IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

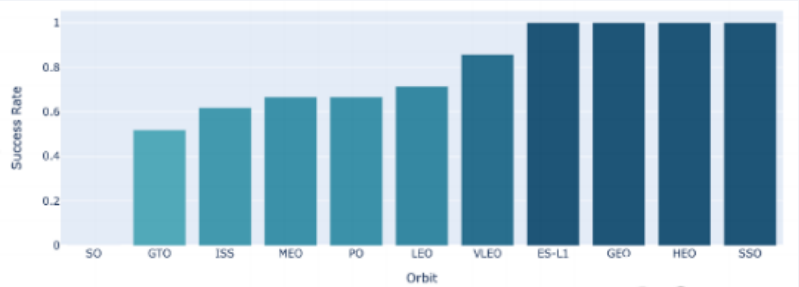
Swaraj Singh Rawat
07-12-2021

Outline

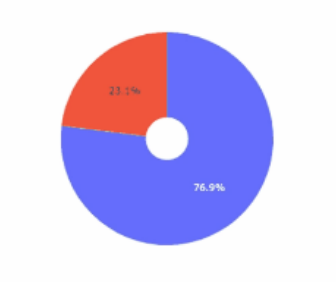
- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

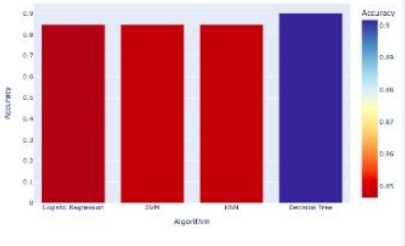
- Summary of methodologies
  Data collection via API, SQL and Web Scraping
  Data wrangling and analysis
  Interactive Maps with Folium
  Predictive Analysis for each classification model



- Summary of all results
  Data analysis along with interactive Visualizations
  Best model for Predictive analysis



Algorithm	Accuracy
Logistic Regression	0.846429
SVM	0.848214
KNN	0.848214
Decision Tree	0.901786



Introduction

- Project background and context
  Here we will predict if the Falcon 9 first stage will land successfully.
  SpaceX advertises Falcon9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollar each, much of the savings is because SpaceX can reuse its first stage. Therefore, if we can determine if the first stage will land successfully. This information can be used if an alternate company wants to bid against SpaceX for rocket launch
- Problems you want to find answers
  With what factors, the rocket will land successful?
  The effect of each relationship of rocket variables on outcome.
  Condition which will aid SpaceX have to achieve the best results.

Section 1

Methodology

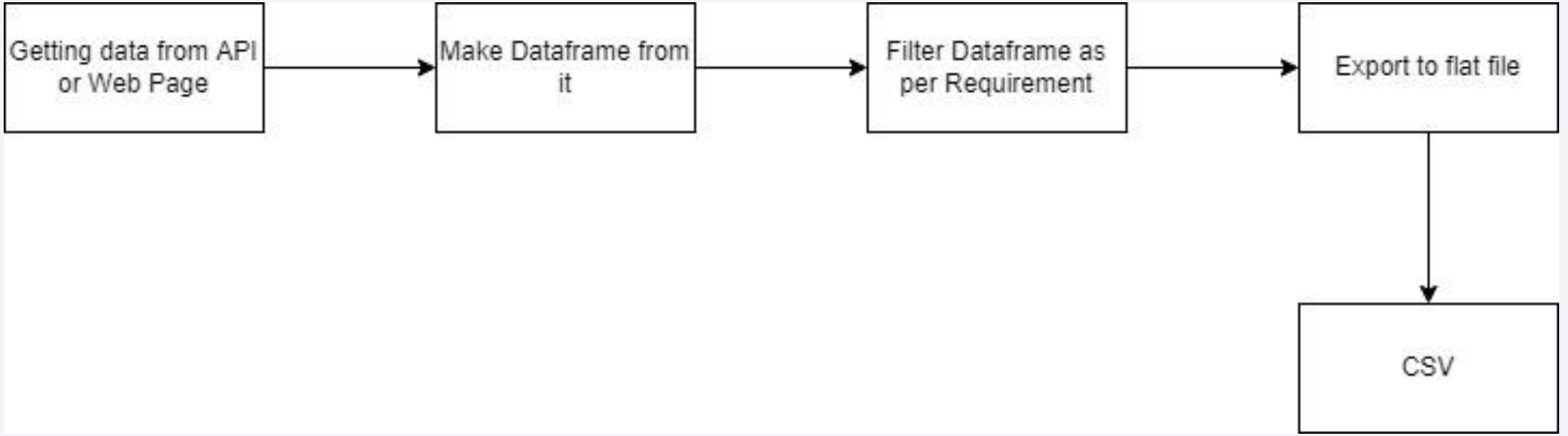
Methodology

Executive Summary

- Data collection methodology:
  - Via SpaceX Rest API
  - Web Scraping from Wikipedia
- Perform data wrangling
  - One hot encoding data fields for machine learning and dropping irrelevant columns (Transforming data for Machine Learning)
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Scatter and Bar graphs to show pattern between data
- Perform interactive visual analytics using Folium and Plotly Dash
  - Using Folium and Plotly dash visulaizations
- Perform predictive analysis using classification models
  - Build, tune, evaluate classification models

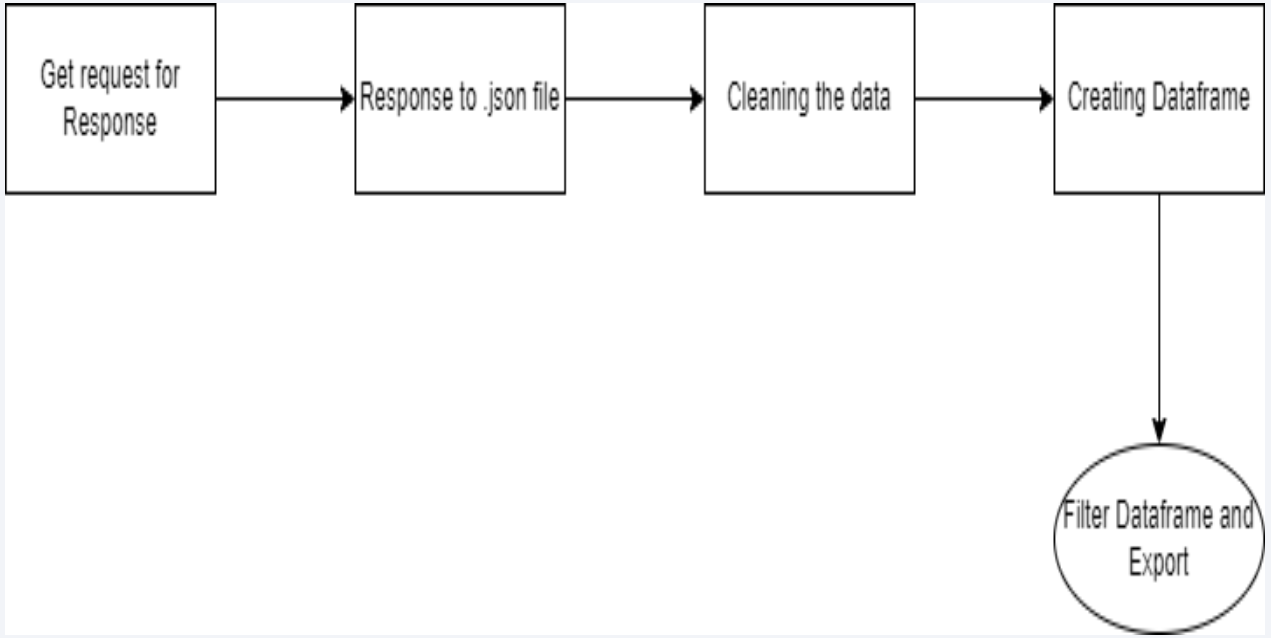
Data Collection

- Data Collection is the process of gathering and measuring information on targeting variables in an established system, which then enables one to answer relevant questions and evaluate outcomes



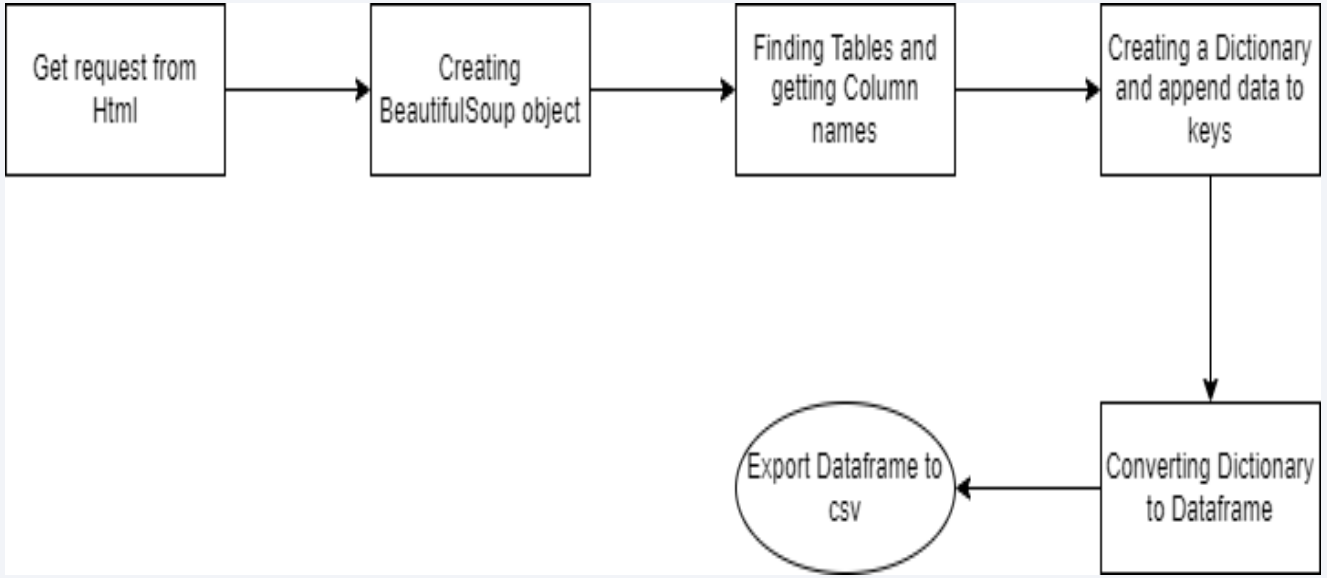
Data Collection – SpaceX API

- First, we create a response object by get request on SpaceX url.
- Then check the status and convert the response to json file.
- Then we apply custom function to clean the data and saved it to a dataframe.
- Then we filter the dataframe and export the file
- URL:- Dala Collection API file



Data Collection - Scraping

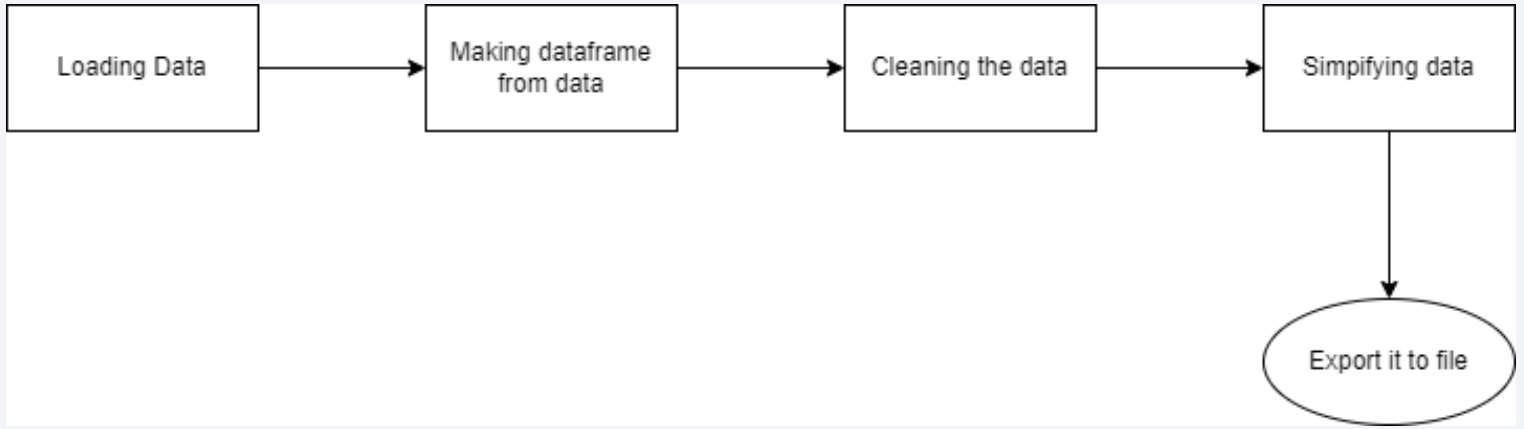
- First, Getting response from html
- Then, creating a BeautifulSoup object
- Then, find the table and Column names.
- Then, Creating a dictionary and append data to keys
- Converting data to dataframe and export it to csv
- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose



Web Scraping file

Data Wrangling

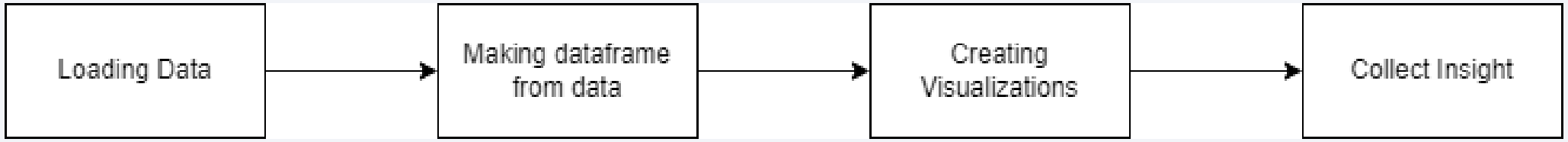
- Data wrangling is the process of cleaning, structuring and enriching raw data into a desired format for better decision making in less time.
- Converted the data to 0 (for bad outcomes) and 1 (for good outcomes)
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose



Data Wrangling file

EDA with Data Visualization

- Data wrangling is the process of cleaning, structuring and enriching raw data into a desired format for better decision making in less time.



- Scatter Plot:-
  Payload vs Flight Number
  Flight Number vs Launch Site
  Payload vs Launch Site
  Flight Number and Orbit type
  Payload and Orbit type

Scatter plots' primary uses are to observe and show relationships between two numeric variables. The dots in a scatter plot not only report the values of individual data points, but also patterns when the data are taken as a whole.

EDA with Data visualization file

EDA with SQL

- SQL (Structured Query Language) is used for performing various operations on the data stored in the databases like updating records, deleting records, creating and modifying tables, views, etc. SQL is also the standard for the current big data platforms that use SQL as their key API for their relational databases.
- We are using IBM's db2 for cloud, which is fully managed SQL Database provided as a service.
- `!pip install sqlalchemy==1.3.9`
- `!pip install ibm_db_sa`
- `!pip install ipython-sql`
  `%load_ext sql`
  `%sqlibm_db_sa://my-username:my-password@my-hostname:my-port/my-db-name`
  `%sql <your-query>`

EDA with SQL

We performed SQL queries to gather information from given dataset:

- ***Display the names of the unique launch sites in the space mission***
- ***Display 5 records where launch sites begin with the string 'CCA'***
- ***Display the total payload mass carried by boosters launched by NASA (CRS)***
- ***Display average payload mass carried by booster version F9 v1.1***
- ***List the date when the first successful landing outcome in ground pad was acheived.***
- ***List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000***
- ***List the total number of successful and failure mission outcomes***
- ***List the names of the booster_versions which have carried the maximum payload mass. Use a subquery***
- ***List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015***
- ***Rank 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***

Build an Interactive Map with Folium

- Folium is a Python library used for visualizing geospatial data. It is easy to use and yet a powerful library. Folium is a Python wrapper for Leaflet.js which is a leading open-source JavaScript library for plotting interactive maps.

Map Objects	Code	Result
Map Marker	folium.Marker(	Map object to make a mark on map
Icon Marker	folium.Icon(	Create an icon on map
Circle Marker	folium.Circle(	Create a circle where marker is being placed
Polyline	folium.Polyline(	Create a line between points
Marker Cluster Object	MarkerCluster(	This is a good way to simplify a map containing many markers having the same coordinate
AntPath	Folium.plugins.AntPath(	Create a animated line between points

Map with folium file

Build a Dashboard with Plotly Dash

- Pie chart showing the total success for all sites or by certain launch site
  Percentage of success in relation to launch site

- Scatter plot showing the correlation between payload and success for all sites or by certain launch site
  It shows the relationship between success rate and Booster version category

Map Objects	Code	Result
Dash and its Components	import dash import dash_html_components as html import dash_core_components as dcc From dash.dependencies import input,Output	Plotly stewards python's leading data viz and UI libraries. With dash open source, dash app runs on your local laptop or server. The dash core component library contains a set of higher level components like sliders, graphs, dropdowns, tables, and more dash provides all the available HTML tags as user friendly python classes
pandas	import pandas as pd	Fetching values from csv and creating dataframe
plotly	Import plotly.express as px	Plot a graphs with interactive plotly library
Dropdown	dcc.Dropdown(	Create a dropdown for launch sites
Rangeslider	Dcc.RangeSlider(	Create a rangeslider for payload mass range selection
Pie chart	Px.pie(	Creating a pie graph for success percentage display
Scatter plot	Px.scatter(	Creating the scatter graph for correlation display

Plotly file

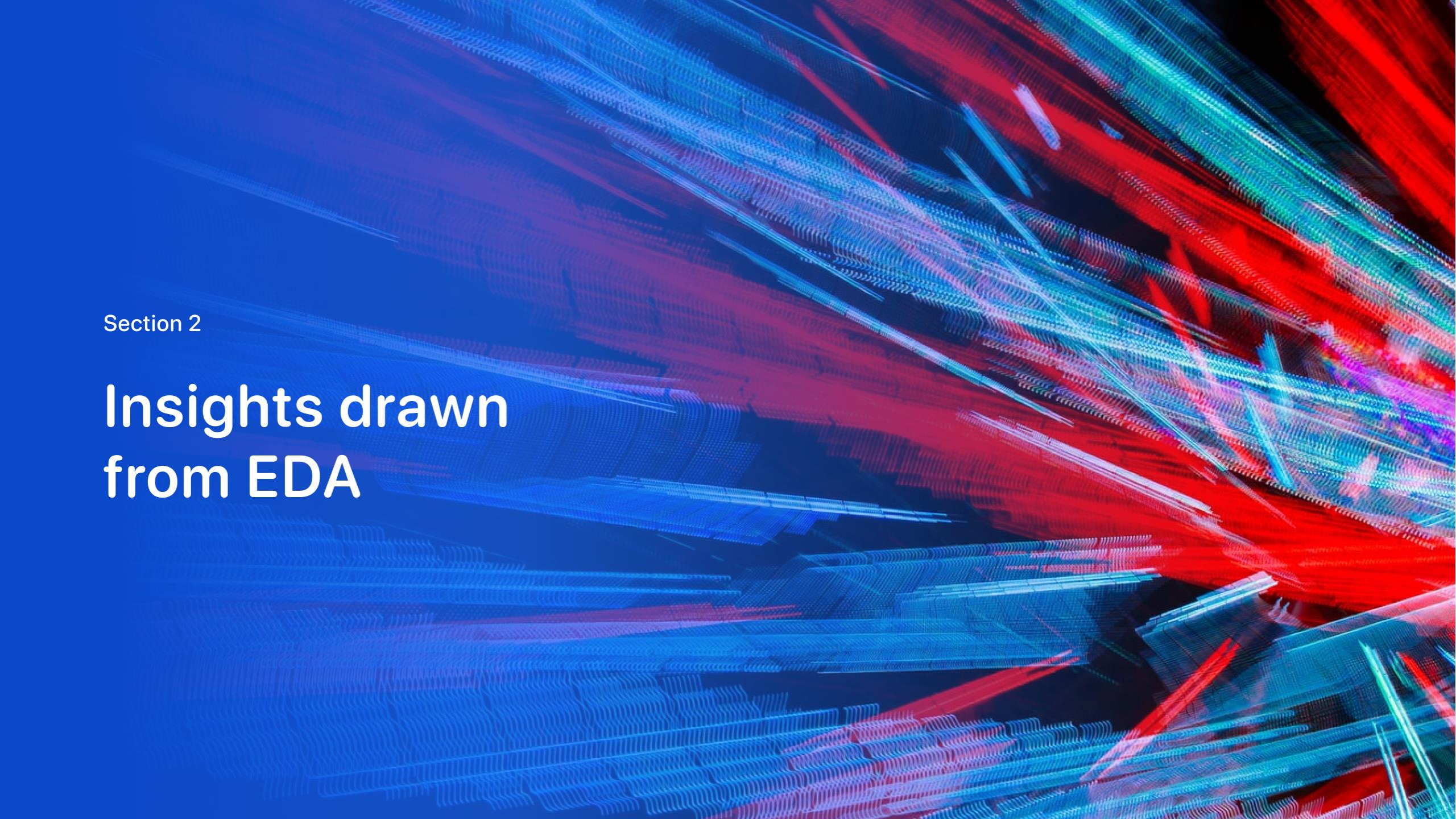
Predictive Analysis (Classification)

Building Model	Evaluating Model	Finding the Best Model
• Load our feature engineered data into dataframe • Transform it to numpy arrays • Standardize the transformed data • Split data into training and test data sets • Check how many test data samples has been created • List down machine learning algorithm we want to use • Set our parameters and algorithm to GridSearchCV objects and train our model	• Check accuracy for each model • Get best hyperparameters for each type of algorithms • Plot confusion matrix	• The model with best accuracy score wins the best performing model

Machine learning file

Results

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

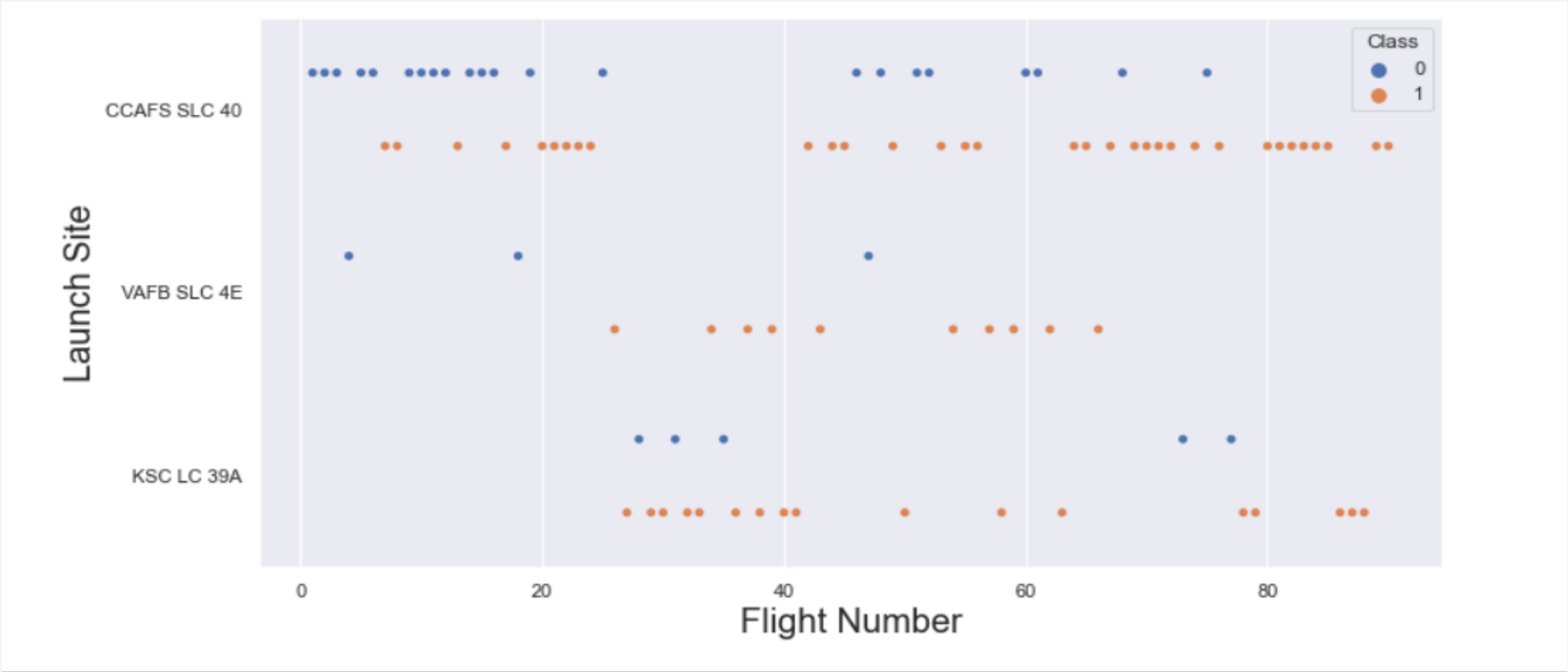


Section 2

Insights drawn from EDA

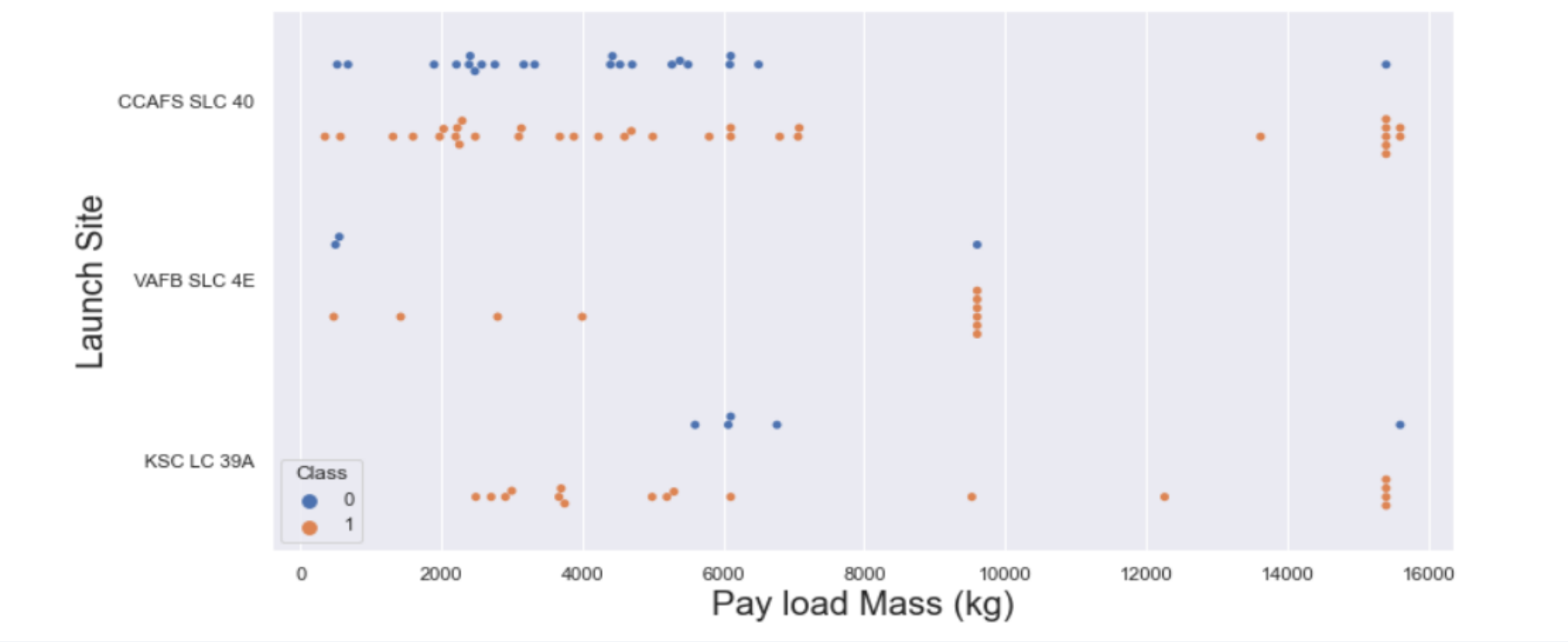
Flight Number vs. Launch Site

- With high Flight number (Greater than 30) the success rate for the Rocket is increasing



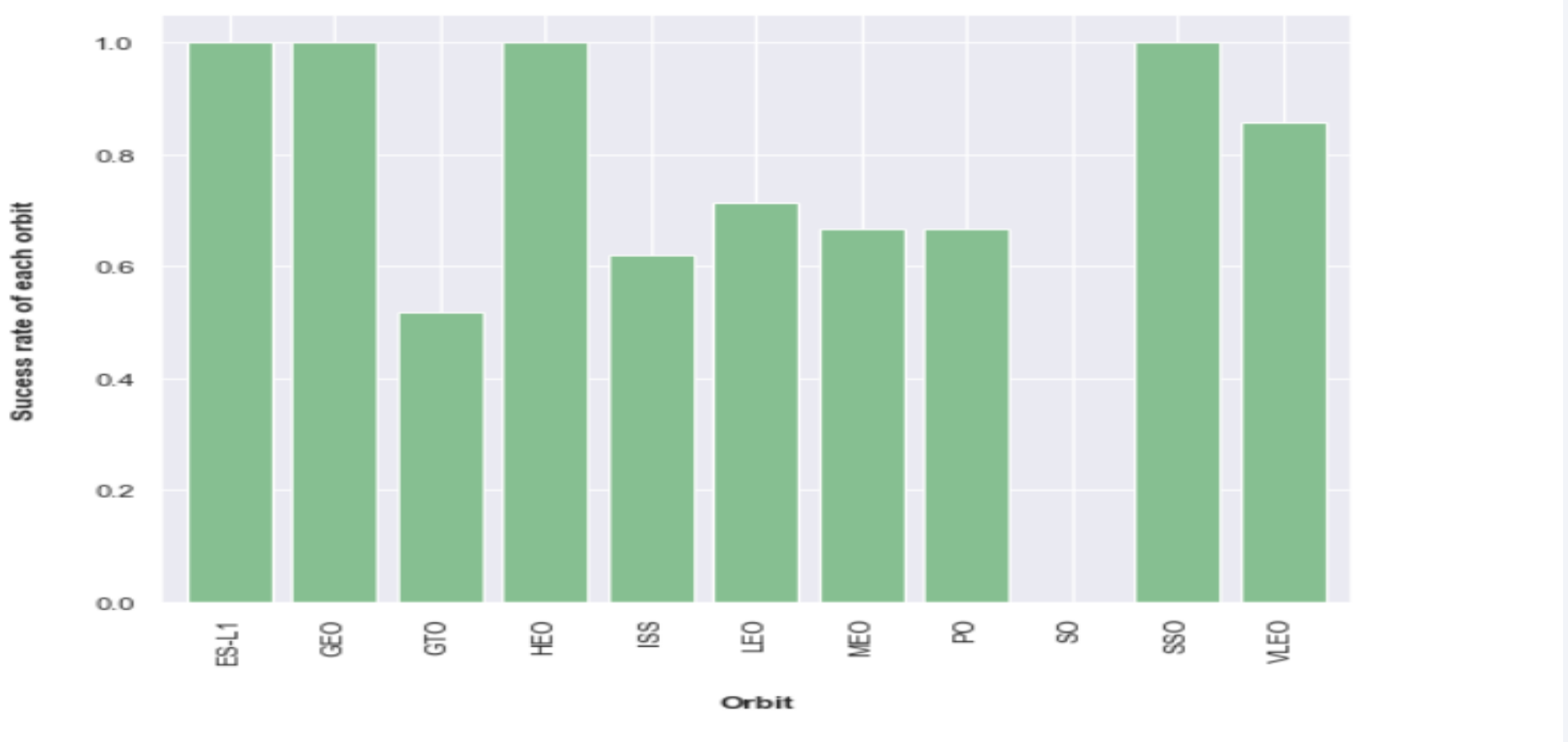
Payload vs. Launch Site

- The greater the payload mass(greater than 7000kg) higher the success rate for the Rocket. But there is no clear patter to take a decision



Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO has high success rates.



Flight Number vs. Orbit Type

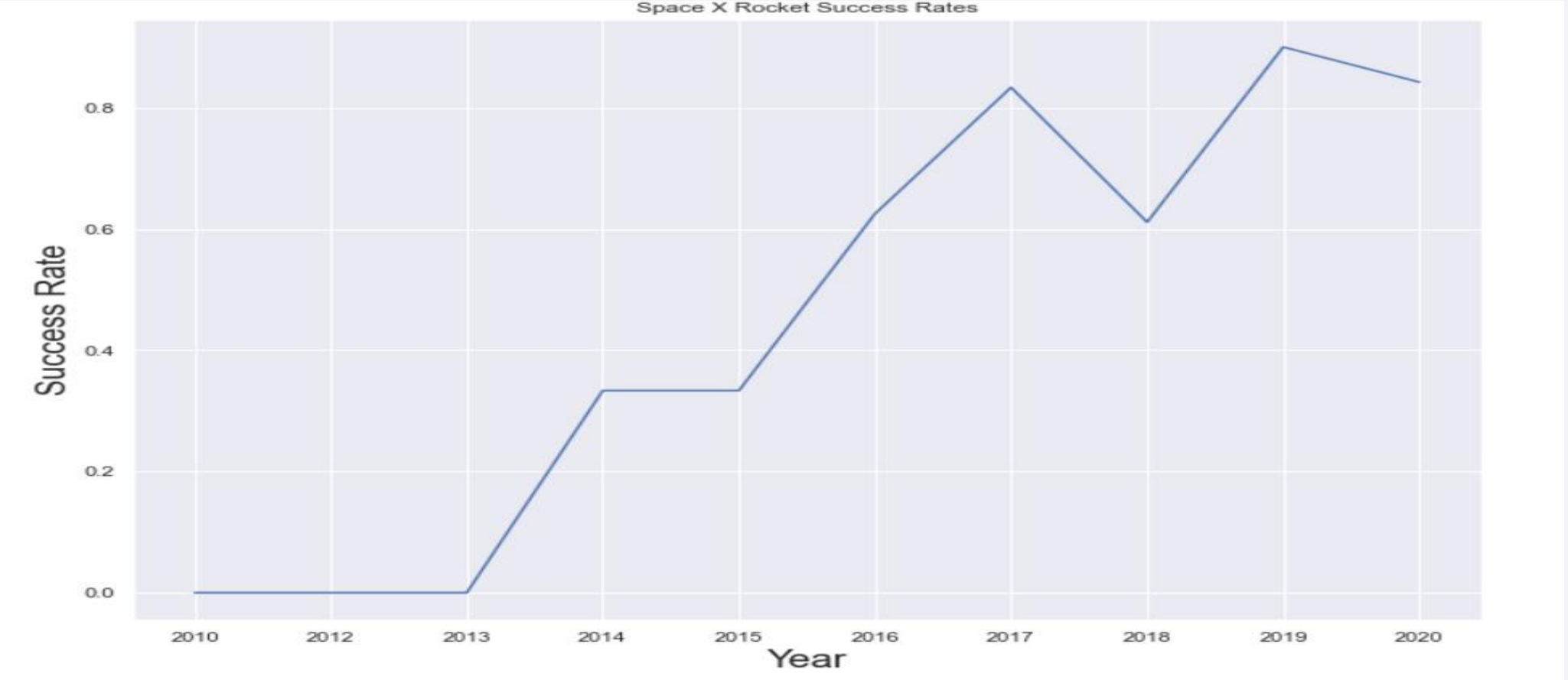
- We see that for LEO orbit the success rate increases with the number of flights.
- There seems to be no relationship between flight number and the GTO orbit

Payload vs. Orbit Type

- Heavy payloads have a negative influence on MEO, GTO, VLEO orbits
- Positive on LEO, ISS orbits

Launch Success Yearly Trend

- Success rate increases since 2013 however there is slight dip after 2019



All Launch Site Names

- Using the word DISTINCT in the query we pull unique values for Launch Site column from the table SPACEX

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
```

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

Launch Site Names Begin with 'CCA'

- Using keyword LIMIT5 in the query we fetch 5 records from table spacex and with condition LIKE keyword with wild card :- CCA% . The percentage in the end suggests that the launch_site name must start with CCA

```
%sql SELECT * FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- Using the function SUM, calculates the total in the column PAYLOAD_MASS_KG and WHERE clause filters the data to fetch Customer's by name 'NASA (CRS)'.

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)" FROM SPACEX WHERE CUSTOMER = 'NASA (CRS)';
```

Total Payload Mass by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

- Using the function AVG works out the average in the column PAYLOAD_MASS_KG_
- The WHERE clause filters the dataset to only perform calculations on Booster_version 'F9 v1.1'.

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster Version F9 v1.1" FROM SPACEX \
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

Average Payload Mass by Booster Version F9 v1.1
2928

First Successful Ground Landing Date

- Using the function MIN works out the minimum date in the column Date and WHERE clause filters the data to only perform calculations on Landing_outcome with value 'Success (ground pad)'

```
%sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad" FROM SPACEX \
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

First Succesful Landing Outcome in Ground Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Selecting only Booster_Version,
  WHERE clause filters the dataset to Landing_Outcome = Success (drone ship)
- AND clause specifies additional filter conditions
  PAYLOAD_MASS_KG_> 4000 AND Payload_Mass_KG_<6000

```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

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

Total Number of Successful and Failure Mission Outcomes

- Selecting multiple count is a complex query. I have used case clause within sub query for getting both success and failure counts in same query.
- Case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end returns a Boolean value which we sum to get the result needed

```
%sql SELECT sum(case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end) AS "Successful Mission", \
    sum(case when MISSION_OUTCOME LIKE '%Failure%' then 1 else 0 end) AS "Failure Mission" \
FROM SPACEX;
```

Successful Mission	Failure Mission
100	1

Boosters Carried Maximum Payload

- Using the function MAX works out the maximum payload in the column PAYLOAD_MASS_KG_ in sub query.
- WHERE clause filters Booster Version which had that maximum payload

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEX \
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

Booster Versions which carried the Maximum Payload Mass
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

- We need to list the records which will display the month names, failure, landing_outcomes in drone ship, booster versions, launch site for the month in year 2015
  via year function we extract the year and future where clause 'Failure (drone ship)' fetches our required values.
- Also, I am using (fn MONTHNAME(DATE)) to get the month name

```
%sql SELECT {fn MONTHNAME(DATE)} as "Month", BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE year(DATE) = '2015' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

Month	booster_version	launch_site
January	F9 v1.1 B1012	CCAFS LC-40
April	F9 v1.1 B1015	CCAFS LC-40

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

- Selecting only Landing__Outcome.
  Where clause filters the data with DATE between '2010-06-04' and '2017-03-20'
- Grouping by LANDING__OUTCOME
  Order by COUNT(LANDING__OUTCOME) in descending order.

```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

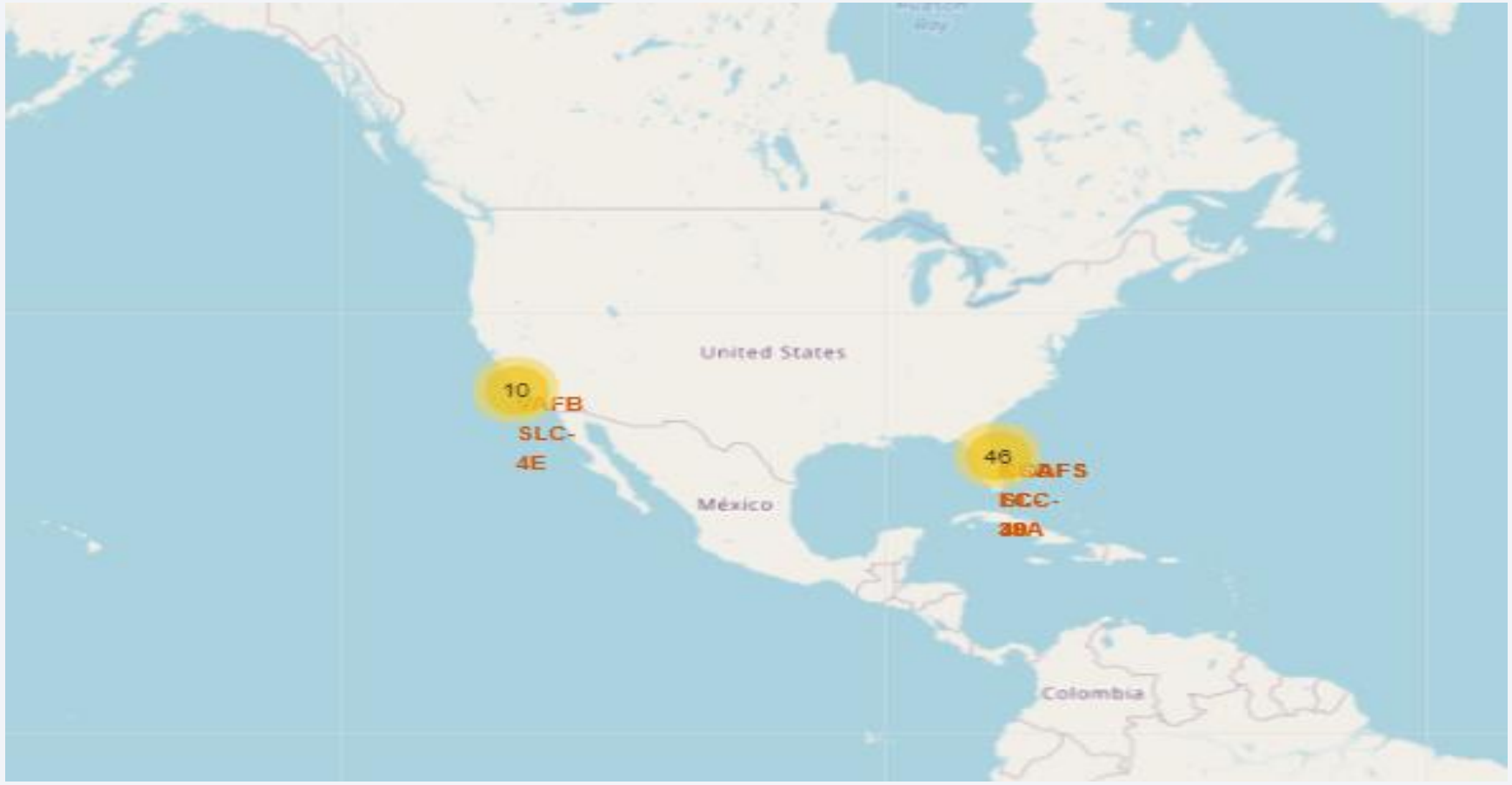
Landing Outcome	Total Count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

Launch Sites Proximities Analysis

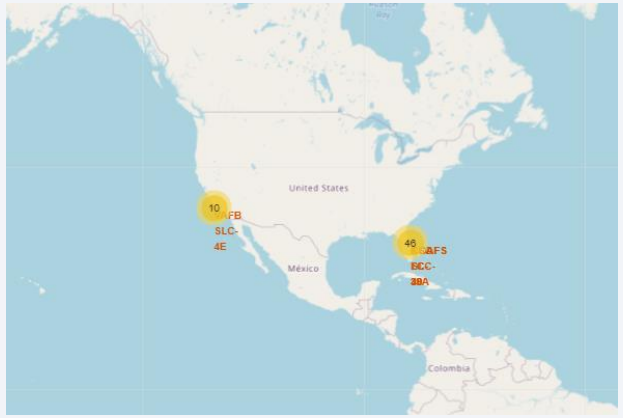
All Launch site on Folium map

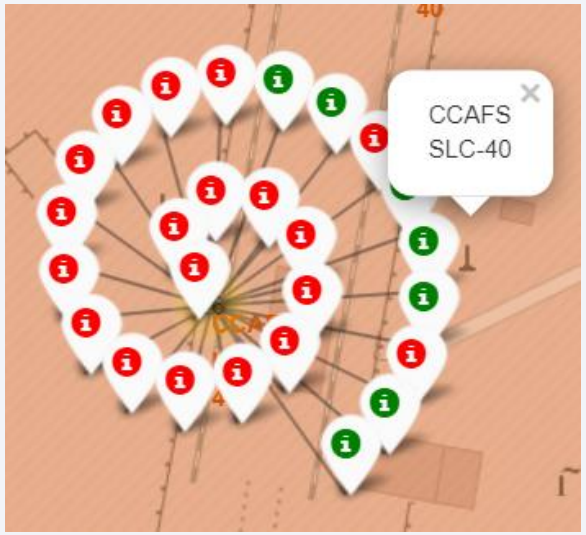
- We can see that the SpaceX launch sites are near to the United states of America coast i.e, Florida and California regions.

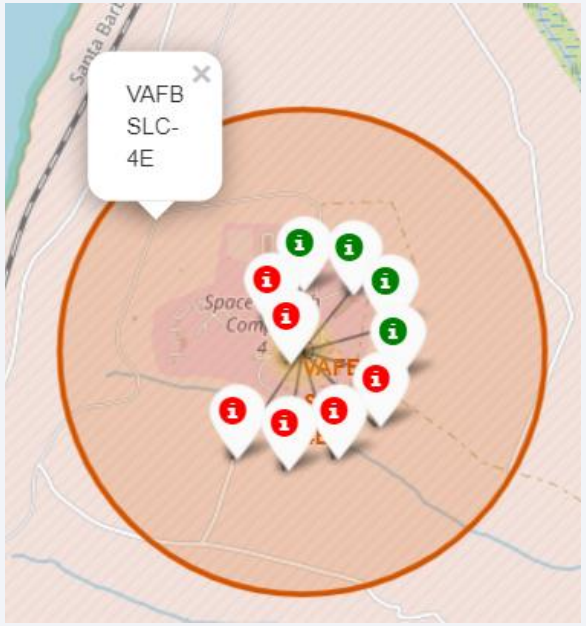


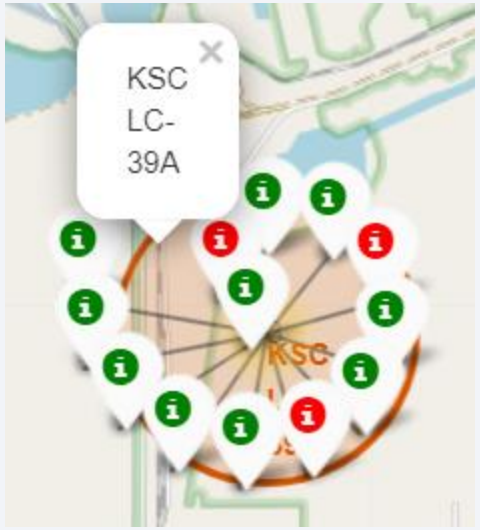
Color labeled launch Records

- Green marker shows successful launches and Red marker shows failures. From these screenshots its easily understandable that KSC LC-39A has the maximum probability to success.



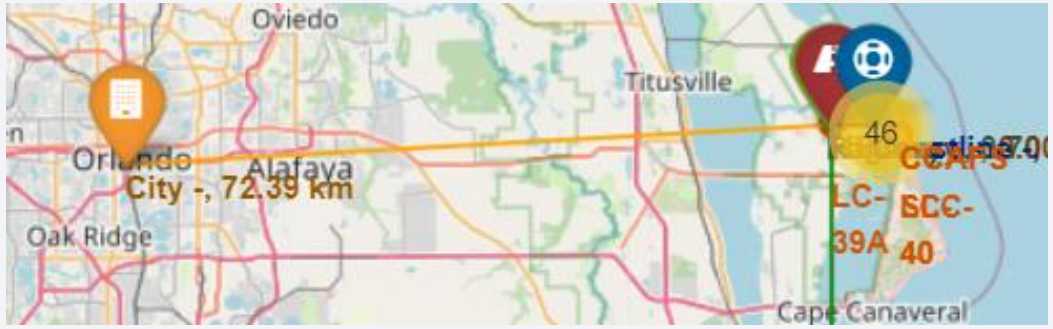


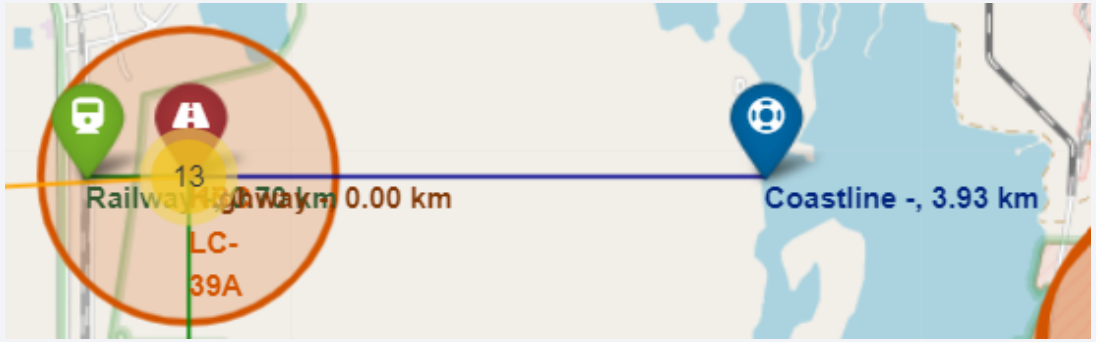




Launch site distance from Coastline & cities

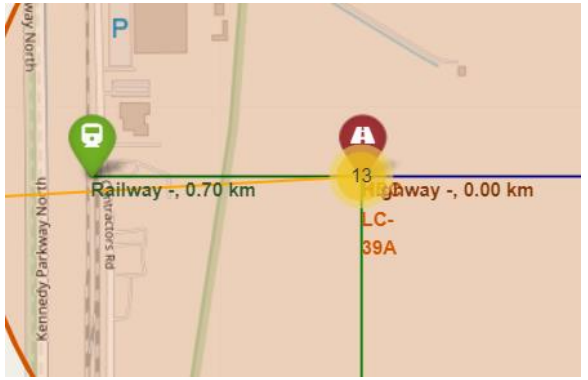
- Distance from all launch sites from cites is greater than 14km for all sites, So, launch sites are far away from cities

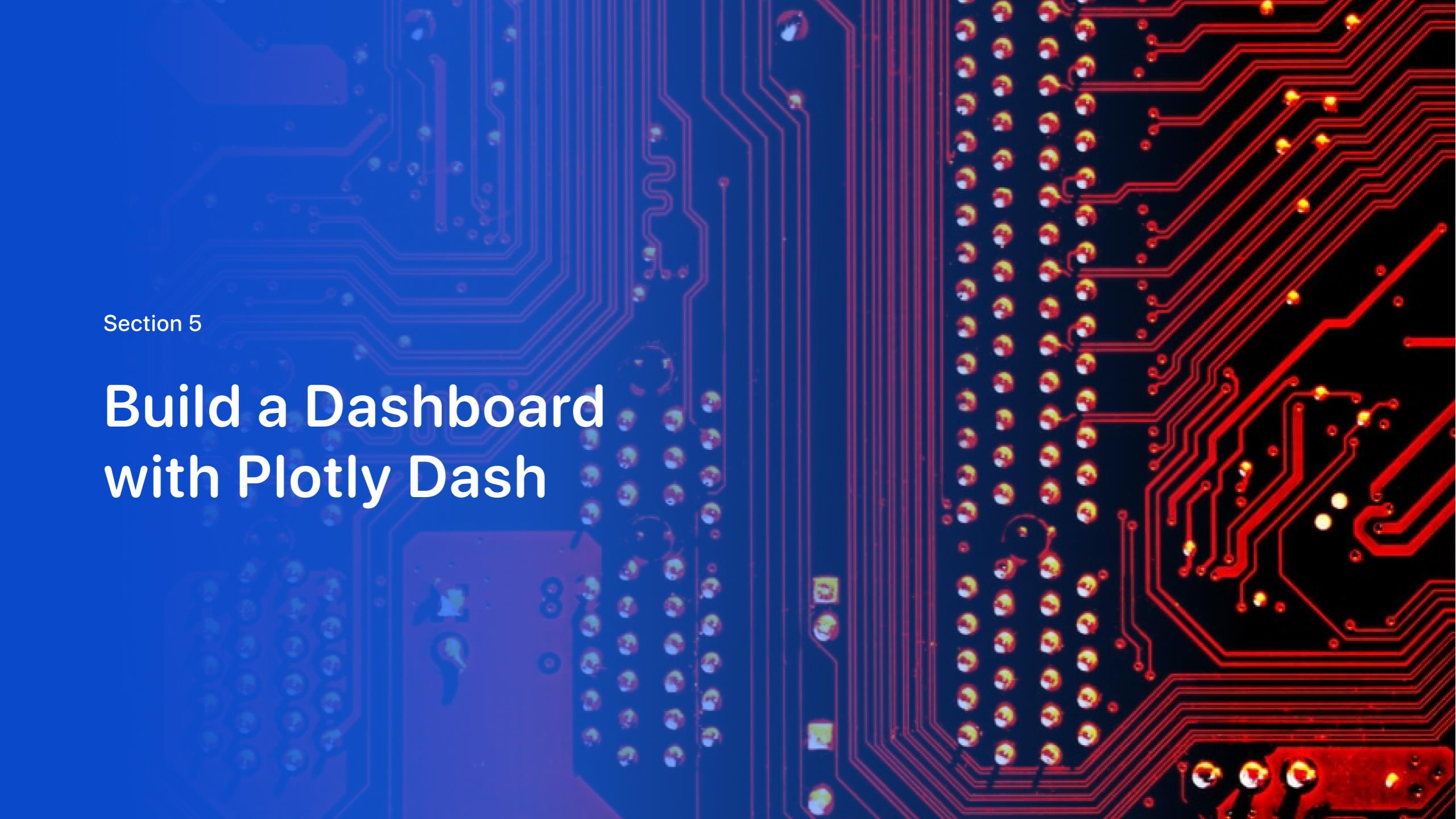




Launch site distances from Equator & Railways

- Distance from equator is 3000km for all sites
- Distance for all launch sites from railway track are greater than 0.7km for all sites. So, launch sites are not so far away from railway tracks





Section 5

Build a Dashboard with Plotly Dash

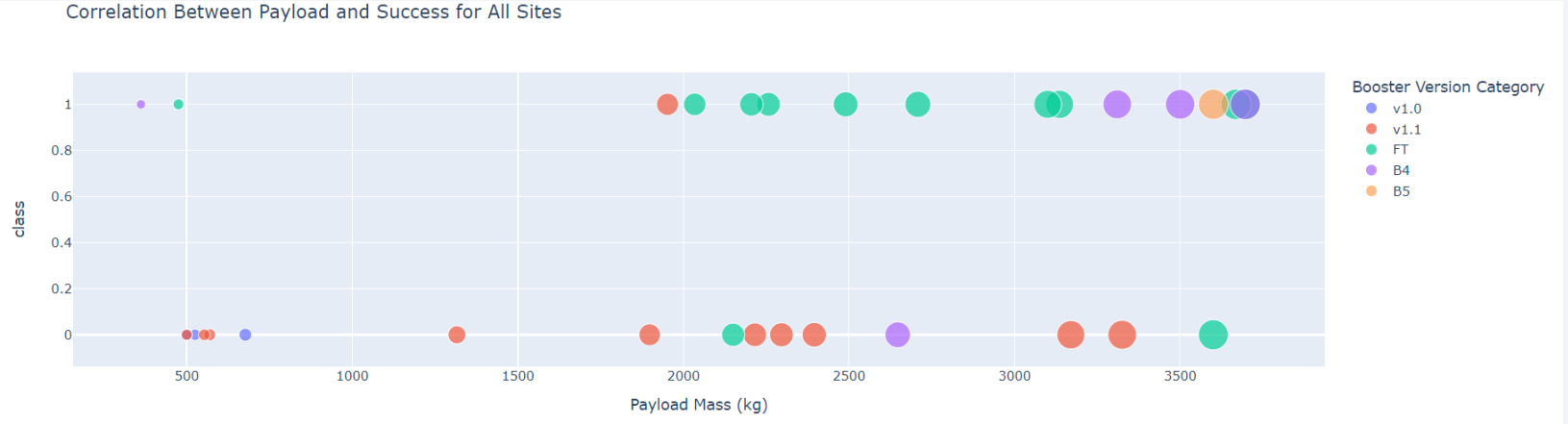
Launch Success count for all sites

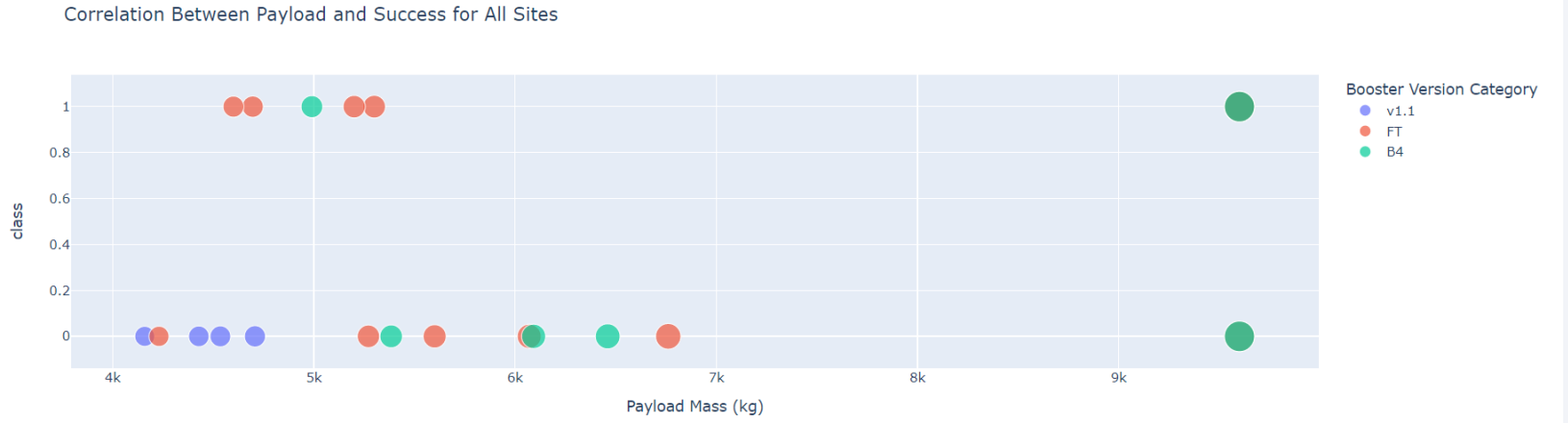
- We can see that KSC LC-39A had the most successful launches from all the sites

[Plotly file]

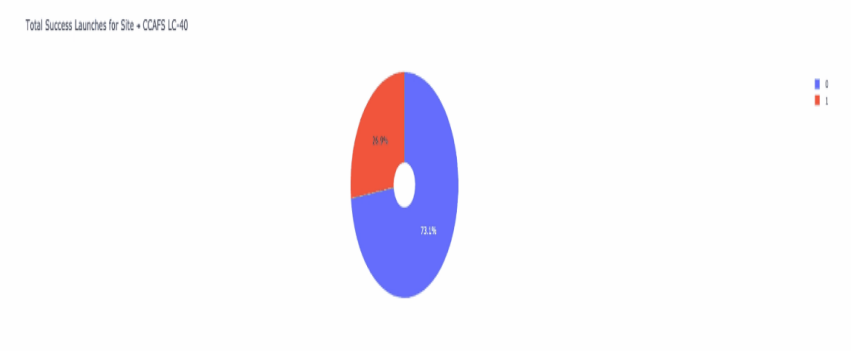
Payload vs Launch Outcomes Scatter Plot for all sites

- We can see the success rates for all low weighted payloads is higher than the heavy weighted payloads.





Launch Site with Highest Launch success ratio



- KSC LC –39A achieved a 79.6% success rate while getting a 23.1% failure rate.
- After visual analysis using the dashboard, we are able to obtain some insights to answer these questions:
- Which site has the highest launch success rate?
  KSC LC –39A
- Which payload range(s) has the highest launch success rate?
  2000kg - 10000kg
- Which payload range(s) has the lowest launch success rate?
  0-1000kg
- Which F9 Booster version (v1.0, v1.1, FT, B4, B5 etc.) has the highest launch success rate?
  FT

Section 6

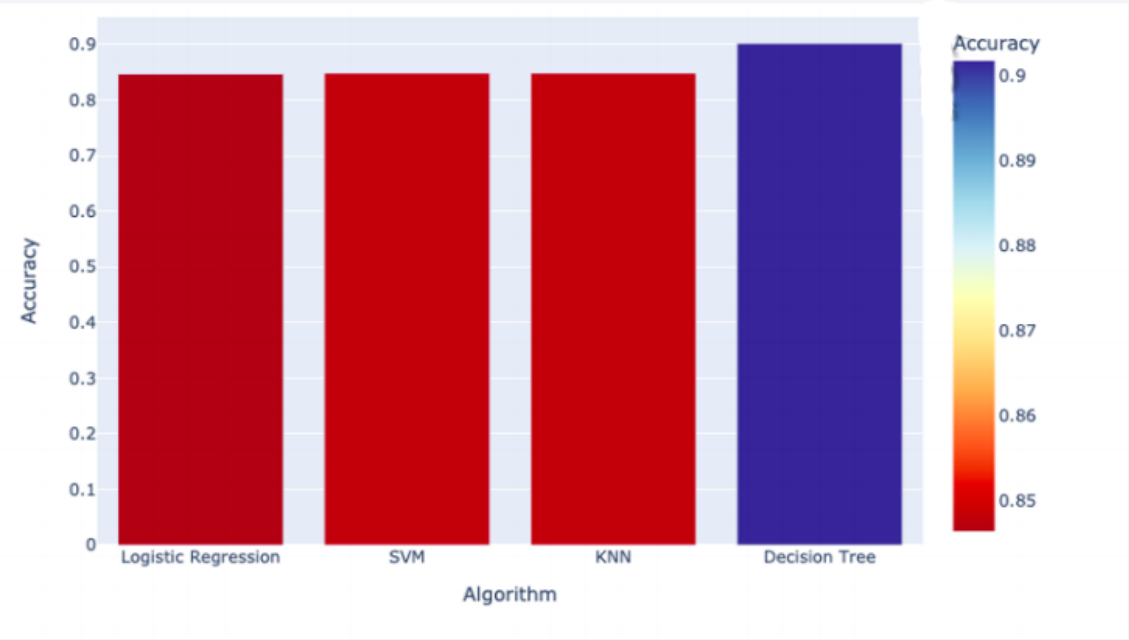
Predictive Analysis (Classification)

Classification Accuracy

- As you can see our accuracy is extremely close, but we do have a clear winner which perform best - "Decision Tree" with a score of 0.90178

We trained four different models which each had an 83%accuracy rate.

Algorithm	Accuracy	Accuracy on test data	Tuned Hyperparameter
Logistic Regression	0.846429	0.833334	{'C':0.01, 'penalty':'12','solver':'lbfgs'}
SVM	0.848214	0.833334	{'C':1.0,'gamma':0.03162277,'kernel':'sigmoid'}
KNN	0.848214	0.833334	{'algorithm': auto, 'n_neighbours': 10, 'p':1}
Decision Tree	0.901786	0.833334	{'criterion':gini, 'max_depth': 10, 'max_features':'sqrt', 'min_samples_leaf': 1, 'min_samples_split':2, 'splitter': best}

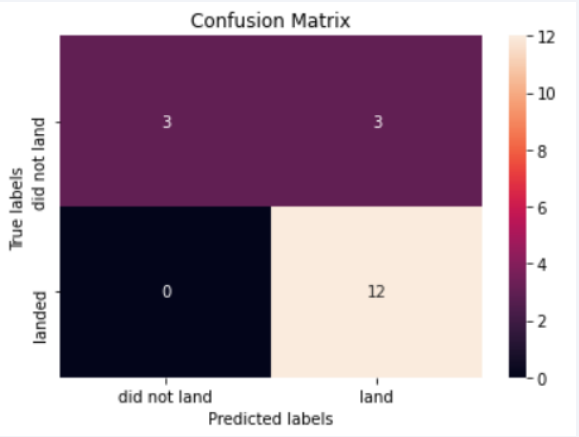


Confusion Matrix

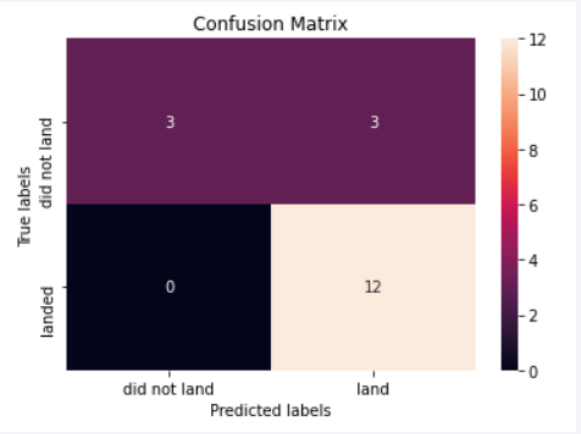
- Out here for all models unfortunately, we have same confusion matrix

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

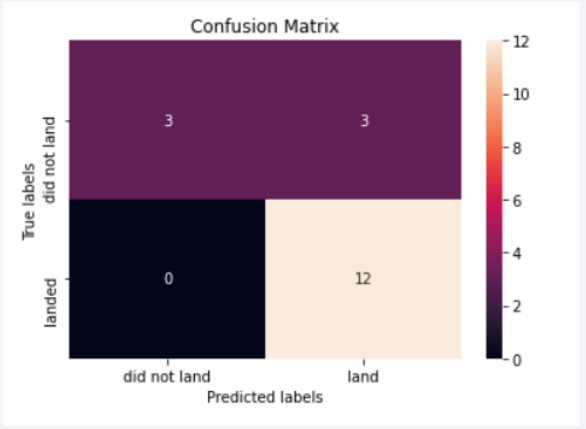


Decision Tree

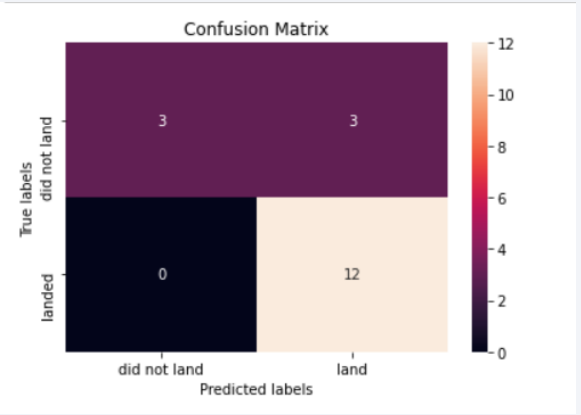


KNN

Accuracy: (TP+TN)/Total = (12+3)/18=0.8333
Misclassification Rate:(FP+FN)/Total = (3+0)/18 = 0.1667
True Positive Rate : TP/Actual Yes = 12/12 = 1
False Positive Rate : FP/Actual No = 3/6 = 0.5
True Negative Rate : TN/Actual No = 3/6 = 0.5
Precision : TP/Predicted Yes = 12/15 = 0.8
Prevalence : Actual Yes/ Total = 12/18 = 0.6667



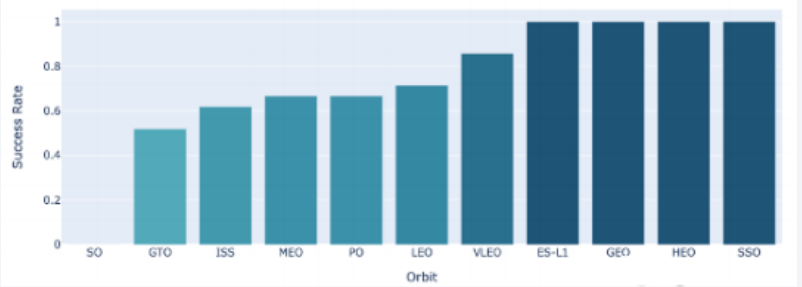
Logistic Regression

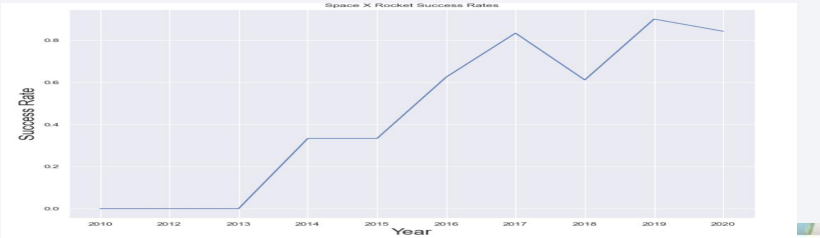


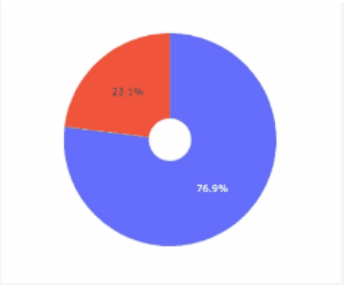
SVM

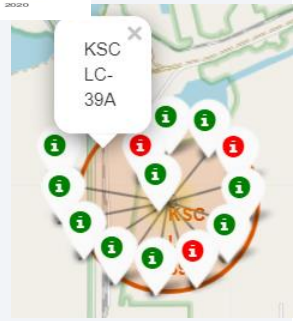
Conclusions

- Orbits ES-L1, GEO, HEO, SSO has highest Success rates
- Success rates for SpaceX launches has been increasing relatively with time and it looks like soon they will reach the required target
- KSC LC-39A had the most successful launches but increasing payload mass seems to have negative impact on success
- Decision tree classifier algorithm is the best for machine learning model for dataset

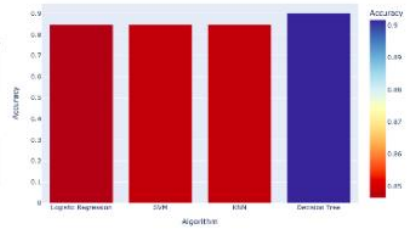








Algorithm	Accuracy
Logistic Regression	0.846429
SVM	0.848214
KNN	0.848214
Decision Tree	0.901786

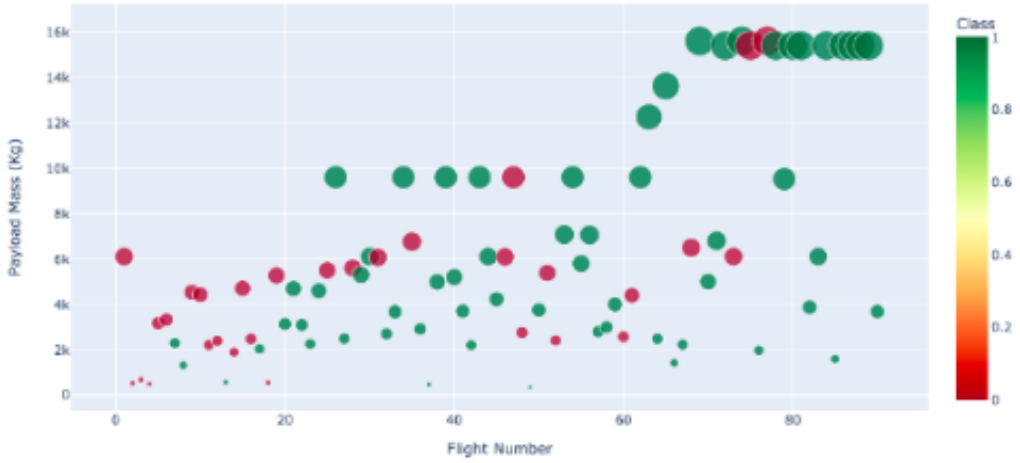


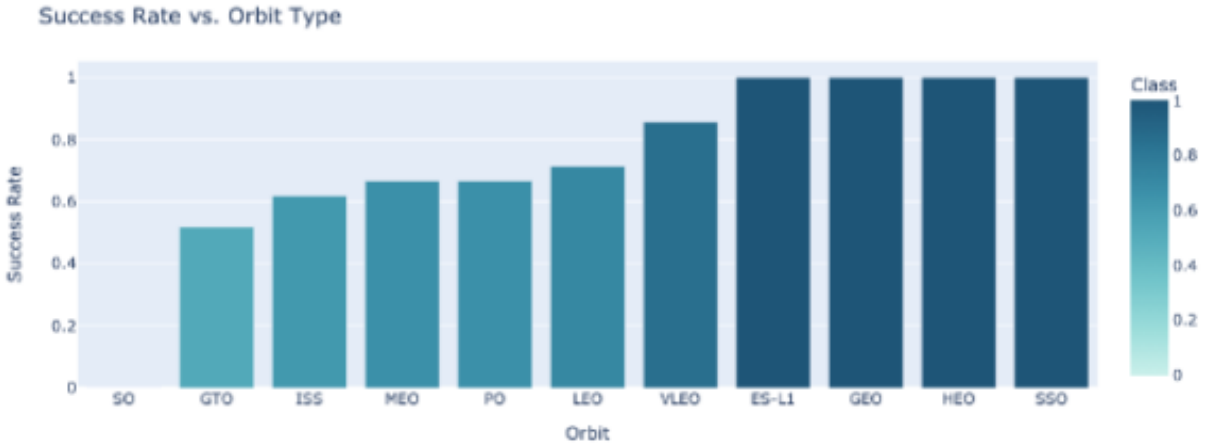
Appendix

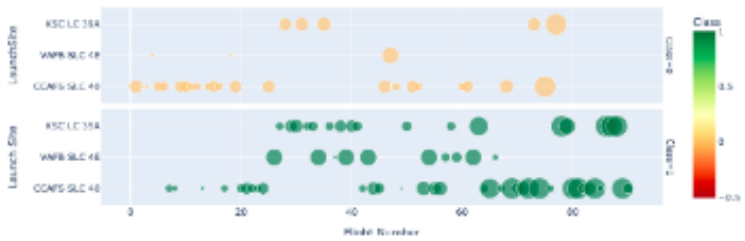
- Interactive Plotly
- Folium Measure Control plugin tool
- Folium custom title layers with labels
- Ibm Cognos Visualization tool
- Basic Decision tree construction

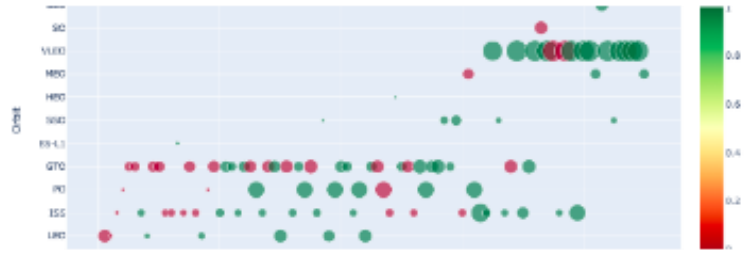
Interactive Plotly

- Used plot instead of seaborn. They are more interactive and easily customizable as well.







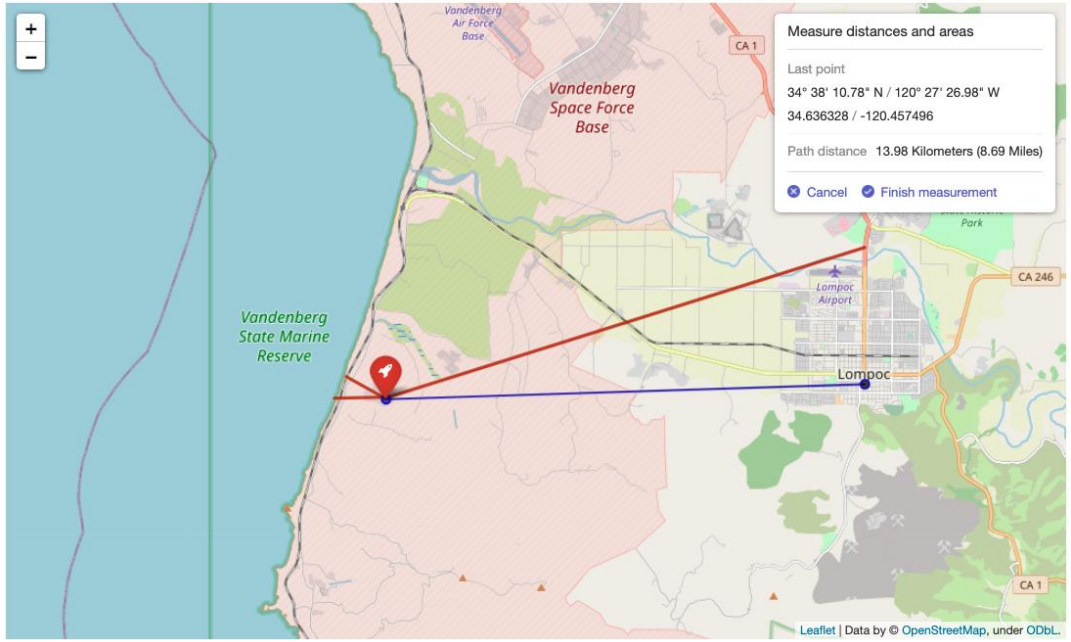


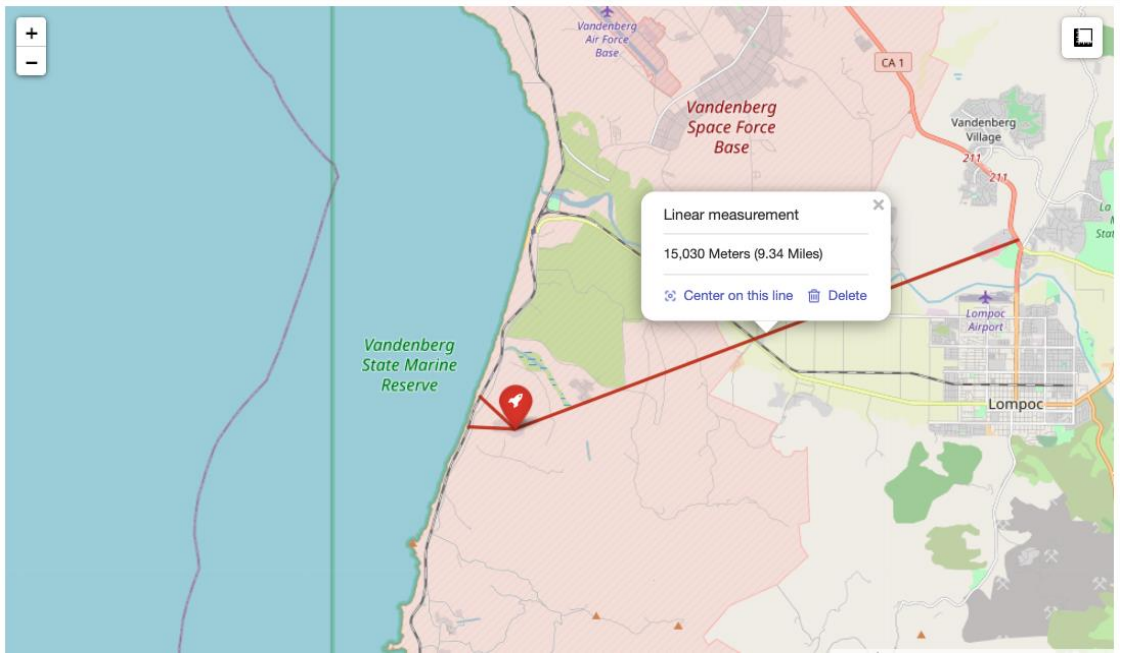
[Plotly file]

Folium Measure Control Plugin Tool

- With measure Control plugin tool, we don't need to write manual distance calculation code and it's very easy to use.

```
from folium.plugins import MeasureControl
site_map.add_child(MeasureControl(primary_length_unit='Kilometers', active_color='#0900ba', completed_color='#ba2f00'))
site_map
```





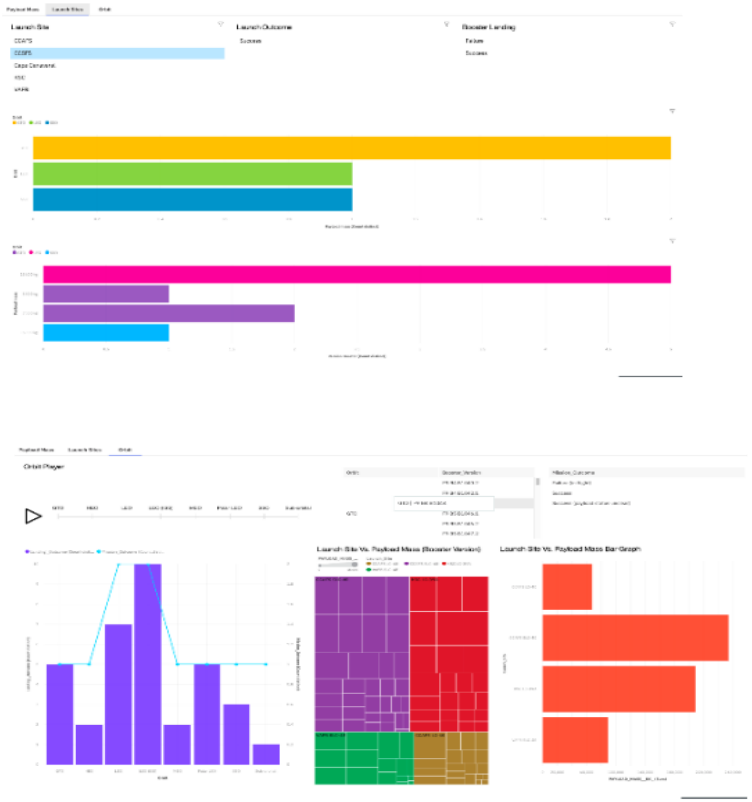
Folium Custom Title layers with labels

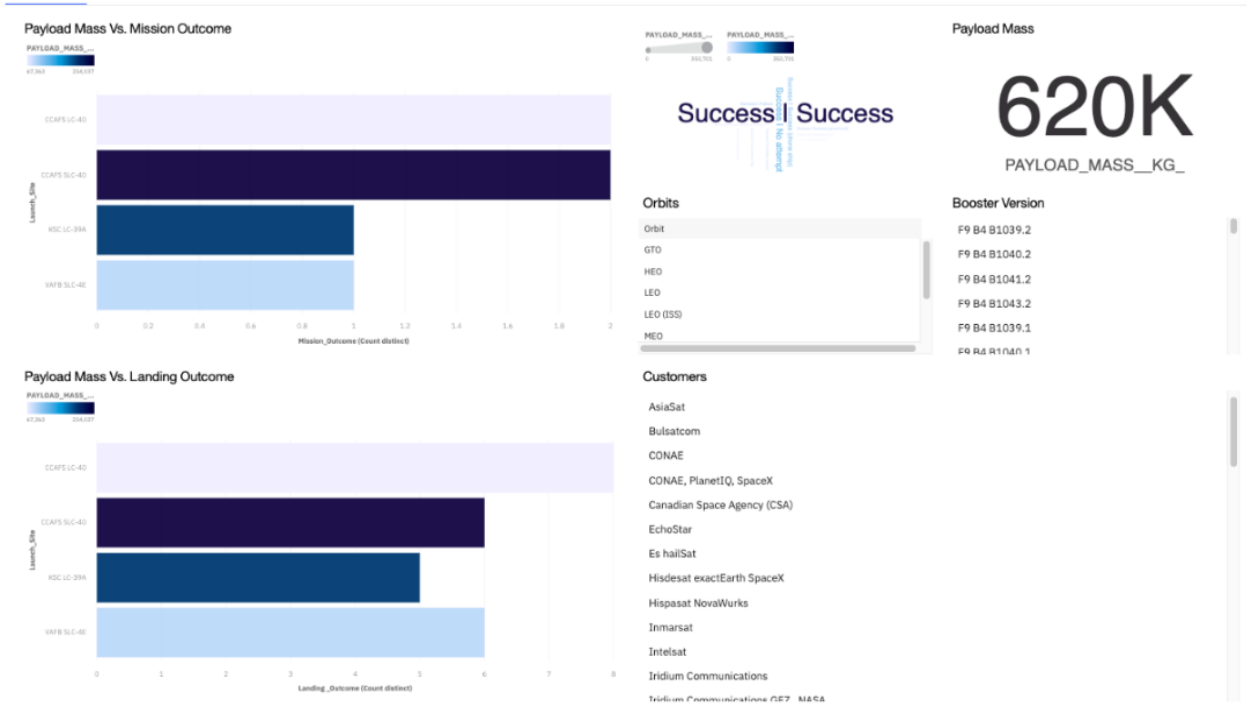
- Created custom title layer to understand the locations of launch site in a better way.

```
folium.GeoJson(geo_json_data).add_to(site_map)
folium.map.CustomPane('labels').add_to(site_map)
folium.TileLayer('stamentonerlabels' pane='labels').add_to(site_map)
site_map
```

IBM Cognos Visualization Tool

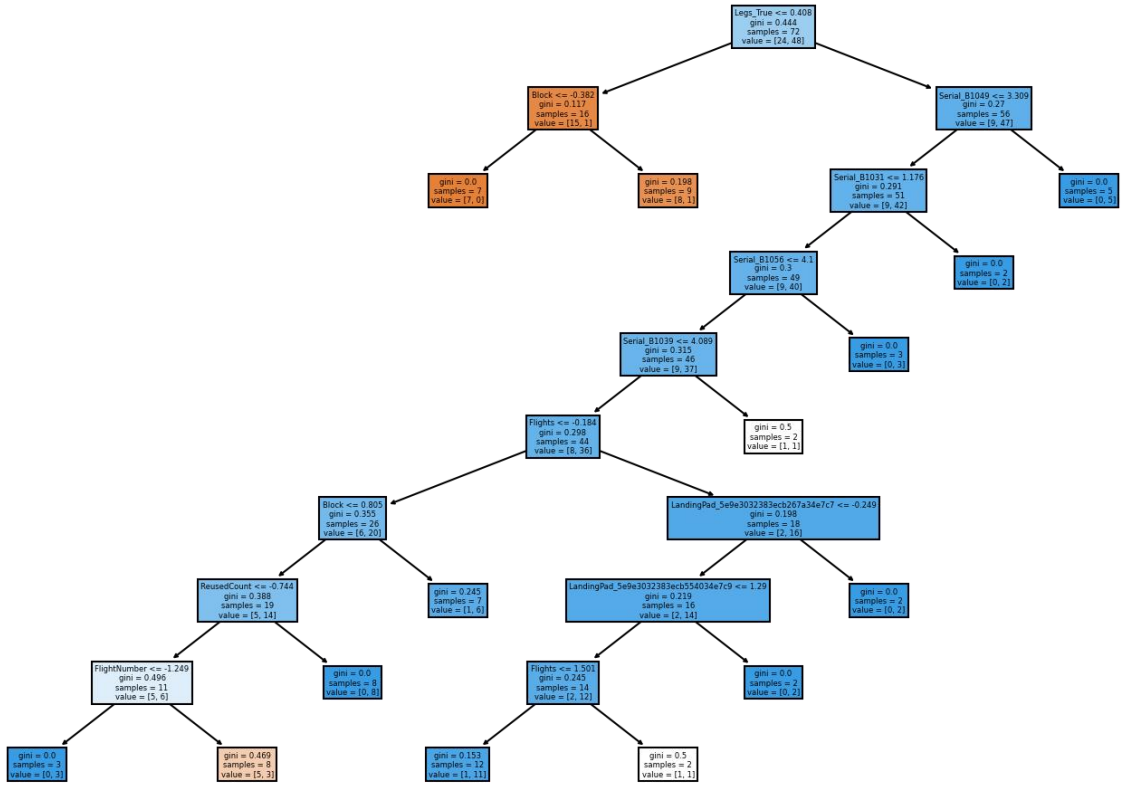
- IBM Cognos Analytics provides analytic insights that help you to detect and validate important relationships and meaningful differences based on the data that is presented by the visualization





Basic Decision Tree Construction

- Decision tree has been constructed, with decision tree model .We can see that we have reached Gini impurity almost near to 0 via the tree model . From this we can determine the correct combination of condition where the probability of the success will be highest.



Thank you!