IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dev Patel
07/30/2025

Outline

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

Executive Summary

SpaceX, a competitor of SpaceY is known for launching many successful space missions and at a lower cost as it can reuse the first stage of rocket launch. Therefore, if we can determine whether the first stage will land or not, we can also determine the cost of the launch. To determine this, we collected necessary data from the SpaceX REST API with the help of get requests method. We used the features Rocket, Payload, Launchpad and Core to extract the necessary features for our project. We created a pandas dataframe from the cleaned data obtained from extracted features. During Data wrangling we created a new 'class' column to represent the categorical launch outcomes in the form of 1s and 0s, where 1 represents success and 0 represents the otherwise. In Exploratory Data Analysis we used inline SQL queries to extract insights from the data. With the help of scatter and bar plots we visualized the relationships between different features. We then plotted the launch sites based on successful on folium maps and created a dashboard to view the annual charts for each launch site. Finally, the data was then used to train Logistic Regression, SVM and KNN models, with class as the target variable. Based of the predicted and actual values, a confusion matrix for each model was created to see which one performed better. Score method was used to compare model's accuracy.

Initially the data obtained from the API was in JSON format. We cleaned the data using HTML parser. After that we found null values in the feature Payload Mass, which were replaced with the mean values. We kept the nulls in landing pad as it represented the events when the pad wasn't used. In the Data Wrangling part, filtered the dataframe to include only Falcon 9 entries. We also created a class label for the launch outcomes that was classified into 6 different categories. In EDA we found that for medium range payloads, booster version B1 had highest success rate in drone ships. Whereas booster version B5 is able to carry maximum payload mass. Visually, we found a relationship between payload mass and high number of successful flights. We also found a relationship between launch site LC-40 and total number of successful flights. For visual analytics, we plotted the launch sites in folium maps with marker clusters representing the launching sites. We calculated the distance between the site 39A, and drew a marker line, visually representing the distance. Finally, we created an interactive dashboard displaying a pie chart of success rate of each launch site followed by a slider for adjusting the weight range for the scatter plot showing its relationship with payload mass, under it.

Introduction

The commercial space age is here! Companies are making space travel affordable for everyone. Virgin Galactic is providing suborbital spaceflights. Rocket Lab is a small satellite provider. Blue Origin manufactures sub-orbital and orbital reusable rockets. Perhaps the most successful is SpaceX. SpaceX's accomplishments include Sending spacecraft to the International Space Station, providing satellite Internet access through Starlink, a satellite internet constellation and, sending manned missions to Space. One reason SpaceX can do this is because the rocket launches are relatively inexpensive. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars while other providers cost upwards of 165 million dollars each. Much of the savings is because SpaceX can reuse the first stage of its rockets. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. Therefore, in this project we will predict if the Falcon 9 first stage will land successfully.

As a data scientist working for SpaceY company. I will be determining the price of each launch by gathering information about Space X and creating dashboards for the team. Then we will determine if SpaceX will reuse the first stage and then train a machine learning model to determine if first stage will land successfully.

Section 1

Methodology

Methodology

Data Collection

We used GET requests method to extract data from the SpaceX REST API. The extracted data was normalized using the pandas normalize feature and converted into a pandas dataframe. We extracted 17 variables from the API data. Some important variables extracted were, Flight Number, Booster version, Orbit, Launch pad, Outcome, Latitude and Longitude. We replaced the null values for Payload Mass with the mean value but kept the nulls for landing pad. Finally, a new pandas dataframe, filtered to include only Falcon 9 entries, was created with the selected variables and saved for the next step.

Data Wrangling

In this section we found that the Outcome column had 8 categorical outcome values, True RTLS, False RTLS, True ASDS, False ASDS, None ASDS, None, True Ocean and False Ocean. For these values, we created a separate column in the dataframe that showed the class labels for each entry where, 1 represented all ‘True’ outcomes and 0 represented False and None outcomes. The data is now ready for next step.

Methodology (contd.)

Exploratory Data Analysis through Visualization and SQL

The data obtained from the previous step is now loaded into the SQL database and queries were passed using python's inline magic commands. By utilizing sub queries, we were able to get valuable insights like max payload mass carried by rockets and counts of landing outcomes from the year 2010 through 2017. To visually analyze the relationships of variable Launch Site with Flight Number and Payload, we plotted scatter plots with the help of seaborn library from python. Similarly, we determined the relationship of the variable Orbit type with Flight Number and Payload. To visualize the launch success trend, we created a new variable that stored grouped counts of successful launches per year. The index and values of this new variable were used to create a line plot with the help of matplotlib library. Finally, we feature engineered 4 categorical columns, Orbits, Launch Site, Landing Pad, and Serial. With the help of OneHotEncoder we converted the categorical columns into numeric columns. The new columns will be replaced with the previous object type versions and then saved. These converted columns will be used for plotting on folium maps.

Visual analysis with Folium and Plotly Dash

In a folium map, we folium markers for all the launch sites in our data, with an additional marker on NASA Jhonson Space Center, as the center point of the map. We created 1000 kms radius circles with each launch site as a center point. With the help of a marker cluster object, we created markers for launch records of each launch site. We added the mouse pointer to the map and selected 4 random locations near launch site KSC LC-39A. We calculated the distance from launch site to each selected point and plotted lines, representing the distance.

Methodology (contd..)

Predictive Analysis with Classification Models

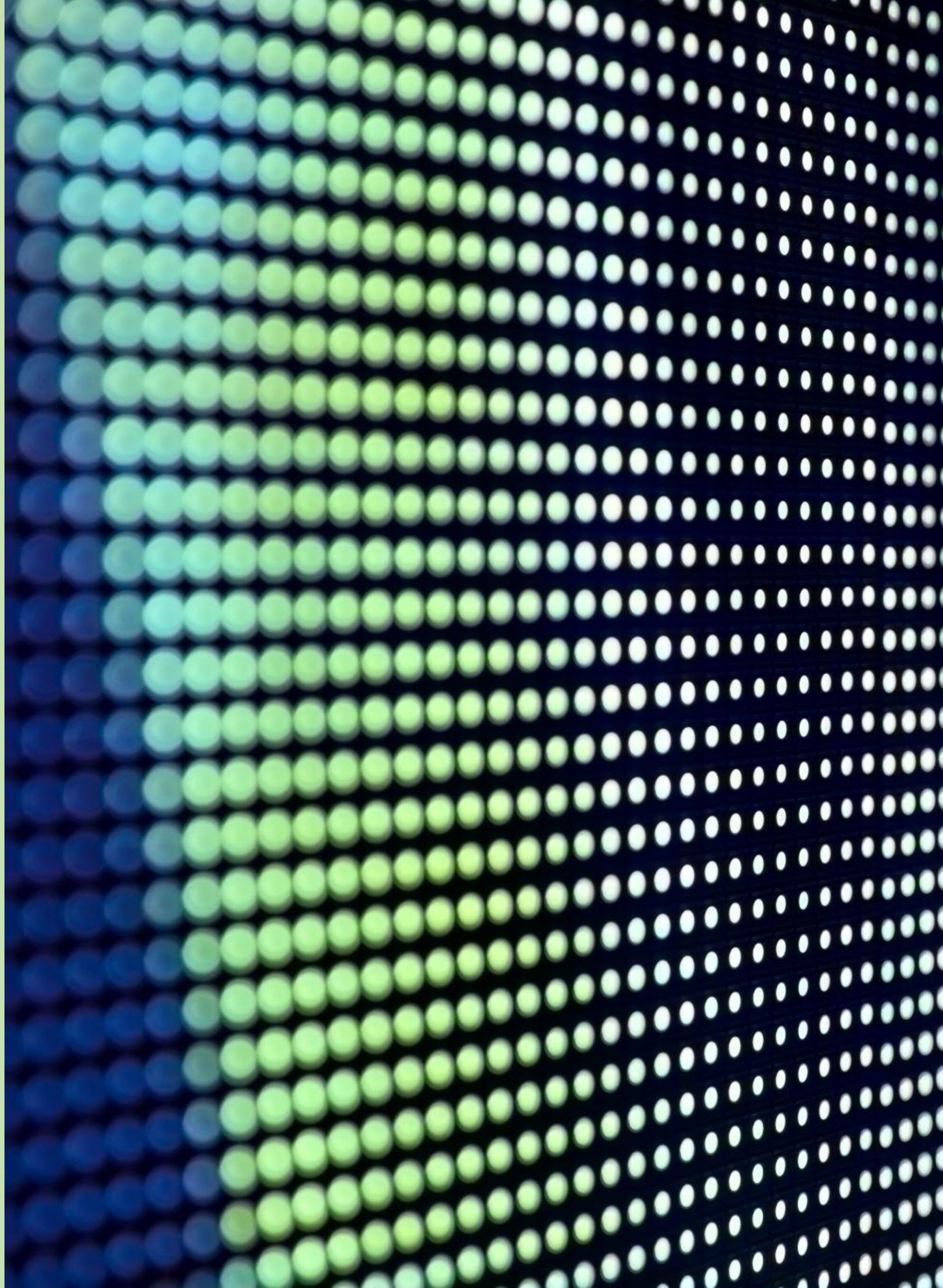
To build and train classification models, we used 2 datasets. First, we used the dataset obtained from data wrangling as our input variable. For the target variable, we used the 'class' columns from encoded data from EDA step, as it represents the result of launch outcome. Before splitting the data, we standardized the input variables using Standard Scalar. Now we split the input and target variables into training and testing sets with the help of 'train_test_split' function from scikit learn. The data was split such that 80% of it was used for testing and the remaining fraction for training. First, we trained the logistic regression model by using hyperparameter tuning and 10-fold cross validation method. Here, the data is split into 10 parts, of which the model is trained on 9 parts and validated on the 10th, rotating through all combinations. After the model was trained and predictions were made with '.predict' method, we determined the best parameters for the model using 'best_params_' method. Accuracy of the predictions made was known by 'best_score' method. Finally, we plot the confusion matric between the actual and predicted values to obtain insights from the trained model. Similarly, we trained SVM, Decision Tree and KNN-Classifier models.

Data Collection

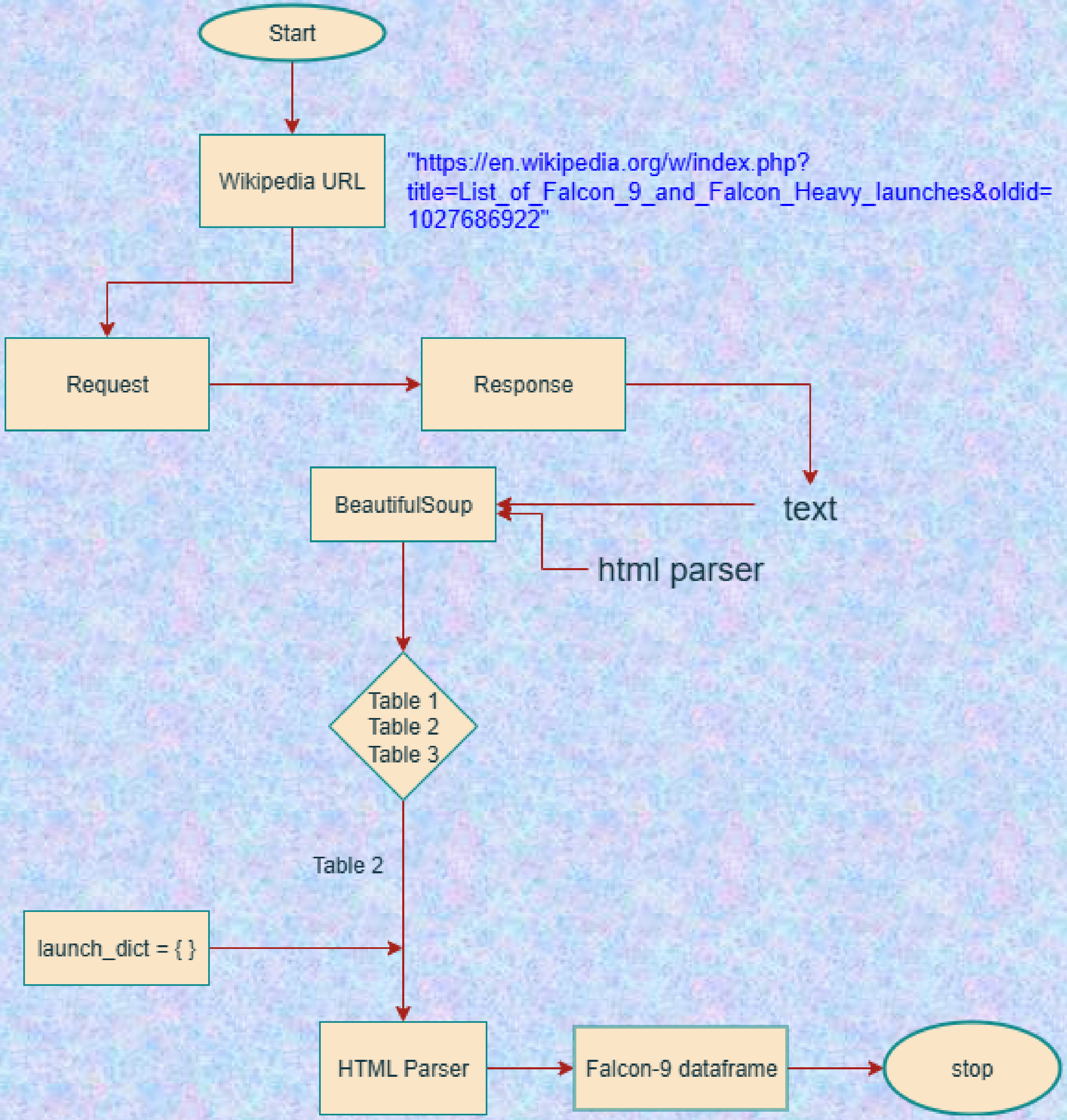
We collected data through 2 sources:

SPACEX REST API

WIKIPEDIA

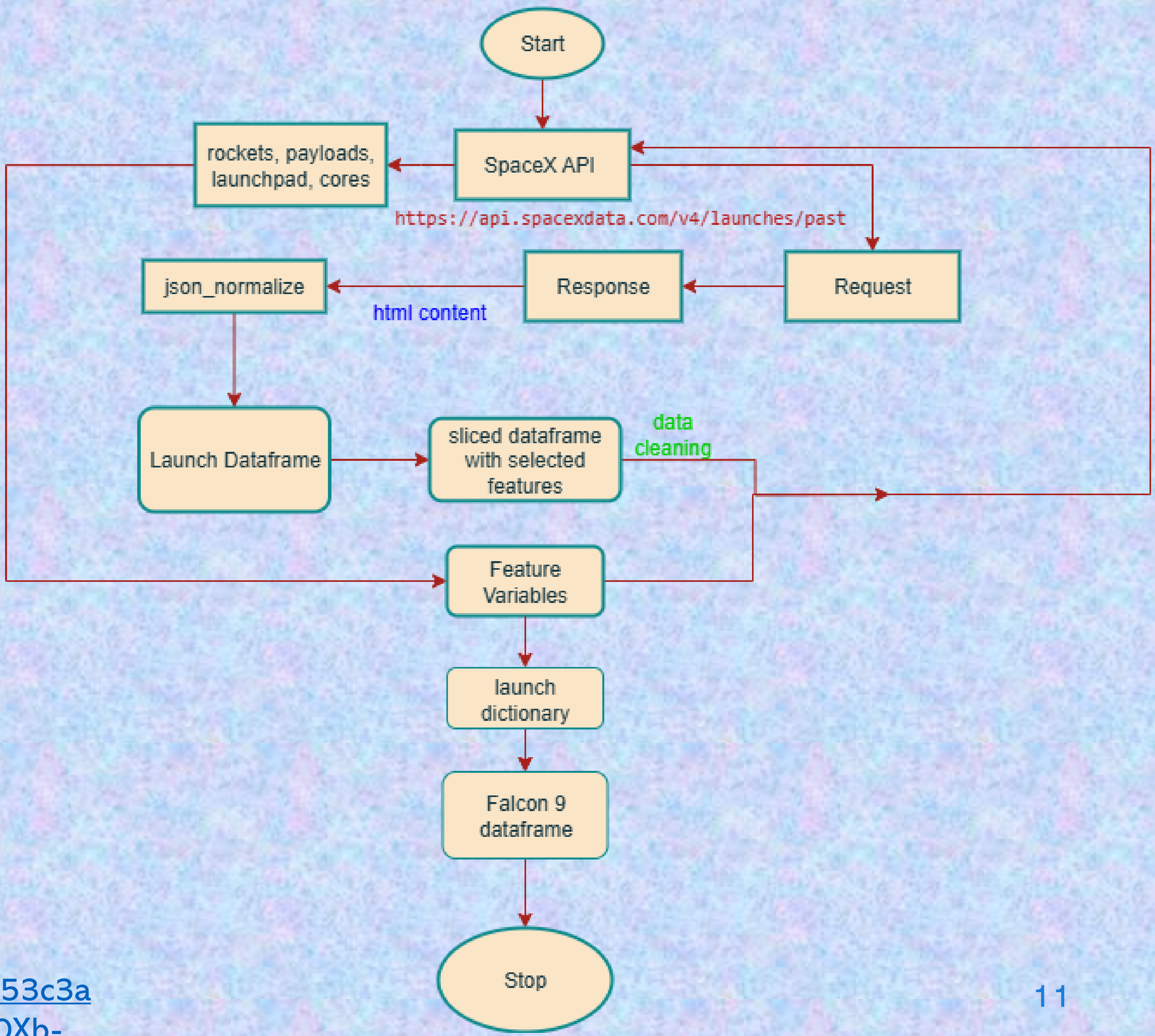


Web scrapping from Wikipedia



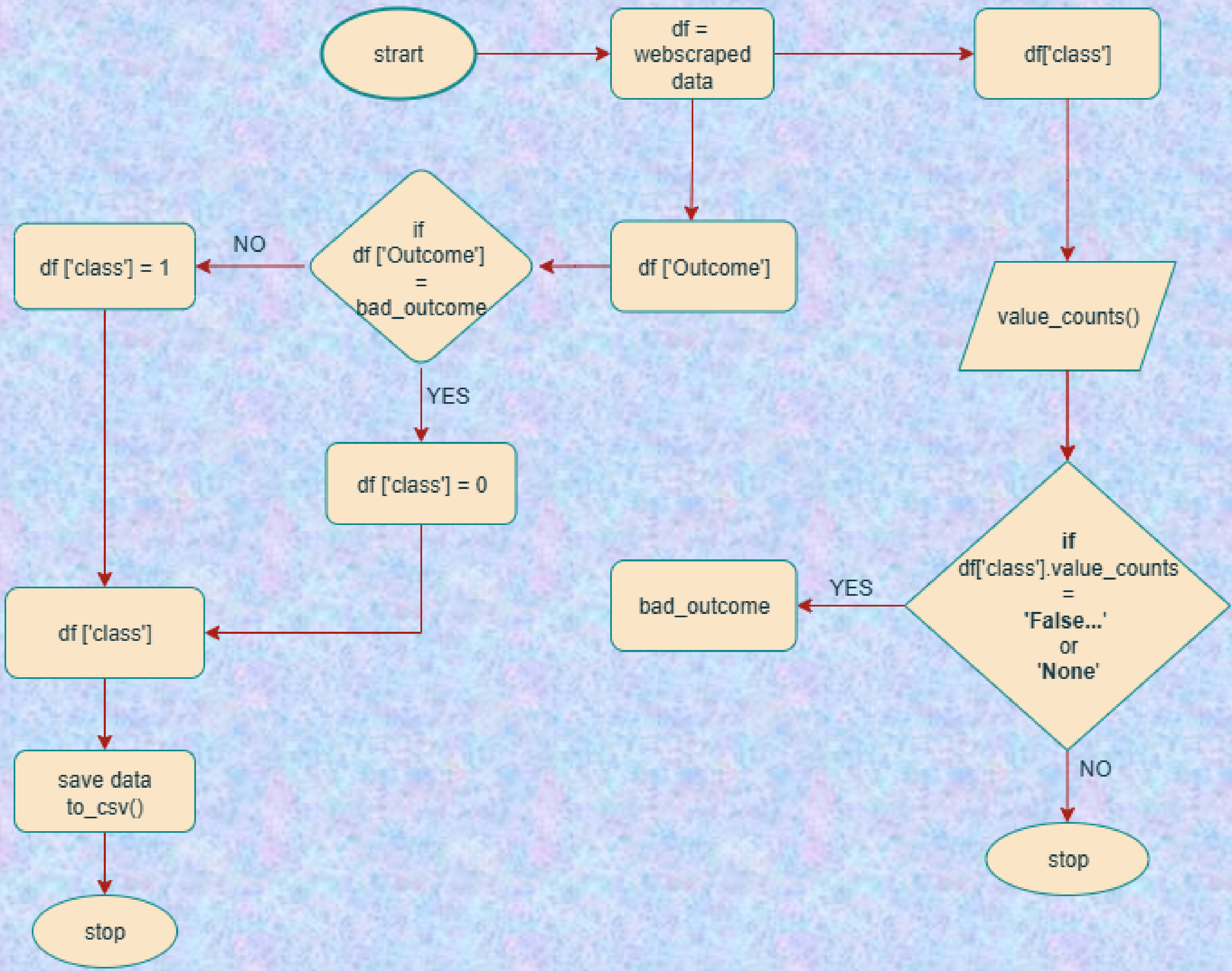
GitHub URL:
https://github.com/devpatel0415/DS0625/blob/be253c3a24230415275cfdbaa8cf5299442f9e35/Project%20Xa-Webscrapping%20HTML%20tables.ipynb

Web scrapping from SpaceX API



GitHub URL: https://github.com/devpatel0415/DS0625/blob/be253c3a24230415275cfdbaa8cf5299442f9e35/Project%20Xb-Webscrapping%20with%20an%20API.ipynb

Data Wrangling

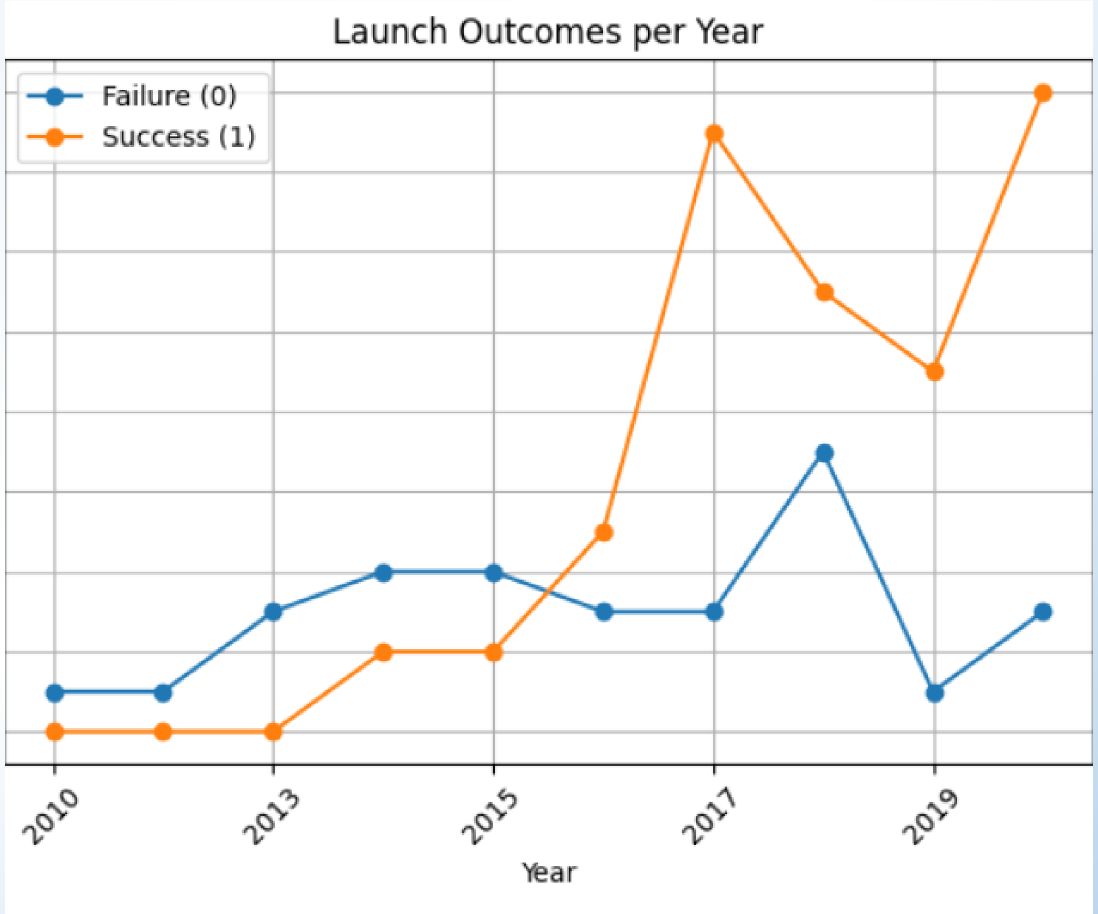


From the web scraped data, we replaced the nulls in Orbit column with the mean value and assigned class labels for landing outcomes such that success is 1 and failure is 0.

GitHub URL:

https://github.com/devpatel0415/DS0625/blob/be253c3a24230415275cfdbaa8cf5299442f9e35/Project%20Xc-%20Data%20Wrangling.ipynb

EDA with Data Visualization



- Scatter plots were used to analyze relations:
  - Payload mass vs Flight number
  - Launch site vs Flight number
  - Payload mass vs Launch site
  - Orbit type vs Flight number
  - Orbit type vs Payload mass
- We used bar plots to visualize the relationship between Orbit types and Launch success rate.
- Finally, we used a line plot to visualize yearly launch success trend
- GitHub URL: https://github.com/devpatel0415/DS0625/blob/be253c3a24230415275cfdbaa8cf5299442f9e35/Project%20Xe-EDA%20with%20Visualization.ipynb

EDA with SQL

From the SQL queries, we were able to derive the following insights:

The dataset consisted of 4 unique launch sites, CCAFS LC-40, VAFB SLC-4E, KSC LC-39A, and CCAFS SLC-40.

The total payload mass carried by NASA boosters was 45596 kgs.

The average payload mass carried by booster version 1.1 was about 2535 kgs.

First successful landing occurred on 4th June 2010.

Booter versions 1.1, FT, B4 and B5 have successfully landed drone ships with payload range of 4000 to 6000 kgs.

Booster version B5 has carried maximum payloads during launches.

GitHub URL:
https://github.com/devpatel0415/DS0625/blob/be253c3a24230415275cfdbaa8cf5299442f9e35/Project%20Xd-%20EDA%20.ipynb

Build an Interactive Map with Folium

In the folium map, we used map objects like markers, circles, marker clusters, mouse pointer and PolyLine to visualize launch data

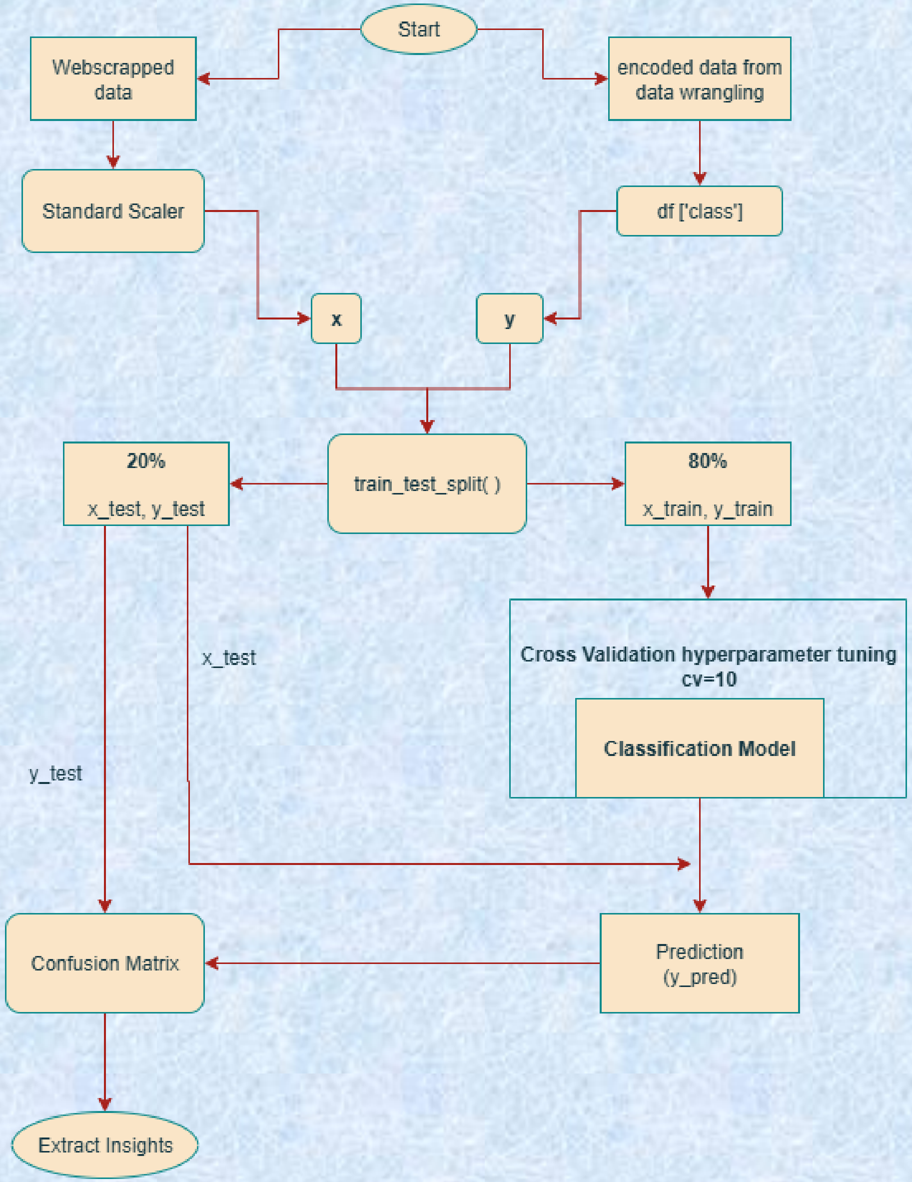
We used folium marker object to mark all the launch sites and added circles with markers as center points, to represent a proximity around the launch site. Now we needed to add markers for the launch outcome for each site. With the help of marker cluster object, we added launch outcome markers, which were red for failure and green for success. This enables all outcome markers to appear as a single interactive marker. With the help of mouse pointer on map, we located for proximities and noted their location coordinates. After calculating the distance of proximities from the launch site, we used the PolyLine to connect the location coordinates, to represent the distance of each proximity from the launch site LC-39A

GitHub URL:
https://github.com/devpatel0415/DS0625/blob/be253c3a24230415275cfdbaa8cf5299442f9e35/Project%20Xf-%20Visual%20analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- The dashboard consists of 4 blocks in total, 2 interactive and 2 visual. For the interactive blocks, we have a dropdown menu and the other interactive is a slider bar. The first visual provides an interactive pie chart and the second visual provides with interactive scatter plot
- The first block is the dropdown menu that allows the user to choose the launch site to view success rate. The default option is set to all sites. The next block is the interactive pie chart, that displays the success rate for the launch site selected earlier. Success is shaded in red, whereas failure in blue. The third block is the slider bar that can be adjusted from both ends. This allows the user to chose their desired payload mass range. Finally, we have the block that shows an interactive scatter plot between payload mass and success, for the payload range selected in the slider.
- **GitHub URL:** https://github.com/devpatel0415/DS0625/blob/be253c3a24230415275cfdbaa8cf5299442f9e35/ProjectXg.py

Predictive Analysis (Classification)



- To build and train classification models, we used 2 datasets. First, we used the dataset obtained from data wrangling as our input variable. For the target variable, we used the 'class' columns from encoded data from EDA step, as it represents the result of launch outcome. Before splitting the data, we standardized the input variables using Standard Scalar. Now we split the input and target variables into training and testing sets with the help of 'train_test_split' function from scikit learn. The data was split such that 80% of it was used for testing and the remaining fraction for training. First, we trained the logistic regression model by using hyperparameter tuning and 10-fold cross validation method. Here, the data is split into 10 parts, of which the model is trained on 9 parts and validated on the 10th, rotating through all combinations. After the model was trained and predictions were made with '.predict' method, we determined the best parameters for the model using 'best_params_' method. Accuracy of the predictions made was known by 'best_score' method. Finally, we plot the confusion matric between the actual and predicted values to obtain insights from the trained model. Similarly, we trained SVM, Decision Tree and KNN-Classifier models. To select the best model, we compared the accuracy score and confusion matrix and found that Logistic Regression was the best one.

- **GitHub URL:** https://github.com/devpatel0415/DS0625/blob/be253c3a24230415275cfdbaa8cf5299442f9e35/Project%20Xh-%20ML%20prediction.ipynb

Results

EXPLORATORY DATA ANALYSIS RESULTS

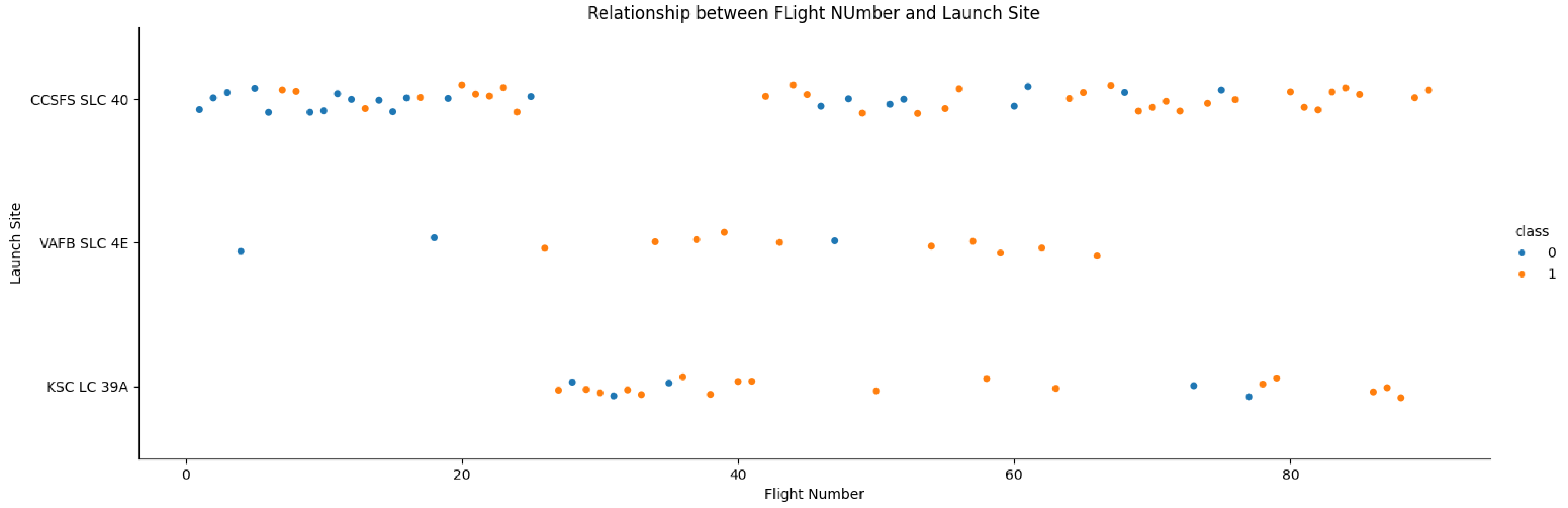
INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS

PREDICTIVE ANALYSIS RESULTS

Section 2

Insights drawn from EDA

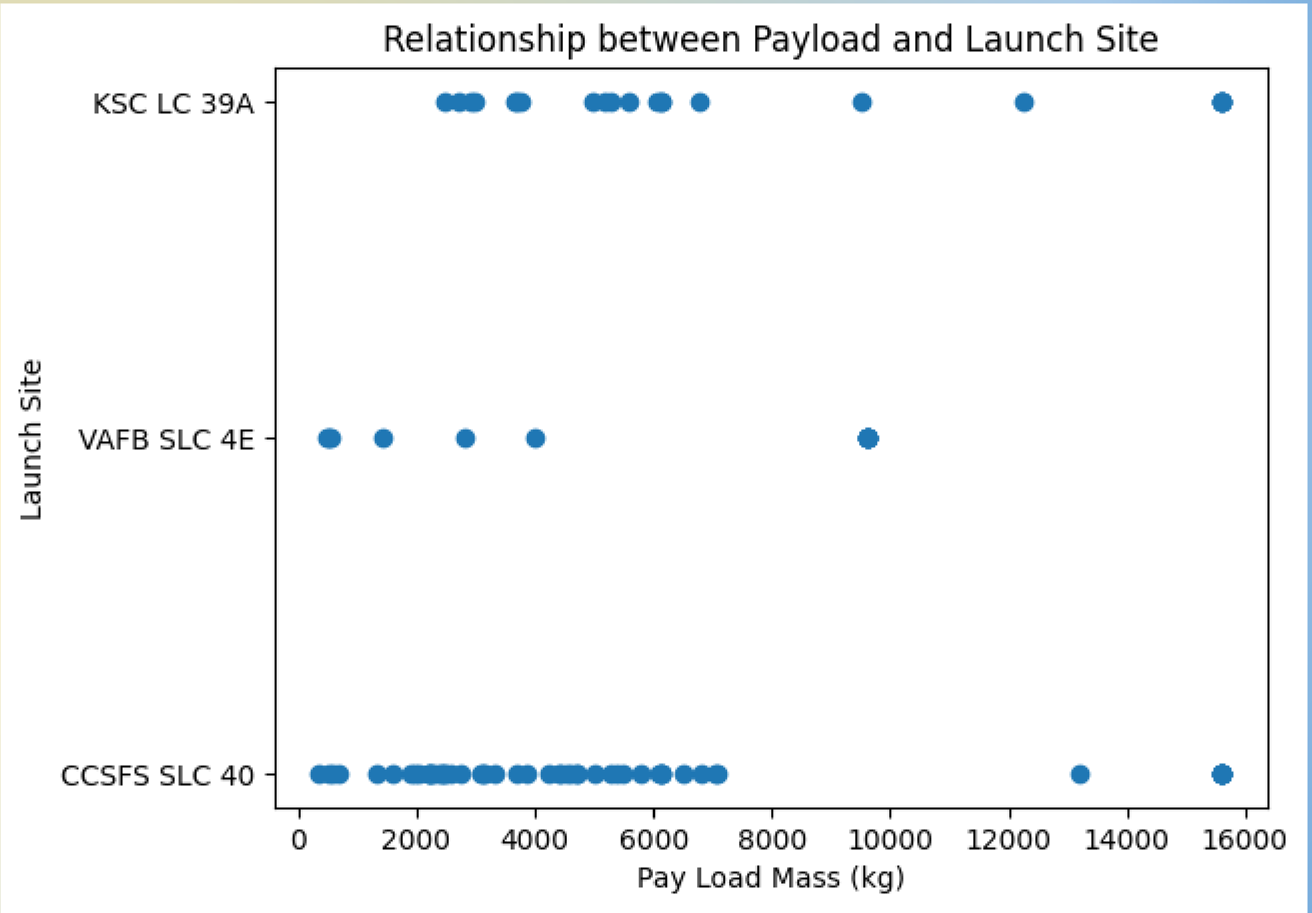
Flight Number vs. Launch Site



The red dots in the plot represent successful landings and blue dots represent failures. From the plot, we can say that launch site CCFS SLC-40 had most successful and unsuccessful landings. Launch site VAFB SLC 4E has a better success rate among all.

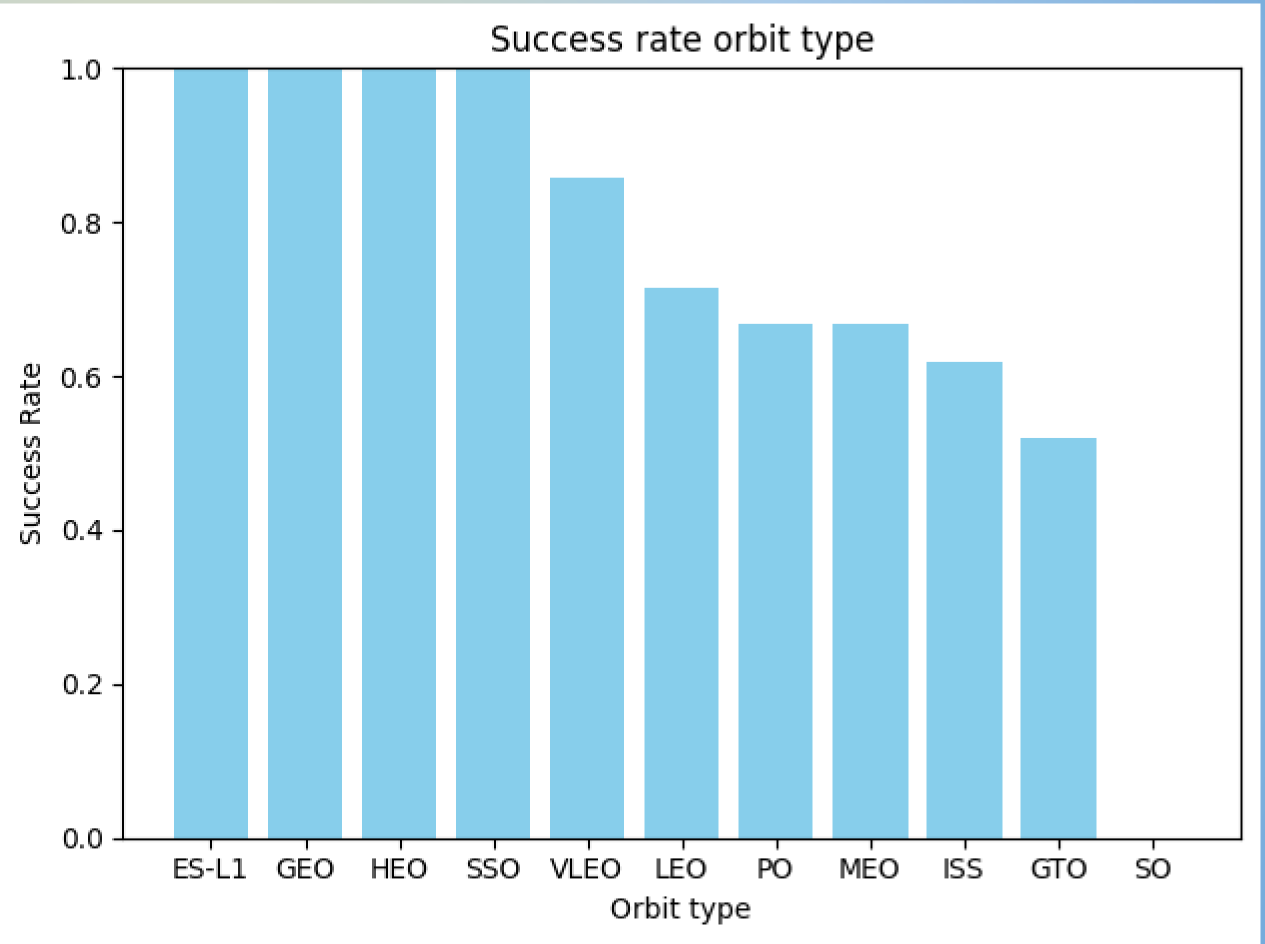
Payload vs. Launch Site

From the plot, we can say that launch site VAFB SLC 4E can carry low to medium payload ranges. Whereas KSC LC 39A and CCFS SLC 40 can carry low to high payloads.



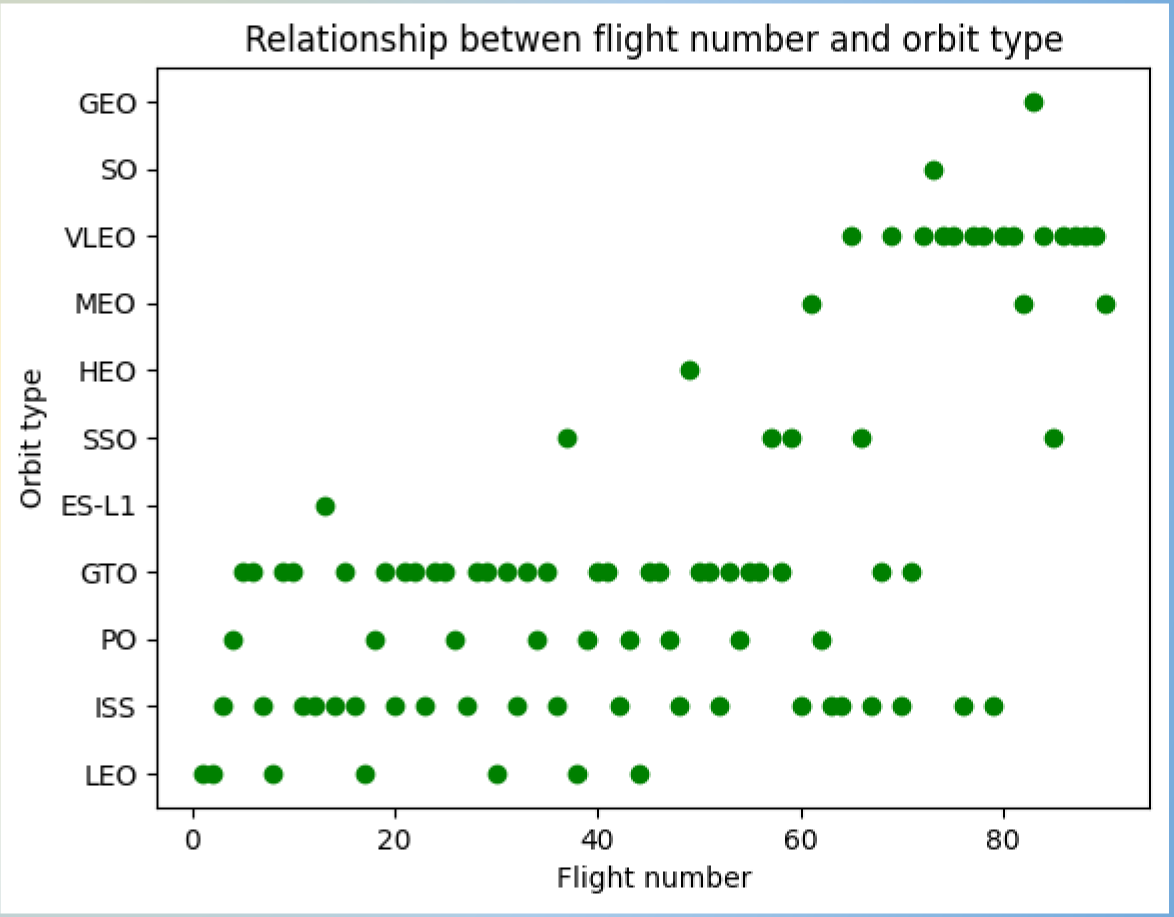
Success Rate vs. Orbit Type

- From the bar chart we can say that Falcon-9 rocket has most successful landings in orbits ES-L1, GEO, HEO and SSO.



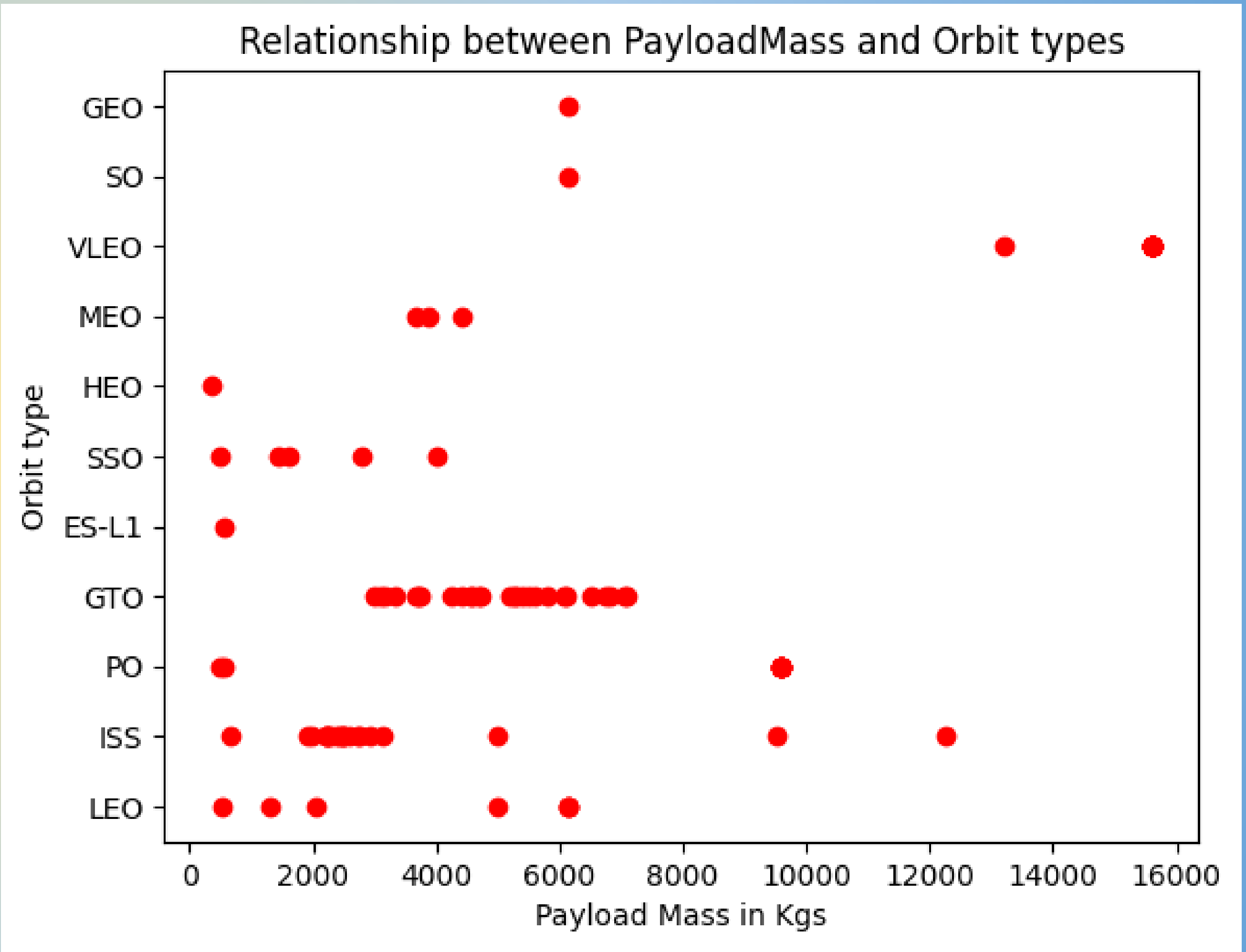
Flight Number vs. Orbit Type

Form the plot we can say that, in the LEO orbit the success appears related to the number of flights. On the other hand, there seems to be no relationship between flight number when in GTO orbit.



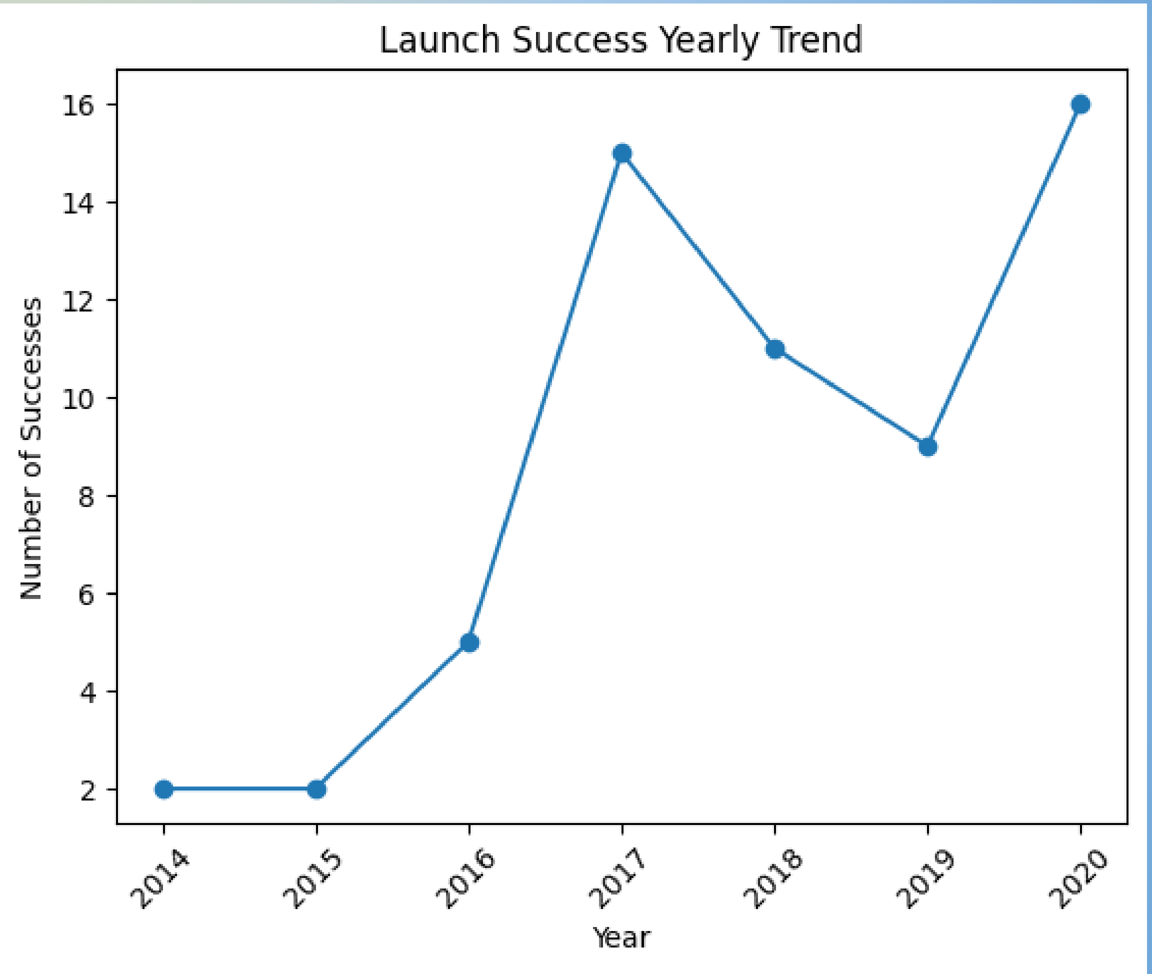
Payload vs. Orbit Type

- With heavy payloads the successful landing rates are more for Polar, LEO and ISS orbits. However, for GTO orbit we cannot distinguish this clearly as both positive landing rate and negative landing are both there here.



Launch Success Yearly Trend

- The average success rate increased from the years 2015 through 2017, then again increased since 2019.



All Launch Site Names

- We used the ‘distinct’ query feature to extract all the unique launch sites in out data. Here are the results obtained.

```
%sql select distinct Launch_Site from spacextable;
```

* sqlite:///ProjectX.db
Done.

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

Launch Site Names Begin with 'CCA'

- The first 5 entries of launch sites that begin with CCA have the same launch site and mission outcome. They also have booster version 1.0 in common. Considering the payload masses, it can be said that the launch site is good for low to medium payload ranges.

```
%sql select * from spacextable where Launch_Site like '%CCA%' limit 5;
```

* sqlite:///ProjectX.db
Done.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

The following query adds up all the payload weights for NASA's Commercial Resupply Services (CRS) missions. Here we are using sum() function on Payload mass column of spacextable and filtering the rows where customer field matches ‘NASA (CRS)’

```
%sql select sum("PAYLOAD_MASS__KG_") from spacextable where Customer like 'NASA (CRS)';
```

* sqlite:///ProjectX.db
Done.

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- This query calculates the average payload mass (in kilograms) for all launches using the F9 v1.1 booster version. The ‘%’ wildcard means it includes any variants starting with 'F9 v1.1'—like 'F9 v1.1 B1010'. It pulls data from the spacextable.

```
%sql select avg(PAYLOAD_MASS__KG_) from spacextable where Booster_Version like 'F9 v1.1%';
```

* sqlite:///ProjectX.db
Done.

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- This query returns the earliest launch date from the spacextable where the mission outcome was marked as 'Success'. It uses the min() function to find the first successful mission date in the dataset.

```
%sql select min(date) from spacextable where Mission_Outcome = 'Success';
```

* sqlite:///ProjectX.db
Done.

min(date)
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000 kgs

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

```
%sql select Booster_Version from spacextable where Mission_Outcome='Success' and PAYLOAD_MASS__KG_ between 4000 and 6000;
```

F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 FT B1032.2
F9 B4 B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

- This query retrieves the names of booster versions that had a successful mission and carried a payload mass between 4000 and 6000 kilograms. It filters the spacextable based on those two conditions and returns the matching Booster_Version entries. It serves as a quick way to spot high-performing boosters handling mid-range payloads.

Total Number of Successful and Failure Mission Outcomes

- The first query returns the total number of missions where the outcome includes "Success" (even partial or conditional successes).
- The second query returns the total number of missions that include "Failure" somewhere in the outcome description.
- They help you quickly compare how many SpaceX missions were successful versus unsuccessful.

```
%sql select count(*) from spacextable where Mission_Outcome like '%Success%';
```

* sqlite:///ProjectX.db
Done.

count(*)
100

```
%sql select count(*) from spacextable where Mission_Outcome like '%Failure%';
```

* sqlite:///ProjectX.db
Done.

count(*)
1

Boosters Carried Maximum Payload

- This query finds the booster version that carried the heaviest payload ever recorded in the spacextable. It does this by:
- Using a subquery to get the maximum payload mass
- Then selecting the Booster_Version that matches that exact mass

```
%sql select Booster_Version from spacextable where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextable);
```

* sqlite:///ProjectX.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

- This SQL query retrieves specific launch records from the spacextable where the launch occurred in the year 2015 and ended in a failed landing on a drone ship. Since SQLite doesn't support month names directly, the query uses substr(Date, 6,2) to extract the numeric month (like "01" for January) from the Date field, while substr(Date, 0, 5) isolates the year for filtering. The final output displays the month number, launch site, booster version, and the landing outcome—but only for launches that match both the year 2015 and the failure status involving a drone ship. It's a useful slice of data for analyzing when and where these specific failures happened.

```
%sql select substr(Date, 6,2) as month, Launch_Site, Booster_Version, Landing_Outcome from spacextable where substr(Date,0,5)='2015' and La
```

* sqlite:///ProjectX.db
Done.

month	Launch_Site	Booster_Version	Landing_Outcome
01	CCAFS LC-40	F9 v1.1 B1012	Failure (drone ship)
04	CCAFS LC-40	F9 v1.1 B1015	Failure (drone ship)

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

```
%sql select Landing_Outcome, count(*) as outcome_count from spacextable where date between '2010-06-04' and
```

* sqlite:///ProjectX.db
Done.

Landing_Outcome	outcome_count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- This query analyzes landing results over a specific period—from June 4, 2010, to March 20, 2017—by grouping all entries in the spacextable based on the type of Landing Outcome. It then counts how many times each outcome occurred using count(*) and ranks the results in descending order of frequency with order by outcome count desc. The goal is to show which types of landings (like "Success (drone ship)", "Failure", or "No attempt") were most and least common during that stretch of SpaceX launches.

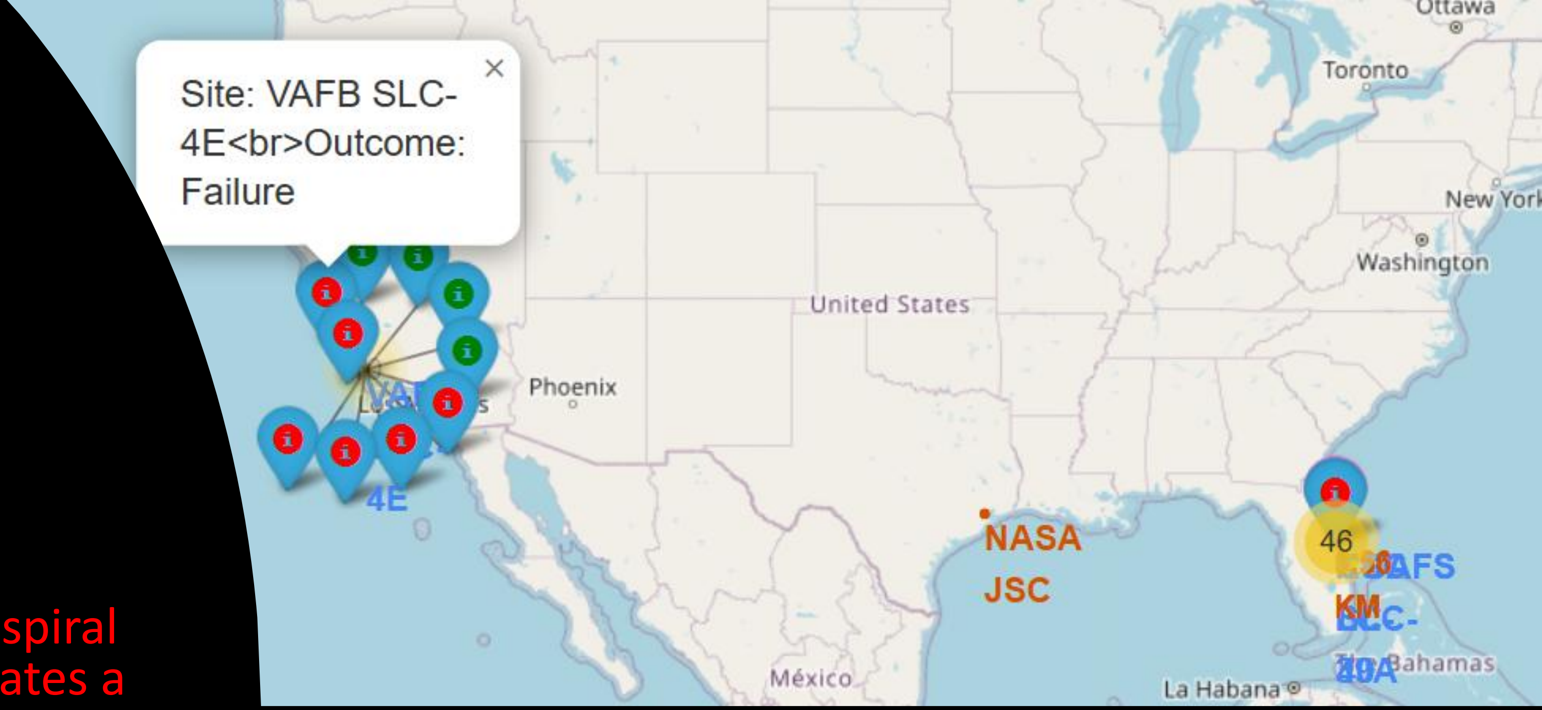
Section 3

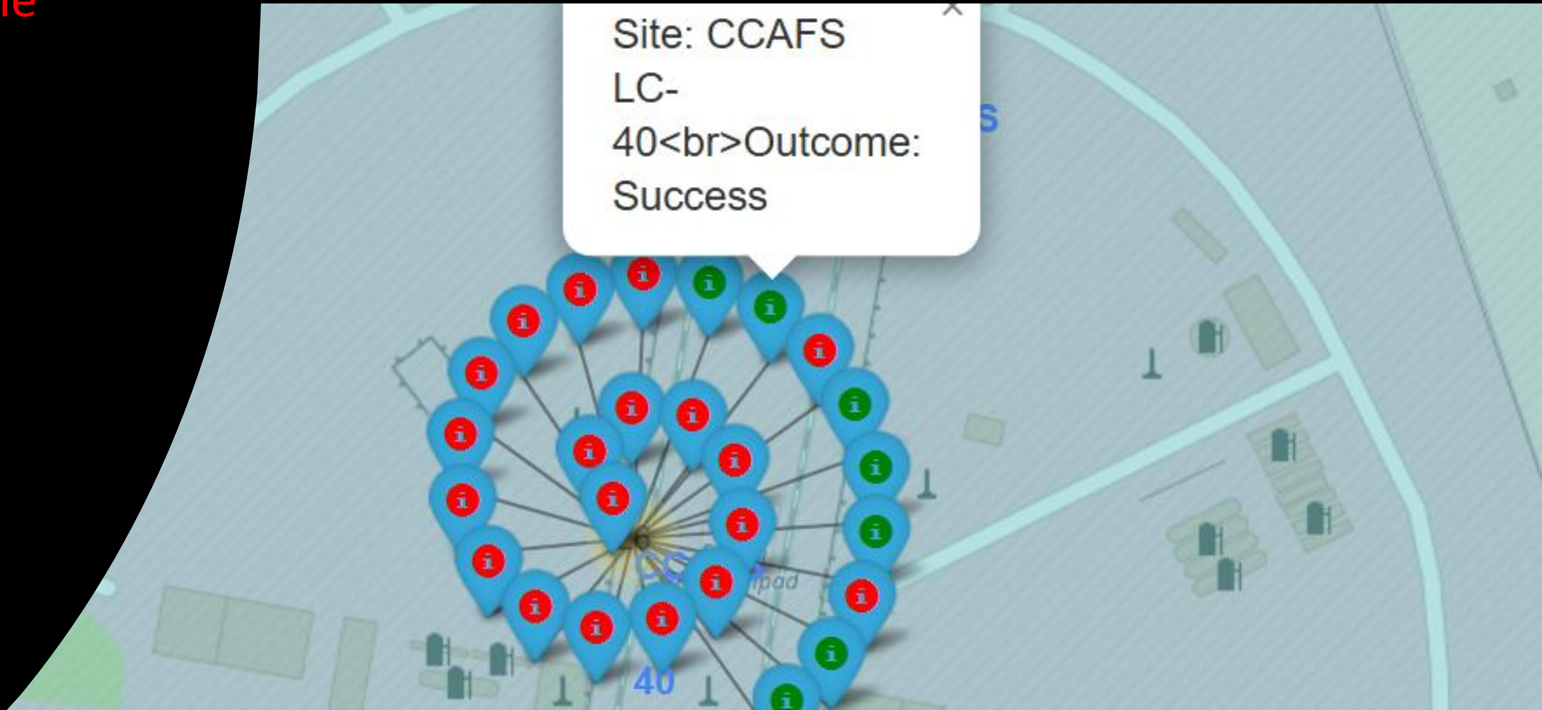
Launch Sites Proximities Analysis

Falcon 9 Launch Sites

- In the map, the yellow blobs represent the marker cluster for launch outcome markers. The blue text represents the label for the maker used to pinpoint launch sites.

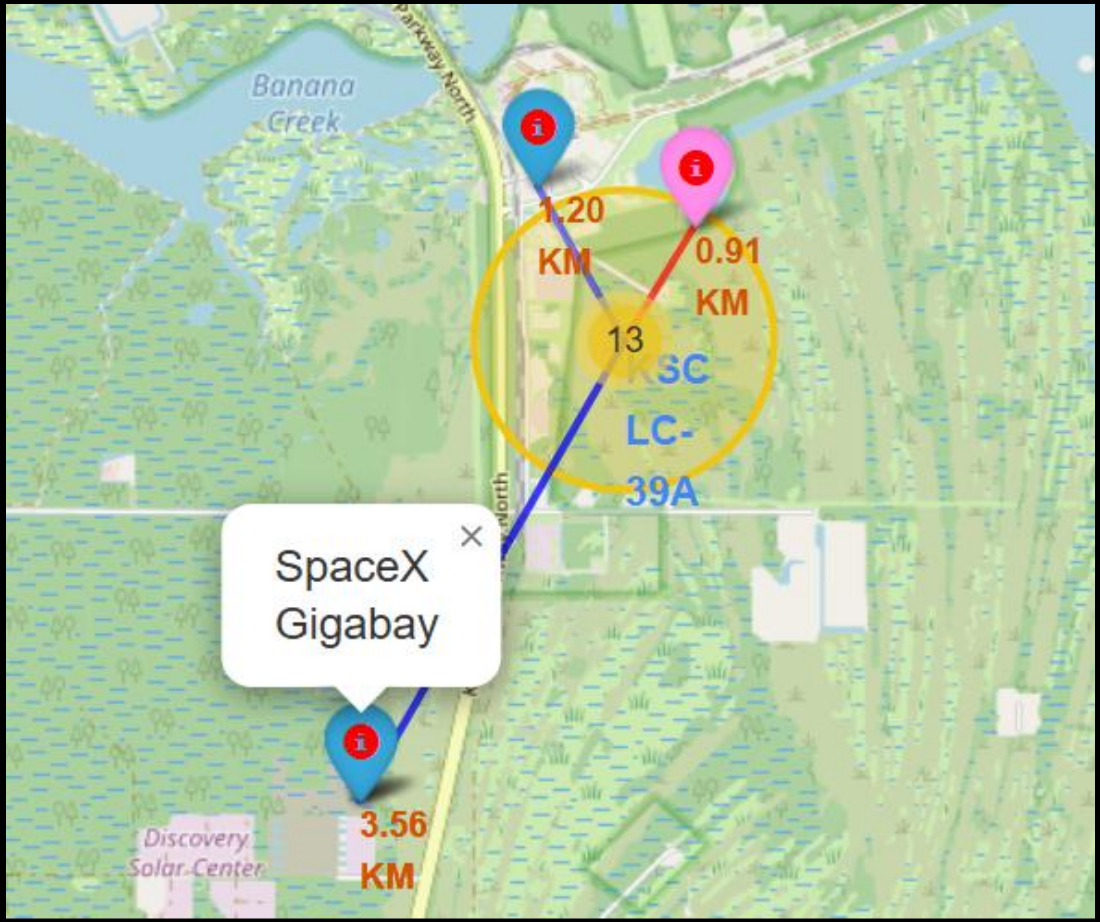
Launch Outcomes





The yellow blobs when clicked, results into a spiral web of markers. Here the green marker indicates a success outcome and red indicates failure. The marker cluster function makes it possible to represent many markers as a single blob.

Proximities near KSC LC-39A



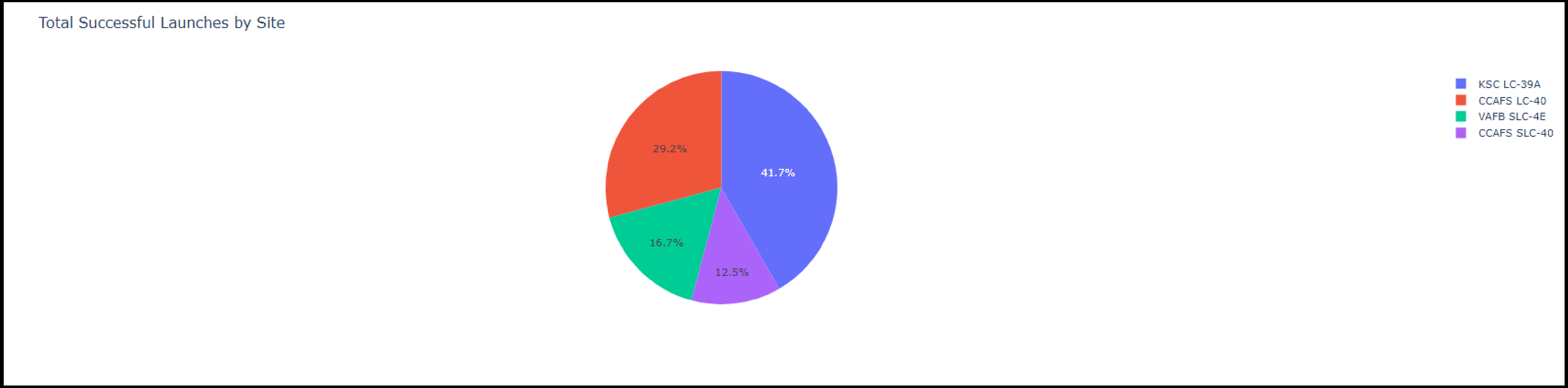
We added markers for each proximity along with the popup to display the calculated distance and a PolyLine connecting the proximity with the launch site, to represent the distance between the two locations.

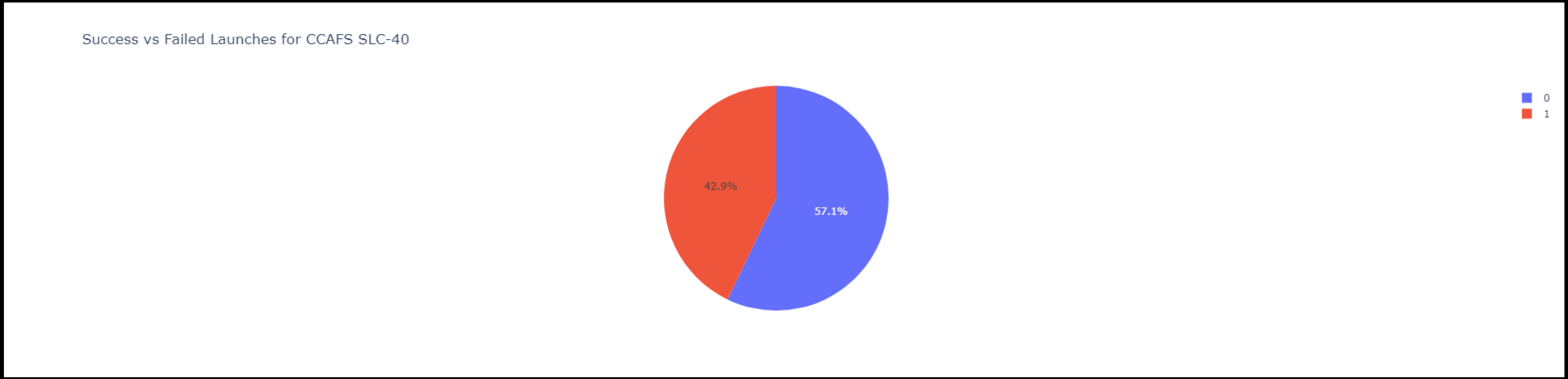
Section 4

Build a Dashboard with Plotly Dash

Total successful launches by site

KSC LC-39A had most successful launches among all sites. With a success rate of approximately 42%

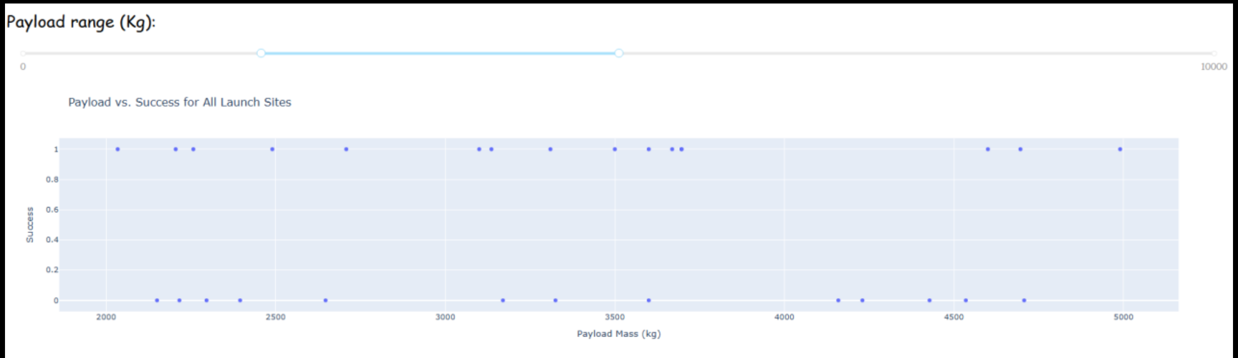


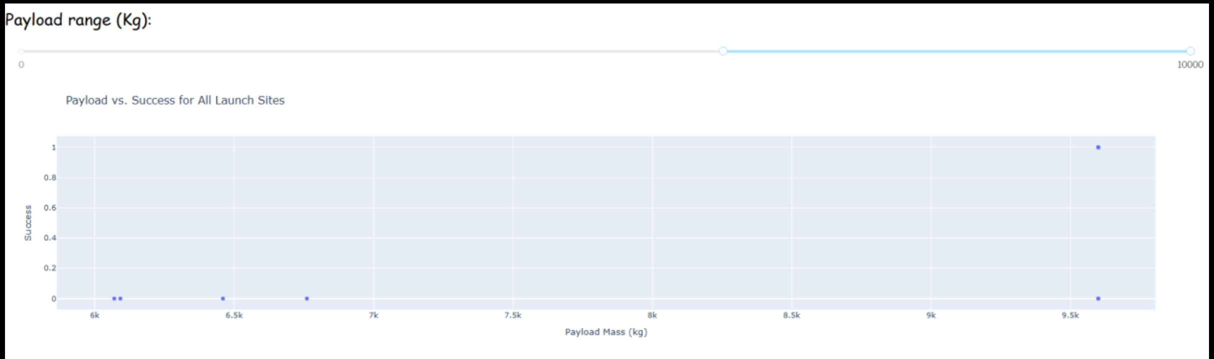


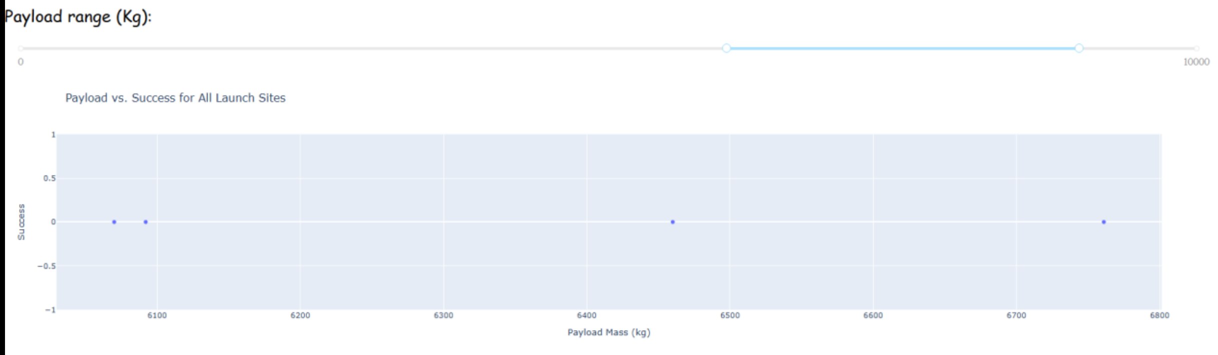
Most successful Launch site

0 (red) represents failure outcomes and 1 (blue) represents successful outcomes.

Launch site CCAFS SLC-40 has a success rate of 43%.







Payload vs Launch Outcome

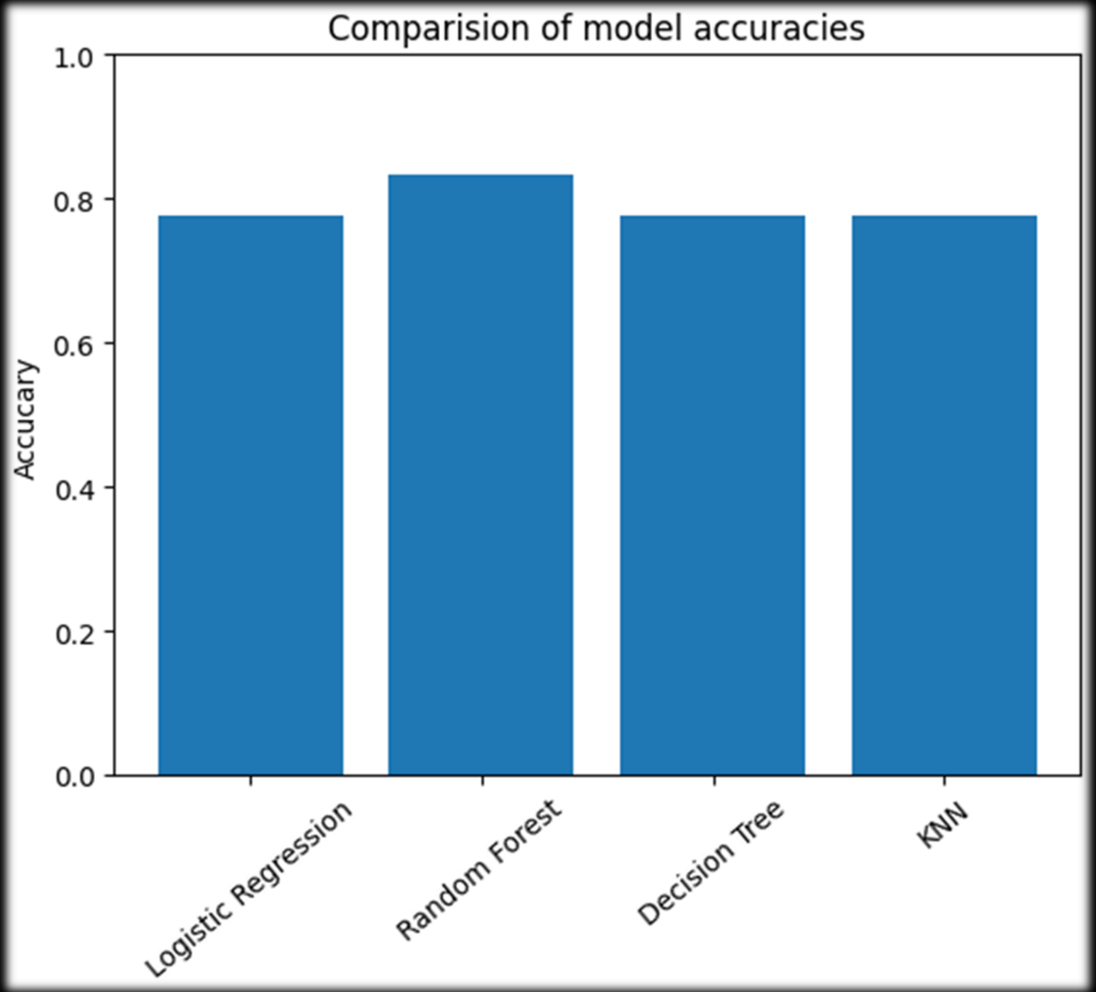
Among all Launch sites, the payload range of 2000 – 5500 kgs had the most successful launches.

Payload range of 8000 to 10000 kgs resulted in least successful outcomes.

Section 5

Predictive Analysis (Classification)

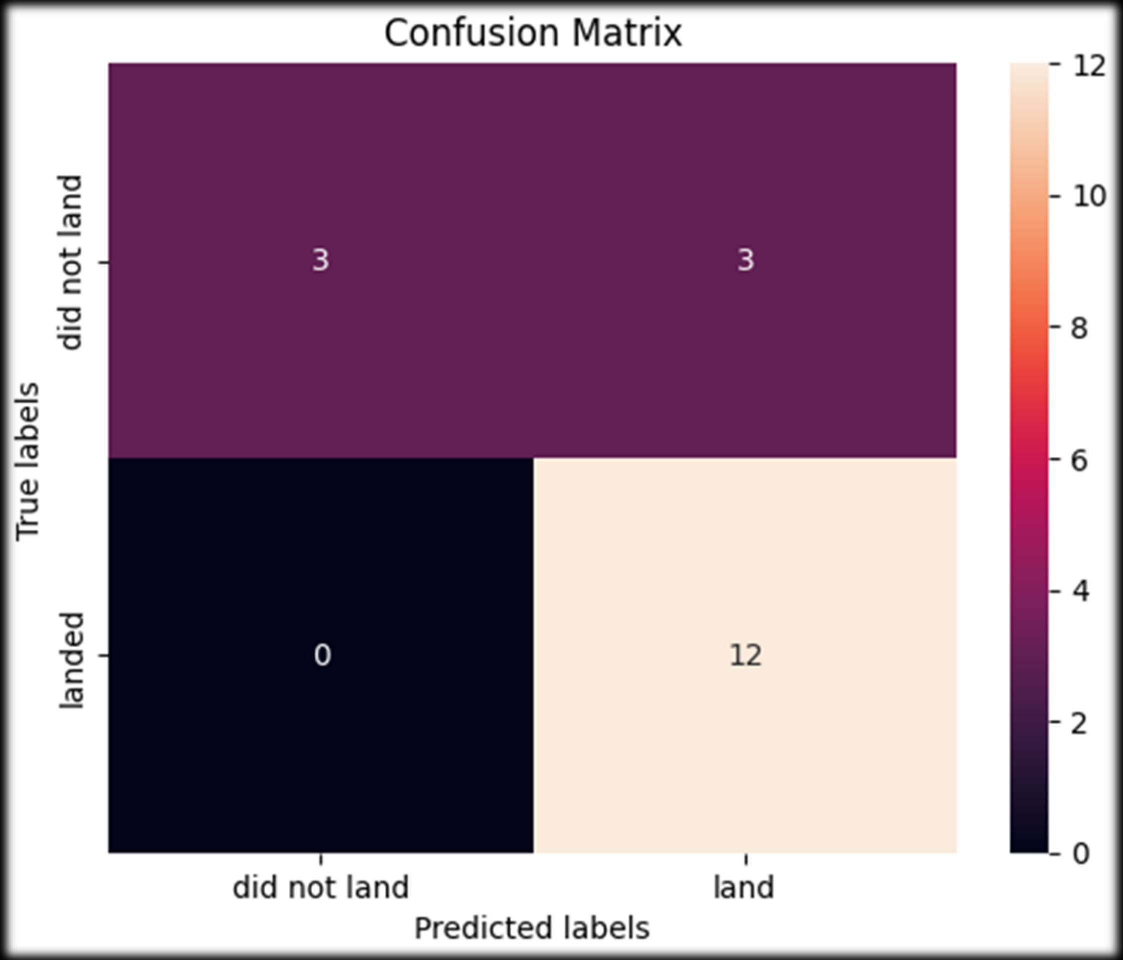
Classification Accuracy



We trained 4 classification models on our data. Logistic Regression, Random Forest, Decision tree and KNN models were trained. We found that the Random Forest model does a better job at predicting the landing outcomes, than other models, with an accuracy of 83%.

Confusion Matrix

The Random Forest model correctly predicted 12 successful landing outcomes that were true but failed to predict any of the 3 unsuccessful landing outcomes.



Conclusions

- Falcon-9 booster version B5 is a good choice for carrying maximum payload with a higher chance of successful outcome.
- Launch site CCSFS SLC-40 has a good chance of successful rocket landing in the GTO orbit with payload mass ranging from 3000 to 7000kgs.
- Launch site KSC LC-39A has a higher success rate in payload range of 2000-4000 kgs.
- Falcon-9 rocket Block-4 has the highest launch success rate.
- Random Forest classification model is a good choice to predict the landing outcome class, as it has an accuracy rate of 83%.

Thank you!