Data Wrangling with SpaceX Data

Sunera Wanninayaka
17th March 2024

OUTLINE

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

EXECUTIVE SUMMARY

- Summary of methodologies
- \- Data Collection through API
- \- Data Collection with Web Scraping
- \- Data Wrangling
- \- Exploratory Data Analysis with SQL
- \- Exploratory Data Analysis with Data Visualization
- \- Interactive Visual Analytics with Folium
- \- Machine Learning Prediction
- • Summary of all results
- \- Exploratory Data Analysis result
- \- Interactive analytics in screenshots
- \- Predictive Analytics result from Machine Learning Lab

INTRODUCTION

- The goal of this project is to develop a machine learning pipeline to predict the results of the first stage of rocket launches, especially with a focus on a startup competing with SpaceX. SpaceX has disrupted the space industry by offering cost-effective rocket launches, primarily due to the innovative reuse of the first stage of the Falcon 9. By re-landing and reusing the rockets, SpaceX has significantly reduced launch costs, making it important for competitors to understand and predict the landing results for strategic bidding.

- The key objectives of the project include:

Identification of factors: Identify all relevant factors that affect the result of the first stage of planting. This includes a comprehensive analysis of variables that contribute to a successful or unsuccessful landing.

- Relationship analysis: Study the relationships between each variable and its impact on the landing result. Understanding these relationships is crucial for building an effective predictive model.
- Optimization conditions: Determine the optimal conditions necessary to increase the likelihood of a successful landing. This includes using the information obtained from the analysis to identify factors that positively affect the landing results.

METHODOLOGY

- Executive Summary
- • Data collection methodology:
  • Data was collected using SpaceX REST API and web scrapping from Wikipedia
- • Perform data wrangling
  • Data was processed using one-hot encoding for categorical features
- • Perform exploratory data analysis (EDA) using visualization and SQL
- • Perform interactive visual analytics using Folium and Plotly Dash
- • Perform predictive analysis using classification models • How to build, tune, evaluate classification models

Data Collection

- The data collection in this project included a dual approach: the use of REST API requests and the use of web scraper methods on Wikipedia. For the REST API, the process started with a GET request, decoded the JSON response and converted it into a panda data frame using json_normalize(). The cleaning procedure was then applied to remove the missing values.
- Web scraping included using BeautifulSoup to extract executable entries from Wikipedia's HTML tables, parse tables, and convert data into panda data frames. This combined methodology provides a comprehensive data set for careful analysis of missile launch results.

Data Collection – SpaceX API

Get request for rocket launch data using API

Use json_normalize method to convert json result to dataframe

Performed data cleaning and filling the missing value

```python
spacex_url="https://api.spacexdata.com/v4/launches/past"

response = requests.get(spacex_url)

# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())

# Lets take a subset of our dataframe keeping only the features we want a
nd the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number',
'date_utc']]

# We will remove rows with multiple cores because those are falcon rocket
s with 2 extra rocket boosters and rows that have multiple payloads in a
single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the s
ingle value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then ex
tracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Data Collection - Scraping

Request the Falcon9 Launch Wiki page from url

```python
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url).text
```

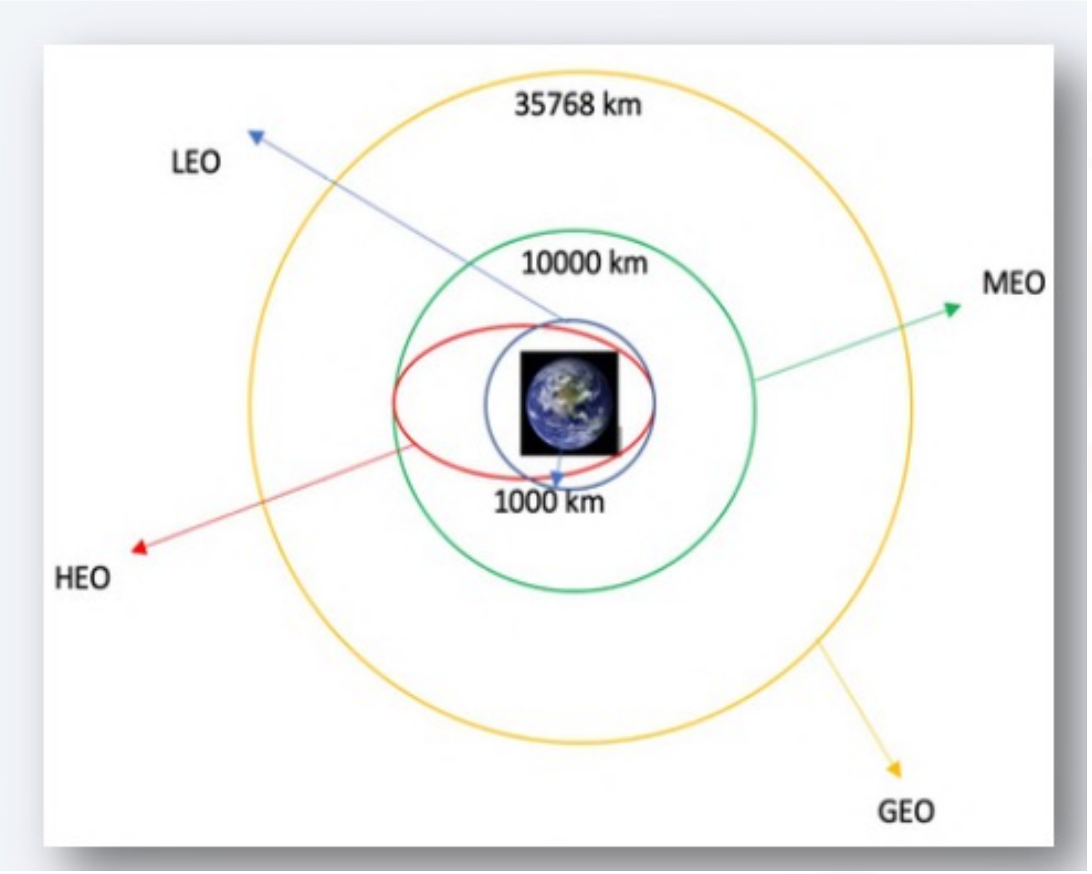
Create a BeautifulSoup from the HTML response

```python
# Use BeautifulSoup() to create a BeautifulSoup object from a response te
xt content
soup = BeautifulSoup(data,'html.parser')
```

Extract all column/variable names from the HTML header

```python
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plai
nrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding t
o launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
```

Data Wrangling

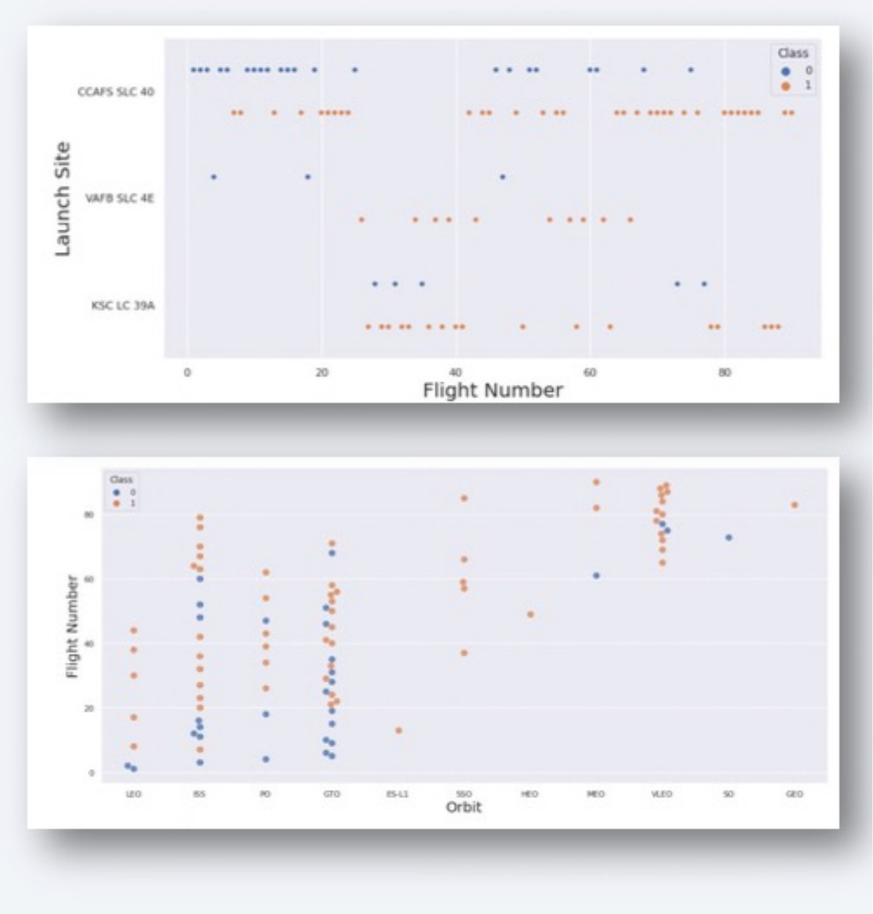


- Data Wrangling is the process of cleaning and unifying messy and complex data sets for easy access and Exploratory Data Analysis (EDA).

- We will first calculate the number of launches on each site, then calculate the number and occurrence of mission outcome per orbit type.

- We then create a landing outcome label from the outcome column. This will make it easier for further analysis, visualization, and ML. Lastly, we will export the result to a CSV.

EDA with Data Visualization



- We first started by using scatter graph to find the relationship between the attributes such as between:

- PayloadandFlightNumber.
- FlightNumberandLaunchSite.

PayloadandLaunchSite.

- FlightNumberandOrbitType.

PayloadandOrbitType.

- Scatter plots show dependency of attributes on each other. Once a pattern is determined from the graphs. It's very easy to see which factors affecting the most to the success of the landing outcomes.

EDA With Data Visualization

After the initial study of the relationships using diffusion graphs, a more in-depth analysis is carried out using histograms and linear graphs. Bar graphs provide a simple interpretation of attribute relationships, in particular, to identify the orbits with the highest probability of success.

This visual representation helps to identify influential factors. In addition, linear graphs are used to illustrate trends and attribute patterns over time, in particular, by focusing on the annual trend of launch success.

This temporary visualization facilitates the identification of models and differences in success rates over the years. Functionality engineering is introduced to improve predictive modeling for future results. This includes the creation of dummy variables for categorical columns, the refinement of the data set for optimal use in machine learning models. This strategic training of functions lays the foundations for precise predictions of success in future modules.

EDA with SQL

- Using SQL, we had performed many queries to get better understanding of the dataset, Ex:
- \- Displaying the names of the launch sites.
- \- Displaying 5 records where launch sites begin with the string 'CCA'.
- \- Displaying the total payload mass carried by booster launched by NASA (CRS).
- \- Displaying the average payload mass carried by booster version F9 v1.1.
- \- Listing the date when the first successful landing outcome in ground pad was achieved.
- \- Listing the names of the boosters which have success in drone ship and have payload mass
- greater than 4000 but less than 6000.
- \- Listing the total number of successful and failure mission outcomes.
- \- Listing the names of the booster_versions which have carried the maximum payload mass.
- \- Listing the failed landing_outcomes in drone ship, their booster versions, and launch sites
- names for in year 2015.
- \- Rank the count of landing outcomes or success between the date 2010-06-04 and
- 2017-03-20, in descending order.

Build an Interactive Map with Folium

- To visualize the launch data into an interactive map. We took the latitude and longitude coordinates at each launch site and added a circle marker around each launch site with a label of the name of the launch site.
- We then assigned the dataframe launch_outcomes(failure,success) to classes 0 and 1 with Red and Green markers on the map in MarkerCluster().
- We then used the Haversine's formula to calculated the distance of the launch sites to various landmark to find answer to the questions of:
- • How close the launch sites with railways, highways and coastlines?
- • How close the launch sites with nearby cities?

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash which allowing the user to play around with the data as they need.
- We plotted pie charts showing the total launches by a certain sites.
- We then plotted scatter graph showing the relationship with Outcome and Payload
- Mass (Kg) for the different booster version.

Predictive Analysis (Classification)

Building the Model

- Load the dataset into NumPy and Pandas
- Transform the data and then split into training and test datasets
- Decide which type of ML to use
- set the parameters and algorithms to GridSearchCV and fit it to dataset.

Evaluating the Model

- Check the accuracy for each model
- Get tuned hyperparameters for each type of algorithms.
- plot the confusion matrix

Improving the Model

- Use Feature Engineering and Algorithm Tuning

Find the Best Model

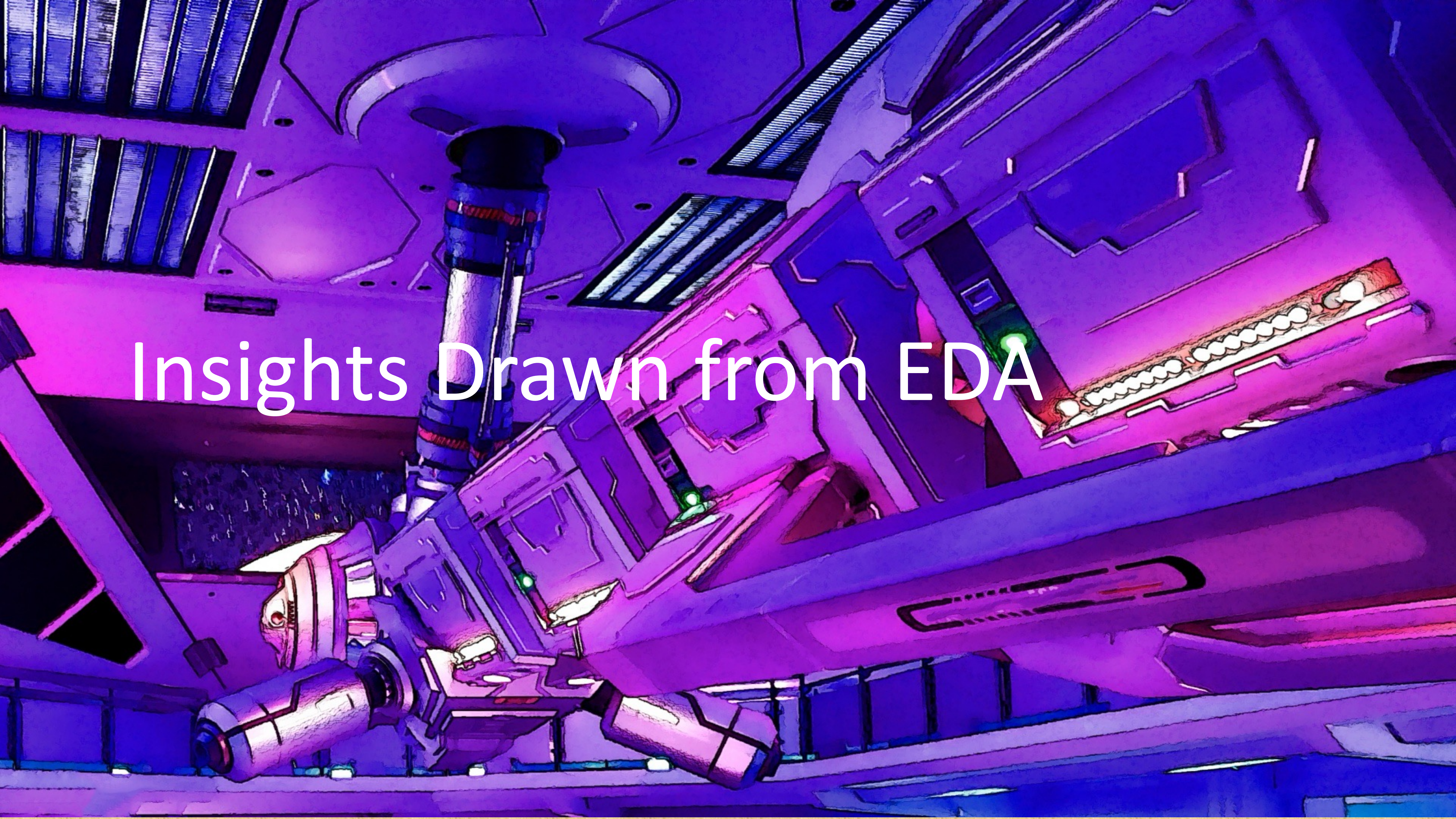
- The model with the best accuracy score will be the best performing model.

Results

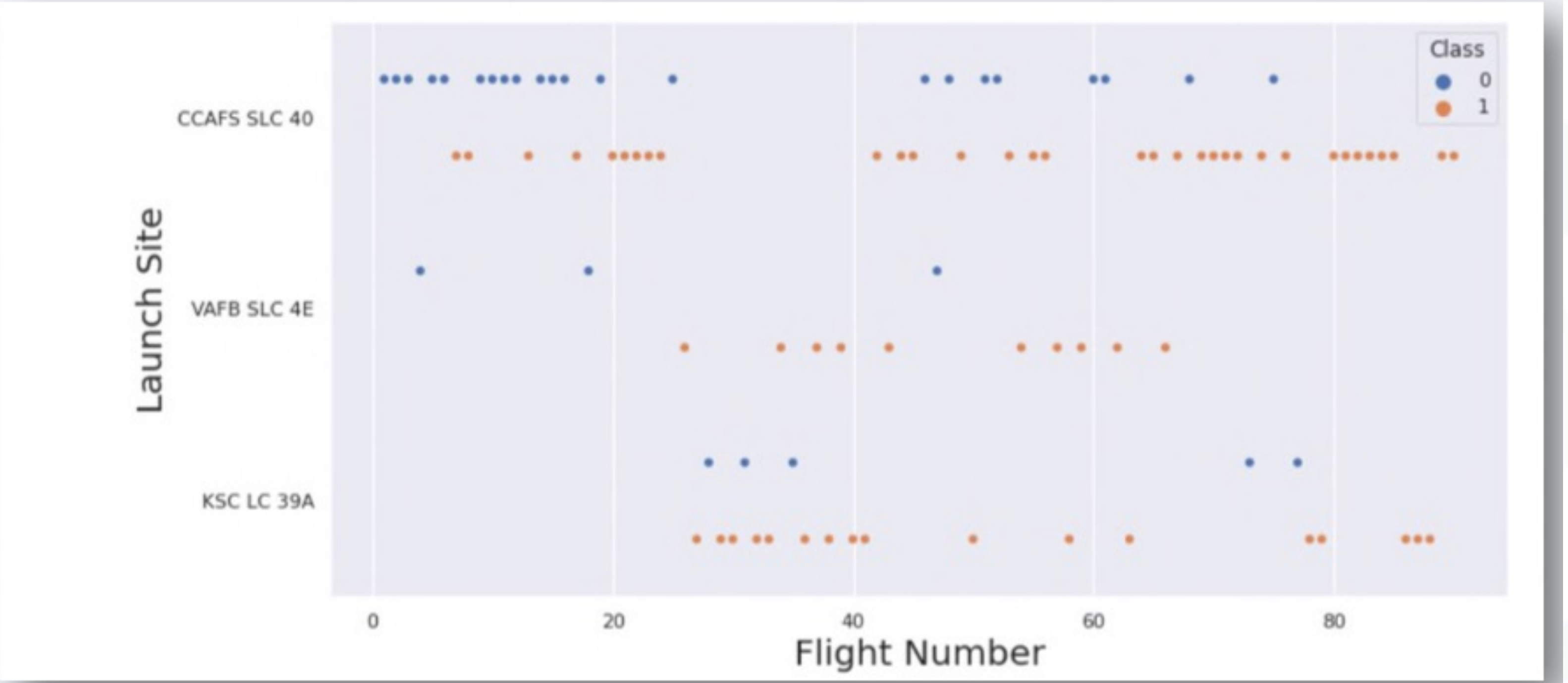
The results will be categorized to 3 main results which is:

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

Insights Drawn from EDA



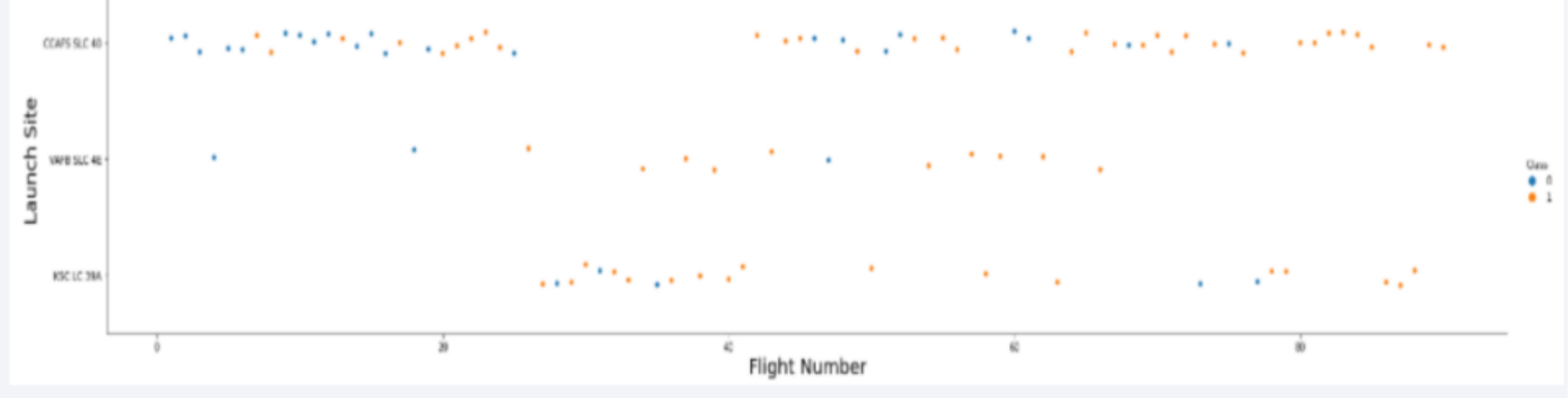
Flight Number vs Launch Site



- This scatter plot shows that the larger the flights amount of the launch site, the greater the the success rate will be.
- However, site CCAFS SLC40 shows the least pattern of this.

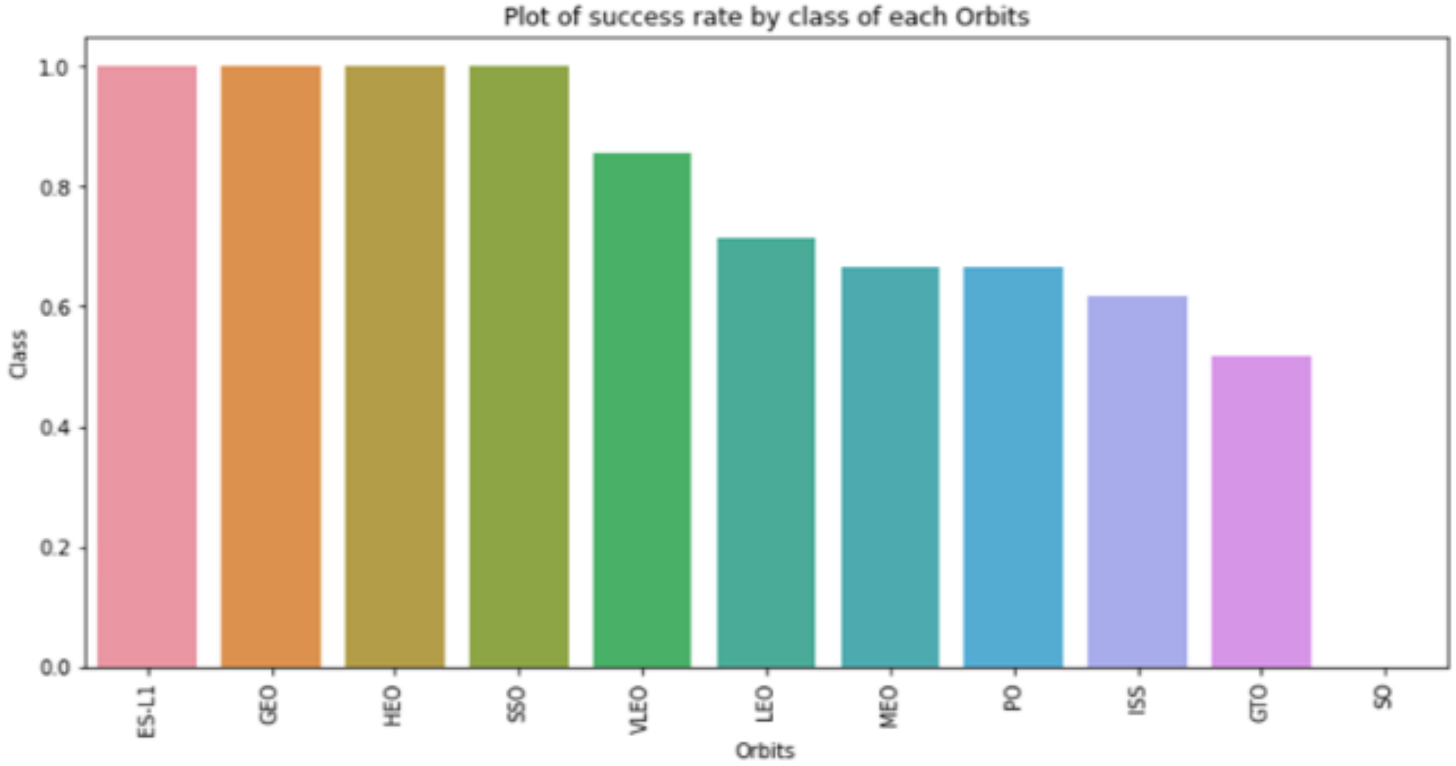
Payload Vs Launch Site

- From the plot we found that larger the flight anount at a launch site, the greater the success it has at a launch site



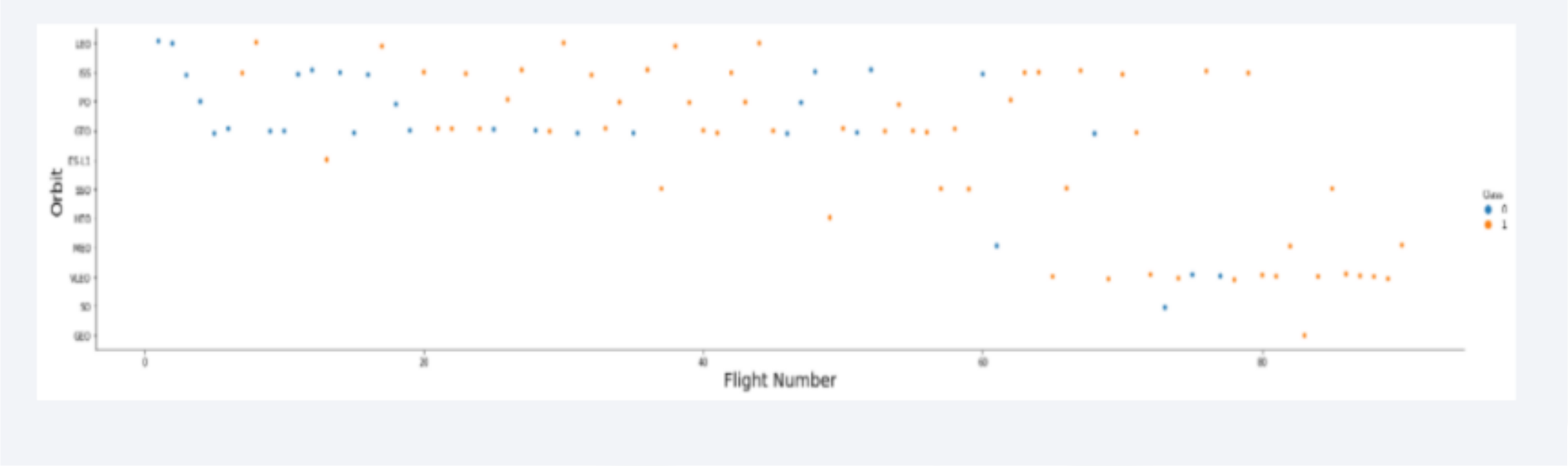
Success Rate vs. Orbit Type

- From the plot,we can see that ES-L1,GEO,HEO,SSO,VLEO had the most success rate



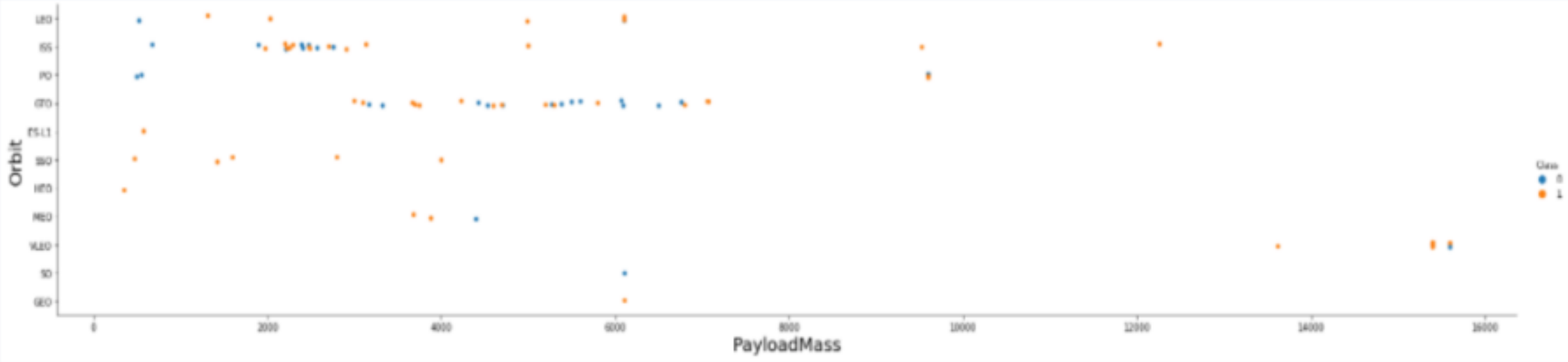
Flight Number vs.Orbit Type

- The plot below shows the flight Number vs Orbit type, we can observe that in the LEO orbit,success is related to the number of flights whereas in the GTO orbit,there is no relationship between flight number and the orbit



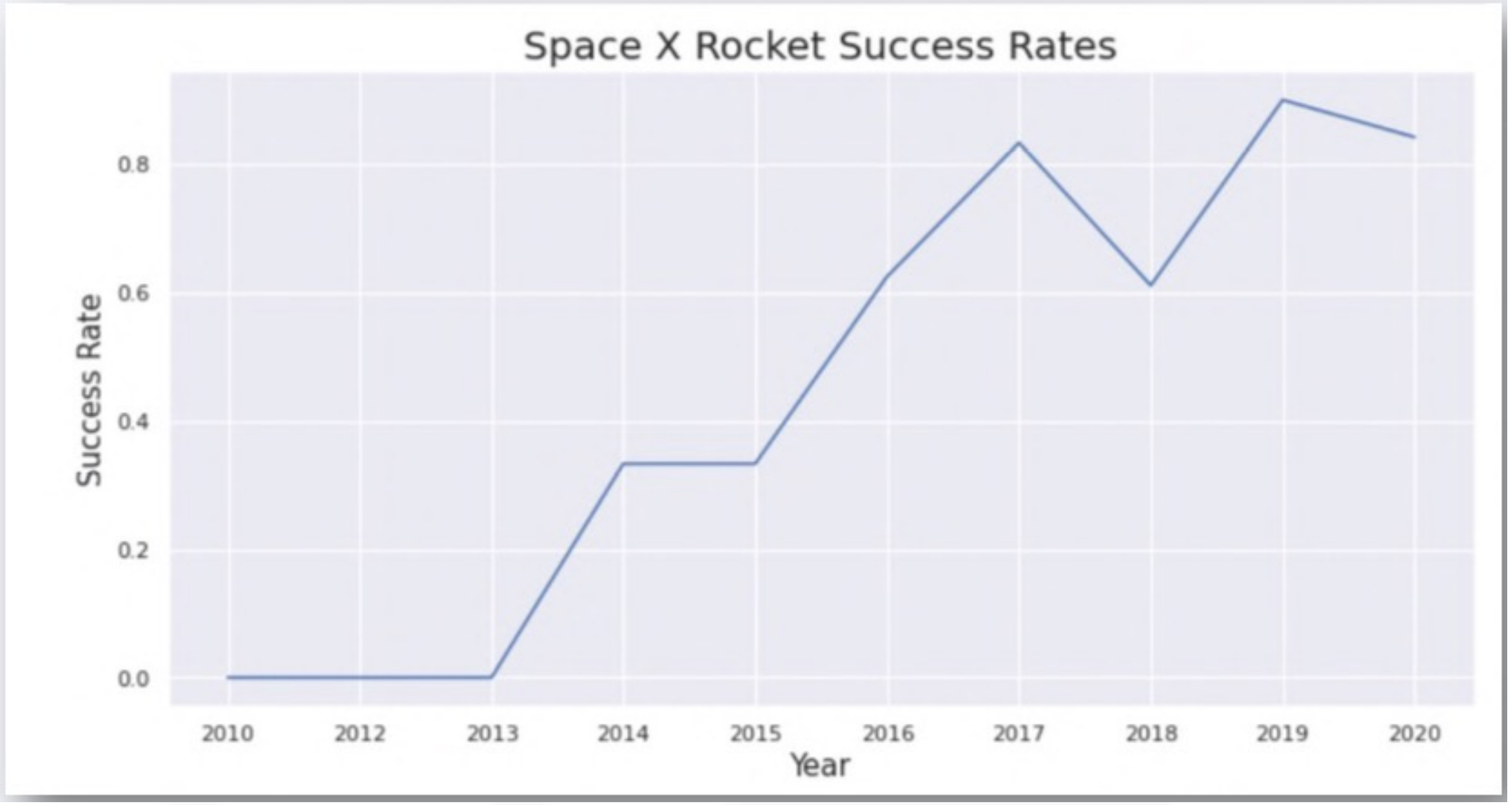
Payload vs Orbit Type

- We can observe that with heavy paloads, the sucessful landing are more for PO,LEO and ISS orbits



Launch Success Yearly Trend

- This figures clearly depicted and increasing trend from the year 2013 until 2020.
- If this trend continue for the next year onward. The success rate will steadily increase until reaching 1/100% success rate.



All Launch Site Names

- We used the keyword **Distinct** to show only unique launch sites from the SPACEX data

Display the names of the unique launch sites in the space mission

In [10]:
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Launch Site Names with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

In [11]:
```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

We used the query above to display 5 records where launch sites begin with 'CCA'

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 using the below query

Display the total payload mass carried by boosters launched by NASA (CRS)

In [12]:
```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

Average Payload Mass by F9 v1.1

- We calculated the average paload mass carried by booster version f9 v1.1 as 2928.4

Display average payload mass carried by booster version F9 v1.1

In [13]:
```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

- We Observed that the dates of the first Successful landing outcome on ground pas was 22nd December 2015

In [14]:

```
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

Out[14]:

	firstsuccessfull_landing_date
0	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000 $

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

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
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

- We used wildcard like '%' to filter for WHERE MissionOutcome was a success or a failure.

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Successful Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Success%';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Successful Mission
100

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Failure Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Failure%';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.databases.appdomain.clou
d:32731/bludb
Done.
```

Failure Mission
1

Boosters Carried 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);
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.databases.appdomain.clou
d:32731/bludb
Done.
```

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

- We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.

2015 Launch Records

- We Used a combinations of the WHERE clause,LIKE, AND and BETWEEN conditions to filter for faild landing outcomes in drone ship for the 2015

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

In [18]:

```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

In [19]:
```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

Out[19]:

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

- We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20

- We applied the groupBY clause to group the landing outcomes and the ORDER by clause to order the grouped landing outcome on descending order

Launch Sites Proximities Analysis

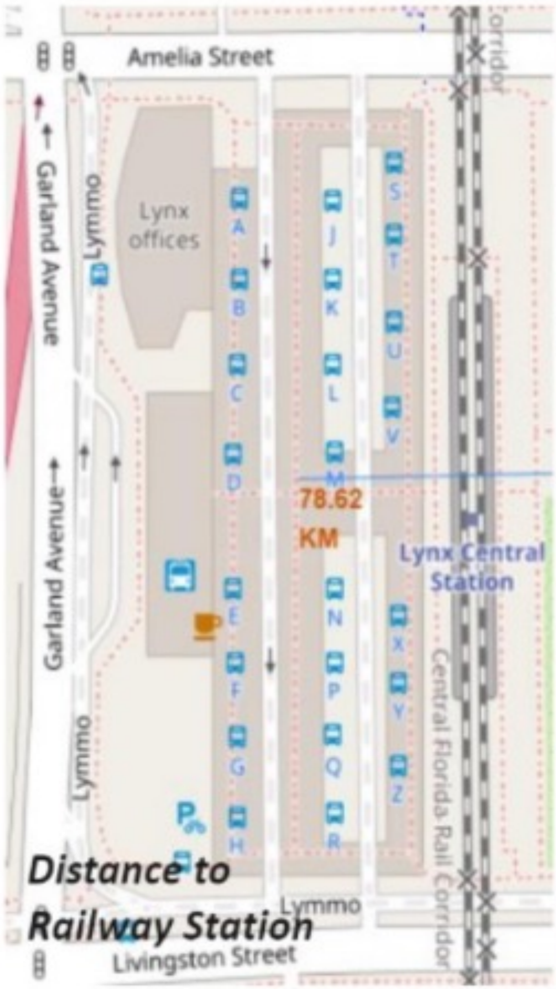
Markers showing launch sites with color labels

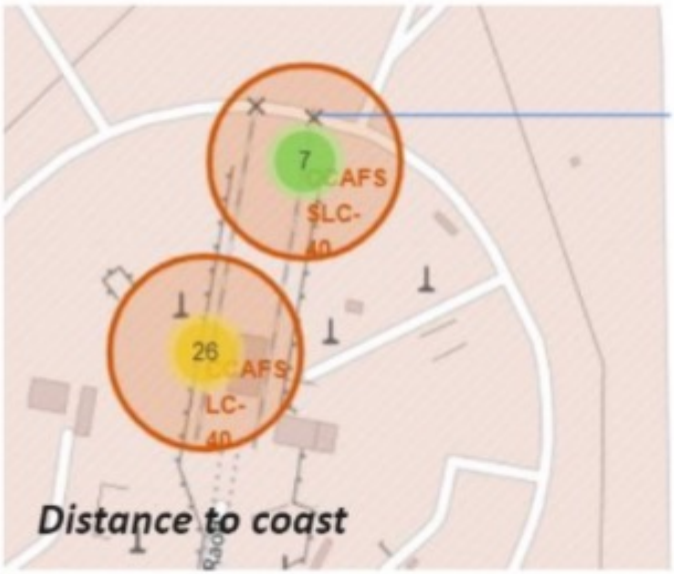
All Launch Sites global map markers

IBM Developer

SKILLS NETWORK

Launch Site distance to Landmarks

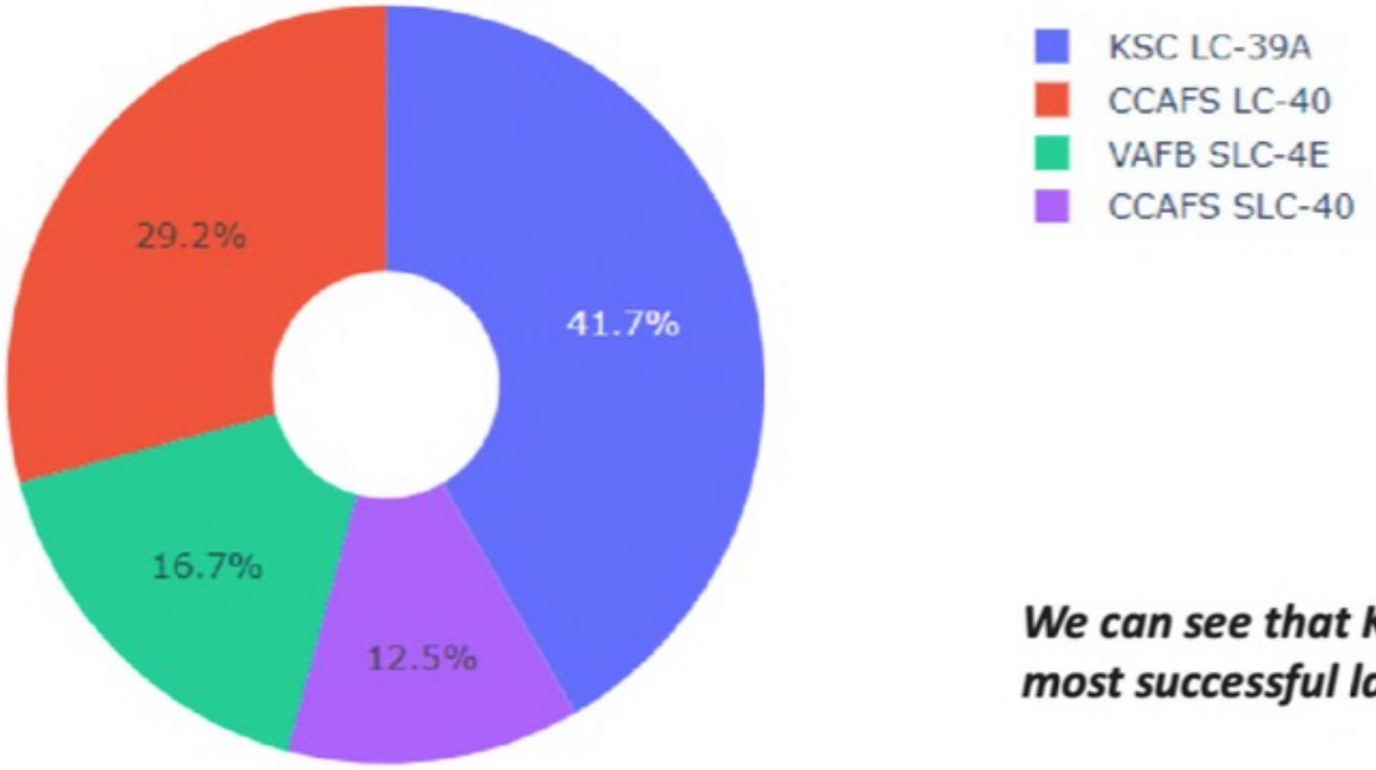




- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

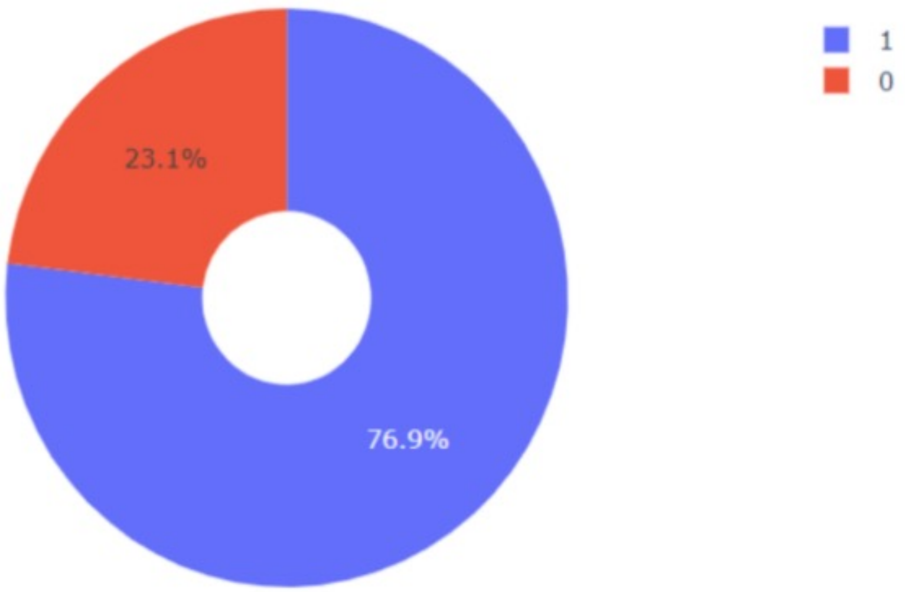
Build a Dashboard with plotly Dash

The success percentage by each sites.



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

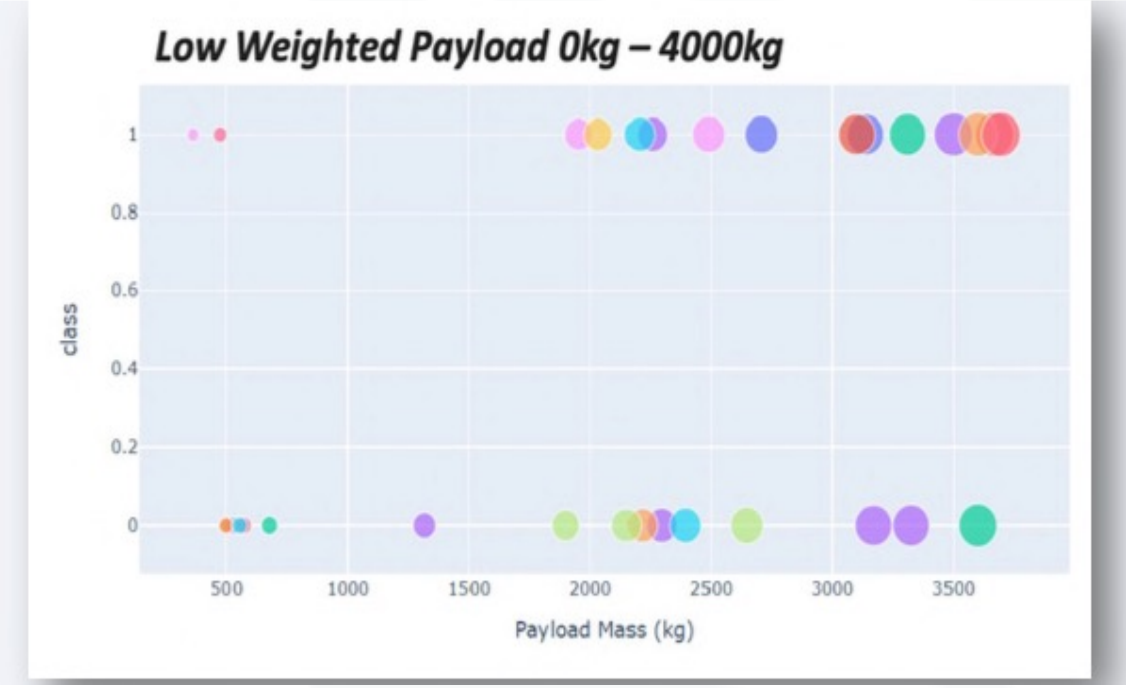
Pir chart to show that launch site with highets launch success ration

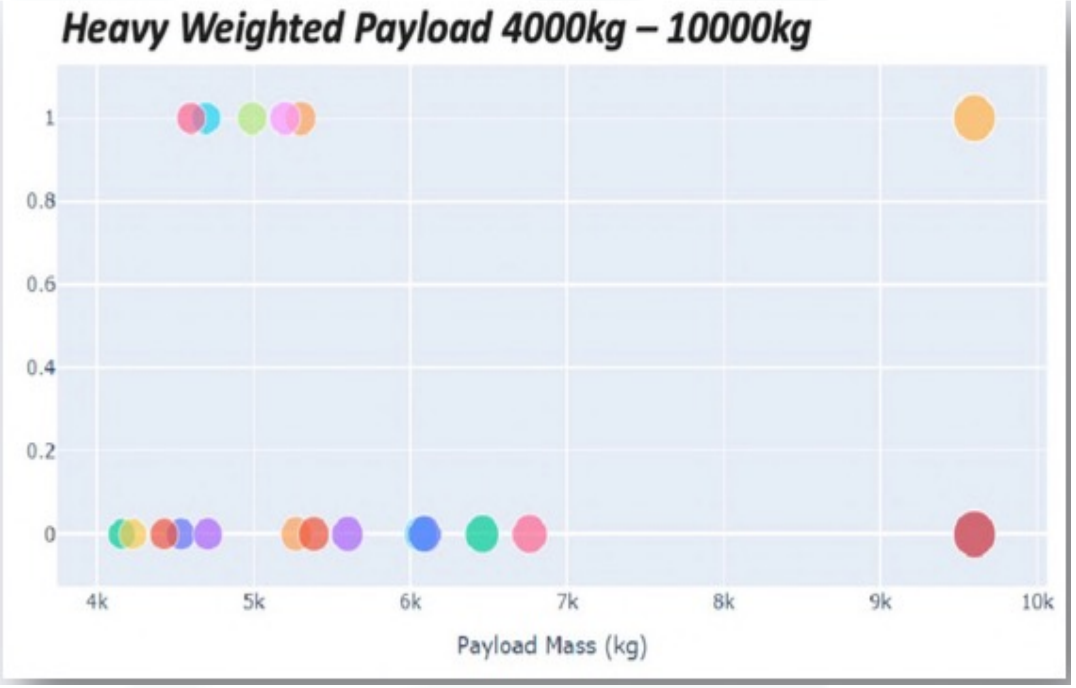


KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload vs Launch Outcome Scatter Plot

- We can see that all the success rate for low weighted payload is higher than heavy weighted payload





IBM Developer

SKILLS NETWORK

Predictive Analysis Classification

Classify Accuracy| the decision tree classifier is the model with the highest classification accuracy

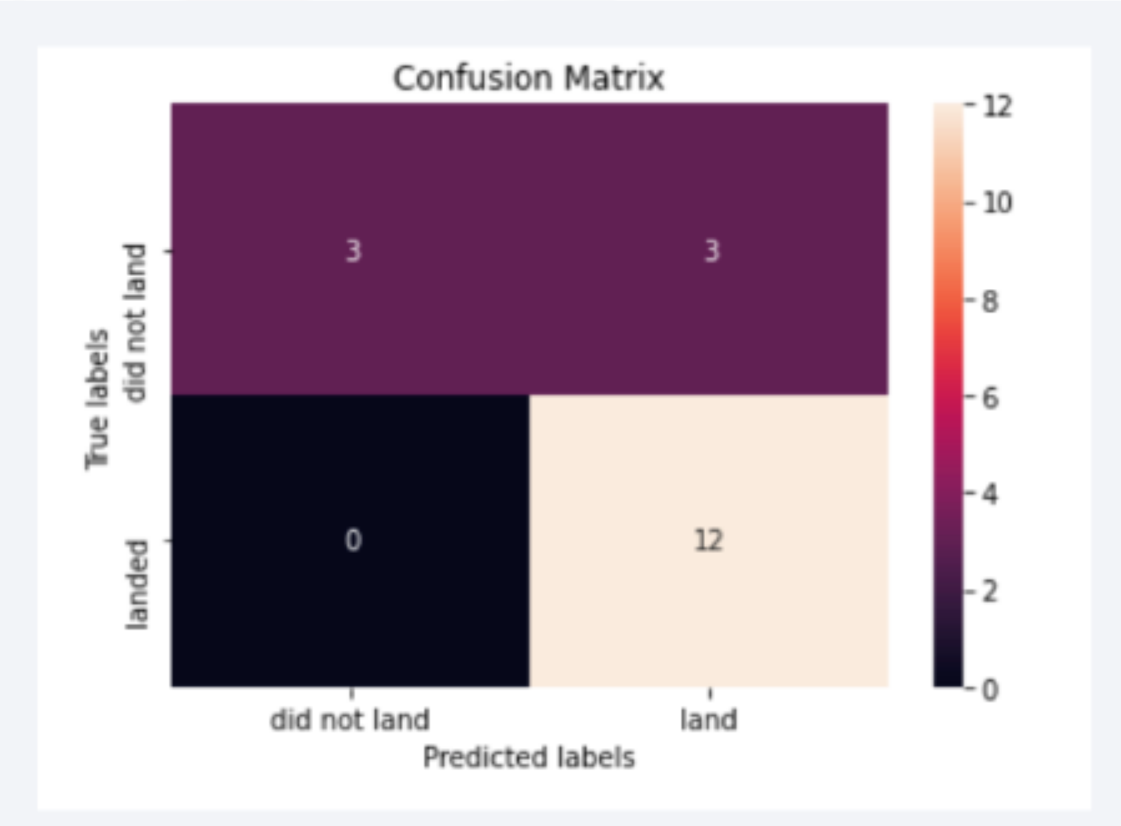
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes

- Major one is flase positives,unsuccessful landing marked as successful landing by the classifier



Conclusions

We can conclude that:

- The Tree Classifier Algorithm is the best Machine Learning approach for this dataset.
- The low weighted payloads (which define as 4000kg and below) performed better than the heavy weighted payloads.
- Starting from the year 2013, the success rate for SpaceX launches is increased, directly proportional time in years to 2020, which it will eventually perfect the launches in the future.
- KSC LC-39A have the most successful launches of any sites; 76.9%
- SSO orbit have the most success rate; 100% and more than 1 occurrence.