IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

David Odutola
8 July 2023

Outline

Executive Summary

- The objective of this project was to develop a predictive model to determine the successful landing of SpaceX's Falcon 9 first stage, with the aim of reducing rocket production and launch costs. Data from the SpaceX API was collected, filtered to focus on Falcon 9 launches, and subjected to exploratory data analysis, feature engineering, and interactive visual analytics.
- Several models were trained and tested, with KNN, SVM, and Logistic Regression consistently achieving an accuracy of 83% and correctly predicting all 12 successful landings. The Decision Trees model achieved an accuracy of 67%, only accurately predicting 9 out of 12 successful landings. These findings provided valuable insights for cost-saving measures and optimization in SpaceX’s rocket production and launches.

Introduction

- SpaceX, known for its cost-efficient operations, advertises Falcon 9 rocket launches at a significantly lower price of 62 million dollars compared to other providers, who charge upwards of 165 million dollars per launch. The key to SpaceX's cost savings lies in their ability to reuse the first stage of the Falcon 9 rocket. Therefore, accurately predicting the landing outcome of the first stage becomes crucial in determining the overall cost of a launch.
- The information generated from our predictive model will be invaluable for alternate companies interested in bidding against SpaceX for rocket launches. By having insights into the probability of successful landings, these companies can make informed decisions on pricing and competitive strategies.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Get request to the SpaceX API
  - Functions to extract useful information such as outcome of previous landings, number of flights, booster names and their versions, payload mass, etc
- Perform data wrangling
  - Replace missing values with mean values
  - Convert Launch outcomes into Training Labels (1 and 0)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models such as SVM, KNN, Logistic Regression and Decision Trees

Data Collection – SpaceX API

Import Libraries and Define Auxiliary Functions

- Libraries such as requests, pandas, numpy, datetime
- Auxiliary functions were defined to extract information such outcome of previous landings, number of flights, booster names and their versions, payload mass, etc.

Request and parse rocket launch data from SpaceX API using GET request

Convert JSON result into a pandas dataframe using .json_normalise() method.

Filter the dataframe using the .drop method to only include Falcon 9 launches

Review the data

Github:
https://github.com/TheNavalScientist/Falcon9_landing_prediction/blob/4a51d5c5f9979898e1cd64fb6eb80f1d9b0daf94/jupyter-labs-spacex-data-collection-api.ipynb

Data Wrangling

- Investigate data for missing values using ***.isnull().sum()*** method
  - Landing pad and Payload mass columns had missing values.
- Calculate mean value of Payload Mass column and replace missing values with mean values.
- Using ***.value_counts()*** method, the following features were determined:
  - Total number of launches with each launch site.
  - Each launch aimed at a dedicated orbit, thus total number of launches per orbit was determined
  - Mission outcome per orbit type

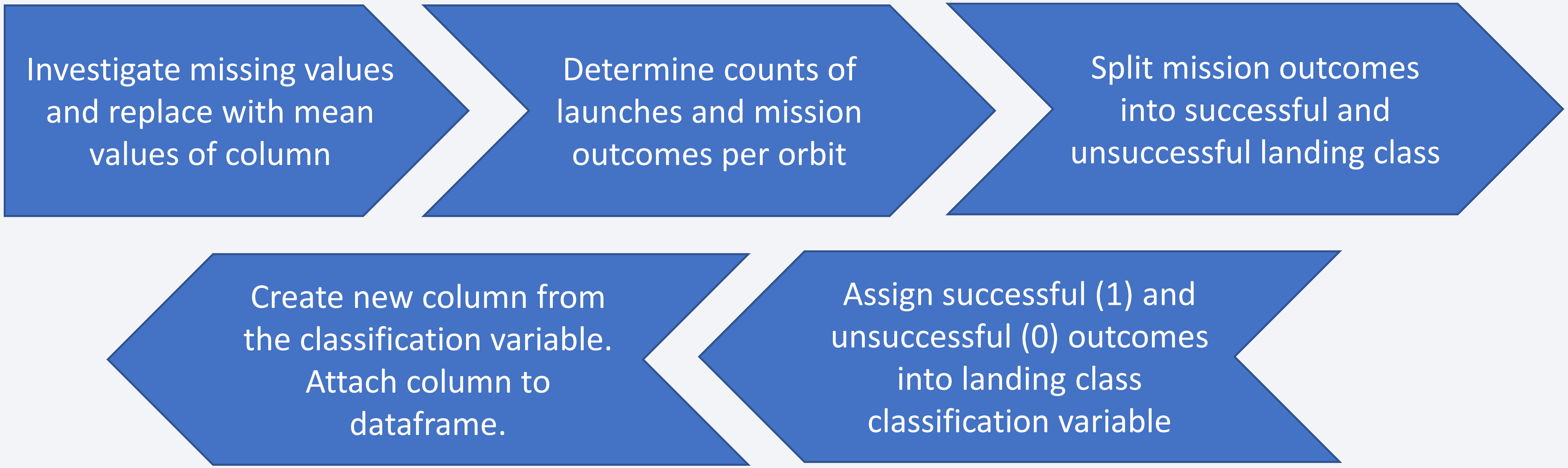
Interpretation of resulting launch outcomes:

- **True Ocean** = mission outcome successfully landed to a specific region of the ocean
- **False Ocean** = mission outcome unsuccessfully landed to a specific region of the ocean.
- **True RTLS** = mission outcome was successfully landed to a ground pad
- **False RTLS** = mission outcome was unsuccessfully landed to a ground pad
- **True ASDS** = mission outcome was successfully landed to a drone ship
- **False ASDS** = mission outcome was unsuccessfully landed to a drone ship.
- **None ASDS** and **None None** = failure to land.

Data Wrangling

- Empty list called bad_outcomes and landing class was created
- Unsuccessful **False Ocean, False RTLS, False ASDS** and **None** were classified as bad_outcomes
- The landing class is the classification variable that represents the outcome of each launch.
- A loop was created which looped through the landing outcomes. If an outcome was found in the bad outcomes list, then the outcome was assigned ‘0’ in the landing class variable. Otherwise, the outcome was assigned ‘1’ in the landing_class variable
- 0 = unsuccessful launch outcome, 1 = successful landing outcome
- New column called ‘Class’ containing the landing outcome of the launches was assigned to the dataframe

Data Wrangling Flowchart



Github:
https://github.com/TheNavalScientist/Falcon9_landing_prediction/blob/825ceed459c942f539c4b5dc88cbb191cdad7490/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

- The Scatter Plot, Bar Chart and Line Plot were used for Data Visualization.
- The Scatter plots helped to understand the correlation between two variables such as Flight Number and Payload Mass, Flight Number and Launch Site, etc. By incorporating different colors, the scatter plots also represented additional dimension or categorical variable such as the launch class
- The Bar chart was used to compare the success rate of the different orbits based on the rocket landing outcomes. This provided a visual hierarchy, allowing quick identification of the highest or lowest success rates as well as facilitating decision-making based on the rankings.
- To understand the trend of the landing outcomes over a period of time, the line plot was used for with data from the periods of 2010 – 2021.

Github: https://github.com/TheNavalScientist/Falcon9_landing_prediction/blob/1d89cd91828c7d9a6455ff09c55fcdff6b8d0c34/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Displayed the names of the unique launch sites in the space mission using keyword DISTINCT
- Displayed 5 records where launch sites begin with the string 'CCA‘ using keywords WHERE, LIKE and LIMIT
- Displayed the total payload mass carried by boosters launched by NASA (CRS) using .fetchall(), SUM(), WHERE and LIKE
- Displayed the average payload mass carried by booster version F9 v1.1 using AVG() and WHERE
- Listed the date when the first successful landing outcome in ground pad was achieved using DATE, WHERE, LIKE and LIMIT
- Listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000 using WHERE and BETWEEN
- Listed the total number of successful and failure mission outcomes using COUNT, WHERE and LIKE with certain conditions

EDA with SQL

- List the names of the booster_versions which have carried the maximum payload mass using WHERE and MAX, with a subquery.
- Listed the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- Ranked 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. Keywords such as COUNT, WHERE and BETWEEN were used.

Github:
https://github.com/TheNavalScientist/Falcon9_landing_prediction/blob/6f3673e93382ff319e0532ed022b1447b8328905/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- A folium.Circle object was created to add a highlighted circle area, with a specific radius, representing the location where the launches are located within a particular launch site.
- A folium.Marker object was created to represent specific locations of the individual launches on the map.
- However, it was observed that many launch records had the exact same coordinates. A MarkerCluster() method was used to simplify the map containing many markers of launch records having the same coordinate.
- With MousePosition() method and an auxiliary function, the positions of reference points such as coastline, railways, towns and their individual distances to launch sites were gotten. The folium.PolyLine was to draw a line from one reference point to another, illustrating the distance between them.

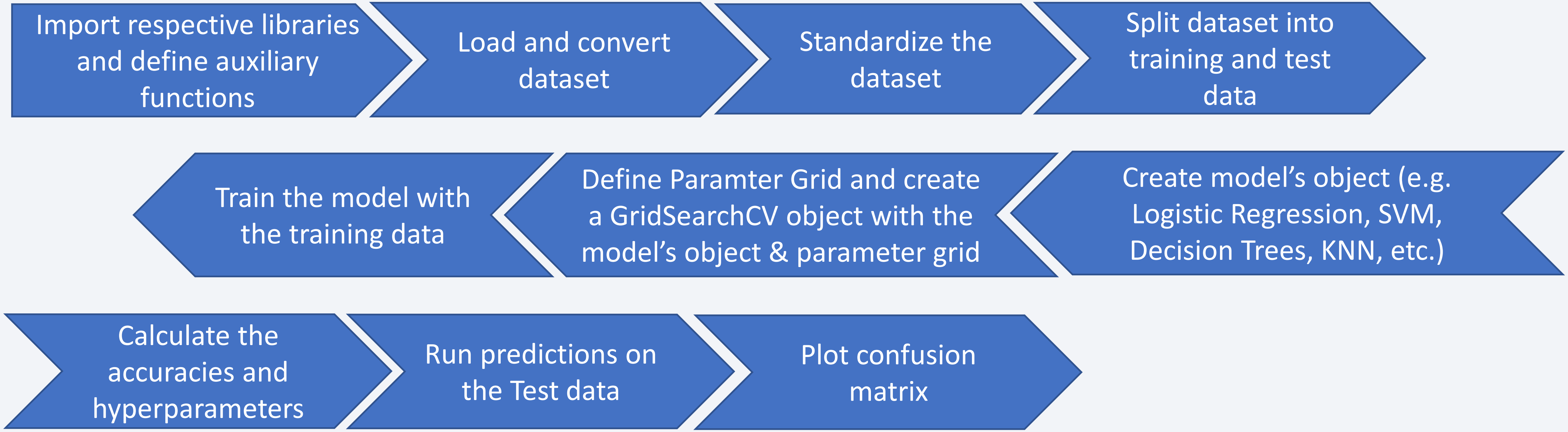
Github:
https://github.com/TheNavalScientist/Falcon9_landing_prediction/blob/1d7df605b2b72679574da39de4e3bb882dfc8cc7/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Launch Site Drop-down Input Component: Used to select all the sites at once, or one specific site from the different launch sites to check its detailed success rate (class=0 vs. class=1).
- Pie Chart: Rendered pie chart was used to illustrate relative proportions or percentages of the successful launch outcomes against the unsuccessful ones in a particular site or all sites. The site of interest was selected from the launch site dropdown input component.
- Range Slider: In order to understand the relationship between payload mass and mission outcome, the range slider was used to select different payload ranges. The slider ranged from 0 – 10,000kg.
- Scatter Plot: The scatter plot was used to visually observe how payload may be correlated with mission outcomes for selected site(s). Additionally, the scatter plots were color-labelled using the Booster version on each scatter point in order to observe mission outcomes with different boosters.

Github:
https://github.com/TheNavalScientist/Falcon9_landing_prediction/blob/91056d6afd870a2d7b6ad1976932b61b6dacaa7e/Space_dash_app.py

Predictive Analysis (Classification Flowchart)



Github:
https://github.com/TheNavalScientist/Falcon9_landing_prediction/blob/621a81e1ff8ca194d58e783bac22319afbdf362c/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Predictive Analysis (Classification)

- Respective libraries such as pandas, numpy, matplotlib, scikit-learn, etc., were imported.
- A function that plots the confusion matrix, taking the test data and predicted data as inputs was defined.
- Data was loaded, converted into a pandas dataframe and assigned to a variable X.
- ‘Class’ column in the data which contained the training labels was converted to a numpy array by using the method to_numpy() and assigned to a variable Y.
- A transform object using the method preprocessing.StandardScaler() was created and used to standardize the data in X.
- With a defined train_test_split function, this function was used to split the data X and Y into training and test data. The parameters test_size was set to 0.2 and random_state to 2. The Training and Test datasets were assigned to the labels: X_train, X_test, Y_train, Y_test.

Predictive Analysis (Classification)

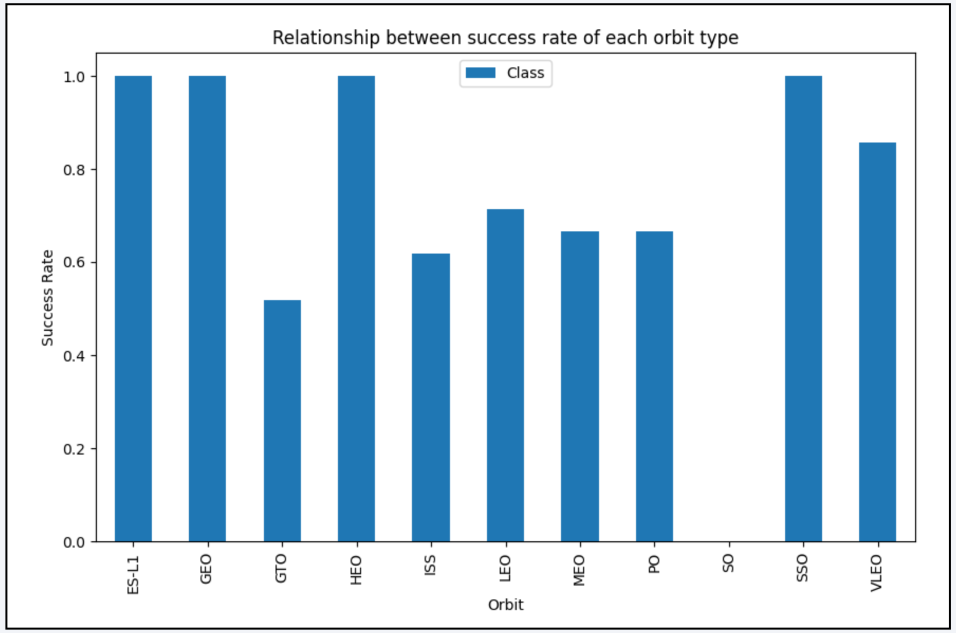
<u>Logistic Regression Model</u>

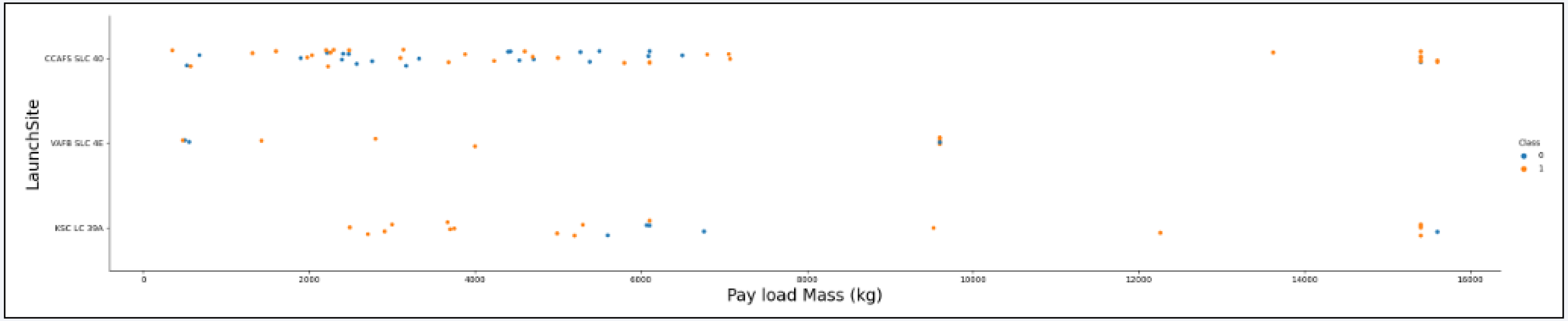
- A logistic regression object was then created using LogisticRegression().
- A parameter grid, which is a dictionary containing the hyperparameters to be tuned, was defined.
- A GridSearchCV object containing the logistic regression object, parameter grid and a cv (cross-validation) of 10 was also created.
- The GridSearchCV object was then fit to training datasets X_train,Y_train using the fit() method.
- The best hyper-parameters and accuracy of the logistic regression's training dataset was gotten using .best_params_ and .best_score_ methods.
- The accuracy of the test datasets, X_test, Y_test was calculated using .score method. This results to how accurate the model is in prediction Falcon 9's first stage landing outcome.

Predictive Analysis (Classification)

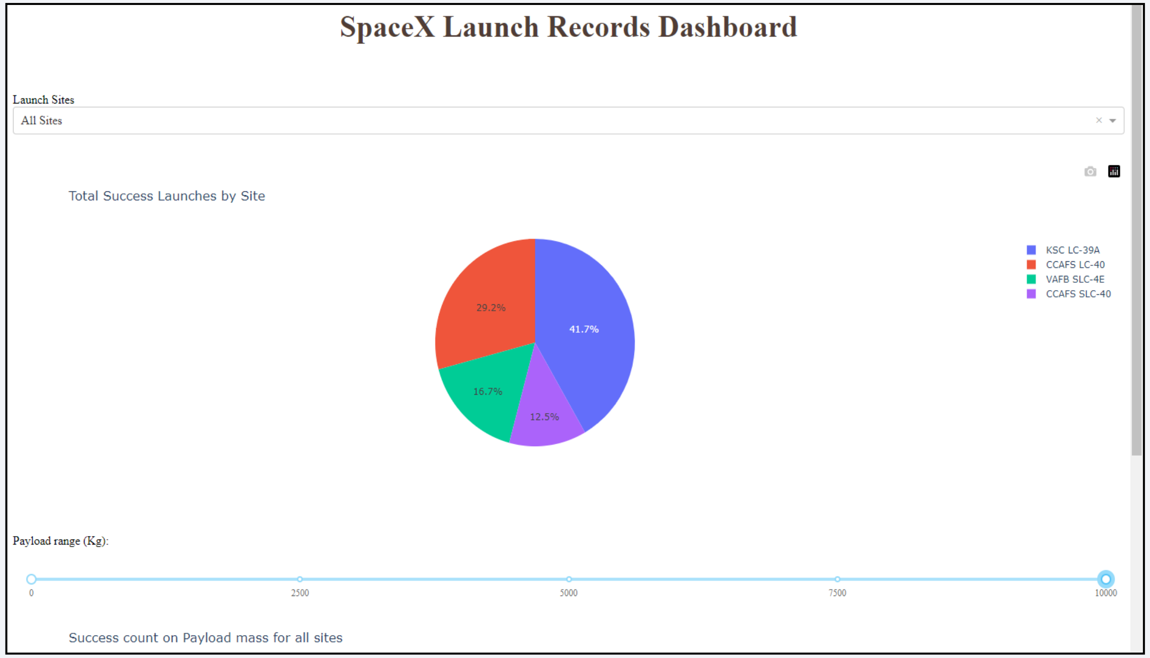
- The model is then used to predict landing outcomes from the test data, X_test, using the method .predict.
- Finally, a confusion matrix is generated from the predicted and actual data in order to evaluate the model's accuracy and understand the types of errors it makes.
- The same steps were used to investigate the hyperparameters and accuracies of other models such as SVM, Decision Trees and KNN.

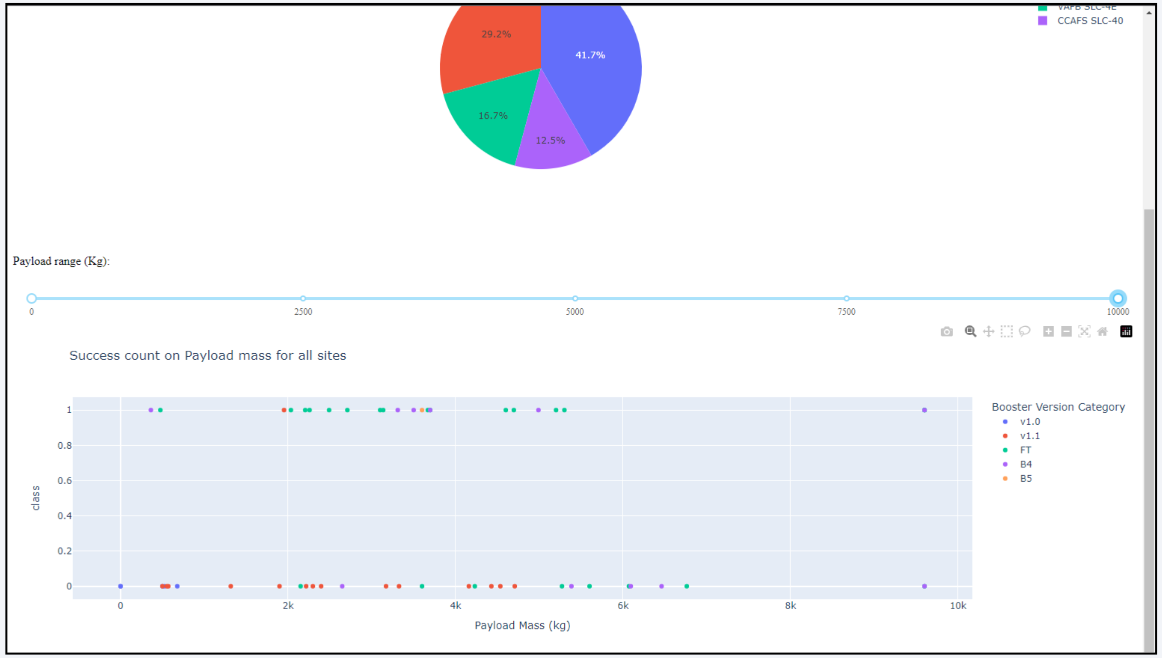
Results – Exploratory Data Analysis



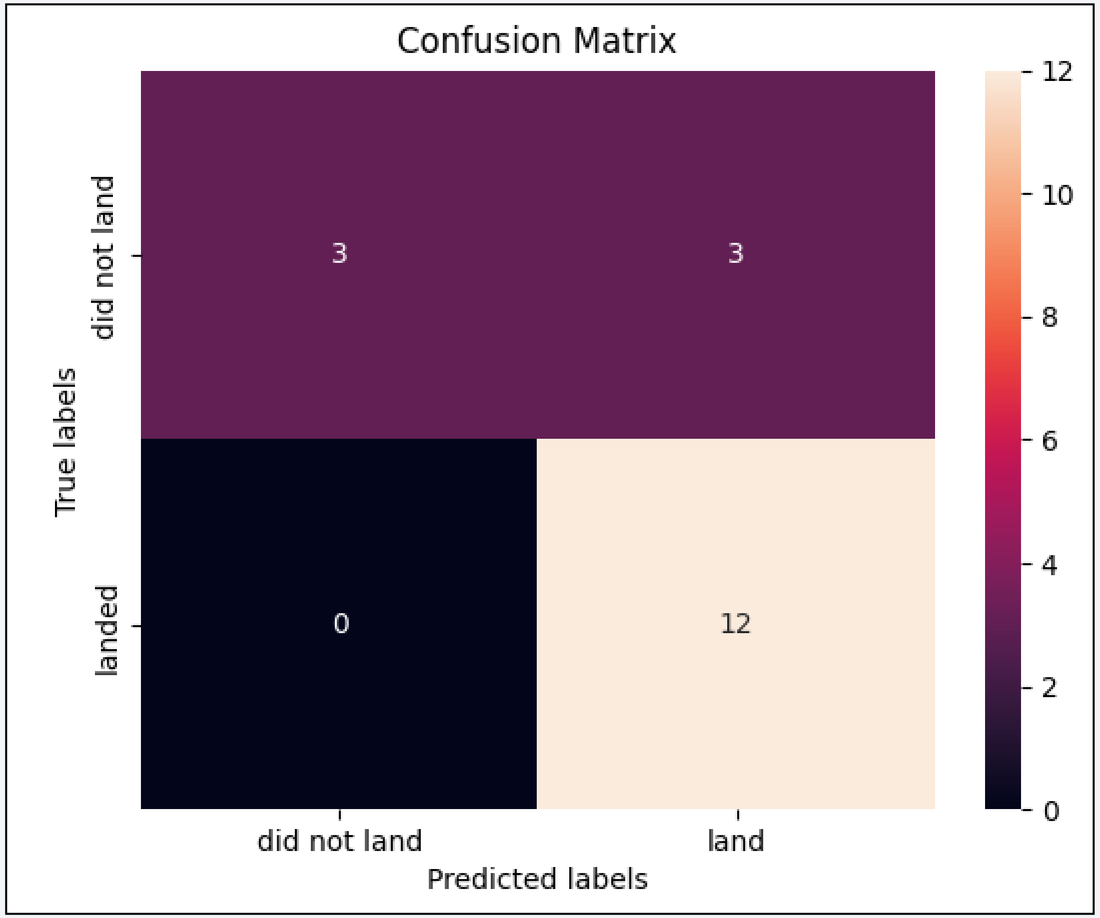


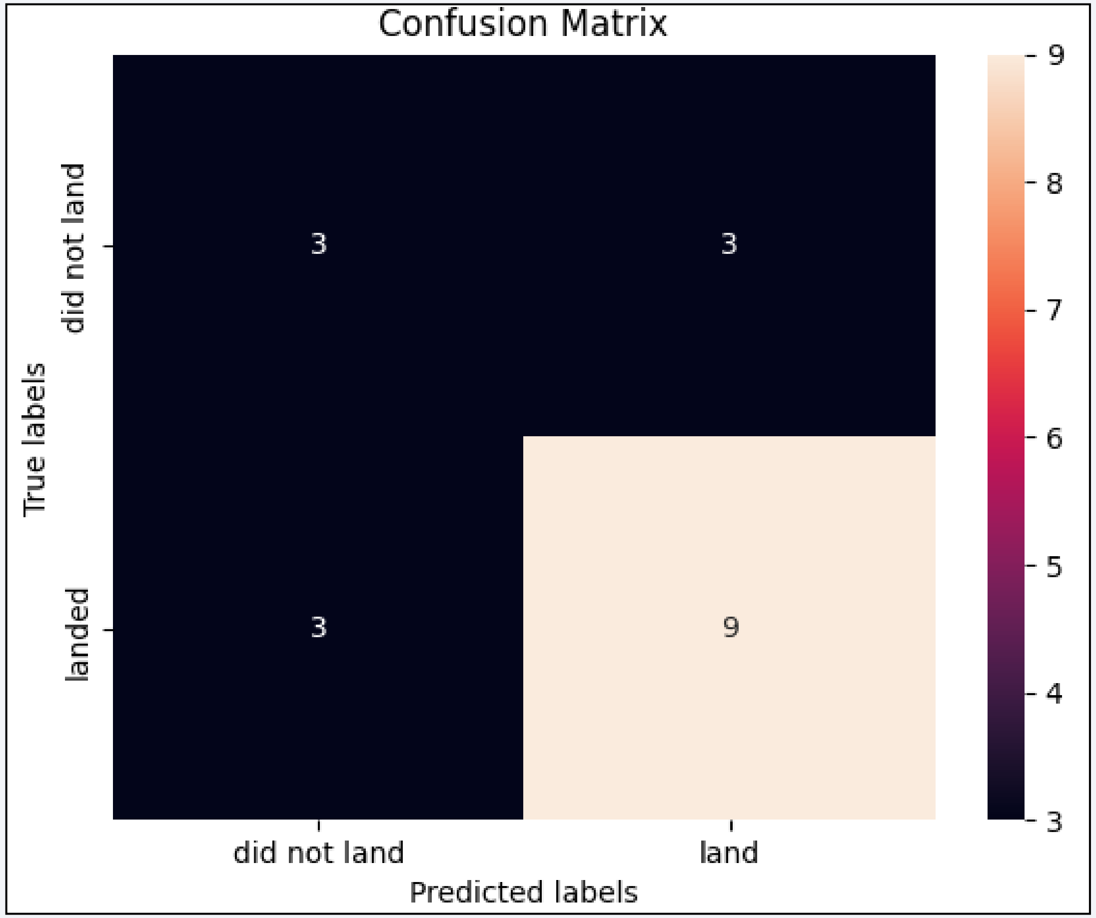
Results - Interactive Analytics Demo





Results – Predictive Analysis Classification

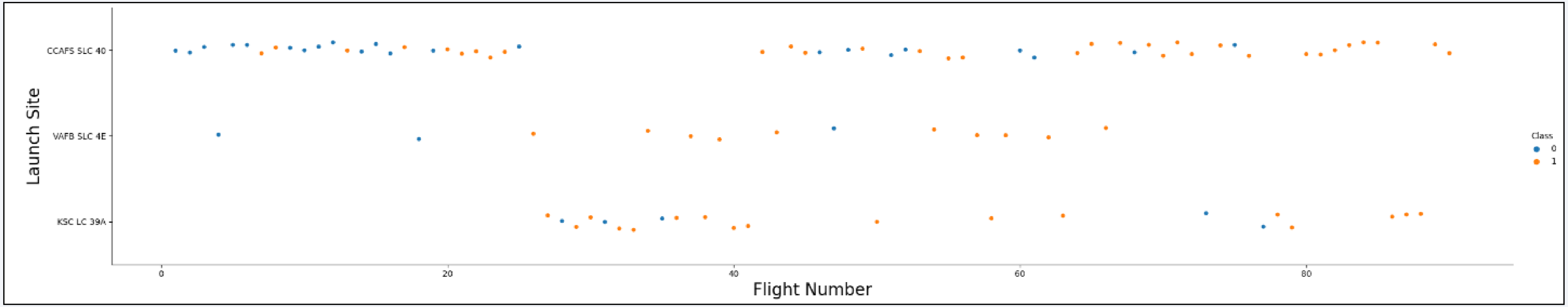




Section 2

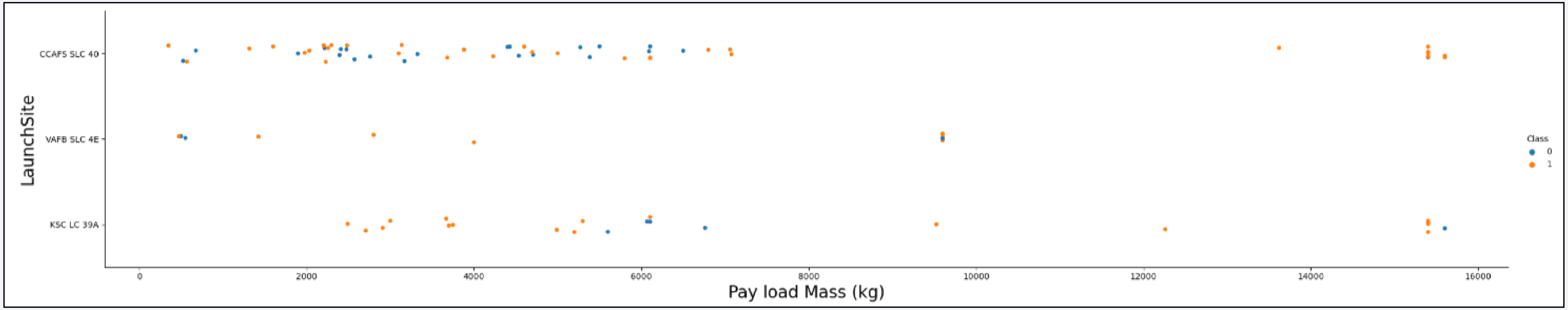
Insights drawn from EDA

Flight Number vs. Launch Site



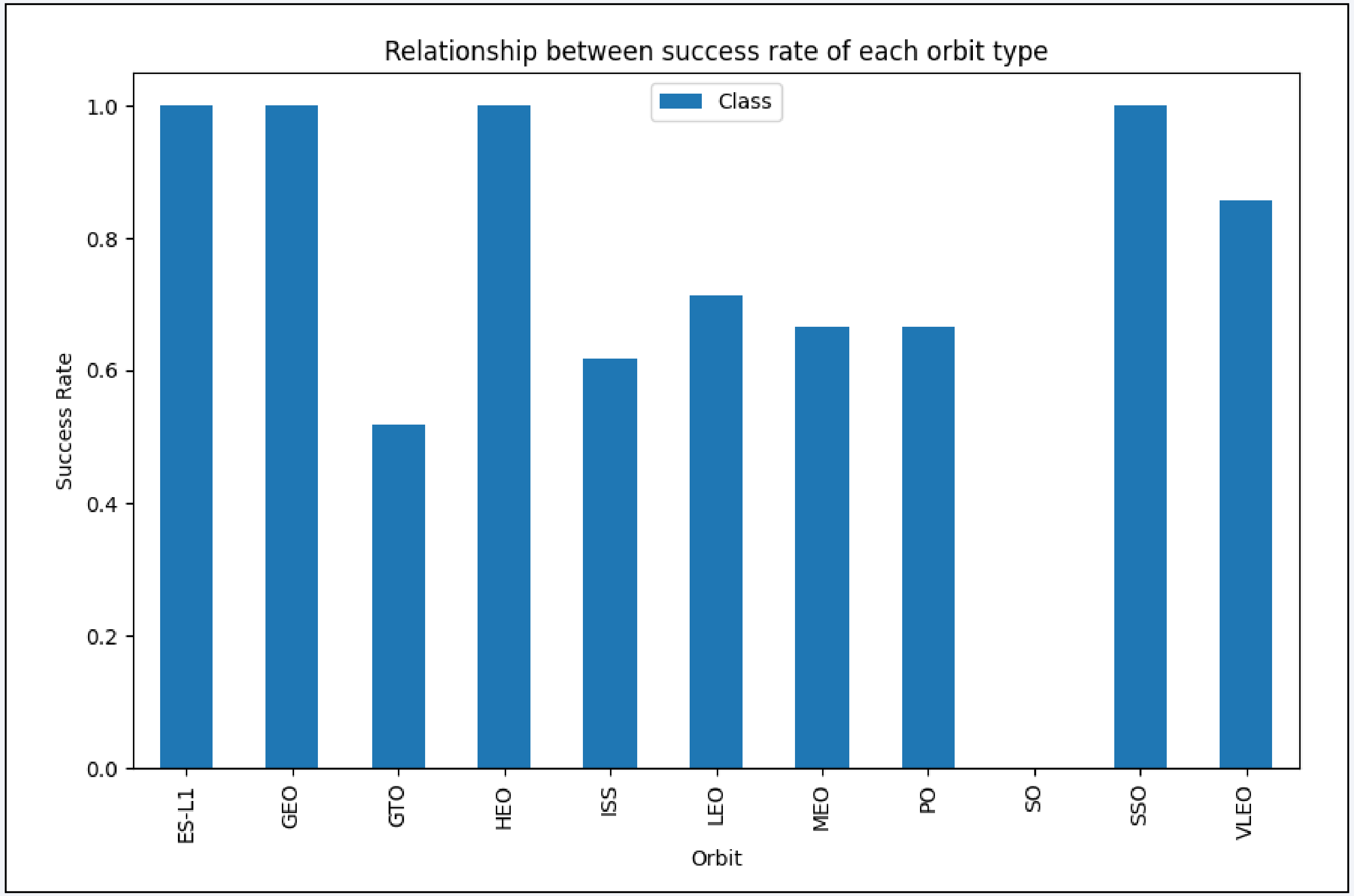
- Above chart shows majority of the launches occurred at launch site CCAFS SLC 40
- About 25% of the launches failed from flight numbers 0 – 25 at launch site CCAFS SLC 40
- Chart shows that adjustments were made to flight numbers 25 – 41 at other launch sites KSC LC 39A and VAFB SLC 4E
- Adjustments were applied and improvements are evident from flight numbers 42 – 90 at launch site CCAFS SLC 40 with the increased success rate of landing outcomes
- KSC LC 39A had the most successful landing outcomes at about 77%

Payload vs. Launch Site



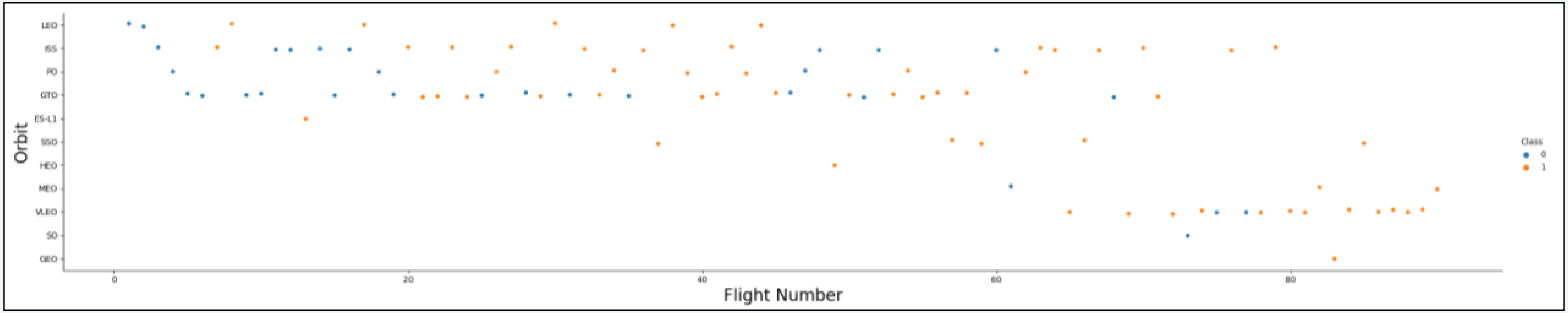
- There is no direct relationship between the launch site and payload mass. However, the most successful landing outcomes occurred between payload mass 2500 – 5000kg at launch site KSC LC 39A.
- Payload mass of about 15500kg resulted to successful landing outcomes at two different launch sites CCAFS SLC 40 and KSC LC 39A.
- There are no rockets launched for heavy payload mass (greater than 10000) at VAFB-SLC launchsite

Success Rate vs. Orbit Type



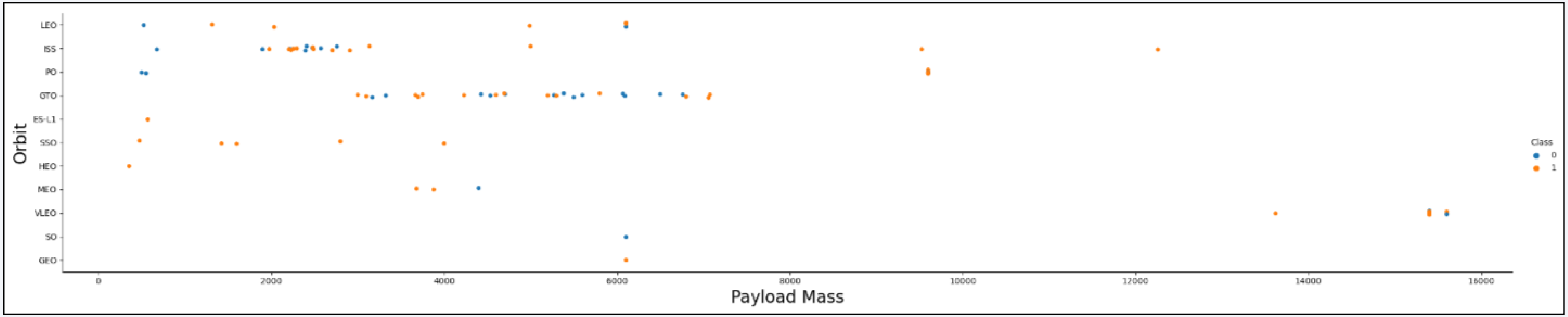
- Bar Chart indicates the most successful landing outcomes occurs at the MEO, GEO, SSO and ES-L1 orbits which totals 10 launch records combined.
- However, at the GTO or ISS orbits, 50 – 60% of the launch outcomes were successful. This sums about 13 out of 27 launch records for the GTO and 21 records for the ISS.
- Hence, while there is no direct relationship between success rates and orbit types, further investigation should focus on launches aimed at the GTO or ISS orbits as their launch ration to success rate is better compared to other orbits.

Flight Number vs. Orbit Type



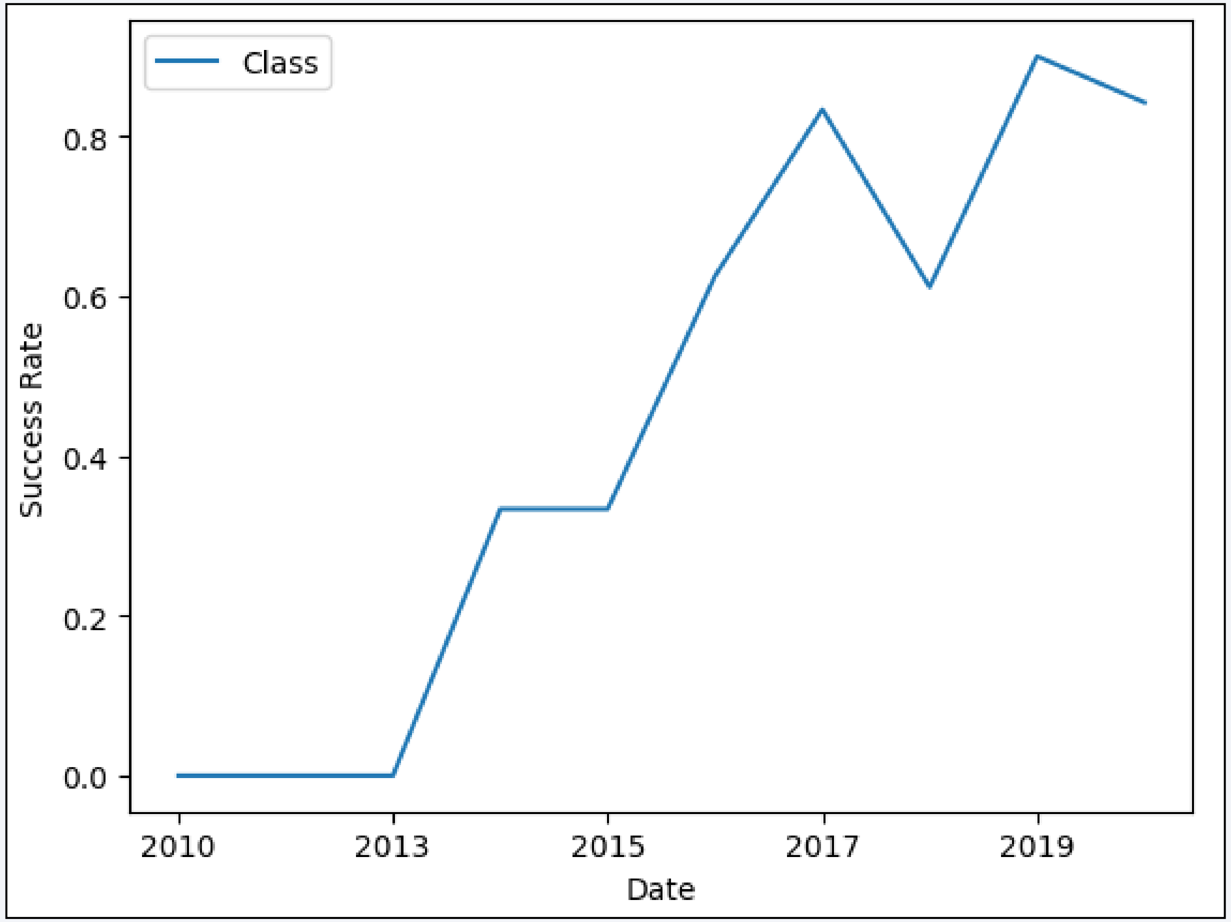
- LEO orbit success rate appears to get better with increasing number of flights
- On the other hand, there seems to be no relationship between flight number and GTO orbit.
- VLEO orbit displays the best success rate to orbit type for launches greater than 10 per orbit

Payload vs. Orbit Type



- For SSO,LEO and ISS orbits, it shows that the success rates improves with increasing payload mass.
- However the above relationship cannot be said for the GTO orbit as we cannot distinguish this as both positive landing rate and negative landing are both there present.
- Maximum payload mass used for the SSS orbit is 4000kg.

Launch Success Yearly Trend



- There was no successful landing outcome recorded until 2013.
- Sucess rate since 2013 kept increasing till 2020.

All Launch Site Names

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL
```

- The above SQL query is asking the database to return all unique/distinct values found in the "Launch_Site" column of the "SPACEXTBL" table.
- The result displayed is a list of all unique launch site values from the specified table column.

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

Launch Site Names Begin with 'CCA'

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

- The above SQL query is asking the database to retrieve up to 5 rows (**LIMIT**) from the table "SPACEXTBL" **WHERE** the "Launch_Site" column contains the substring **"CCA".**

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer LIKE '%(CRS)%'
```

- Above line of code executes an SQL query is selecting the sum of the "PAYLOAD_MASS__KG_" column **FROM** the table "SPACEXTBL" **WHERE** the "Customer" column contains the substring "(CRS)“.
- The result of the query, in this case, is a single value (the sum of the payload masses):

SUM(PAYLOAD_MASS__KG_)
48213.0

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1'
```

- The above SQL query calculates the average value of the "PAYLOAD_MASS__KG_" column **FROM** the "SPACEXTBL" table for rows **WHERE** the "Booster_Version" column is equal to 'F9 v1.1'.
- The result is the average payload mass for the specified condition.

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- The SQL query finds the first date **FROM** the "DATE" column in the "SPACEXTBL" table for rows **WHERE** the "Landing_Outcome" column contains the substring "ground pad".
- The result is the first date associated with landings on ground pads according to the specified condition.

```
%sql SELECT DATE FROM SPACEXTBL WHERE Landing_Outcome LIKE '%ground pad%' LIMIT 1
```

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

Date
22/12/2015

Successful Drone Ship Landing with Payload between 4000 and 6000

- The SQL query retrieves the "Booster_Version" values **FROM** the "SPACEXTBL" table for rows **WHERE** the payload mass is **BETWEEN** 4000 **AND** 6000 (inclusive) **AND** the landing outcome is 'Success (drone ship)'.

- The result is the booster versions associated with successful landings on drone ships within the specified payload mass range.

```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND Landing_Outcome = 'Success (drone ship)'
```

```
 * 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 SQL query **COUNTS** the number of occurrences of "Mission_Outcome" in the "SPACEXTBL" table for rows **WHERE** the "Mission_Outcome" column contains the substring 'Success' **OR** the substring 'Failure'.

- The result is the count of mission outcomes that meet the specified conditions.

```
%sql SELECT COUNT(Mission_Outcome) FROM SPACEXTBL WHERE Mission_Outcome LIKE '%Success%' OR Mission_Outcome LIKE '%Failure%'
```

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

Done.
```

COUNT(Mission_Outcome)
101

Boosters Carried Maximum Payload

```
%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

- The SQL query retrieves the "Booster_Version" values **FROM** the "SPACEXTBL" table for rows **WHERE** the payload mass is equal to the maximum payload mass value in the same table.
- The result is the booster versions associated with the rows having the maximum payload mass.

2015 Launch Records

- The SQL query retrieves the substring of "Date" (**FROM** index 4, spanning 2 characters), "Booster_Version", "Launch_Site", and "Landing_Outcome" values **FROM** the "SPACEXTBL" table for rows **WHERE** the "Landing_Outcome" column contains the substring 'Failure (drone ship)' **AND** the substring of "Date" (from index 7, spanning 4 characters) is equal to '2015'.
- The result will include specific columns associated with the rows meeting the specified conditions.

```
%sql SELECT substr(Date, 4, 2), Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTBL WHERE Landing_Outcome LIKE \
'%Failure (drone ship)%' AND substr(Date,7,4)='2015'
```

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

Done.
```

substr(Date, 4, 2)	Booster_Version	Launch_Site	Landing_Outcome
10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

```
%sql SELECT Landing_Outcome, COUNT(*) AS COUNT_LAUNCHES FROM SPACEXTBL WHERE DATE BETWEEN '06/04/2010' AND '20/03/2017' \
GROUP BY Landing_Outcome ORDER BY COUNT_LAUNCHES DESC
```

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

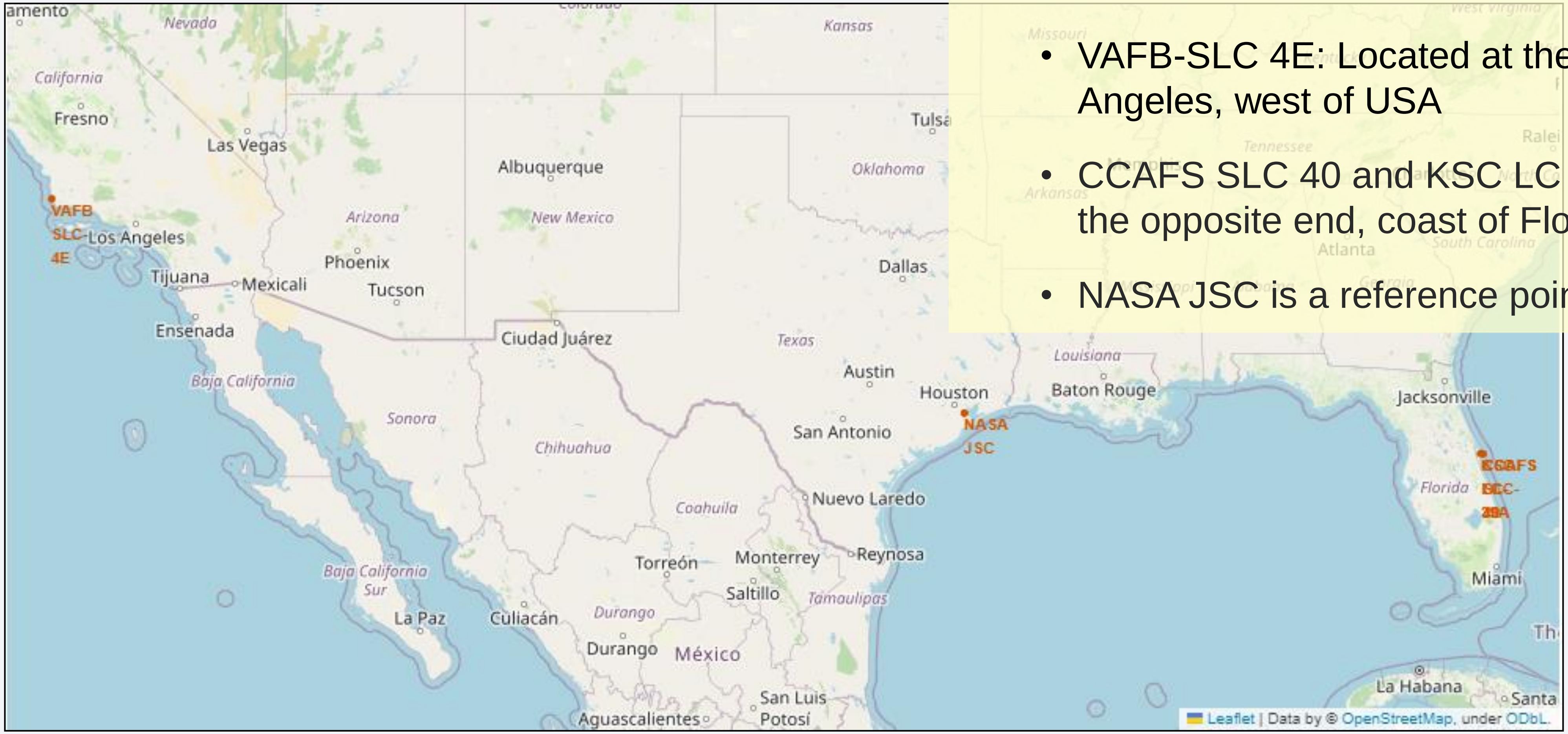
Landing_Outcome	COUNT_LAUNCHES
Success	19
No attempt	9
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	3
Failure	3
Failure (parachute)	2
Controlled (ocean)	2
No attempt	1

- The SQL query retrieves the "DATE" column and the count of "Landing_Outcome" occurrences **FROM** the "SPACEXTBL" table for rows **WHERE** the "DATE" column has values **BETWEEN** '2010-06-04' **AND** '2017-03-20‘.
- The result is then **GROUPED** by " Landing_Outcome " **AND** ordered by the **COUNT** of " Landing_Outcome " occurrences in **DESC** (descending) order.

Section 3

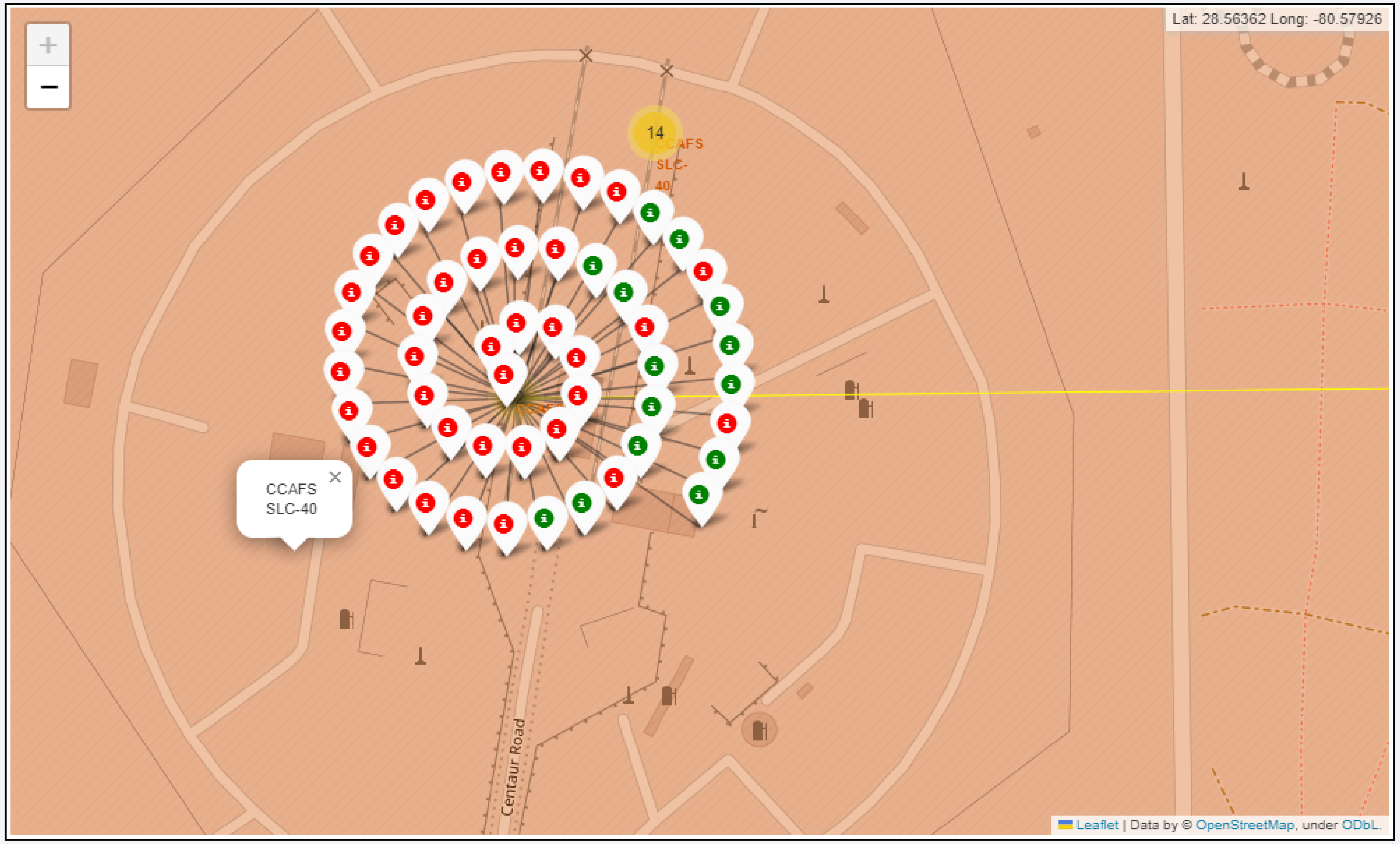
Launch Sites Proximities Analysis

Location of Launch Sites



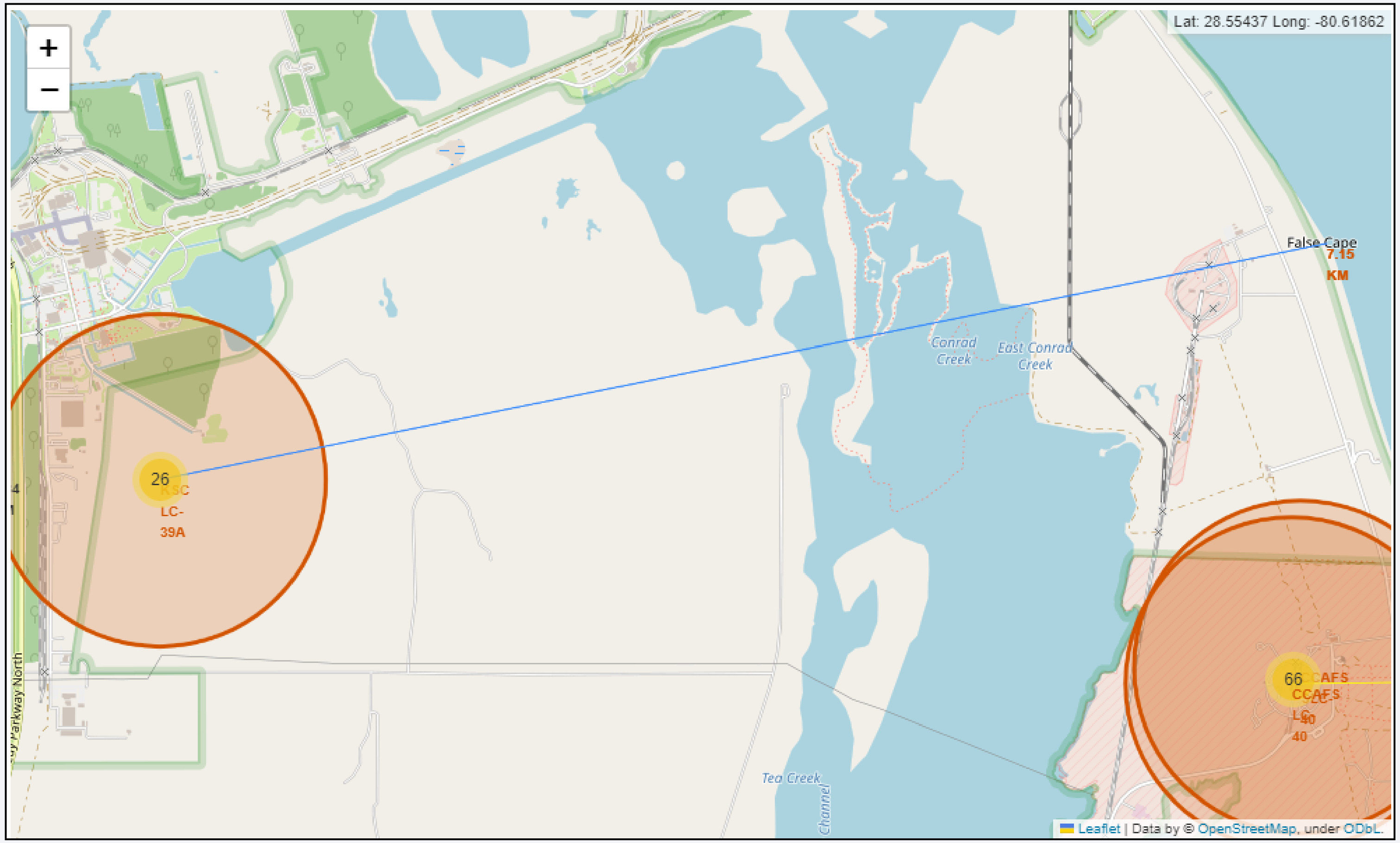
- Screenshot shows the four distinct launch sites
  - VAFB-SLC 4E: Located at the coast of Los Angeles, west of USA
  - CCAFS SLC 40 and KSC LC 39A are located at the opposite end, coast of Florida, east of USA.
  - NASA JSC is a reference point on the map.

Landing outcomes from Launch Site



- Map showing color coded landing outcomes of a particular launch site CCAFS SLC-40
- Green = Successful landing
- Red = Unsuccessful landing
- Out of 52 launches the site recorded, only 14 landed successfully.

Proximity of Launch Site to Landmarks

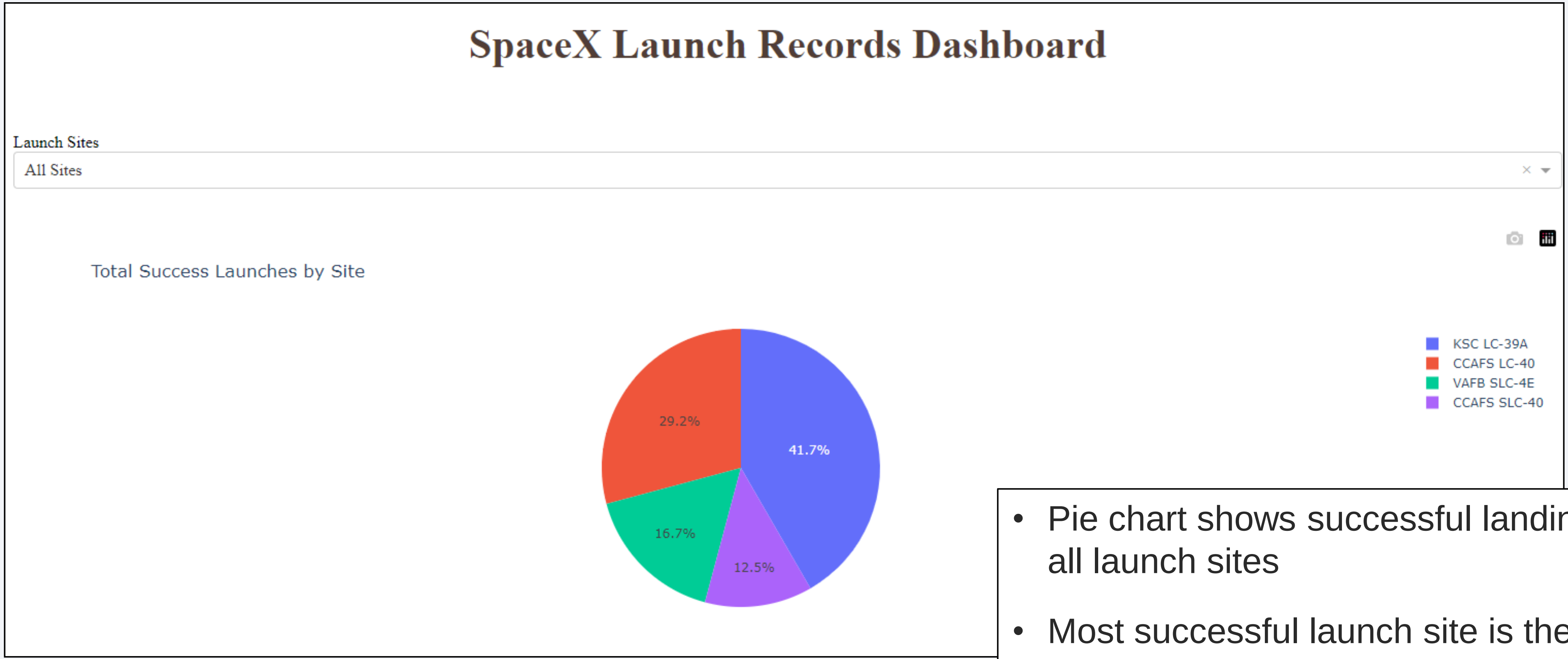


- Map is showing calculated distance from launch site KSC LC-39A to the coast at False Cape. Distance recorded on the map is 7.15km.
- Other recorded distances indicates that the Launch sites are in close proximities to the coast, highways and railways
- However they are kept at a significant distance away from the cities.

Section 4

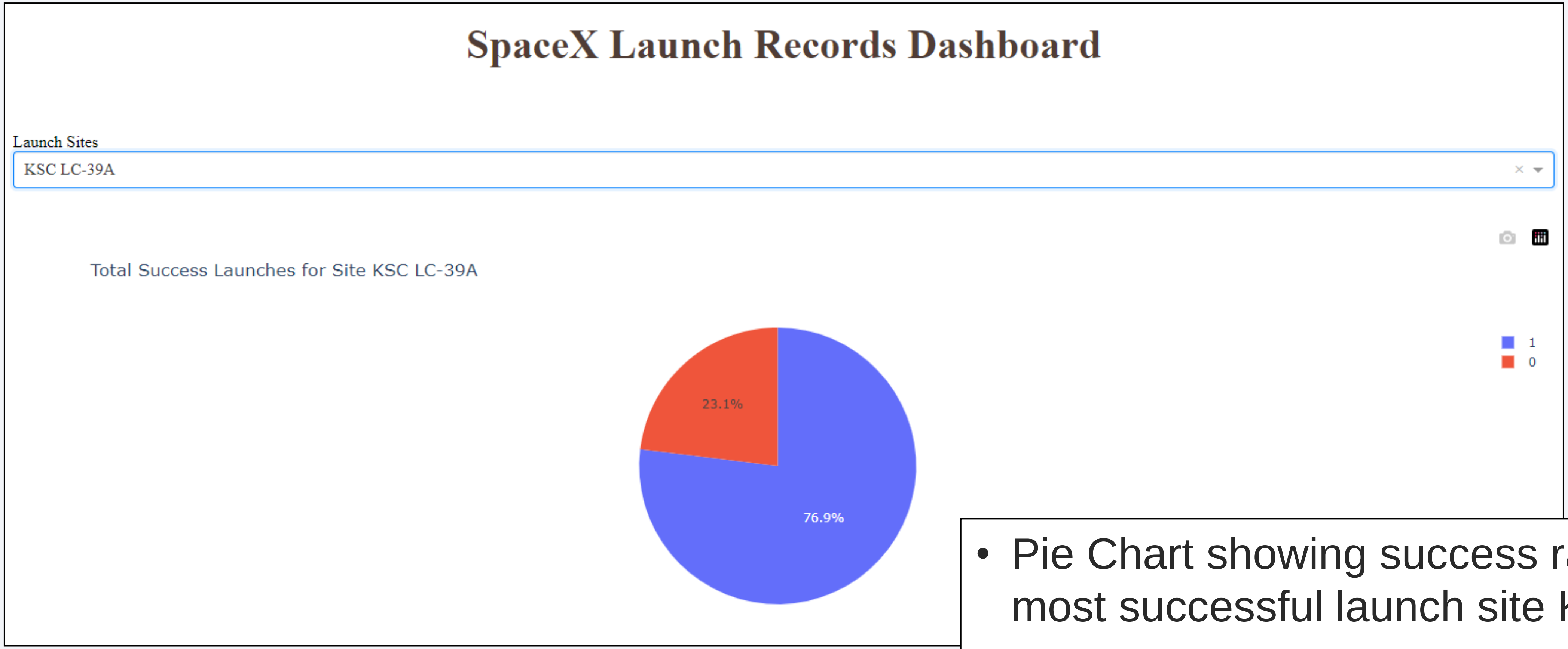
Build a Dashboard with Plotly Dash

Landing Success for Launch Sites



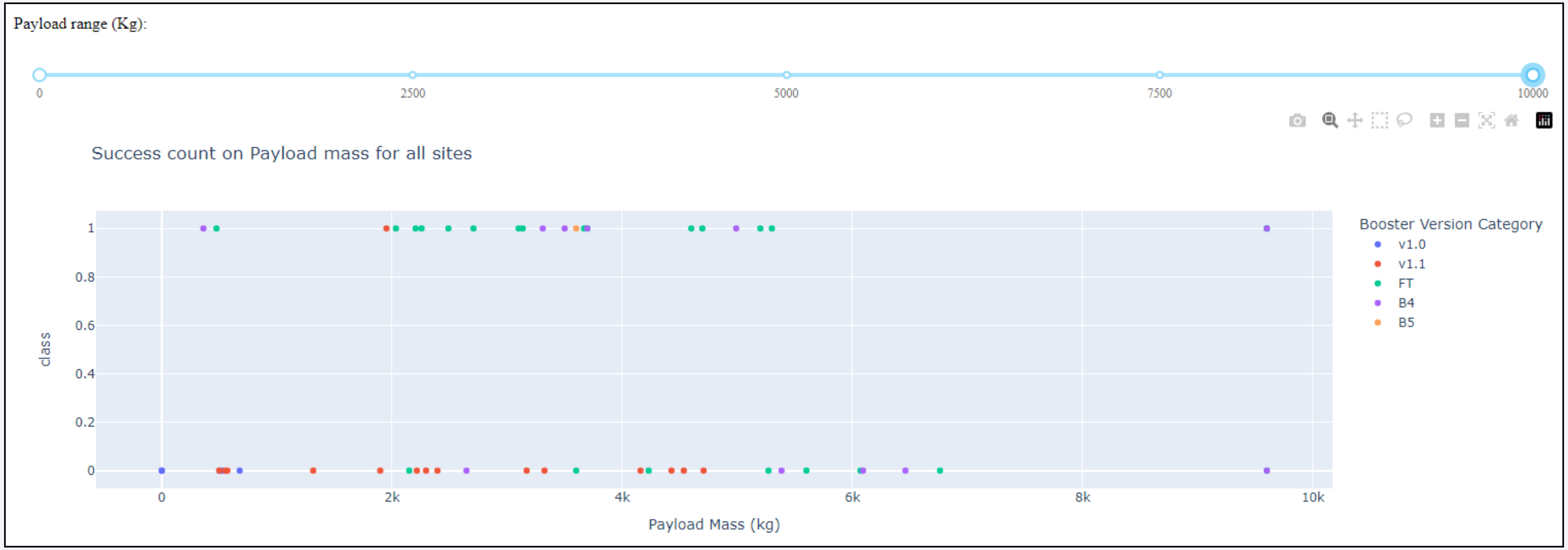
- Pie chart shows successful landing outcomes of all launch sites
- Most successful launch site is the KSC LC-39A with 41.7% success.
- Least successful is CCAFS SLC-40 with 12.5% success.

Success rate of Launch site KSC LC-39A



- Pie Chart showing success rate for the most successful launch site KSC LC-39A.
- 76.9% of landing outcome was successful compared to the 23.1% unsuccessful landings.

Payload mass vs. Launch Outcome

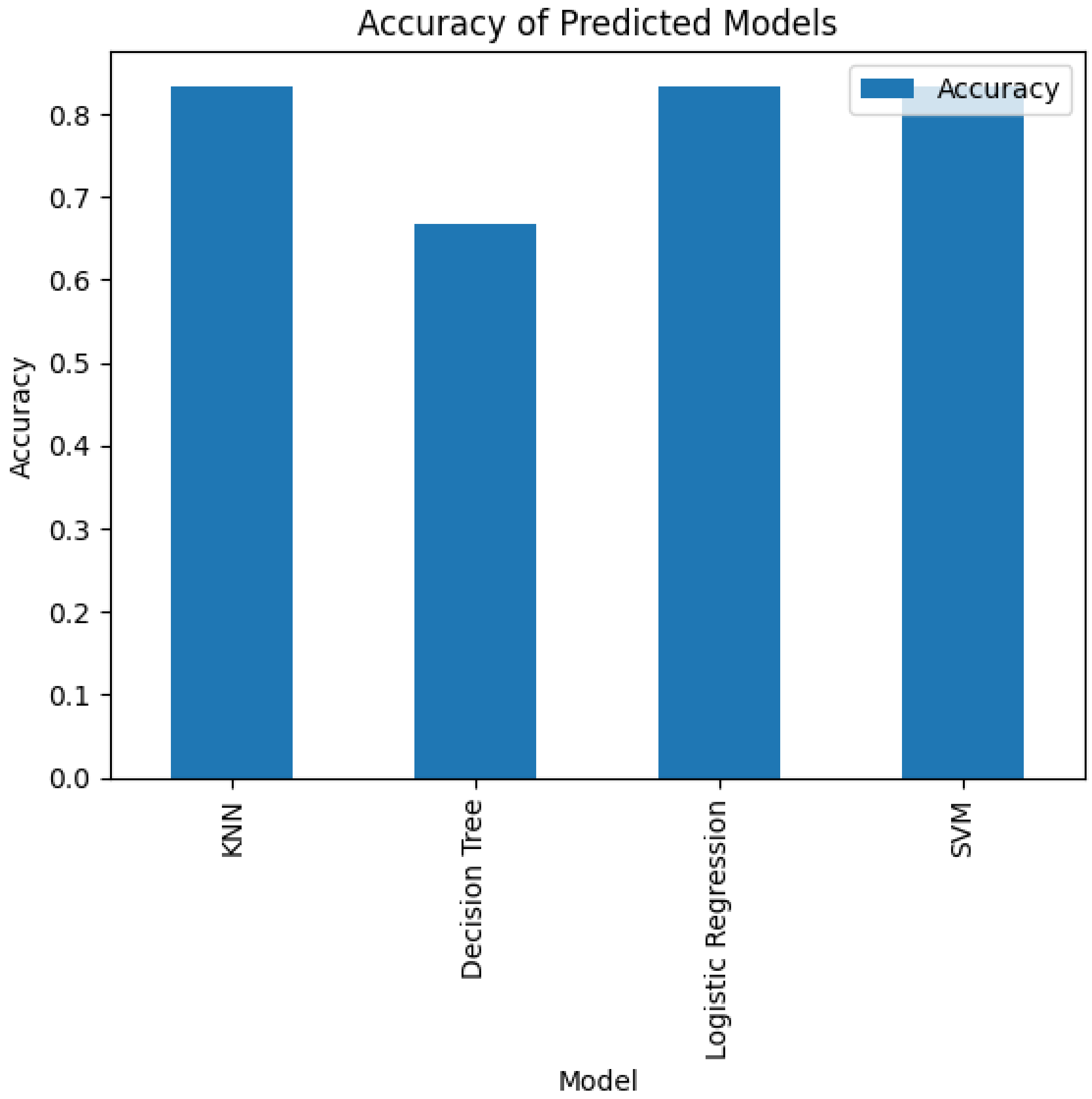


- Scatter Plot showing screenshot of Payload mass vs. Launch Outcome for all sites with payload range 0 – 10,000kg.
- Most successful landing outcome across all sites occurred with payload ranges 2,000 – 4,000kg, with FT being the most prominent Booster version category.
- Payloads with Booster Version Category v1.1 possesses landing outcomes with the least success rate.

Section 5

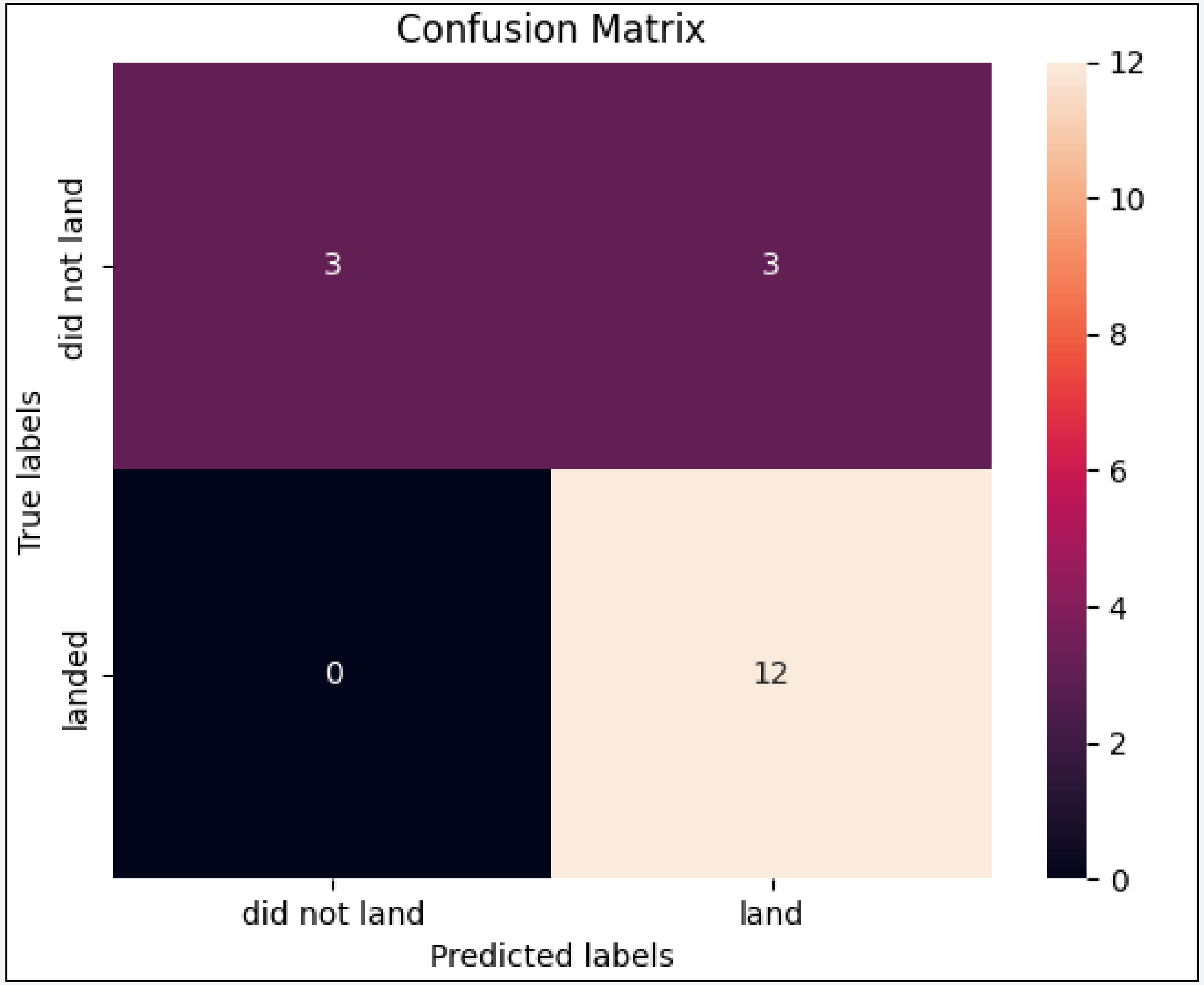
Predictive Analysis (Classification)

Classification Accuracy



- Bar Chart showing the classification accuracies of the predicted models
- Three models including KNN, Logistic Regression and SVM all have similar classification accuracy at 0.83
- Decision Tree has the least accuracy at 0.67

Confusion Matrix



- Image showing the confusion matrix of the Logistic Regression model with classification accuracy of 0.83
- Out of 6 launches that did not actually land, the model predicted 3 successfully landed, and 3 unsuccessful.
- The model correctly predicted the 12 successful landing outcomes that actually landed.

Conclusions

- The project successfully developed a predictive model to determine the successful landing of SpaceX's Falcon 9 first stage.
- Gathered dataset was subjected to exploratory data analysis, feature engineering, and interactive visual analytics.
- Analysis shows that success rate of landing outcomes improved significantly over the years from 2013 till 2021.
- The most successful launch site is the KSC LC-39A with 41.7% success, with the least successful being CCAFS SLC-40 with 12.5% success.
- It was observed that the Launch sites were built in close proximities to the coast, highways and railways. However they are kept at a significant distance away from the cities.
- Relationships between features provided in the data were established, which helped in the classification of landing outcomes.
- Amongst the classification models implemented, three models including KNN, Logistic Regression and SVM produced the best classification accuracy at 0.83, predicting correctly the 12 landing outcomes that successfully landed.

Thank you!