IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ramkumar M
10-Dec-2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection thru API
  - Data collection thru web scrapping
  - Data wrangling
  - Explore the dataset using EDA methods
  - Interactive charts using dash & folium
  - Data preprocessing like treating missing, one-hot encoding
  - Model building
- Summary of all results
  - EDA outputs
  - Interactive Data Visualization
  - Predictive model results

Introduction

- Project background and context

  As a newly entered competitor in the Rocket Launching field determining the launch cost is a tough ask. Most of the providers Rocket Launch cost is greater than 160 million USD. Whereas SpaceX in their website mentioned the Launch cost as 62 million USD.

  This is because SpaceX can recover the first stage and reuse it.

  By determining the whether first stage will land successfully or not we can determine the cost of a launch.

  We deploy classification model to identify whether the first stage was landed successfully or failed.

- Problems you want to find answers

  Number of successful launches

  Factors influencing the launch results

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API & from Wikipedia using web scrapping technique
- Perform data wrangling
  - Checked the data consistency, selected only the required variables
  - Replaced the missing values with mean
  - Created class variable using actual outcome variable to employ binary classification model
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Built multiple models like Logistic Regression, SVM, Decision Tree, KNN and compare the results and pick the best model & perform GridSearchCV and tune the hyperparameters

Data Collection

Data collection process involves gathering data from multiple sources, thru SpaceX API calls and Web scrapping a Wikipedia page.

Data must be collected from both sources to get the complete information.

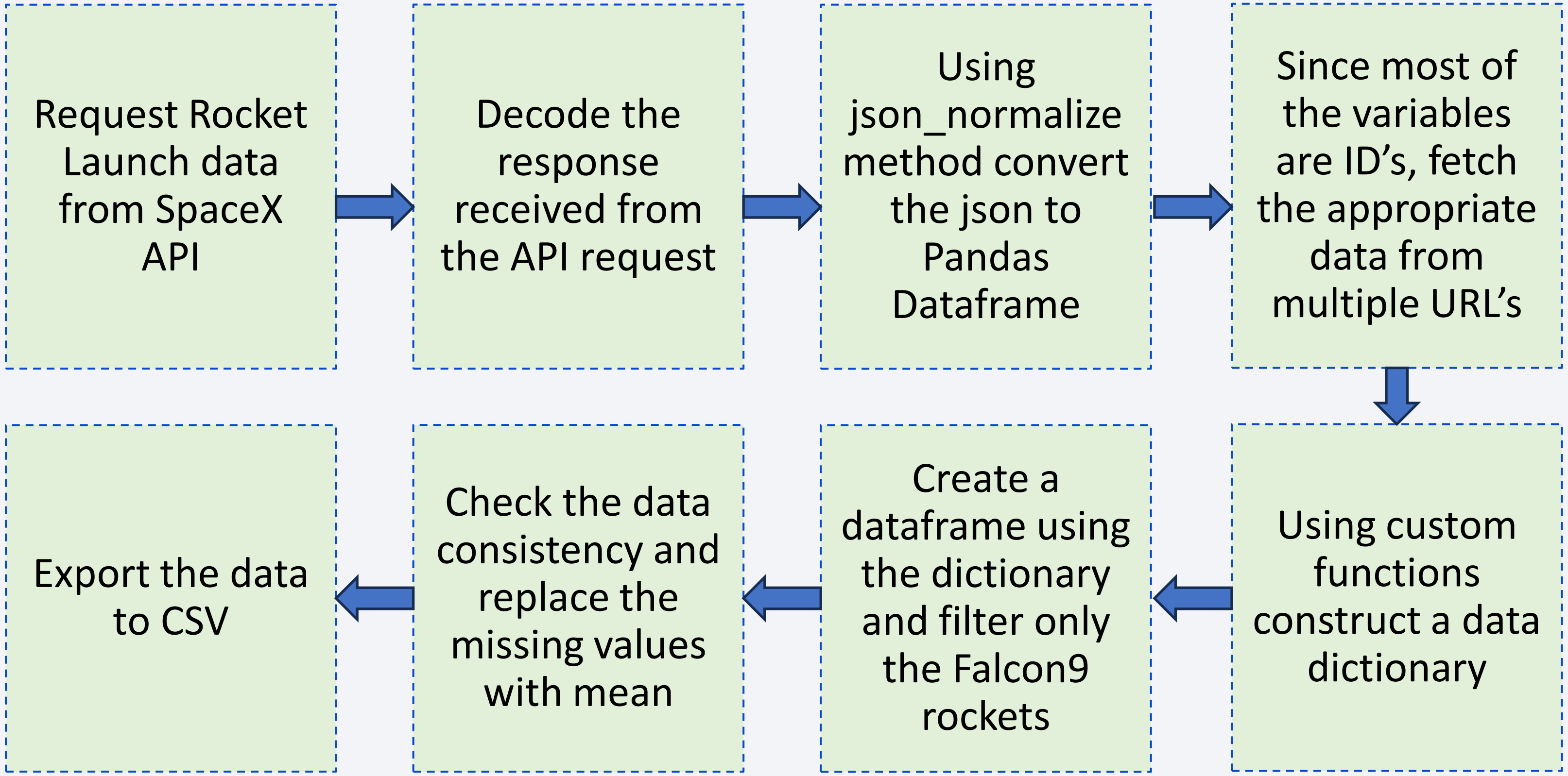
Features retrieved from SpaceX API:

FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

Features retrieved thru webscrap:

Flight, No., Launch, site, Payload, Payload, mass, Orbit, Customer, Launch, outcome, Version, Booster, Booster, landing, Date, Time

Data Collection – SpaceX API



Request Rocket Launch data from SpaceX API

Decode the response received from the API request

Using json_normalize method convert the json to Pandas Dataframe

Since most of the variables are ID's, fetch the appropriate data from multiple URL's

Using custom functions construct a data dictionary

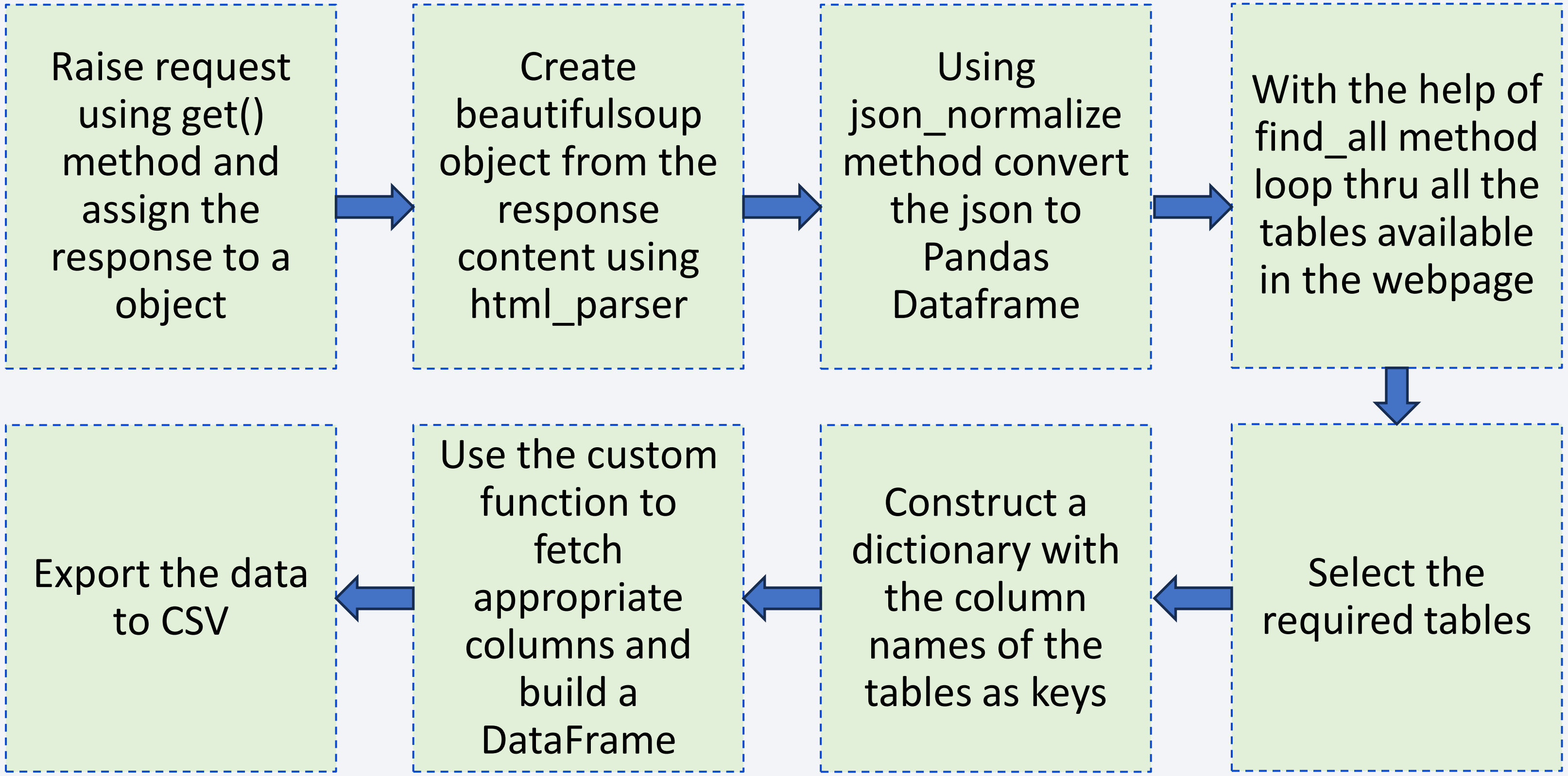
Create a dataframe using the dictionary and filter only the Falcon9 rockets

Check the data consistency and replace the missing values with mean

Export the data to CSV

Data collection using API

Data Collection - Scraping



Raise request using get() method and assign the response to a object

Create beautifulsoup object from the response content using html_parser

Using json_normalize method convert the json to Pandas Dataframe

With the help of find_all method loop thru all the tables available in the webpage

Select the required tables

Construct a dictionary with the column names of the tables as keys

Use the custom function to fetch appropriate columns and build a DataFrame

Export the data to CSV

[Data collection using Web scrapping]

Data Wrangling

- Checked the data for missing values using isna() method
- Verified the types of each variable and identify the list of numerical and categorical variables
- For all the categorical variables determine the number of occurrence of each options
- The landing_outcomes variable contains multiple categories like "True ASDS", "None None", "True RTLS“ etc as mentioned in Table 1
- Identify the list of successful and failure categories as split by Table 2
- Create a binary variable as a final dependent variable

Data Wrangling

Table 1

Categories	Count
True ASDS	41
None None	19
True RTLS	14
False ASDS	6
True Ocean	5
False Ocean	2
None ASDS	2
False RTLS	1

Table 2

Successful landings	Failed landings
True ASDS	None None
True RTLS	False ASDS
True Ocean	False Ocean
	None ASDS
	False RTLS

EDA with Data Visualization

1. Scatter Plot

- A scatter plot is used to visualize the relationship between two continuous variables. Each point in the plot represents an observation, with its position determined by the values of the two variables. To enhance interpretability, we included the output class variable, which visually distinguishes patterns between successful and failed launches. Scatter plots are helpful for identifying trends, correlations and outliers in the data.

2. Line Graph

- A line graph connects data points with a line to show trends over a continuous variable, typically time. It is commonly used to visualize changes and patterns, such as increases or decreases, and is effective for observing data continuity and trends.

3. Bar Diagram

- A bar diagram (or bar chart) is used to compare categorical data. The height (or length) of each bar represents the value or frequency of the corresponding category. It is ideal for visualizing comparisons among distinct groups or categories and highlighting differences in magnitude.

EDA Data Viz

EDA with SQL

- Aim is to read the CSV data and load it into SQL and perform EDA
- Connected to the SQL database using the connection string
- Using pandas read the CSV file and load it to SQL using to_sql method
- Using the magic string we will execute the SQL queries in python
- Slice & view the data using distinct & limit clauses
- Explore each variables by applying various SQL functions. Aggregate functions like sum(), avg(), min(), max(), count()
- With the help of subquery identified the names of 'Booster_Version' which carried maximum payload mass
- With the help of group by clause, identified the number of times of failure or successful launches within a give month / year.

EDA with SQL

Build an Interactive Map with Folium

- As the launch success rate depends on multiple factors including the initial position of rocket trajectories it is essential to map & visualize the actual launch sites in a map
  - Started the map zoom in Houston Area
  - Added a marker & circled the NASA JSC in the map
  - Added a marker & circle for the 4 launch sites in the map zoomed in entire USA
- In each site added a marker_cluster which display the total number of launches in the circle. By clicking on the circle, we can see the number of failed & successful launches.
- Calculated the distance and displayed it in KM's from the launch sites to the proximities. Like closest coastal lines, railroads, highways.

Folium Map

Build a Dashboard with Plotly Dash

Instead of showing the static charts we can utilize the Plotly and create interactive charts letting users to interact directly

Interactive Pie Chart:

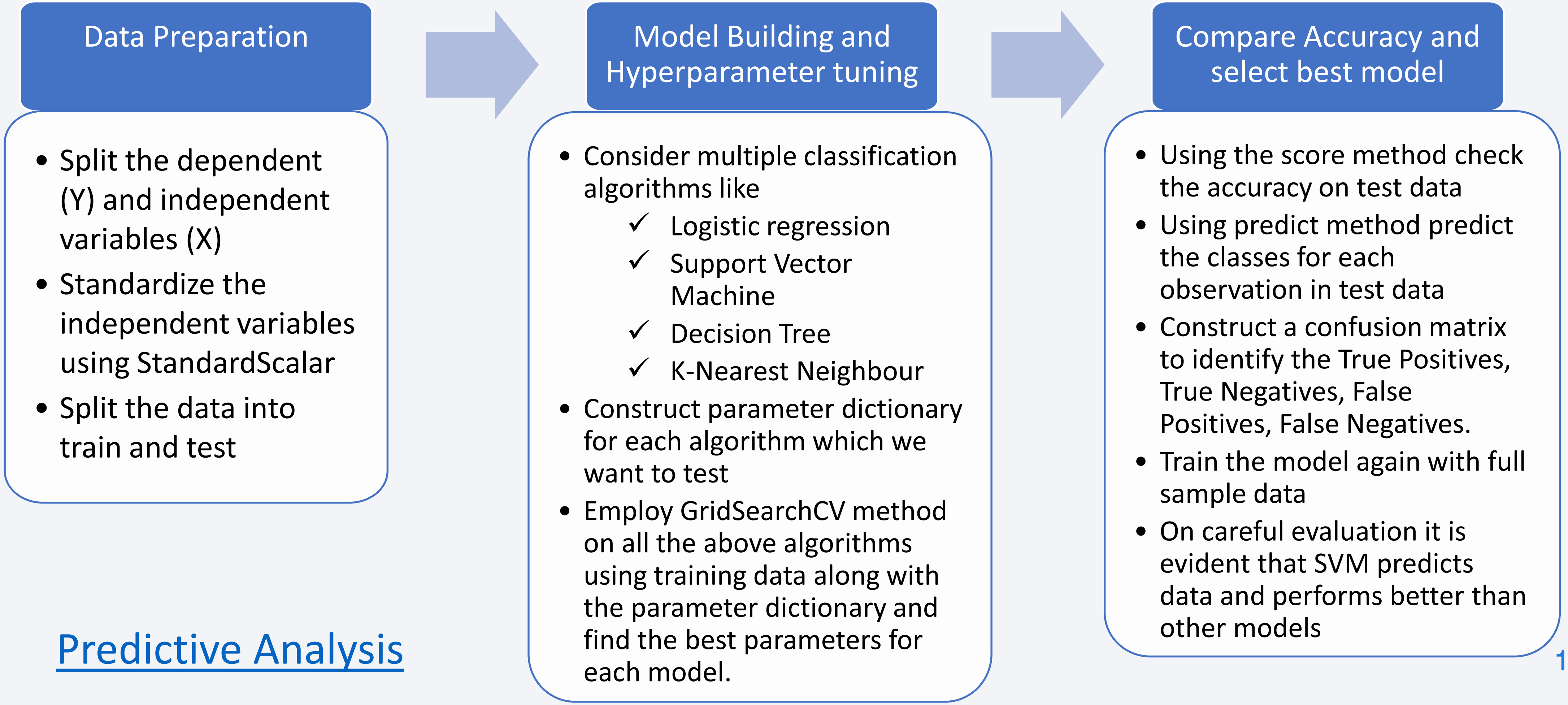
- A dropdown is displayed, allowing users to select the site. By default, “All” sites are selected.
- Initially, a pie chart is displayed, visualizing the total percentage of successful launches for each site.
- When the site name is changed using the dropdown, the pie chart automatically refreshes to show the split between successful and failed launches for the selected site.

Interactive Scatter Plot:

- This scatter plot helps analyze the relationship between payload mass and launch outcomes.
- Data points are color-coded based on the Rocket Booster version, enabling clear distinction of results.
- A range slider is added to adjust the upper and lower limits of the payload mass, allowing users to explore correlations across different ranges.

Plotly Dash App

Predictive Analysis (Classification)



Data Preparation

- Split the dependent (Y) and independent variables (X)
- Standardize the independent variables using StandardScalar
- Split the data into train and test

Model Building and Hyperparameter tuning

- Consider multiple classification algorithms like
  - ✓ Logistic regression
  - ✓ Support Vector Machine
  - ✓ Decision Tree
  - ✓ K-Nearest Neighbour
- Construct parameter dictionary for each algorithm which we want to test
- Employ GridSearchCV method on all the above algorithms using training data along with the parameter dictionary and find the best parameters for each model.

Compare Accuracy and select best model

- Using the score method check the accuracy on test data
- Using predict method predict the classes for each observation in test data
- Construct a confusion matrix to identify the True Positives, True Negatives, False Positives, False Negatives.
- Train the model again with full sample data
- On careful evaluation it is evident that SVM predicts data and performs better than other models

[Predictive Analysis]

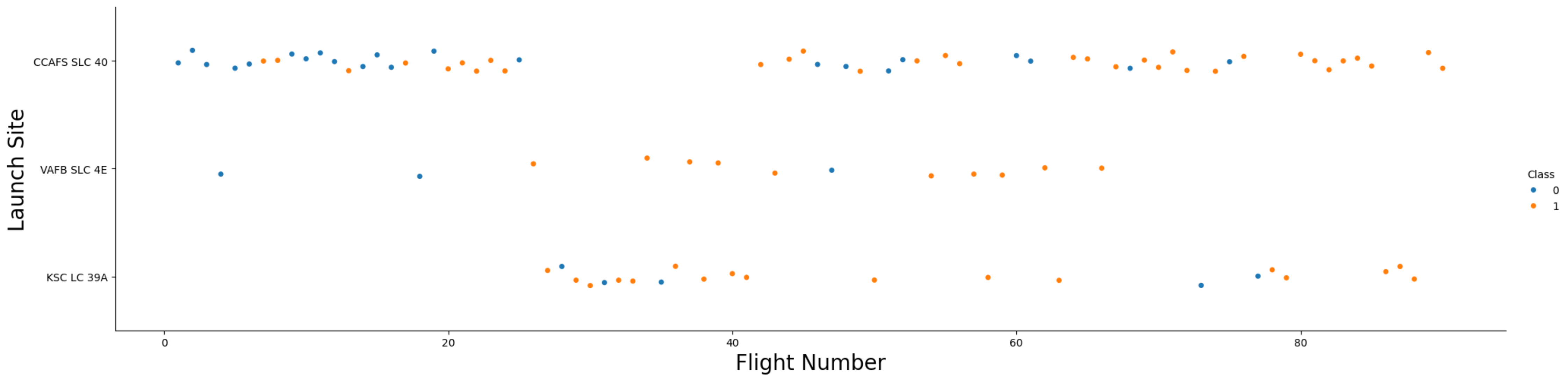
Results

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

Section 2

Insights drawn from EDA

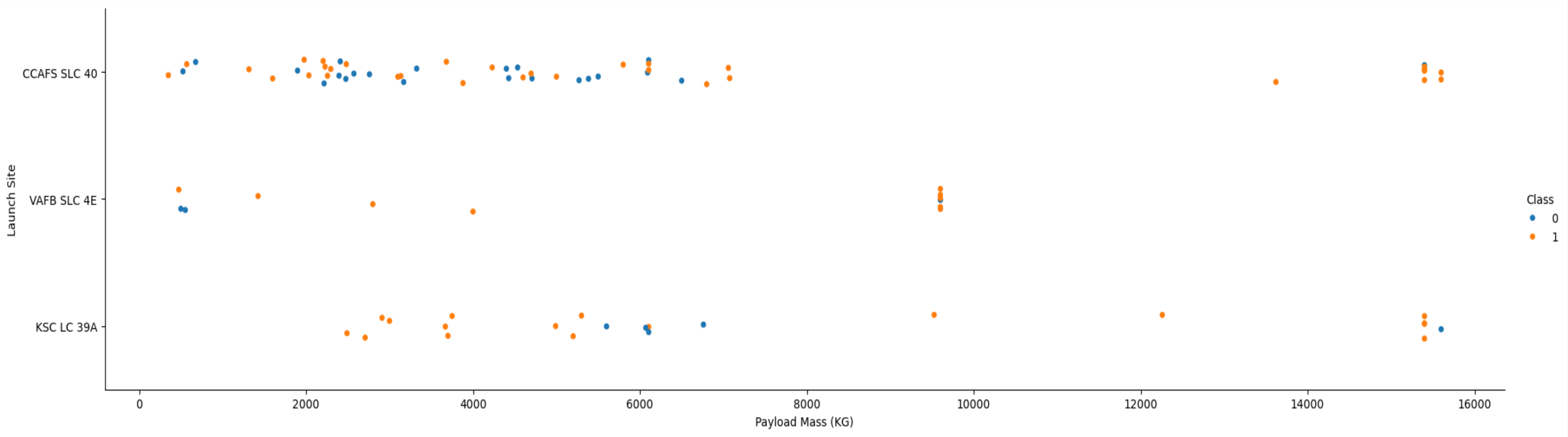
Flight Number vs. Launch Site



Explanation:

- The probability of first stage rocket return landing is more successful in recent test compared to the earlier launches across all 3 launch sites.
- Majority of the test were conducted from site CCAFS SLC 40. This includes the early tests where the failure rate are higher.
- The success rate of VAFB SLC 4E & KSC LC 39A are higher than CCAFS SLC 40.

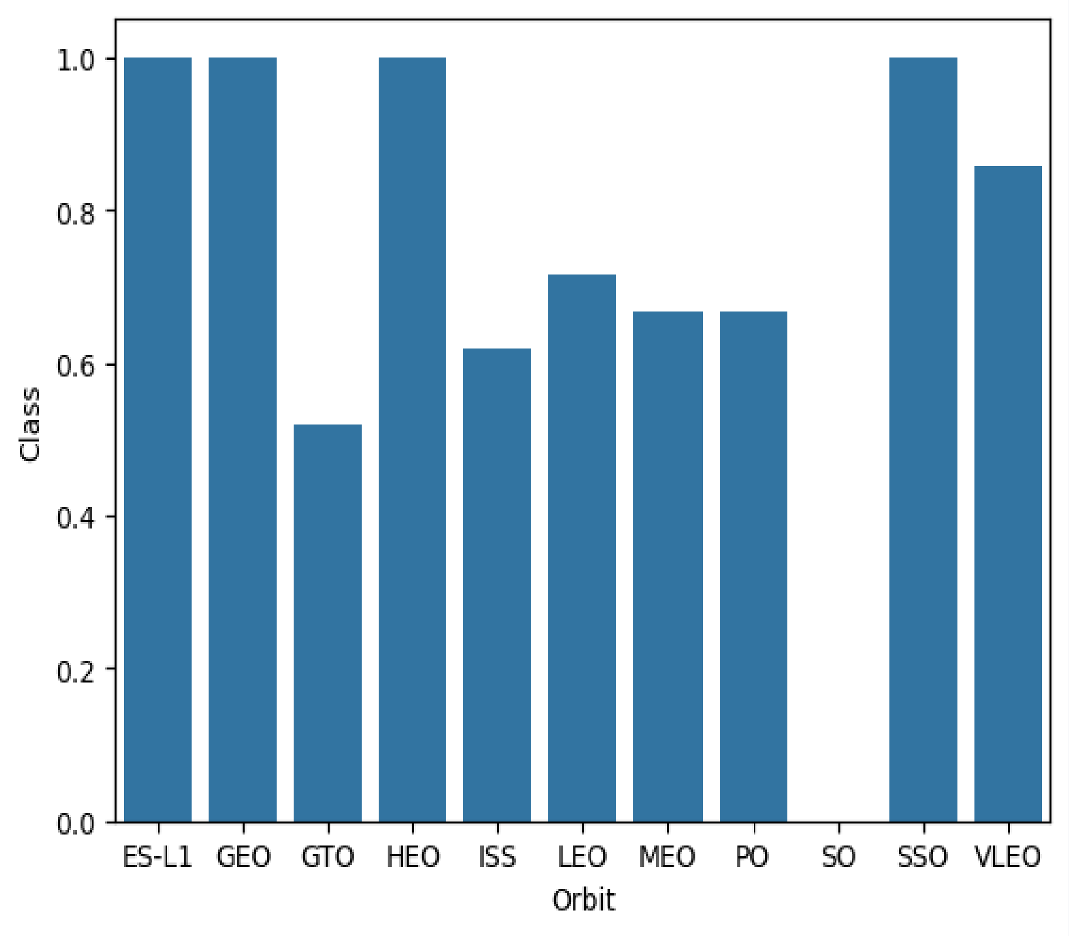
Payload vs. Launch Site



Explanation:

- The success rate increased drastically when the payload mass increased above 9000 KG.
- No rockets launched with payload more than 10000 KG from VAFB SLC 4E site.

Success Rate vs. Orbit Type



Explanation:

- The bar diagram between Orbit and Outcome class clearly indicating a relationship. As some of the orbits have 100% success rate and some have medium and 1 have 0% success.
- High Success rate (>80) found in

  ES-L1, GEO, HEO, SSO, VLEO

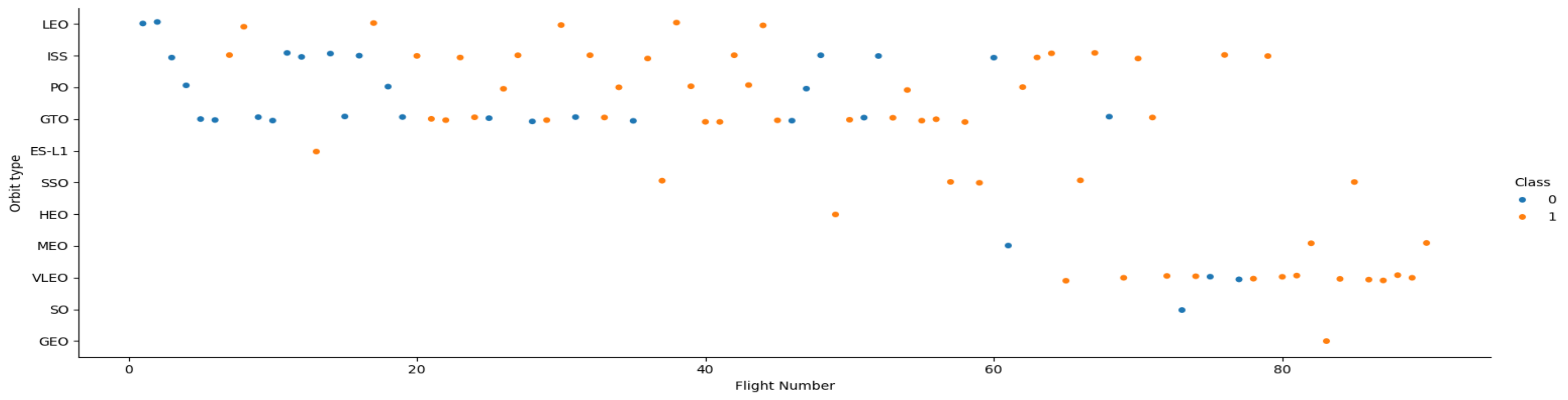
- Average Success rate (50%-80%)

  GTO, ISS, LEO, MEO, PO

- Orbit with 0% Success

  SO

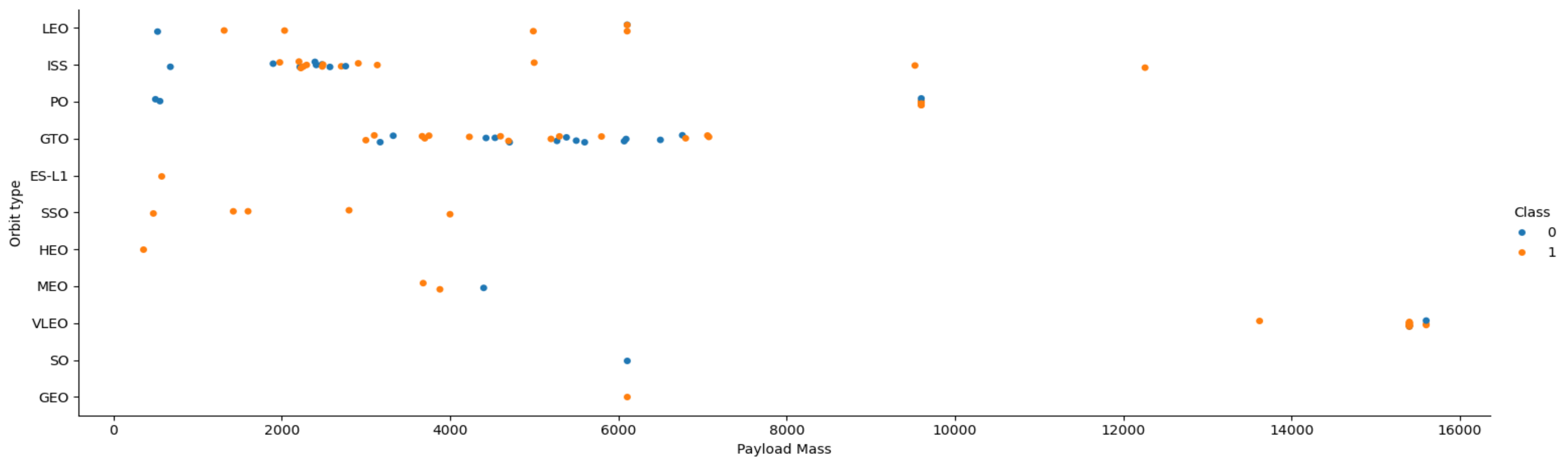
Flight Number vs. Orbit Type



Explanation:

- In the initial days, all the rockets were tested on specific orbits such as LEO, ISS, PO, GTO
- Later tested with different orbits but mostly on VLEO

Payload vs. Orbit Type



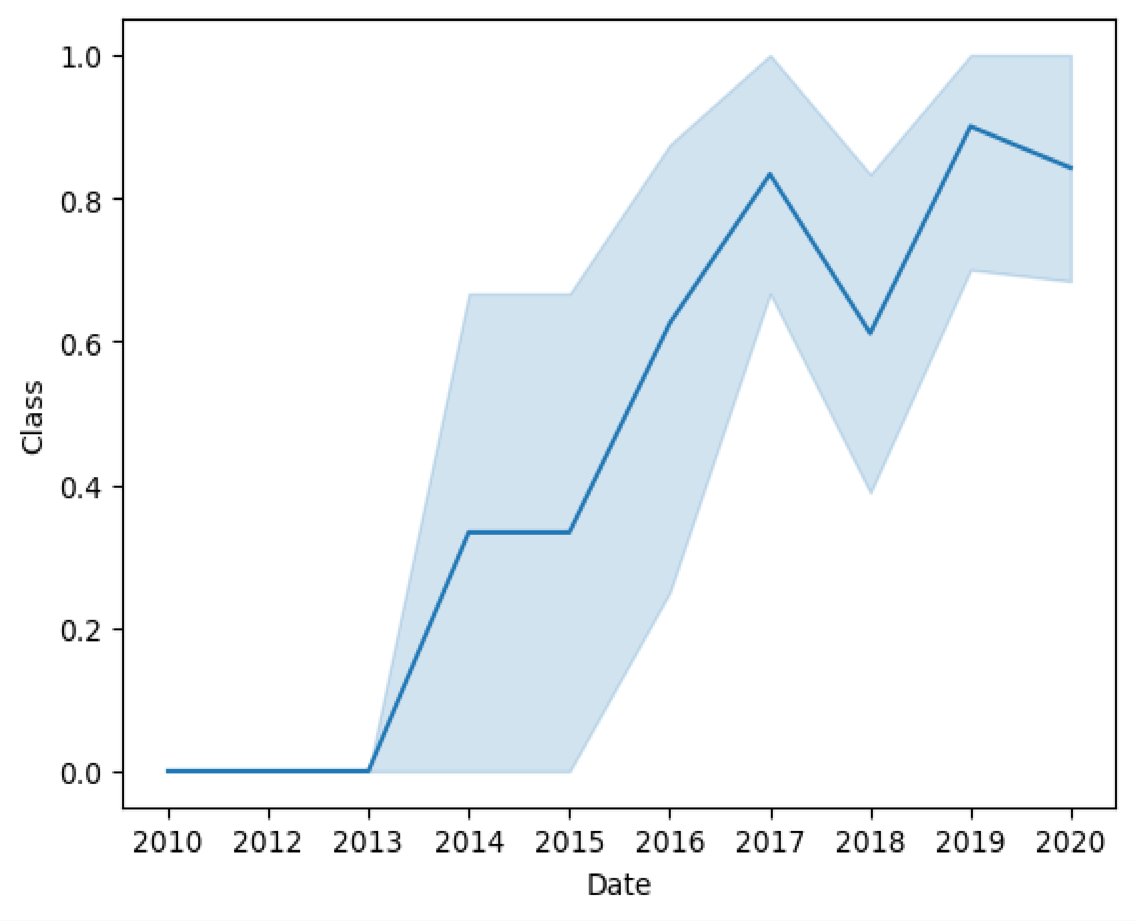
Explanation:

- Even with low payload mass (<5000 KG) the success is higher for the rockets launched in SSO orbit. Contrary to the previous assumption of heavy payload results in higher success rate.
- Only in LEO orbit as the payload increase the success rate also increased where as GRO & ISS showed mixed results.

Launch Success Yearly Trend

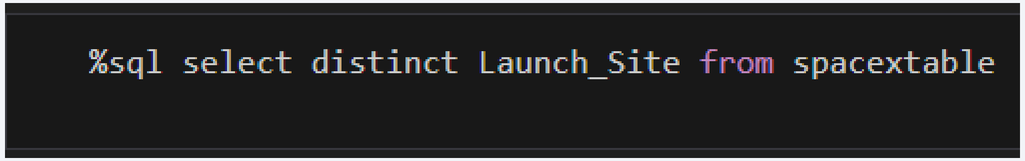
Explanation:

- Till 2013 the success rate is NIL
- From 2013 onwards the success rate is gradually increasing
- In 2018 & 2020 there was a small dip in success rate



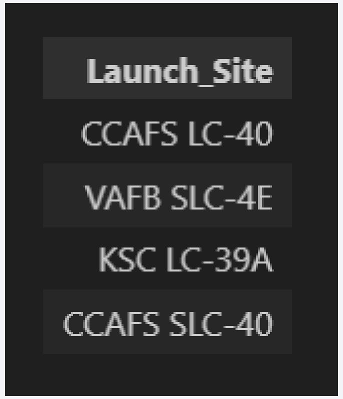
All Launch Site Names

Query



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

Output

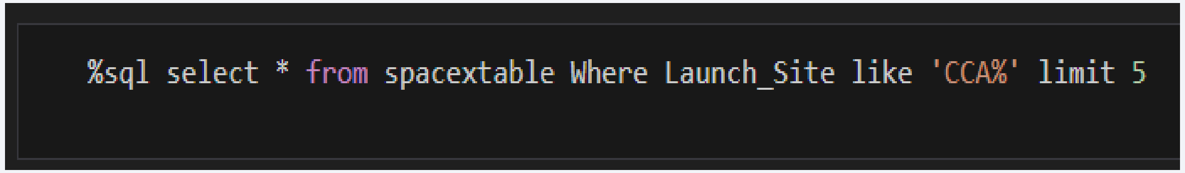


Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

- Select query retrieves the data from the SpaceXtable
- The keyword Distinct fetches only the unique values present from the Launch_site column

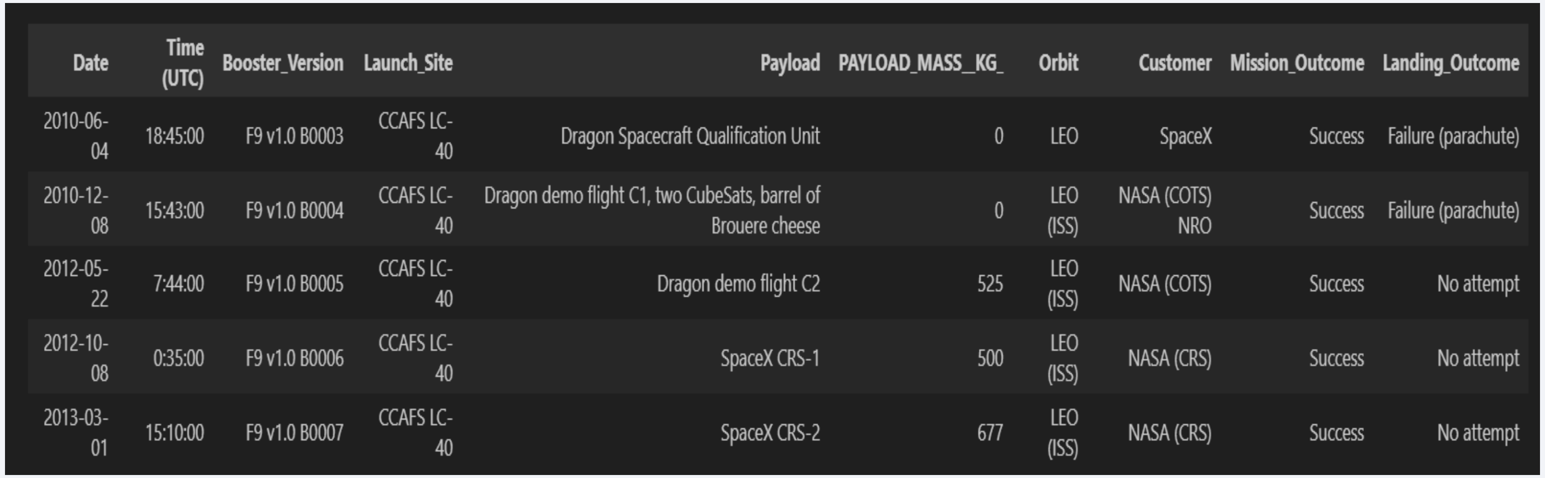
Launch Site Names Begin with 'CCA'

Query



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

Output

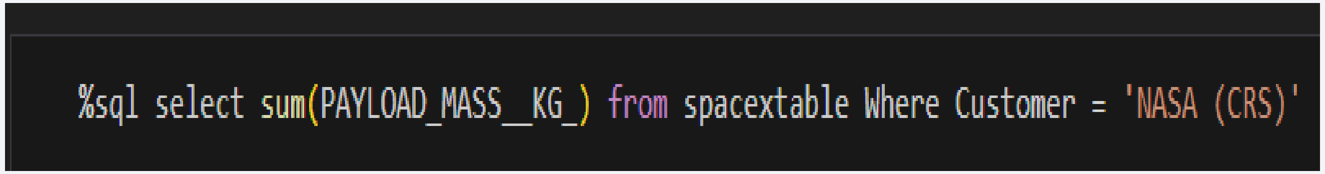


Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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

- Using the WHERE Clause and passing the regular expression (%) we filtered the records to display only the sites starting with ‘CCA%’
- We also limited the number of records using LIMIT to display only 5 records

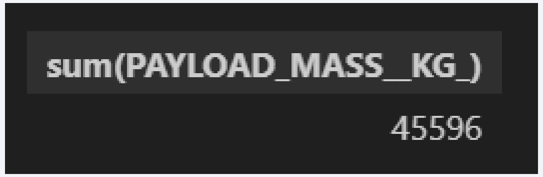
Total Payload Mass

Query



```
%sql select sum(PAYLOAD_MASS__KG_) from spacextable Where Customer = 'NASA (CRS)'
```

Output

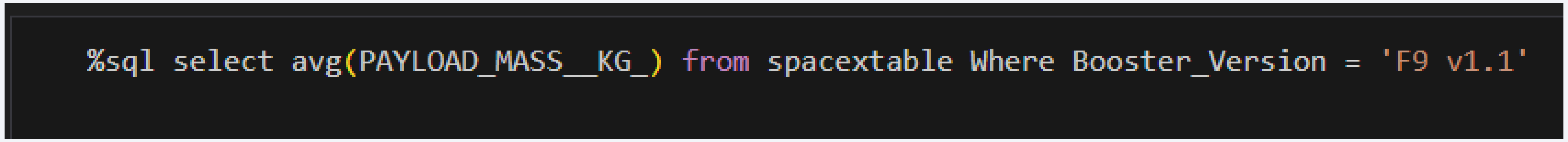


```
sum(PAYLOAD_MASS__KG_)
                 45596
```

- By using the Where clause we filtered out only the NASA (CRS) Customers and calculated the total payload using sum function.

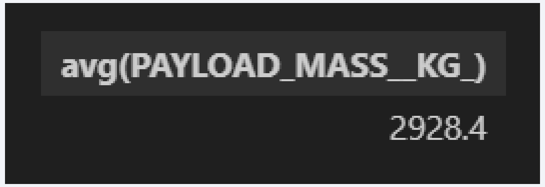
Average Payload Mass by F9 v1.1

Query



```
%sql select avg(PAYLOAD_MASS__KG_) from spacextable Where Booster_Version = 'F9 v1.1'
```

Output

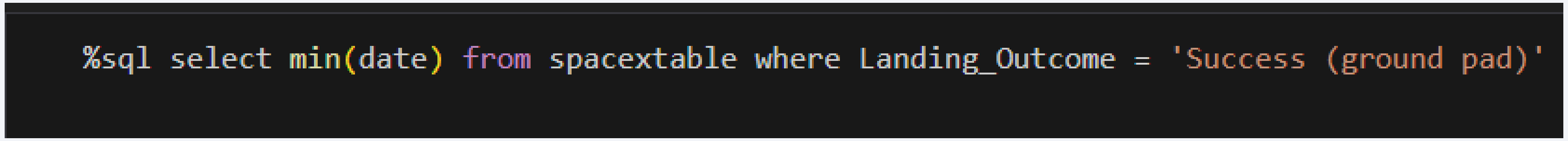


avg(PAYLOAD_MASS__KG_)
2928.4

- By using the Where clause we filtered out only the `'F9 v1.1’` Booster_version and calculated the average payload using avg function.

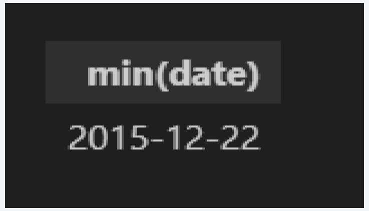
First Successful Ground Landing Date

Query



```
%sql select min(date) from spacextable where Landing_Outcome = 'Success (ground pad)'
```

Output

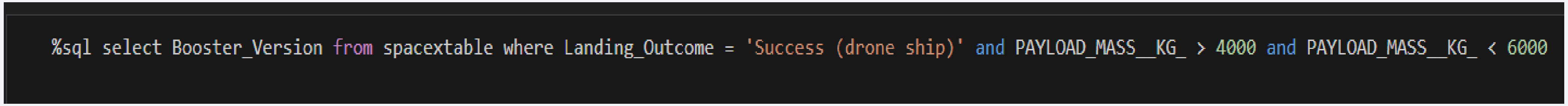


```
min(date)
2015-12-22
```

- By using the Where clause we filtered out only the `'F9 v1.1’` Booster_version and calculated the average payload using avg function.

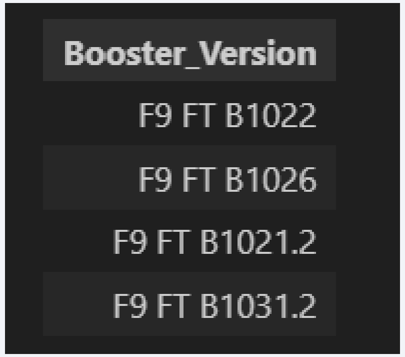
Successful Drone Ship Landing with Payload between 4000 and 6000

Query



```
%sql select Booster_Version from spacextable where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

Output

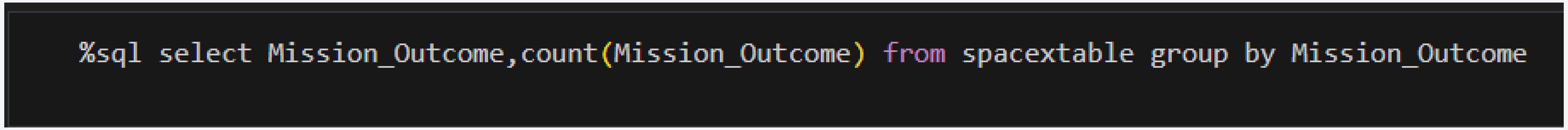


Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- By using the Where clause and logical operator AND condition, we filtered out the landing_Outcome along with PAYLOAD Mass between 4000-6000 range

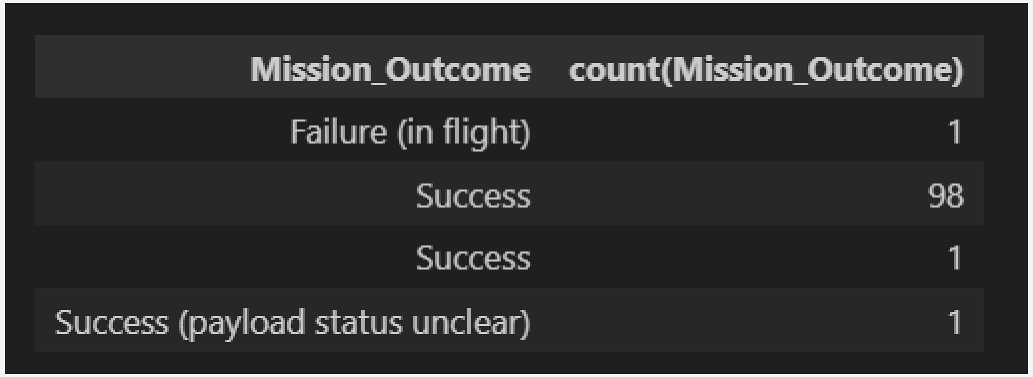
Total Number of Successful and Failure Mission Outcomes

Query



```
%sql select Mission_Outcome,count(Mission_Outcome) from spacextable group by Mission_Outcome
```

Output

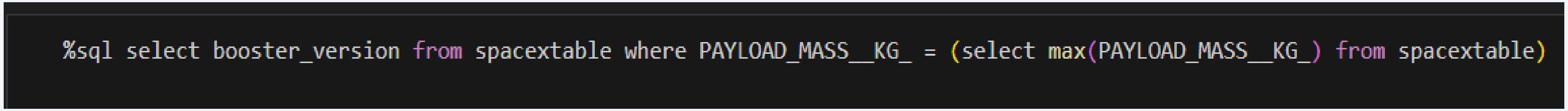


Mission_Outcome	count(Mission_Outcome)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- The COUNT function aggregates data and returns the number of records. Using the GROUP BY clause, the counts are grouped by the distinct values in the Mission_outcome column

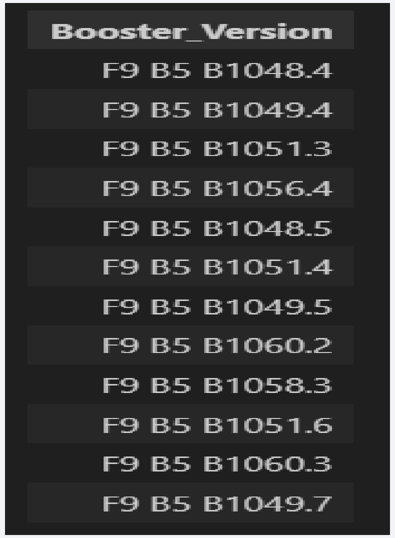
Boosters Carried Maximum Payload

Query



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

Output

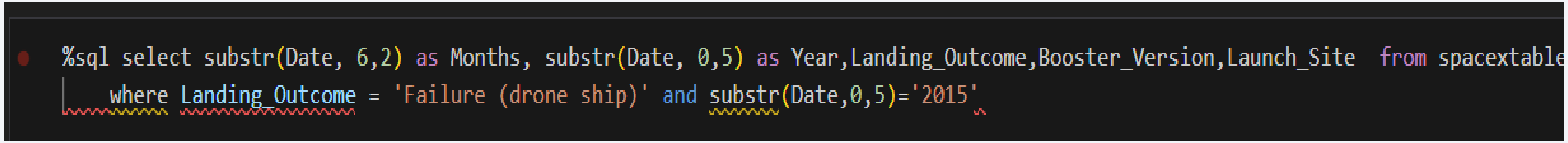


Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

- The subquery will find the maximum value of the Payload column and it matches with the main query to retrieve the corresponding booster version which carried the maximum payload

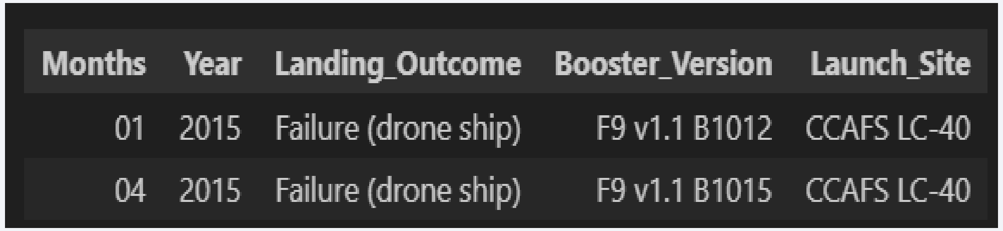
2015 Launch Records

Query



```
%sql select substr(Date, 6,2) as Months, substr(Date, 0,5) as Year,Landing_Outcome,Booster_Version,Launch_Site  from spacextable
    where Landing_Outcome = 'Failure (drone ship)' and substr(Date,0,5)='2015'
```

Output

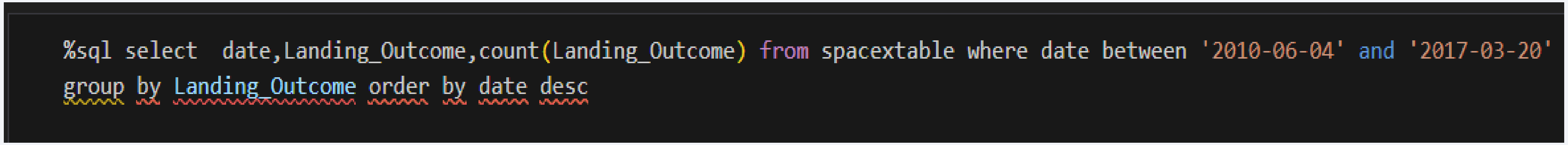


Months	Year	Landing_Outcome	Booster_Version	Launch_Site
01	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Using the substr function retrieves the year and month from the date variable and filtering the records to display only failures in 2015 year.

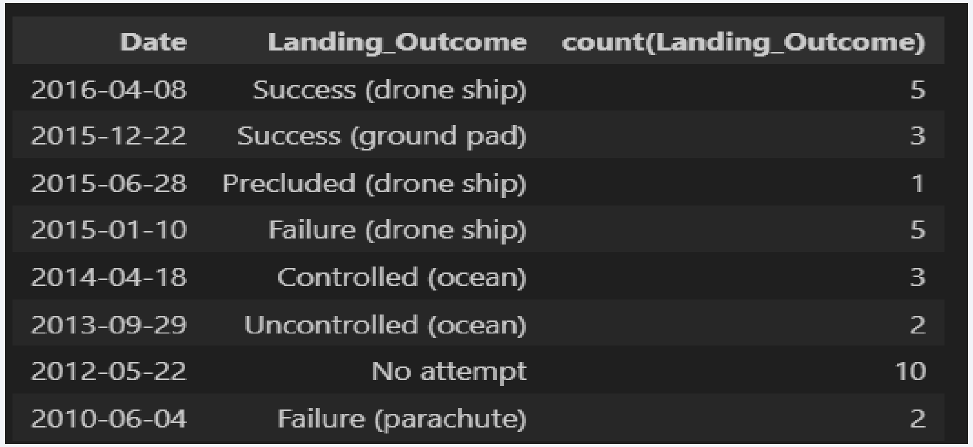
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

Query



```
%sql select  date,Landing_Outcome,count(Landing_Outcome) from spacextable where date between '2010-06-04' and '2017-03-20'
group by Landing_Outcome order by date desc
```

Output



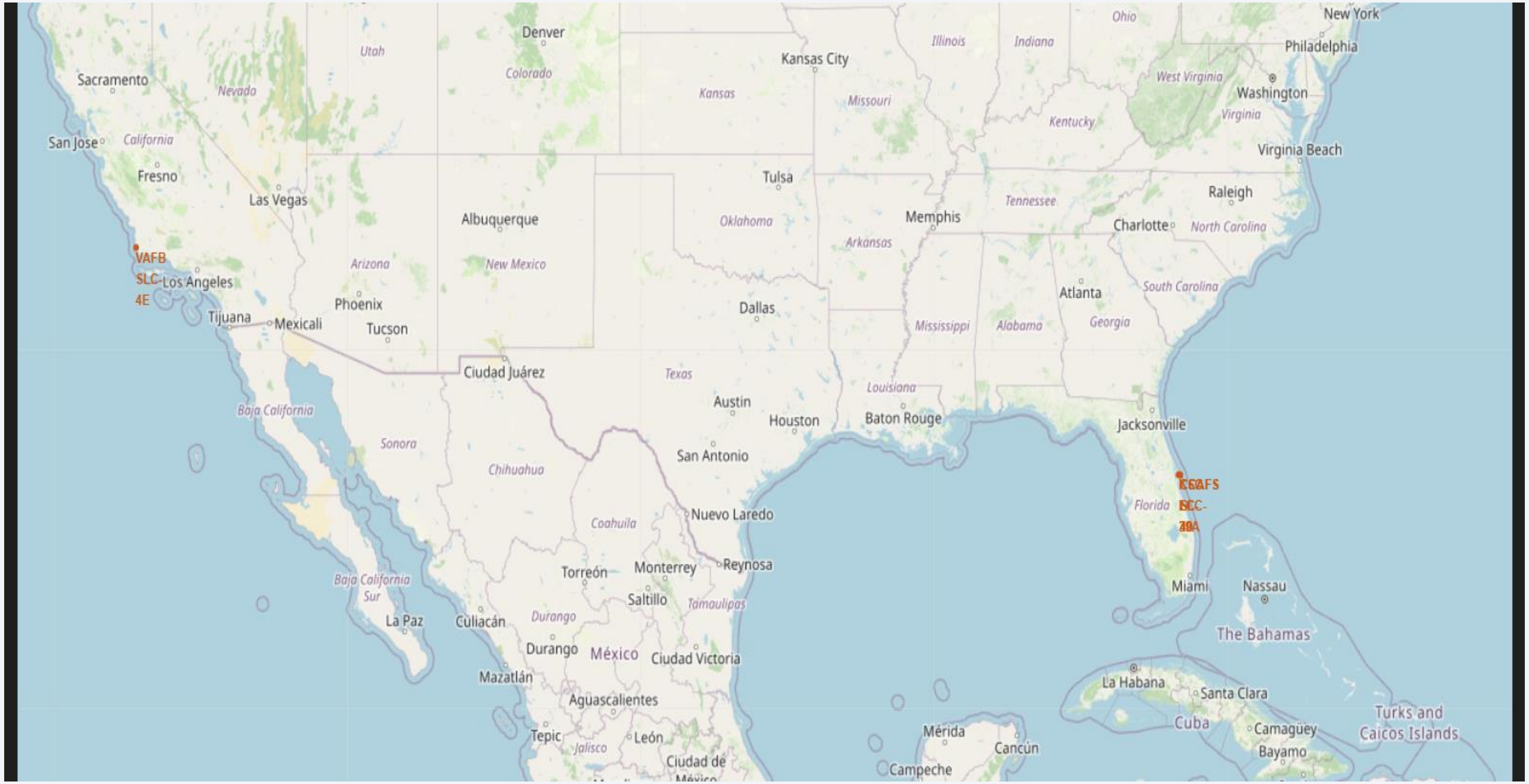
Date	Landing_Outcome	count(Landing_Outcome)
2016-04-08	Success (drone ship)	5
2015-12-22	Success (ground pad)	3
2015-06-28	Precluded (drone ship)	1
2015-01-10	Failure (drone ship)	5
2014-04-18	Controlled (ocean)	3
2013-09-29	Uncontrolled (ocean)	2
2012-05-22	No attempt	10
2010-06-04	Failure (parachute)	2

- Fetching the Landing outcome and their corresponding count using the count and group by clause
- Filtering only the records which are launched from 2010-06-04 till 2017-03-20. Sorting the records from recent to old.

Section 3

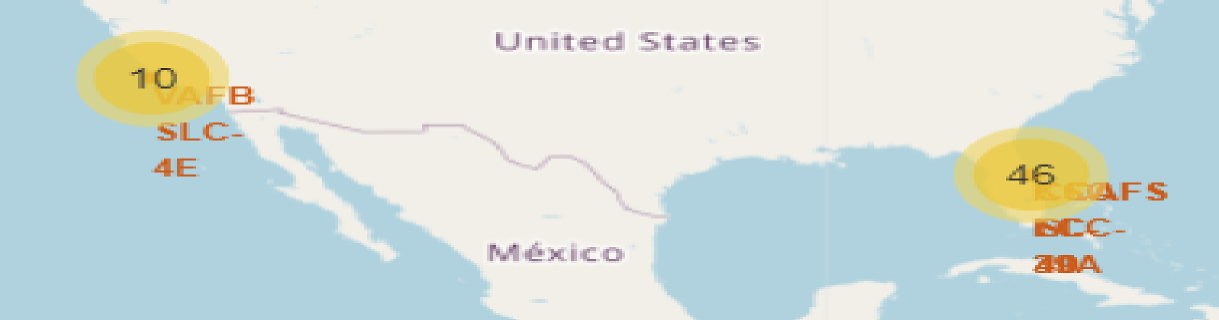
Launch Sites Proximities Analysis

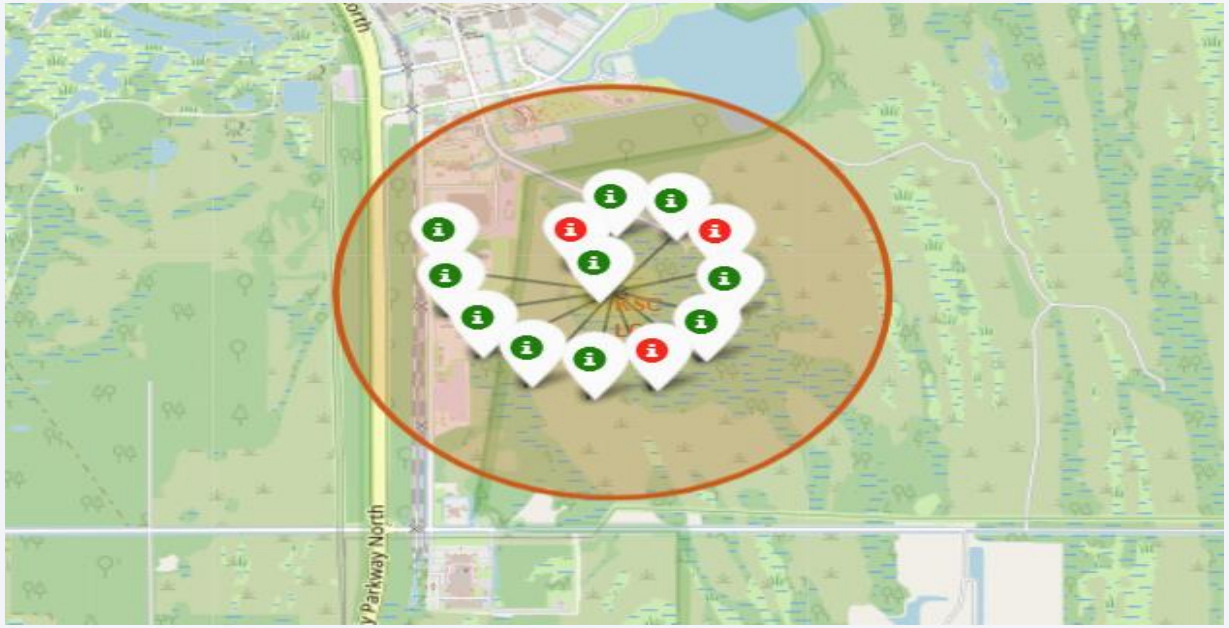
Launch site locations



- All the launch sites are in the coastal areas
- Of the four, three launch sites are close to each other.

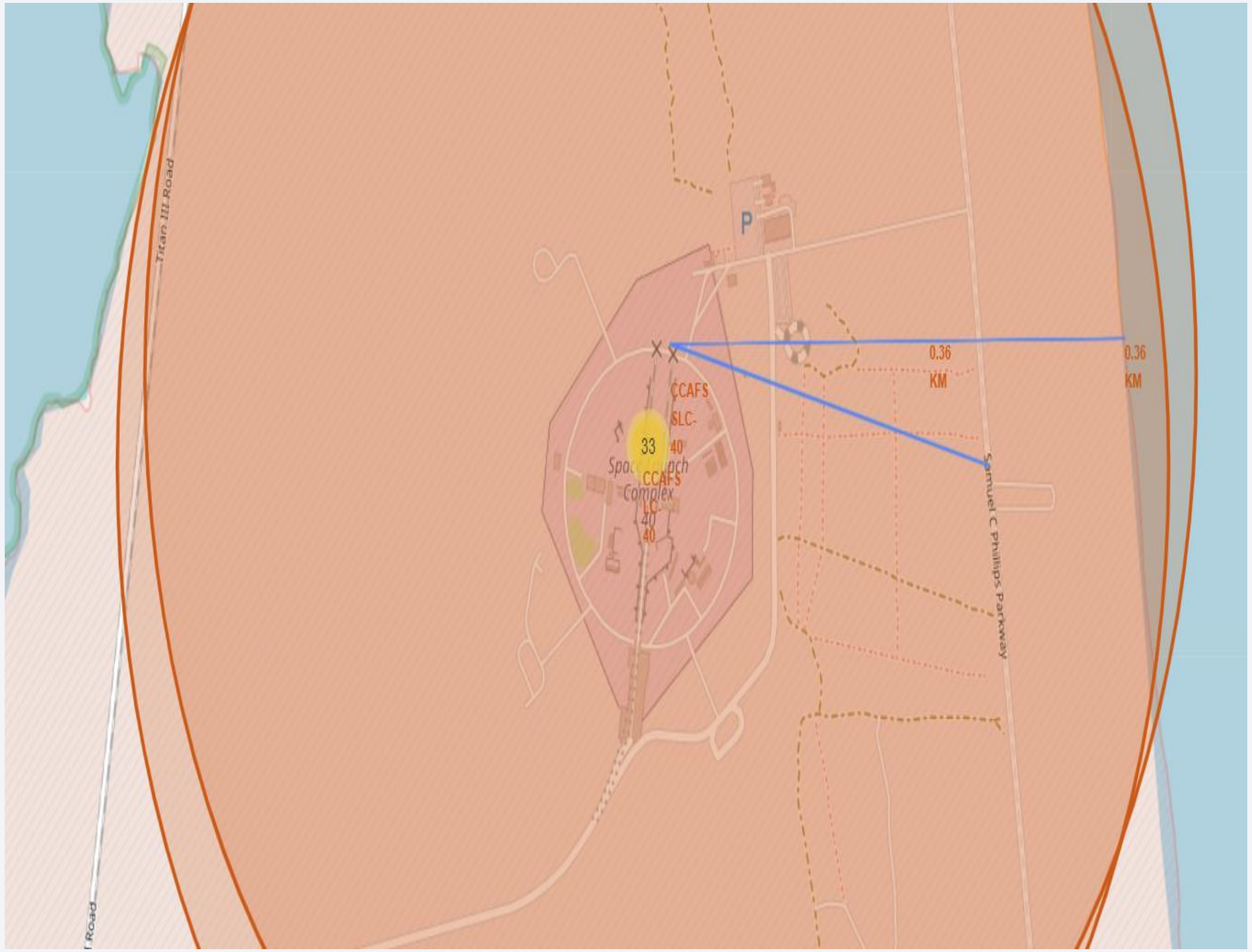
Number of Success/Failed return launches



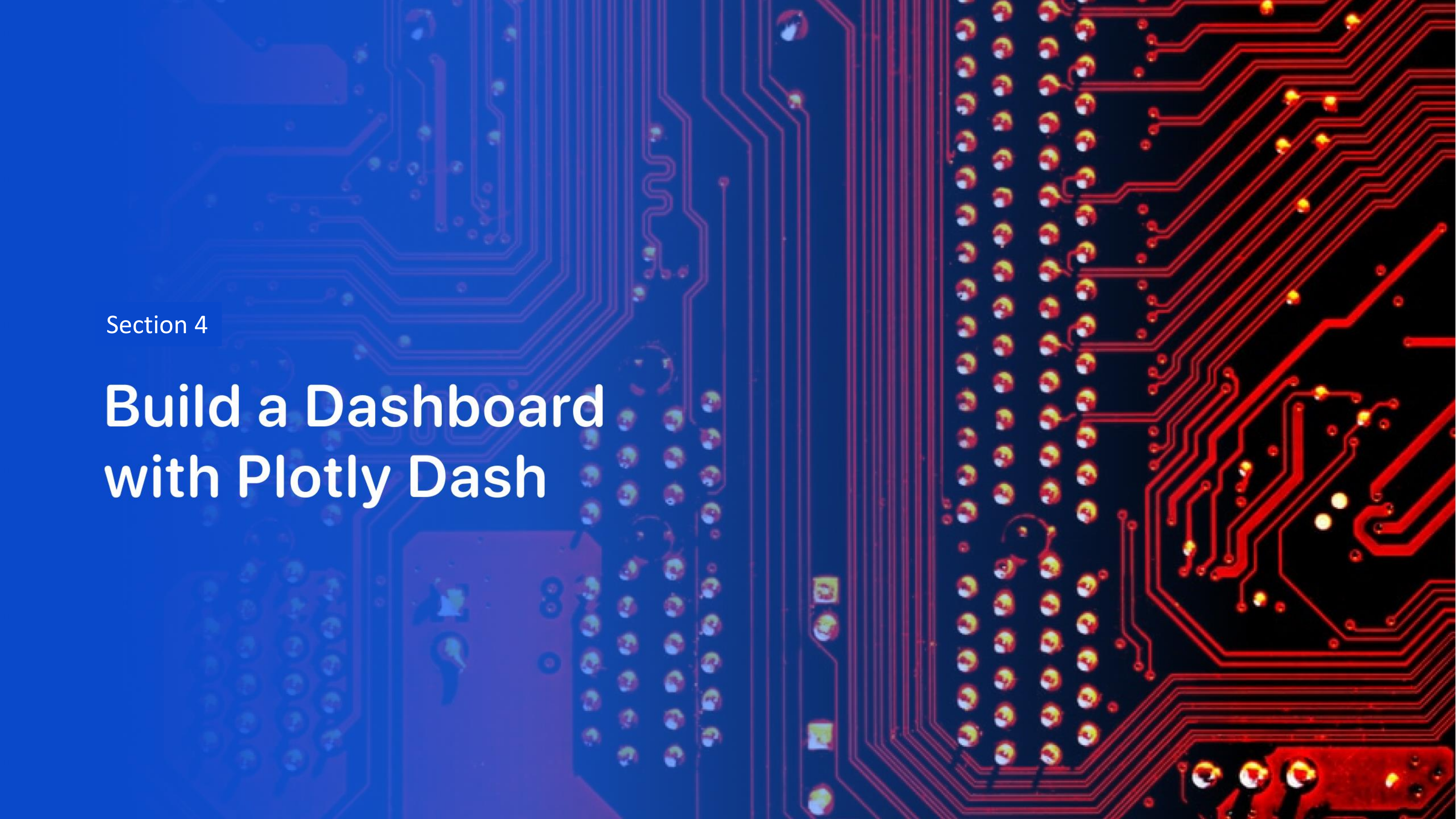


- Displayed the total number of rocket launches from each location
- Upon selecting the individual launch site, we the actual success and failed launches
  - Red icon indicated the failures
  - Green icon indicates the successful launches

Proximity of launch sites



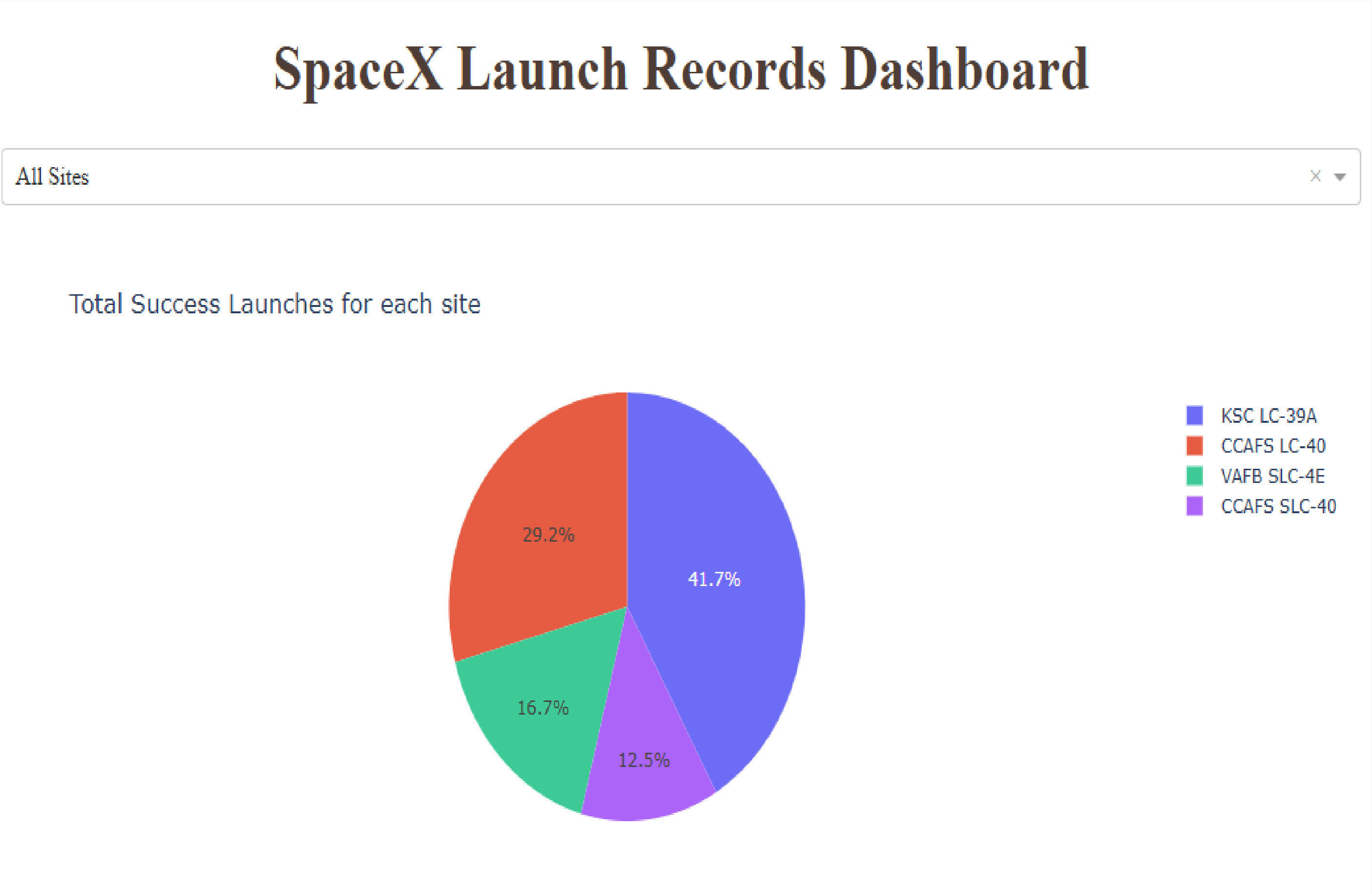
- Created a polyline from the launchsite to near by coastal line and highway.
- Calculated the distance in KM’s



Section 4

Build a Dashboard with Plotly Dash

Success rate of all each Launch site



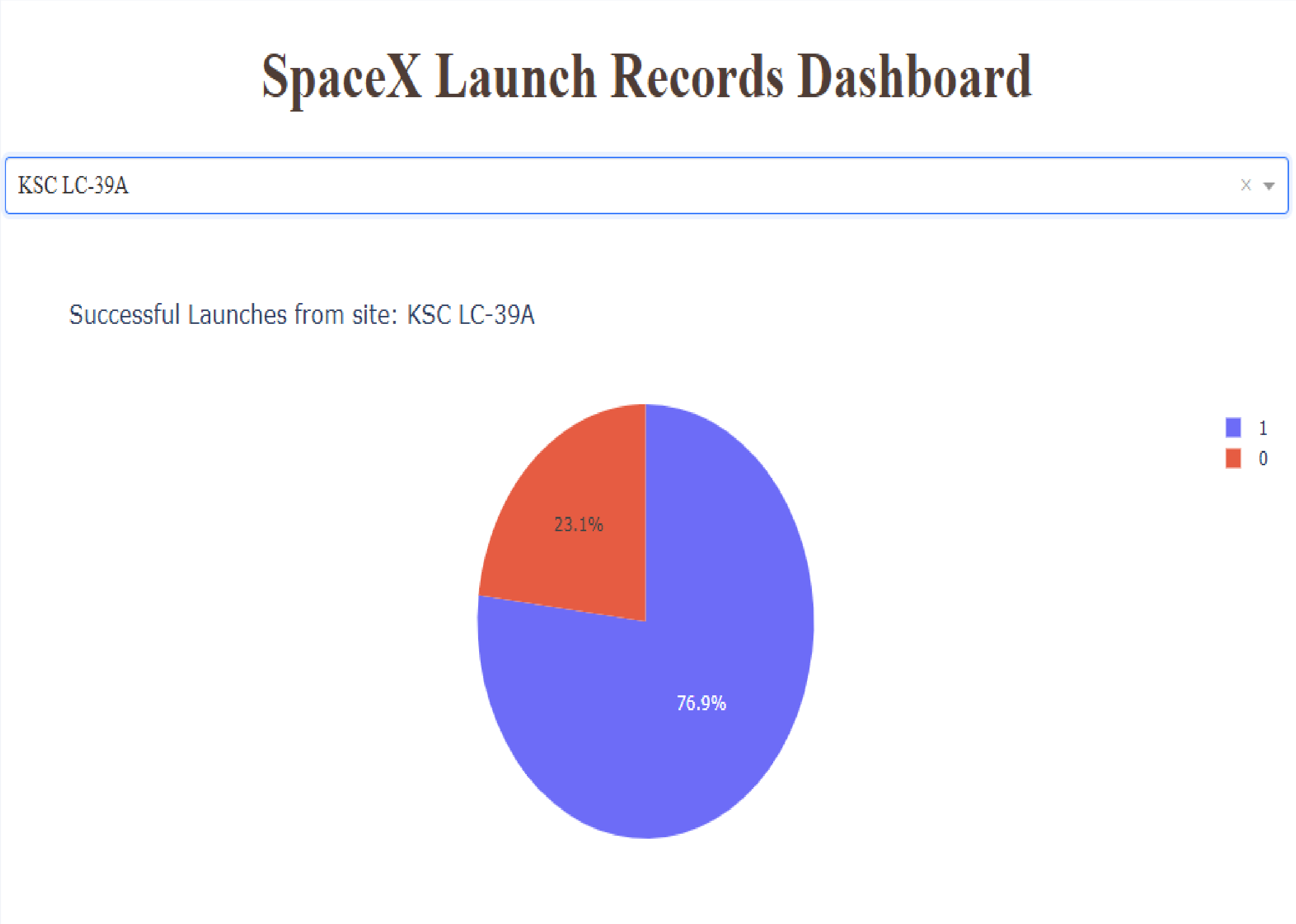
SpaceX Launch Records Dashboard

All Sites

Total Success Launches for each site

- The pie chart displays the total success rate of each launch site
- KSC LC-39A site has a higher success ratio 41.7%

Launch site with highest success ratio



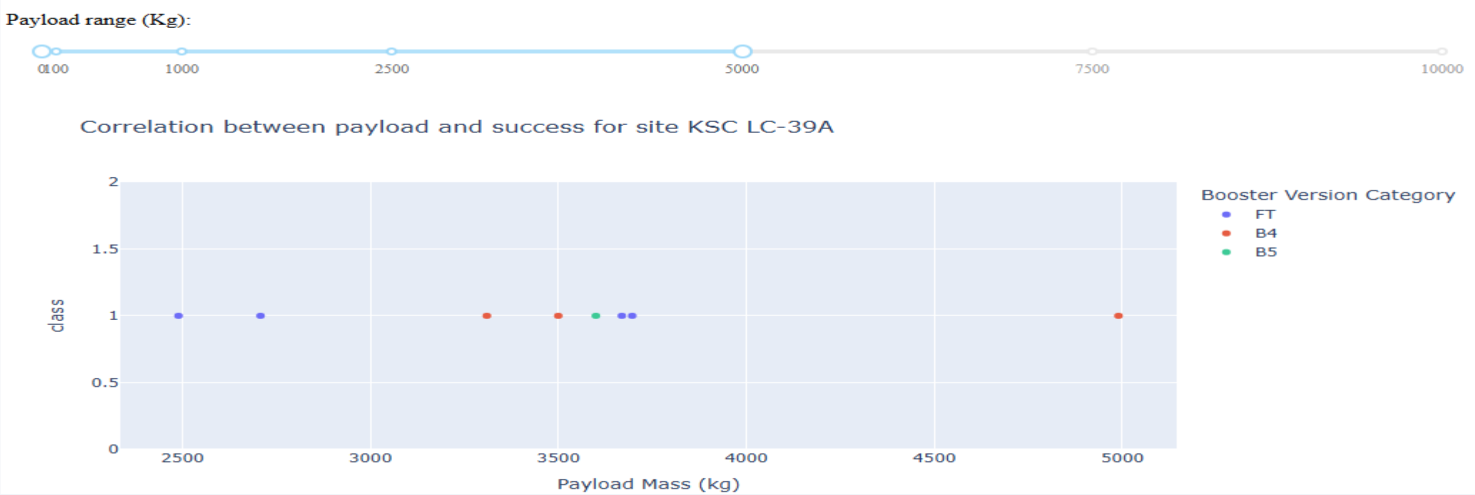
SpaceX Launch Records Dashboard

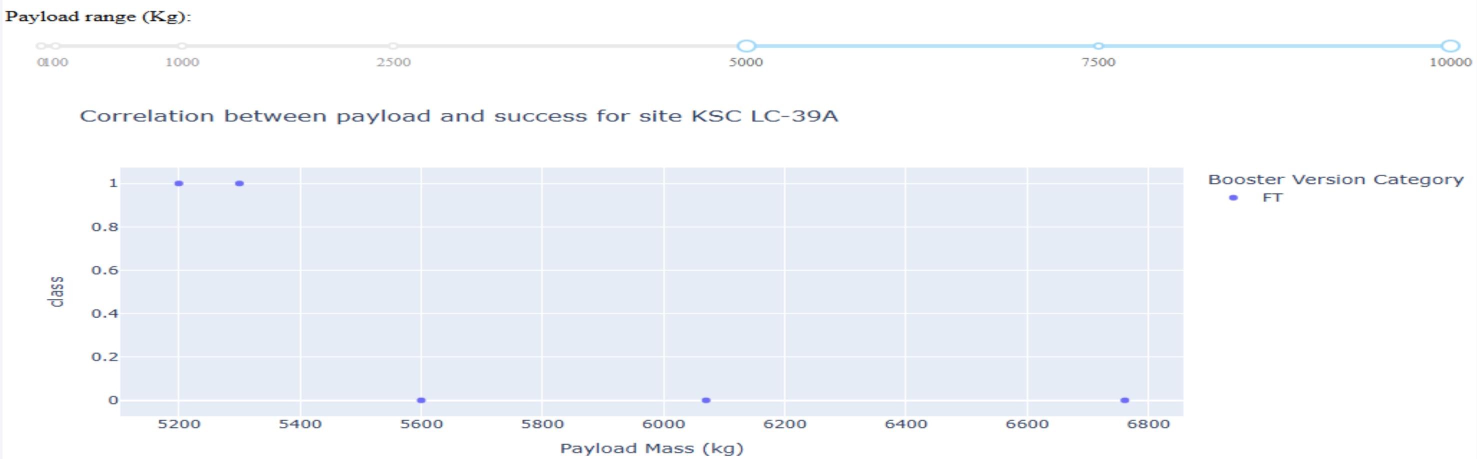
KSC LC-39A

Successful Launches from site: KSC LC-39A

- Almost 77% of the launches were successful for site KSC LC-39A

Relationship between Payload and outcome





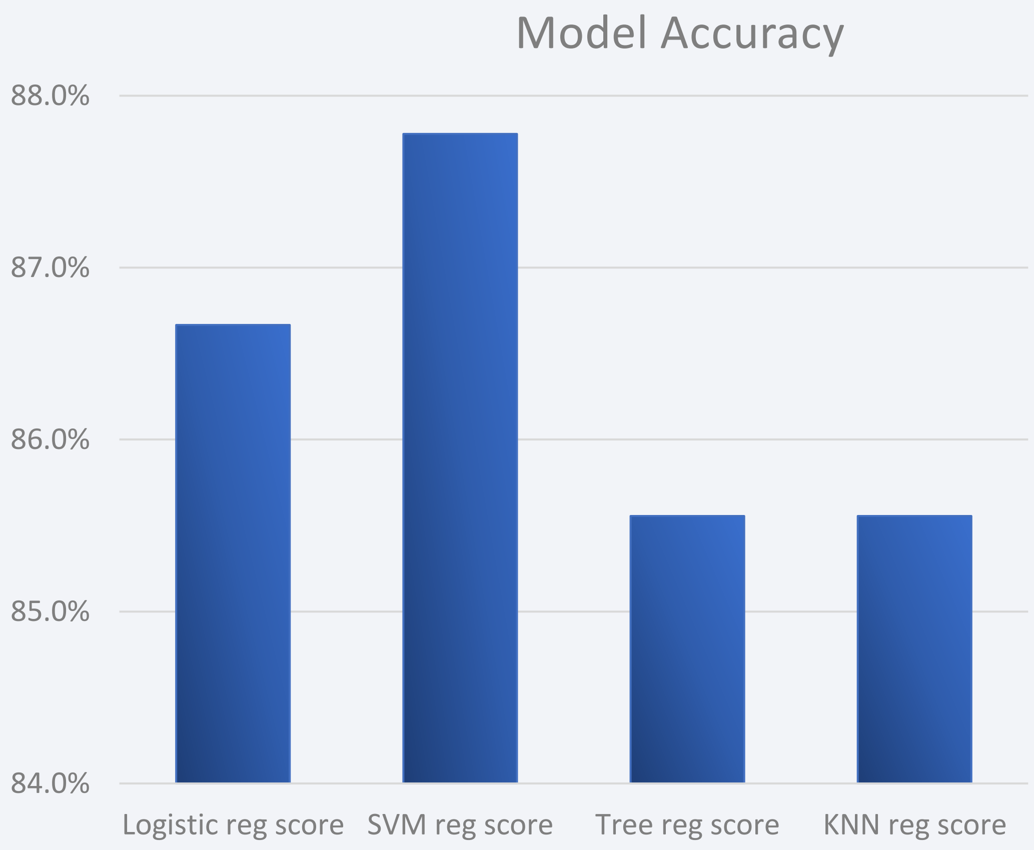
- These scatter plot explains the correlation between payload and their outcome
- First chart we have payload range 0-5000
- Second chart we have payload range 5000-max

Section 5

Predictive Analysis (Classification)

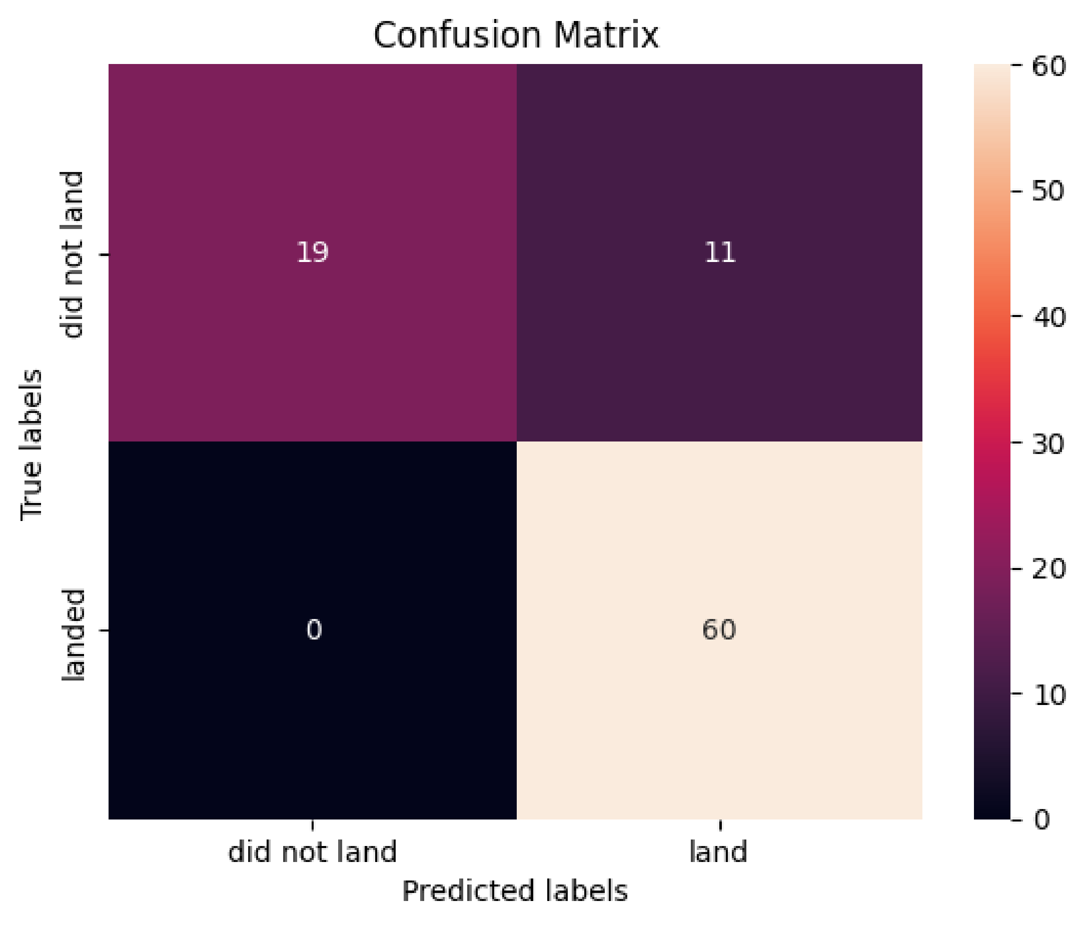
Classification Accuracy

- Trained the model and tuned the hyper parameters for four different models using train data
  - Logistic regression, SVM, Decision Tree, KNN
- Validate the results with test data
- Train the model again with entire data set
- Evaluate the performance using Accuracy method
- SVM, is the best model



Confusion Matrix

- Out of 90 records SVM correctly classified the 79 records
  - True Positive – 60
  - True Negative – 19
- 11 records falls in False positive Type I error those Model predicted as Landed but those were failed in real.
- From this confusion matrix we can derive Precision, Recall & F1 Score
  - Precision = TP/(TP+FP) = 60/(60+11) = 84.5
  - Recall = TP/(TP+FN) = 60/(60+0) = 100
  - F1-Score = 91.59

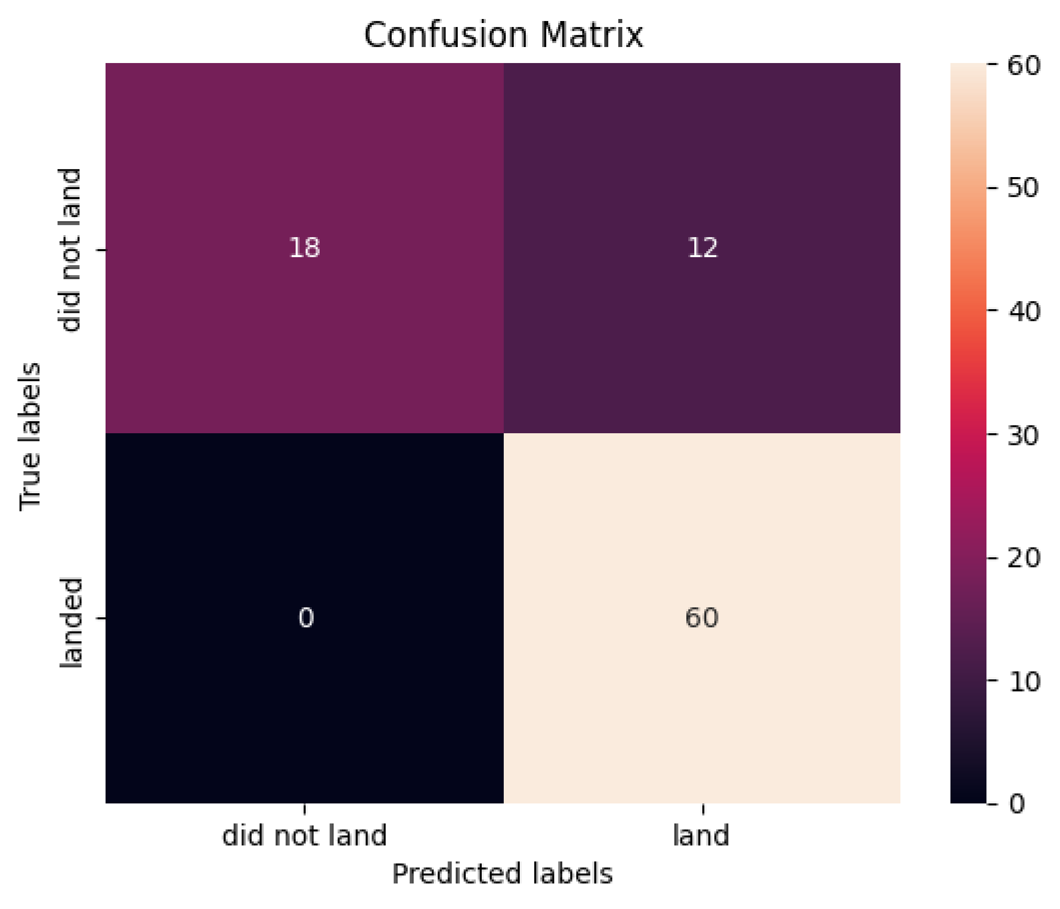


Conclusions

- Even though SpaceX struggled initially to successfully land the first stage of the return rockets in the subsequent launches they got more success.
- There is a strong relation between success rate with payload mass and the orbit of the launch.
- All the launch sites were on coastal areas.
- 41.7% of the successful launches are from KSC LC launch site.
- Out of all the launches from KSC LC site 77% gave success.
- Possibly because of the low volume of data SVM performed better than other classification algorithms.
- After hyper parameter tuning, we got approximately 88% accuracy in SVM.
- On further review of confusion matrix, we noticed that
  - Precision = 84.5
  - Recall = 100
  - F1-Score = 91.6

Appendix

- The accuracy of SVM & Logistic regression are very similar.
- So, trained the logistic regression on full data and tuned its parameter.
- The confusion matrix shows the result.
- Compared to SVM we got 1 additional record in False positive. 11 in SVM vs 12 in Logistic regression.
- Hence the accuracy also dropped.



Thank you!