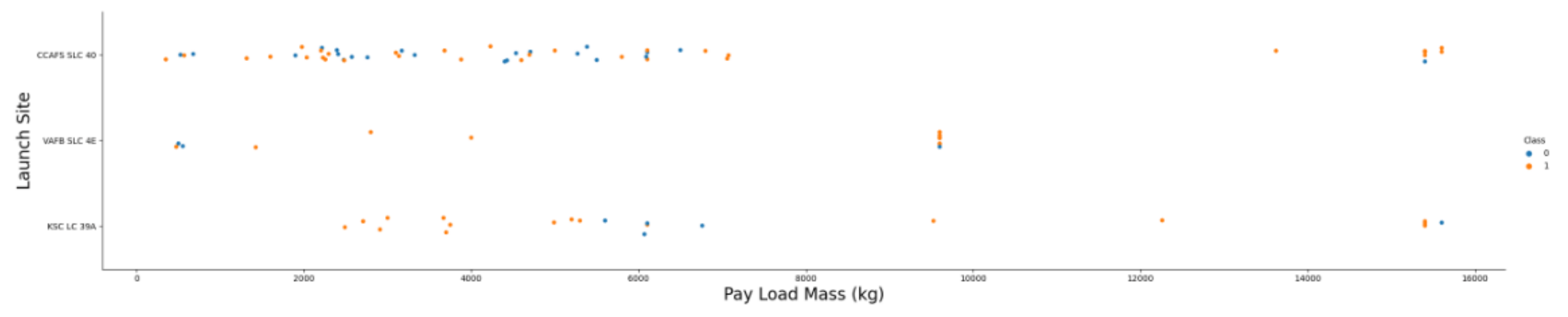

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



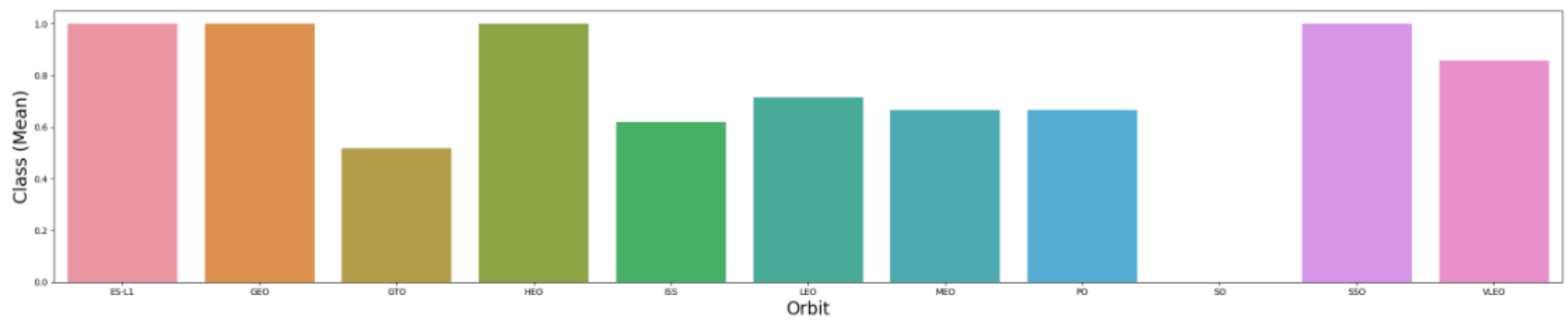
- According to the plot, the success rate improved as the flight number increases in CCAFS SLC 40.
- In contrast, the other 2 launch sites seems to have a steady good success rate. However, there are no enough observations to conclude with certainty.

Payload vs. Launch Site



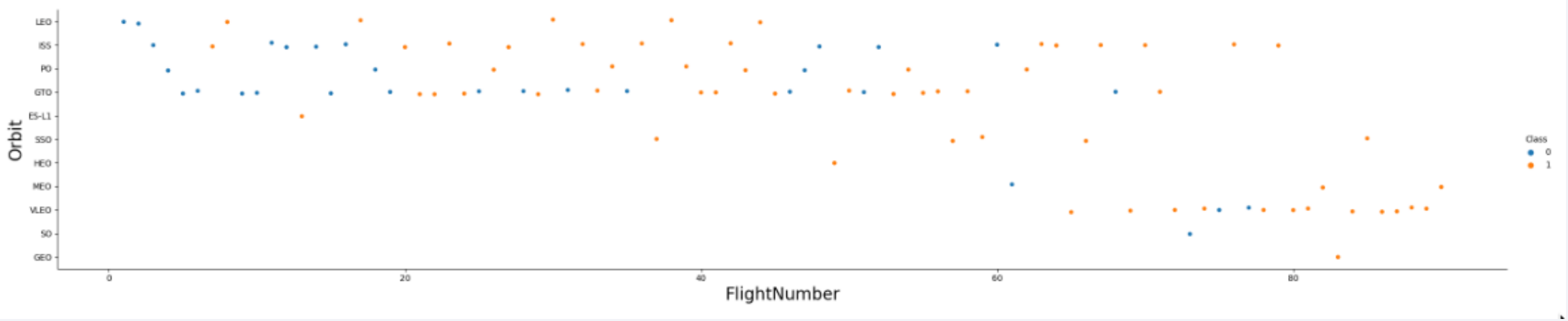
- There are few rockets launched with more than 8,000 kg of Payload Mass.
- There are no rockets launched in the VAFB SLC 4E with more than 10,000 kg of Payload Mass
- We can not observe a relationship between these two variables.

Success Rate vs. Orbit Type



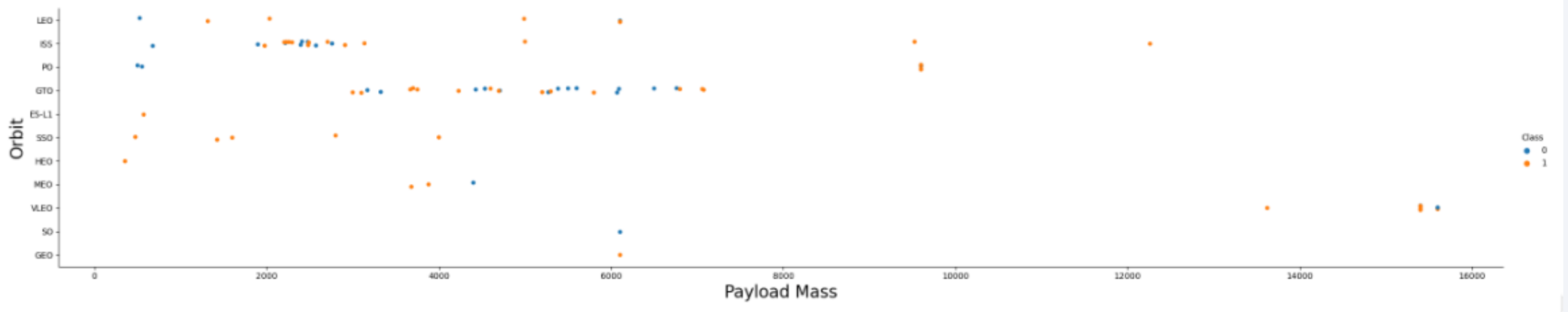
- According to the plot, the orbits with the greatest success rate (100%) are ES-L1, GEO, HEO and SSO.
- In contrast, SO has a success rate of 0%.

Flight Number vs. Orbit Type



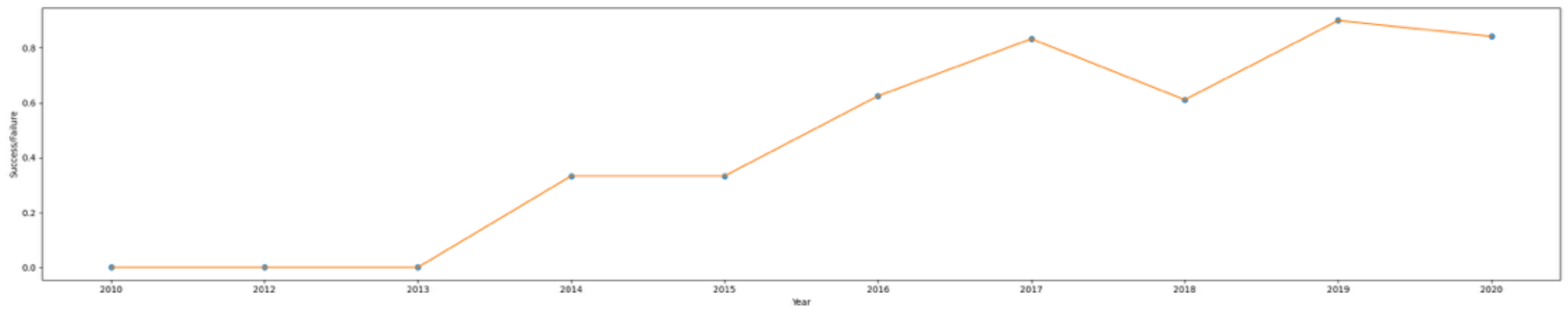
- There has been only one flight to GEO, HEO, and ES-L1, which all were successful. As a consequence, their high success rate is not relevant.
- The success of the LEO orbit seems to be related to the number of flights.

Payload vs. Orbit Type



- The most successful landing rates with heavy payloads are for PO and ISS orbits.
- There are no observations of heavy payloads flights for most of the orbits.

Launch Success Yearly Trend



- The launch success rate has increased since 2013, possibly due to lessons learned and technological advances.

All Launch Site Names

- DISTINCT is used to retrieve the unique values of a column, in this case Launch_Site column

```
%%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

Launch Site Names Begin with 'CCA'

- The LIKE command is used with wildcards represented as % to search for words contained in an entire character string. In this case, we looked for strings that start with CCA in the Launch_Site column

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- SUM is used to retrieve the sum of all values from a column. In this case, we obtained the total sum of PAYLOAD_MASS_KG column for the customer “NASA (CRS)”

```
%%sql

SELECT SUM(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Customer = 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- SUM is used to retrieve the average of all values from a column. In this case, we obtained the average from PAYLOAD_MASS_KG column for the Booster “F9 V1.1”

```
%%sql

SELECT AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Booster_Version like 'F9 V1.1%'
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- MIN is used to retrieve the minimum value from a column. In this case, we obtained the minimum value from Date column. In other words, we got the oldest date for the given WHERE clause
- Since the dates were string type, we also had to transform them from DD-MM-YYYY to YYYY-MM-DD.

```
%%sql
SELECT min(substr(Date, 7, 4) || "-" ||
           substr(Date, 4, 2) || "-" ||
           substr(Date, 1, 2))
           AS 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

- The BETWEEN command is used to filter the records with values within a given range. In this case, we got the boosters' names that successfully landed with a payload mass greater than 4000 but less than 6000.

```
%%sql
SELECT Booster_Version
FROM SPACEXTBL
WHERE "Landing _Outcome" = "Success (drone ship)"
    AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
LIMIT 10
```

```
 * 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

- The CASE expression was used to return values based on the condition met. Therefore, the character strings that contain ”Success” were transformed into “Success” and the rest strings were transformed into “Failure”.
- Afterwards, a GROUP BY statement was used to group the records that have the same values in the Mission_Outcome column. In addition, a COUNT function was used to calculate the number of records with the same Mission_Outcome value.

```
%%sql
SELECT Mission_Outcome, count(1)
FROM (SELECT CASE
                WHEN Mission_Outcome like '%Success%' THEN 'Success'
                ELSE 'Failure'
             END AS Mission_Outcome
      FROM SPACEXTBL)
GROUP BY Mission_Outcome
```

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

Mission_Outcome	count(1)
Failure	1
Success	100

Boosters Carried Maximum Payload

- First, the MAX function was used to calculate the maximum value from the PAYLOAD_MASS__KG_ column
- Then, we used a WHERE clause to retrieve all the Boosters that have that Payload Mass

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

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

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- The SUBSTR function was used to retrieve part of the character string from the Date column. In this case, we got the month part for the SELECT clause, and the year part for the WHERE clause.
- The CASE expression was used to get the month from the Date column and transform it into its name form.

```
%%sql
SELECT CASE WHEN substr("Date", 4, 2) = '01' THEN 'January'
            WHEN substr("Date", 4, 2) = '02' THEN 'February'
            WHEN substr("Date", 4, 2) = '03' THEN 'March'
            WHEN substr("Date", 4, 2) = '04' THEN 'April'
            WHEN substr("Date", 4, 2) = '05' THEN 'May'
            WHEN substr("Date", 4, 2) = '06' THEN 'June'
            WHEN substr("Date", 4, 2) = '07' THEN 'July'
            WHEN substr("Date", 4, 2) = '08' THEN 'August'
            WHEN substr("Date", 4, 2) = '09' THEN 'September'
            WHEN substr("Date", 4, 2) = '10' THEN 'October'
            WHEN substr("Date", 4, 2) = '11' THEN 'November'
            ELSE 'December'
       END AS month_name,
      "Landing _Outcome", Booster_version, Launch_Site
FROM SPACEXTBL
WHERE "Landing _Outcome" like '%Failure (drone ship)%'
       AND substr(Date, 7, 4) = '2015'
LIMIT 10
```

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

month_name	Landing _Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- The GROUP BY statement was used to group the records that have the same Landing_Outcome values. In addition, a COUNT function was used to calculate the number of records with the same Landing Outcome value.
- With the WHERE clause, we filtered the result only to get the launches with a successful outcome that occurred between 2010-06-04 and 2017-03-20.
- Finally, use the ORDER BY keyword to sort the resulting rows in descending order.

```
%%sql
SELECT "Landing _Outcome", COUNT(1) FROM SPACEXTBL
WHERE "Landing _Outcome" like "%success%"
    AND (substr(Date, 7, 4)||"-"||substr(Date, 4, 2)||"-"||substr(Date, 1, 2)) BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing _Outcome"
ORDER BY COUNT(1) DESC
```

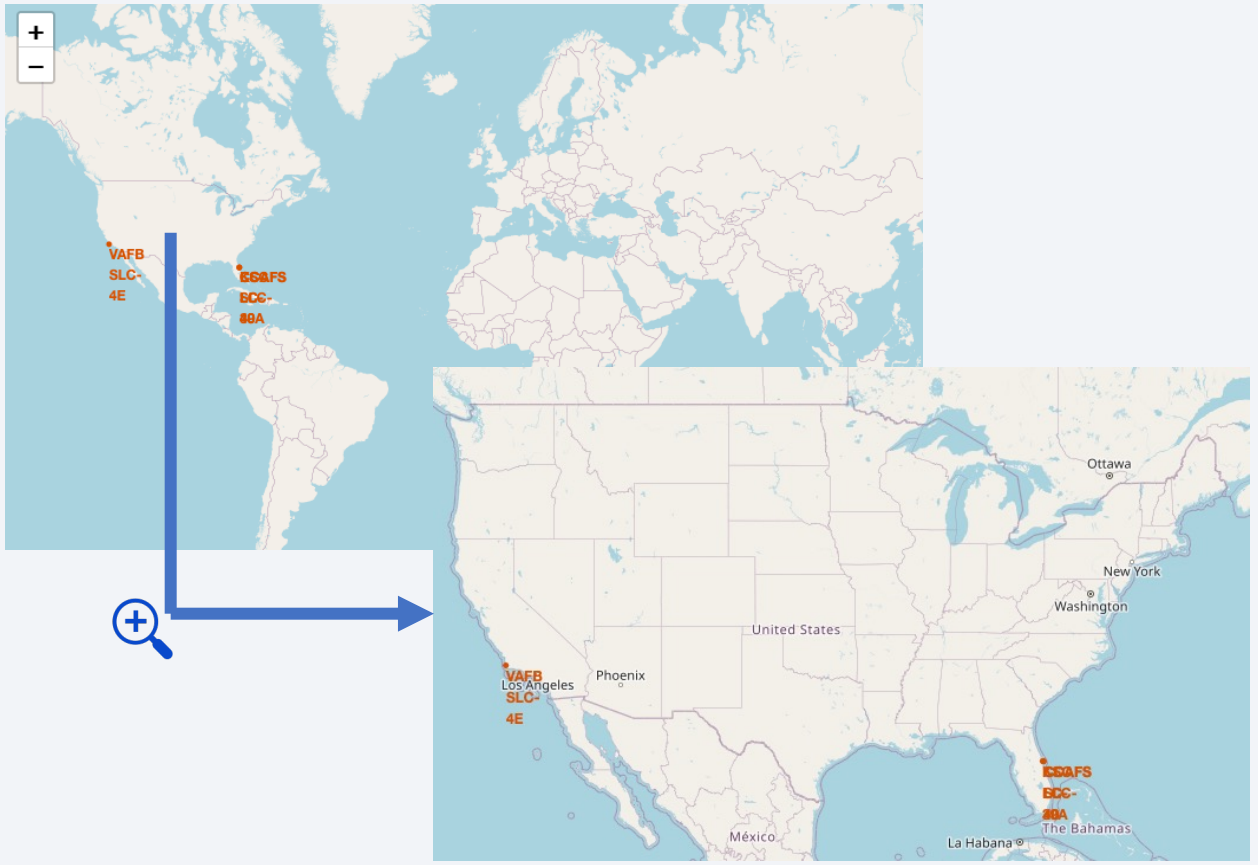
```
 * sqlite:///my_data1.db
Done.
```

Landing _Outcome	COUNT(1)
Success (drone ship)	5
Success (ground pad)	3

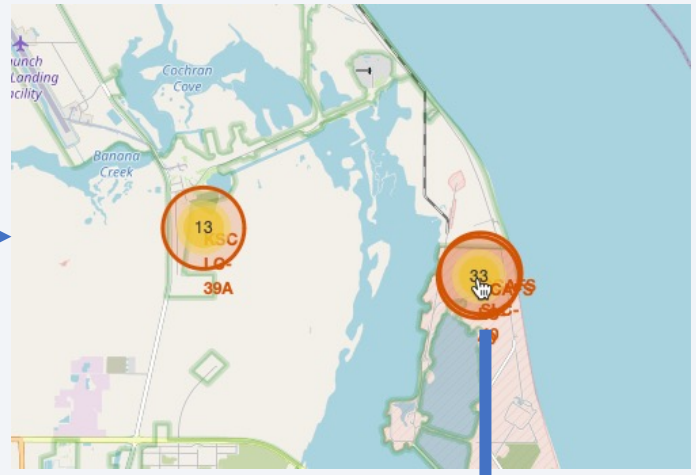

Section 3
Launch Sites
Proximities Analysis

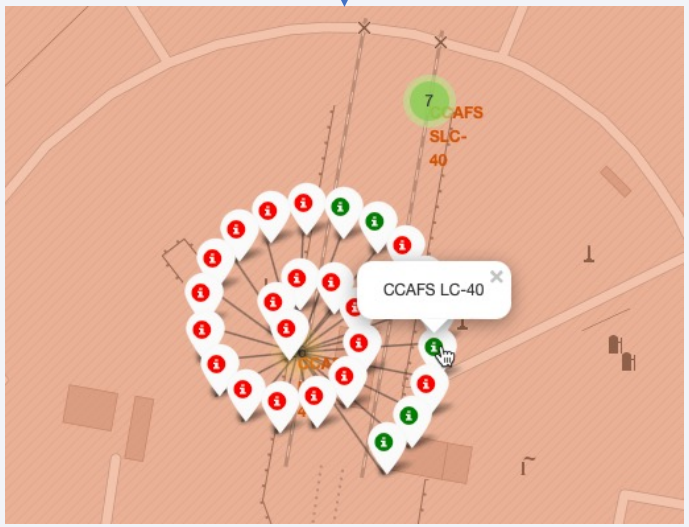
Launch sites marked on a global map

- The map shows the SpaceX launch sites.
- All the sites are in the United States of America, in California and Florida specifically.



Successful and failed launches marked with color labels

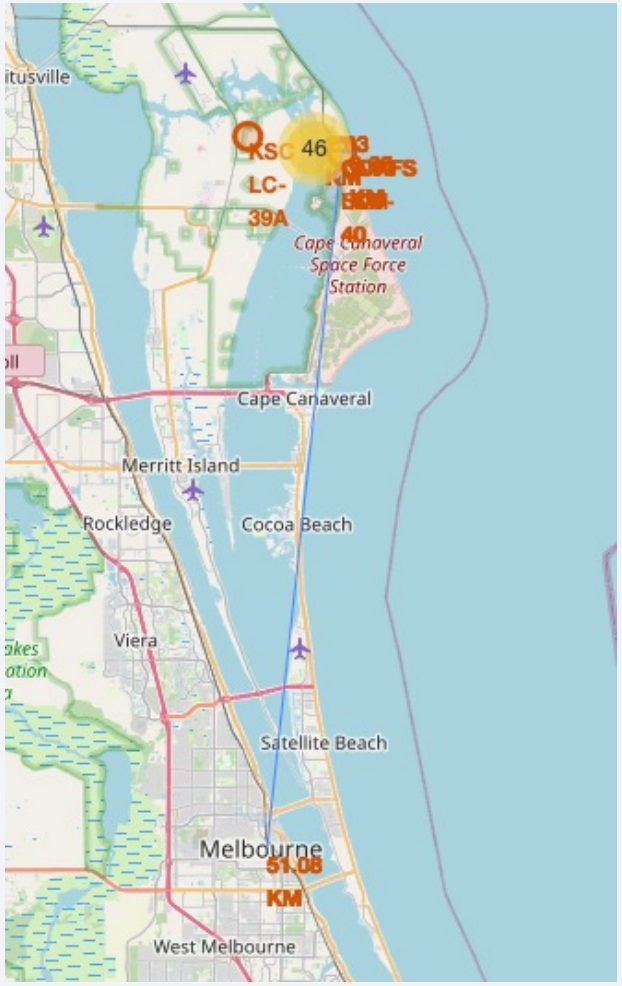


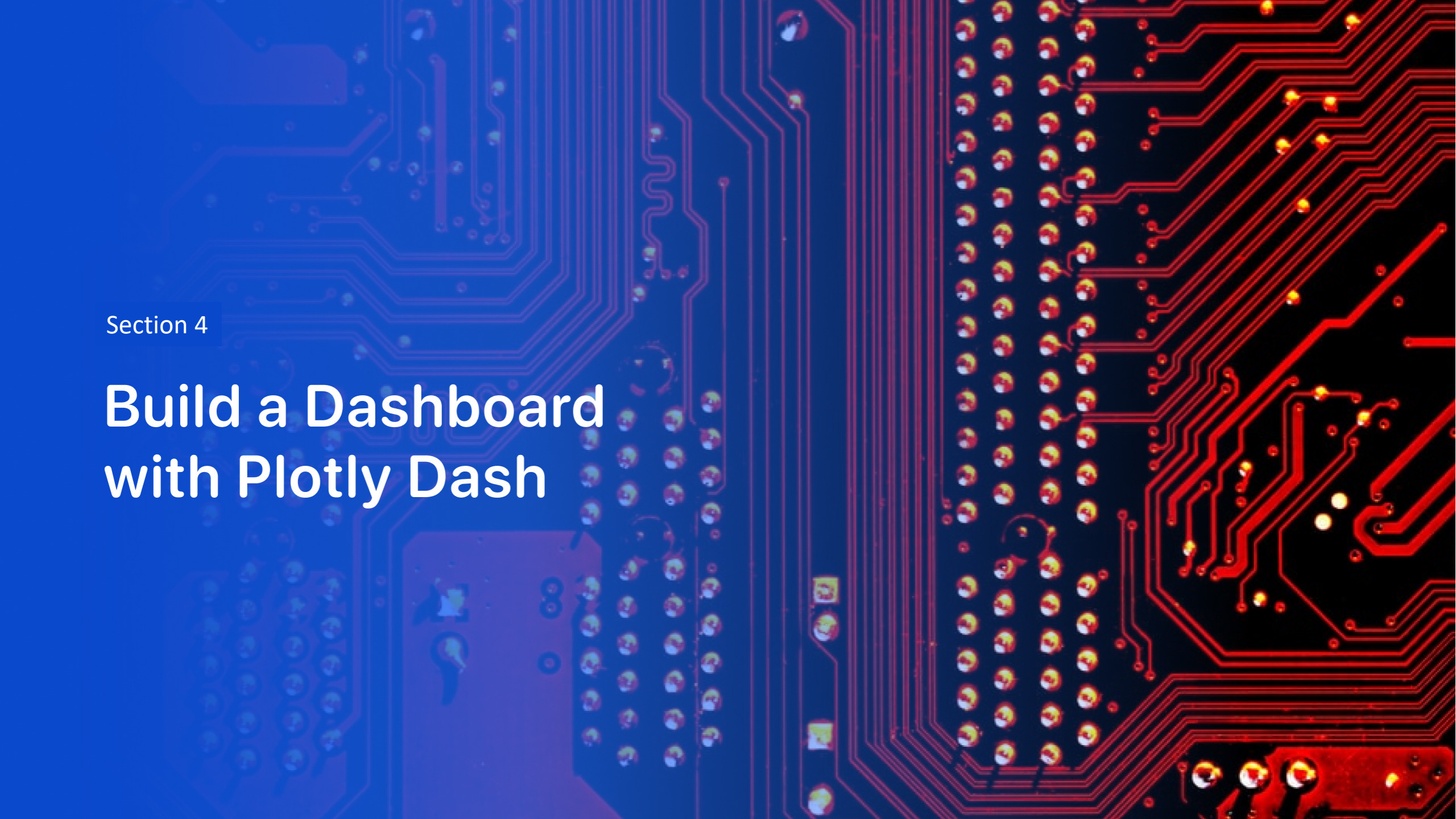


- The map has marks of all the launches grouped by each site.
- Going deep, the outcomes of each launch can be seen. Green markers represent successful launches, and red markers represent failures.

Distances from launch sites to proximities

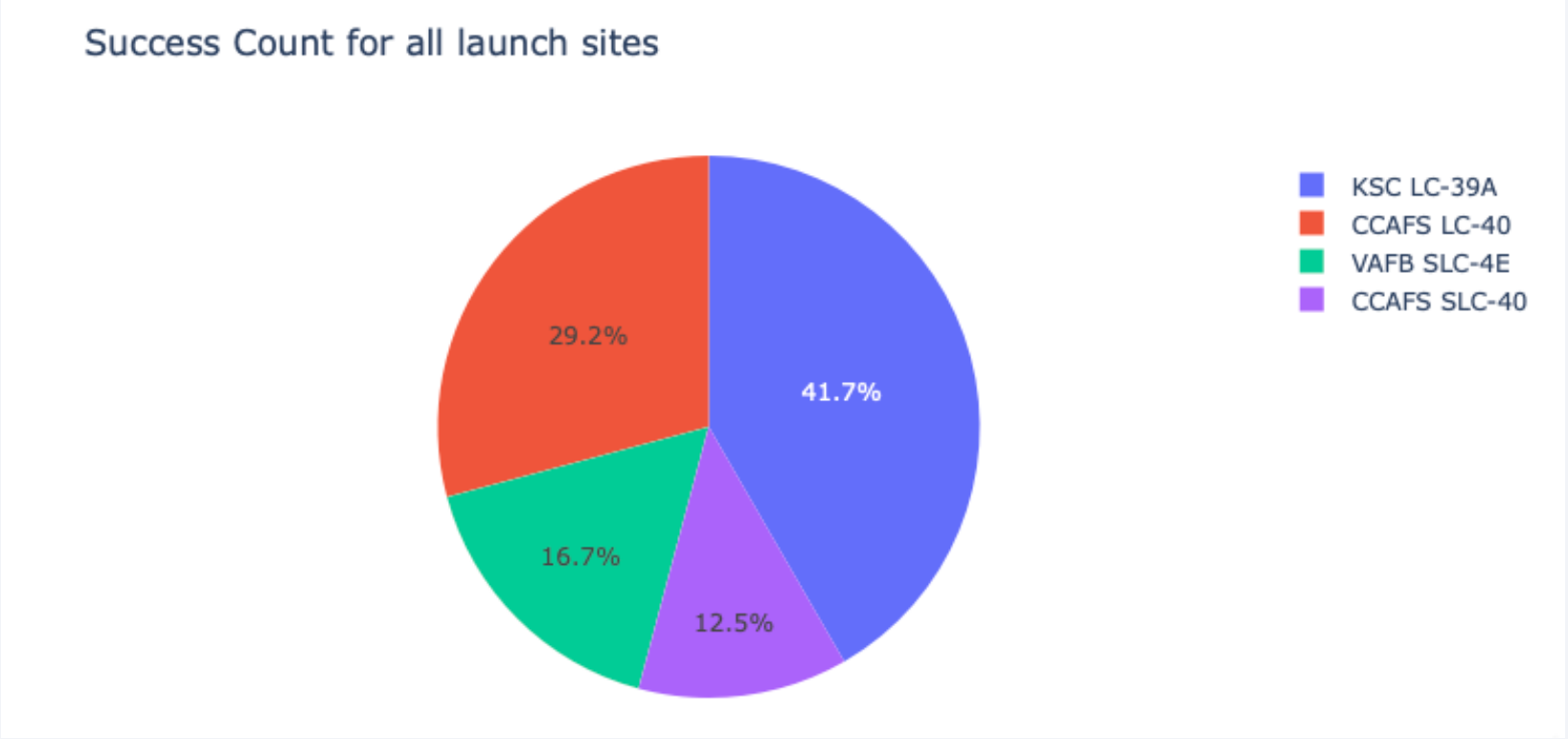
- The map shows that the CCAFS SLC-40 and CCAFS LC-40 launch sites are close to a railway, a highway and the coastline.
- In contrast, the launch sites are far away from the nearest city.




Section 4
Build a Dashboard with Plotly Dash

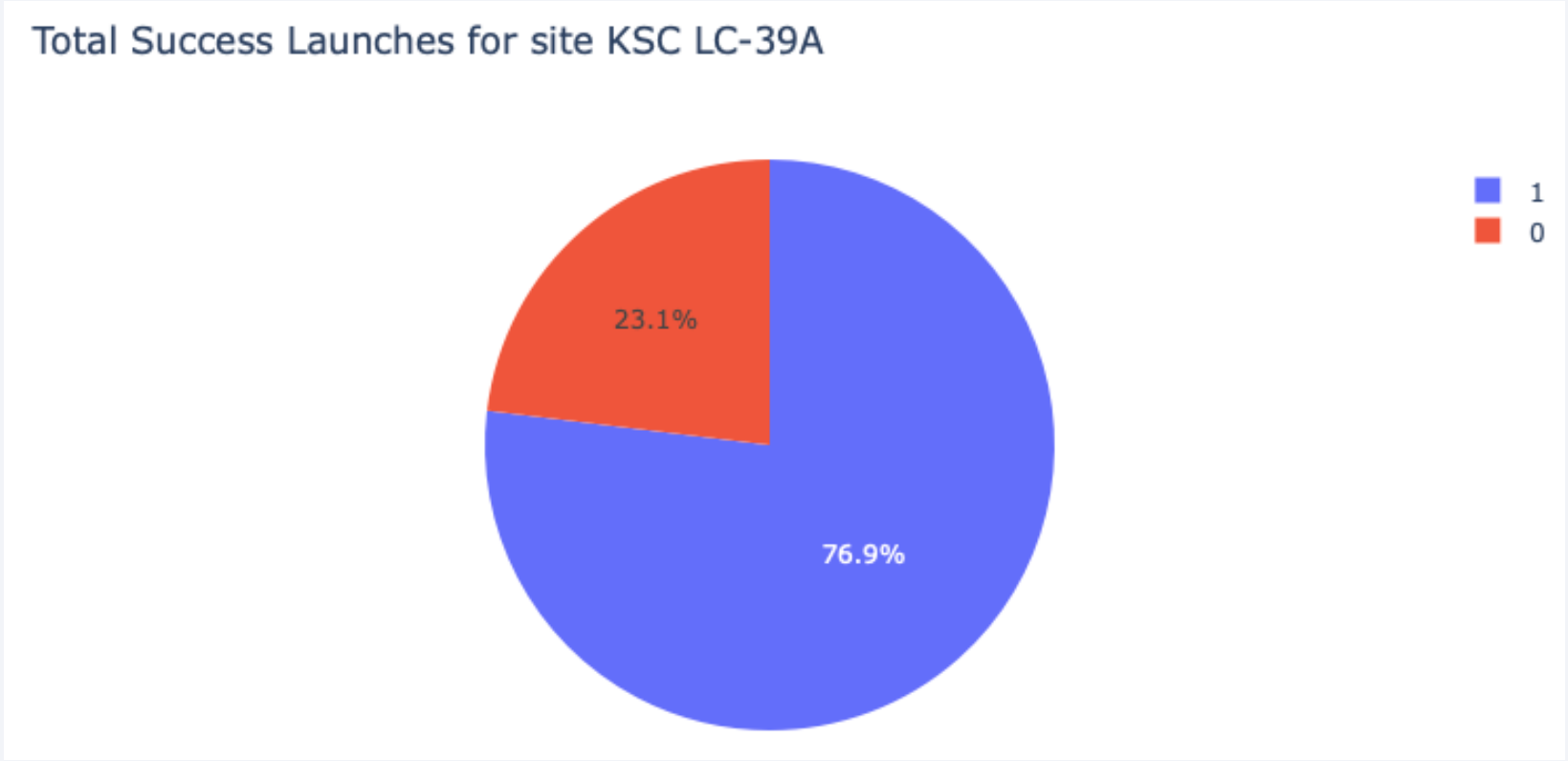
Distribution of successful launches for each site with respect to the total of successful launches

- The pie chart shows that KSC LC-39A is the launch site with more successful launches with respect to the total.



Outcome rate distribution of the site with the greatest success rate

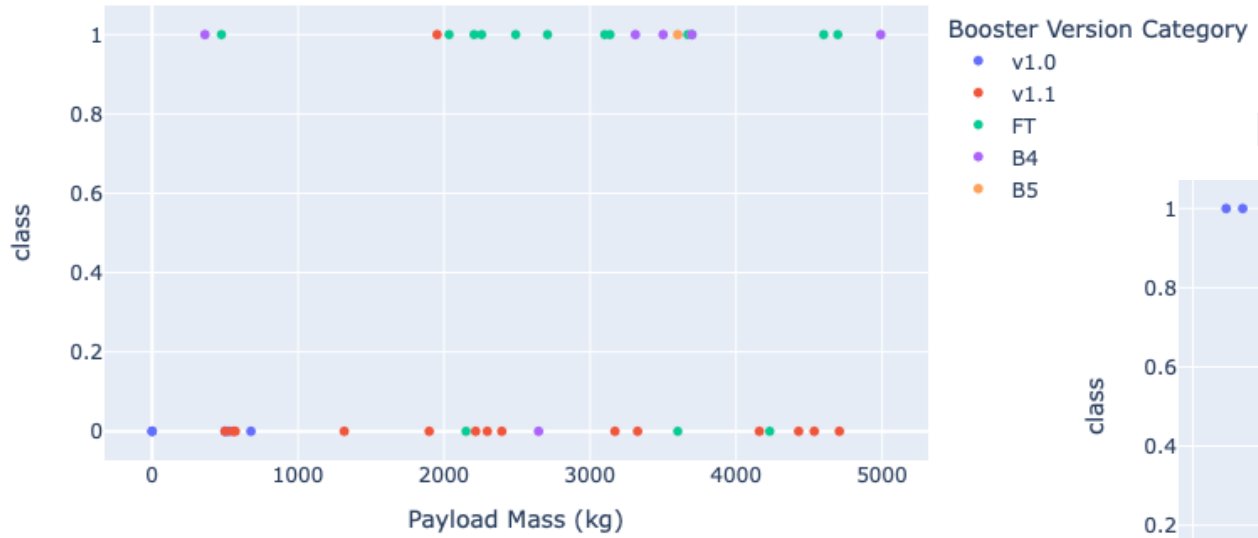
- The pie chart shows the outcome distribution for KSC LC-39A, which is the site with the greatest success rate (76.9%)



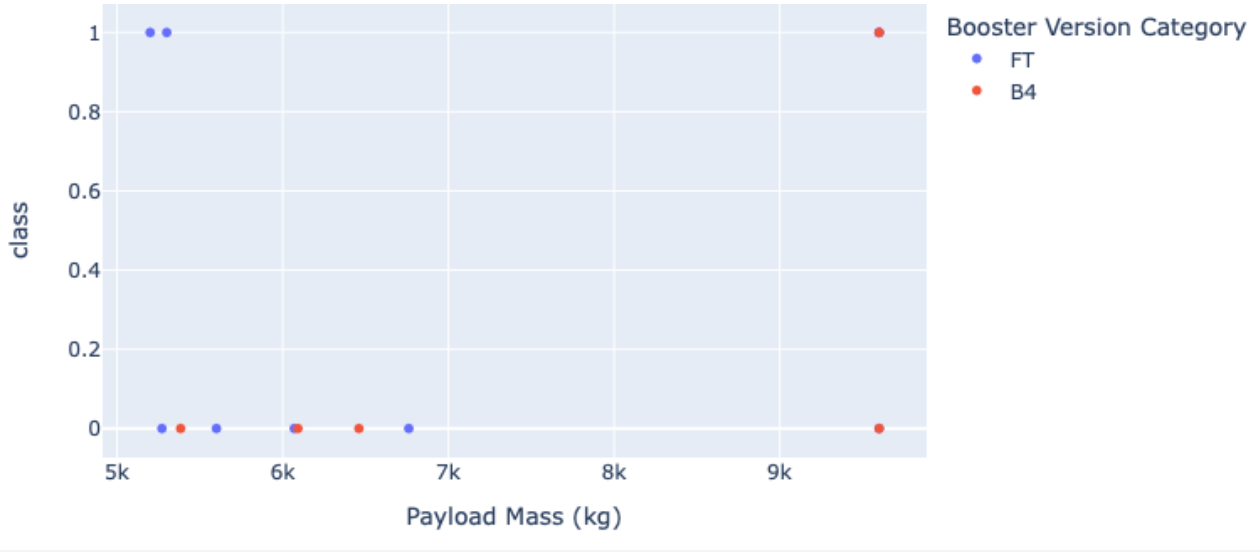
Relation between Payload Mass vs Launch Outcome for all sites

- The scatter plots show that the success rates for launches with light payloads are higher than the ones with heavy loads.

Light payloads (from 0 to 5,000 kg)



Heavy payloads (from 5,000 to 10,000 kg)

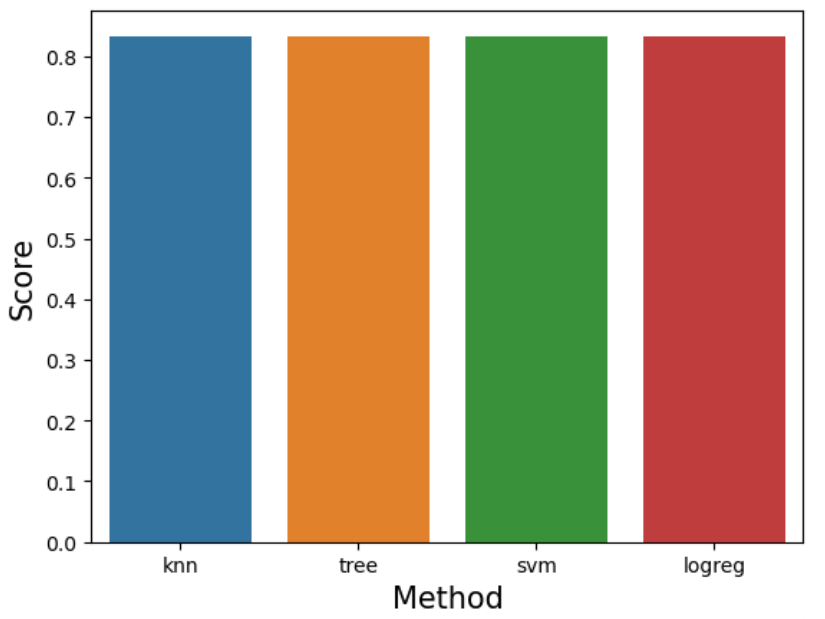


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- All models have an accuracy score of 83.33%



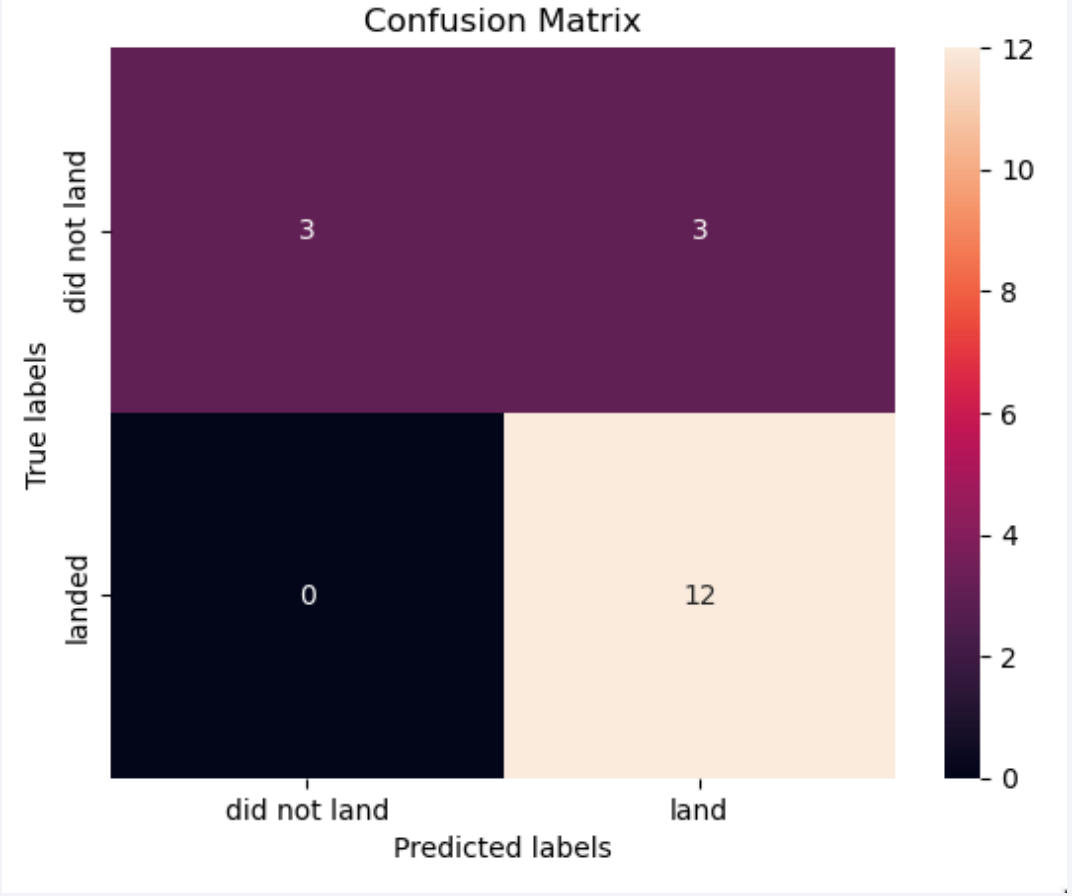

```
knn_score = knn_cv.score(X_test, Y_test)
tree_score = tree_cv.score(X_test, Y_test)
svm_score = svm_cv.score(X_test, Y_test)
logreg_score = logreg_cv.score(X_test, Y_test)
```

```
best_score = 0

if knn_score > best_score:
    best_score = knn_score
    best_method = "KNN"

if tree_score > best_score:
    best_score = tree_score
    best_method = "TREE"
elif tree_score == best_score:
        best_method = "EVEN"

if svm_score > best_score:
    best_score = svm_score
    best_method = "SVM"
elif svm_score == best_score:
        best_method = "EVEN"

if logreg_score > best_score:
    best_score = logreg_score
    best_method = "LOGREG"
elif logreg_score == best_score:
        best_method = "EVEN"

print("The best method is ", best_method," with a score of: ", best_score)
```

```
The best method is  EVEN  with a score of:  0.8333333333333334
```

Confusion Matrix

- All the models share the following Confusion Matrix



- For the test data, the models predicted three false positives. Stated otherwise, there are three predictions of successful outcomes that were failures in reality

Conclusions

- The launch success rate has increased since 2013.
- The success rate improved as the flight number increases in CCAFS SLC 40.
- The site KSC LC-39A has the greatest landing success rate (76.9%).
- Success rates for light payloads launches are higher than the heavy load ones.
- Although few rockets were launched with more than 8,000 kg of Payload Mass, the most successful landing rates are for PO and ISS orbits.
- The site VAFB SLC 4E does not have experience launching rockets with more than 10,000 kg of Payload Mass.
- The orbits ES-L1, GEO, HEO and SSO have a perfect success rate. However, the most relevant is SSO because the other orbits had only one flight.
- Launch sites can be close to railways, highways and coastlines. In contrast, they are far away from the cities.
- For outcome prediction, SVM, KNN, Tree or Logistic Regression models can be used. They all have the same prediction accuracy (83.33%) for this dataset.

Appendix

- All the project documentation can be found in the following GitHub repository: https://github.com/LeWare6/SpaceX_Launch_Analysis.git

Thank you!