IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

RITVIK KUMAR

24th MARCH 2023

Outline

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 using different classification models
- Summary of all results
    - Exploratory Data Analysis result
    - Interactive analytics in screenshots
    - Predictive Analytics result

Introduction

- Project background and context

  Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

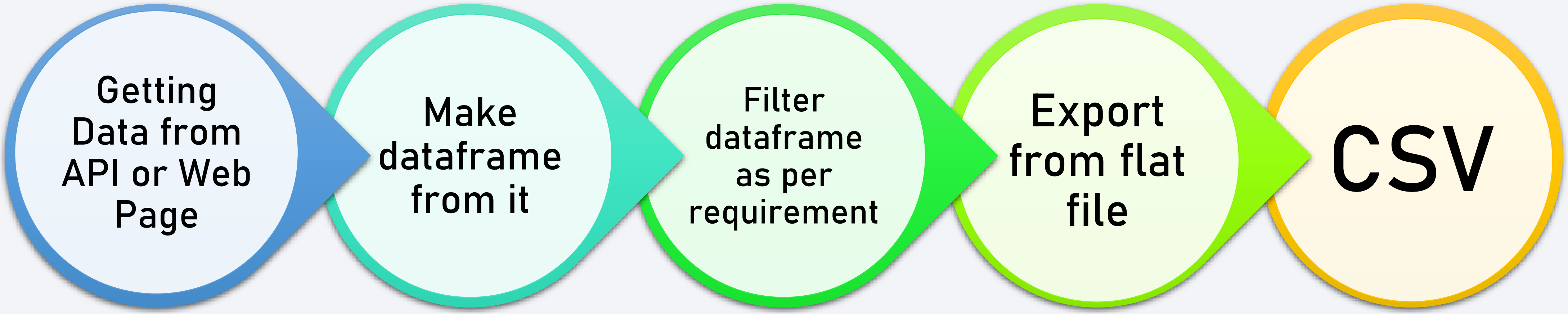
- Data collection methodology
  - ❑ Data was collected using SpaceX Rest API and Web scraping from Wikipedia.
- Perform data wrangling
  - ❑ One-hot encoding was applied to categorical features and dropping irrelevant columns for machine learning.
- Perform exploratory data analysis (EDA) using visualization and SQL
  - ❑ Scatter, bar and line charts to show patterns between data.
- Perform interactive visual analytics using Folium and Plotly Dash
  - ❑ Build a dashboard to analyze launch records interactively with Plotly Dash.
  - ❑ Build an interactive map to analyze the launch site proximity with Folium.
- Perform predictive analysis using classification models
  - ❑ Train different classification models.

Data Collection

Data collection is the process of gathering, measuring, and analyzing accurate data from a variety of relevant sources to find answers to research problems, answer questions, evaluate outcomes, and forecast trends and probabilities.

For our project , the data was collected through these two URL's:

- https://api.spacexdata.com/v4/launches/past
- https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches



Data Collection – SpaceX API

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

```python
response = requests.get(spacex_url)
```

```python
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

```python
data_falcon9 =launch[launch['BoosterVersion']!='Falcon 1']
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```


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

```python
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857

GITHUB Link

Data Collection - Scraping

- Getting Response from HTML
- Creating BeautifulSoup Object
- Finding Tables
- Getting Column names
- Creation of dictionary and appending data to keys
- Converting dictionary to dataframe
- Dataframe to .CSV

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
# assign the response to a object
response = requests.get(static_url).text
```

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, "html.parser")
```

```
# Let's print the third table and check its content
html_tables = soup.find_all('table')
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
column_names = []
table_rows=first_launch_table.find_all('th')
for i,row in enumerate(table_rows):
    try:
     name = extract_column_from_header(table_rows[i])
     if (name is not None and len(name) > 0):
        column_names.append(name)
```

```
launch_dict= dict.fromkeys(column_names)
```

```
df=pd.DataFrame(launch_dict)
```

Flight No.	Launch site	Payload	Payload mass	Orbit
1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO
2	CCAFS	Dragon	0	LEO
3	CCAFS	Dragon	525 kg	LEO
4	CCAFS	SpaceX CRS-1	4,700 kg	LEO
5	CCAFS	SpaceX CRS-2	4,877 kg	LEO

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

GITHUB Link

Data Wrangling

Data wrangling is the process of removing errors and combining complex data sets to make them more accessible and easier to analyze.

Load data and calculate number of launches at each site → Calculate number and occurrence of each orbit → Calculate number and occurrence of mission outcome per orbit type → Create landing outcome label from Outcome column → Export dataset as CSV

```
df=pd.read_csv(dataset_part_1_csv)
```

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
```

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class=[]
for i in df['Outcome']:
    if i in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

```
df.to_csv("dataset_part_2.csv", index=False)
```

[GITHUB Link]()

Data Wrangling

Calculate number and occurrence of mission outcome per orbit type

Create landing outcome label from Outcome column

Class column will represent the classification variable that represents the outcome of each launch

Export dataset as CSV

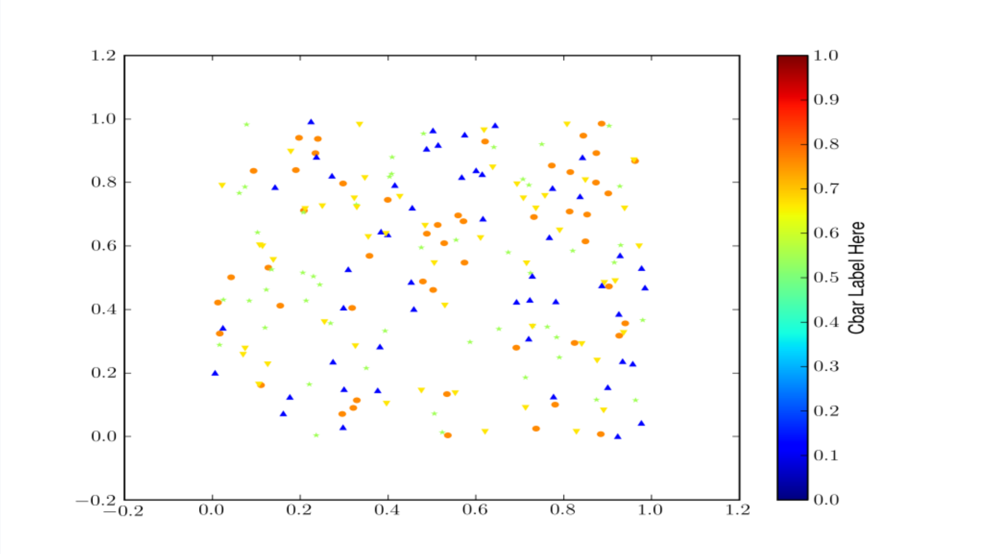
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

EDA with Data Visualization

[GITHUB Link]

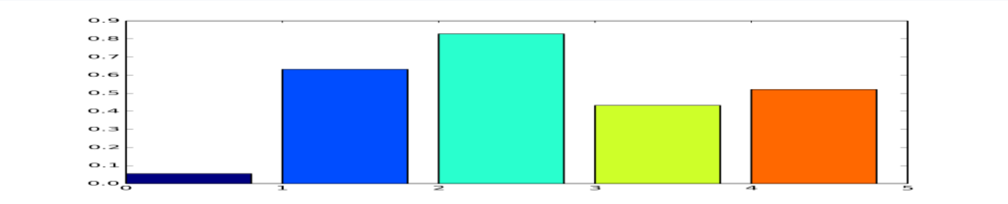
Scatter Graph/Plot :

A scatter plot (2D) is a useful method of comparing variables against each other. The data in a scatter plot is considered to express a trend. Once a pattern is determined from the graph it's very easy to predict which factors will lead to maximum probability of success in both outcome and landing.



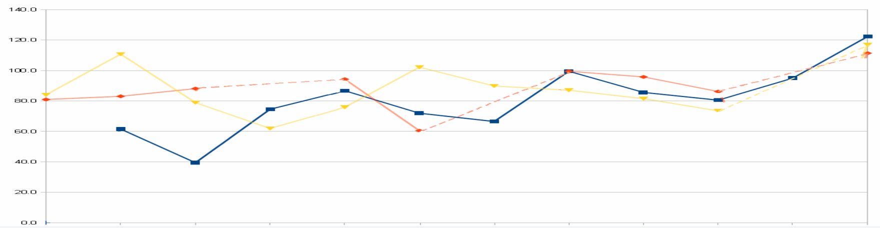
Bar Graph :

Bar graphs are easiest to interpret a relationship between attributes. Via this bar graph we can easily determine which orbits have the highest probability of success.



Line Chart/Plot :

A line plot is a type of plot which displays information as a series of data points called 'markers' connected by straight line segments. Use line plot when you have a continuous data set. Line graphs are useful in that they show trends clearly and can aid in predictions for the future.



EDA with SQL

SQL is an essential tool for data science. It's not just crucial for writing queries and manipulating data, it's also helpful in communicating with other people, building models, and visualizing results.
SQL is a standard language for storing, manipulating and retrieving data in databases.

We performed SQL queries to gather information from given dataset :

1. Display the names of the unique launch sites in the space mission
2. Display 5 records where launch sites begin with the string CCA
3. Display the total payload mass carried by boosters launched by NASA (CRS)
4. Display average payload mass carried by booster version F9 v1.1
5. List the date when the first successful landing outcome in ground pad was achieved
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. List the total number of successful and failure mission outcomes
8. List the names of the booster versions which have carried the maximum payload mass. Use a subquery
9. List the records which will display the month names, failure landing outcomes in drone ship, booster versions, launch site for the months in year 2015
10. Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order

```
%load_ext sql
```

```
import csv, sqlite3

con = sqlite3.connect("my_data1.db")
cur = con.cursor()
```

```
!pip install -q pandas==1.1.5
```

```
%sql sqlite:///my_data1.db
```

```
'Connected: @my_data1.db'
```

```
import pandas as pd
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv")
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False,method="multi")
```

GITHUB Link

Build an Interactive Map with Folium

Folium makes it easy to visualize data that's been manipulated in python on an interactive leaflet map.

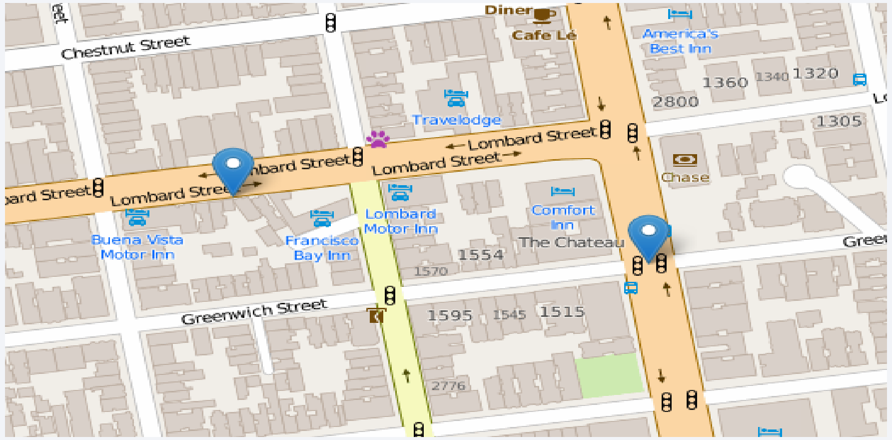
We use the latitude and longitude coordinates for each launch site and added a circle marker around each launch site with a label of the name of the launch site.

It is also easy to visualize the number of success and failure for each launch site by assigning the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success, with Green and Red markers on the map.

Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.

We calculated the distances between a launch site to its proximities.
We answered some question for instance:

- Are launch sites near railways, highways and coastlines.
- Do launch sites keep certain distance away from cities.



[GITHUB Link]

Build a Dashboard with Plotly Dash

We built an Interactive dashboard with Plotly dash

We plotted Pie Charts showing the total launches by all sites as well as Success and Failure Rate for an Individual Launch Site

- Percentage of Success relation to Launch Site

We plotted scatter graph showing the correlation between Payload Mass (Kg) and Success for all sites or by certain launch site

- It shows the relationship between Success Rate and Booster Version Category.

[GITHUB Link]

Predictive Analysis (Classification)

- Building Model
- Evaluating Model
- Finding Best Performance Classification Model
- Best Model

[GITHUB Link]

Predictive Analysis (Classification)

1. Building Model

- ❑ Load our featured engineered data into dataframe
- ❑ Transform in into NumPy arrays
- ❑ Standardize and Transform data
- ❑ Split data into training an testing data sets
- ❑ Check how many test samples have been created
- ❑ List down machine learning algorithms we want to use
- ❑ Fit our parameters and algorithm's to GridSearchCV
- ❑ Fit our datasets into the GridSearchCV objects and train our model to find best parameters and accuracy among the various algorithm's

```
y = data['Class'].to_numpy()
transform = preprocessing.StandardScaler()
X= transform.fit_transform(X)
X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size=0.2, random_state=2)
```

2. Evaluating Model

- ❑ Check accuracy for each model
- ❑ Get best hyperparameters for each type of algorithm's
- ❑ Plot confusion Matrix

```
yhat=algorithm.predict(X_test)
plot_confusion_matrix(Y_test, yhat)
```

3. Finding Best Performing Classification Model

- ❑ The model with best accuracy score wins the best performing model

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
```

Results

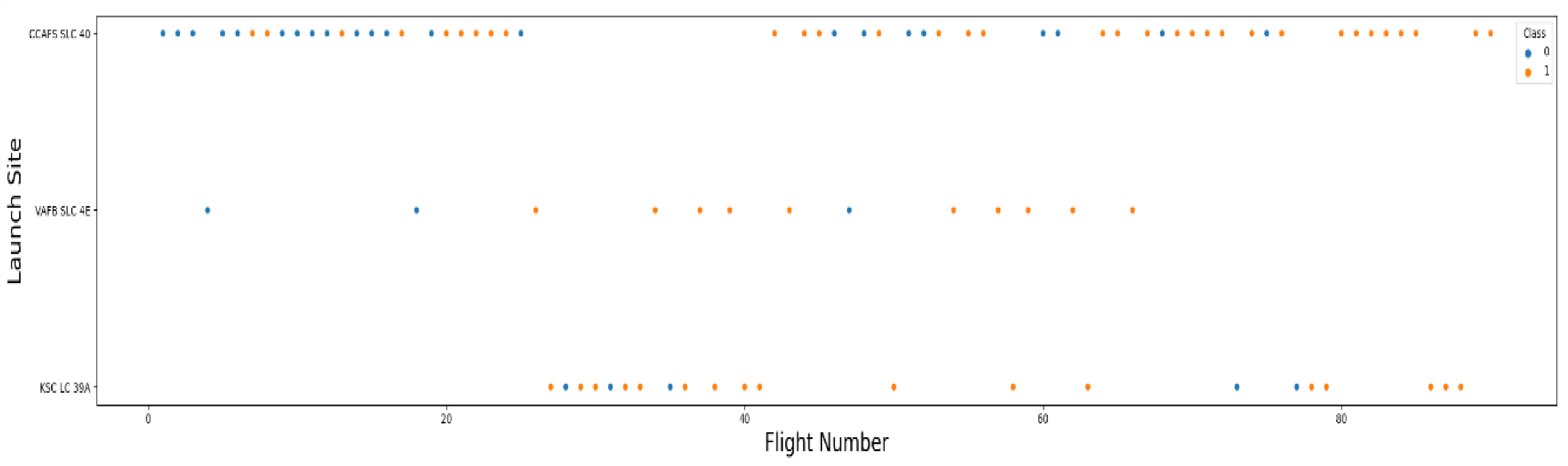
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

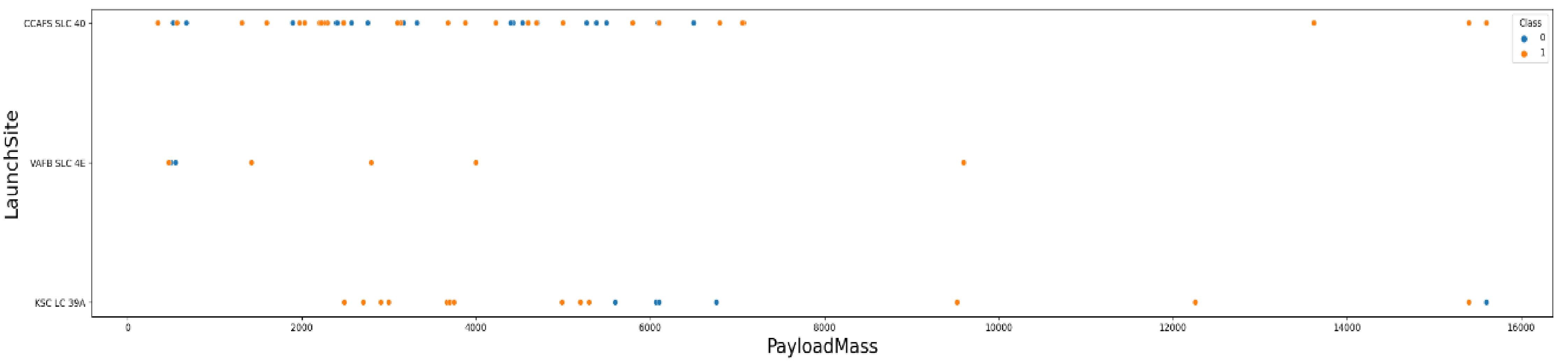
Flight Number vs. Launch Site

- From the scatter plot, it is observed that the larger the flight amount at a launch site, the greater is the success rate at that launch site.



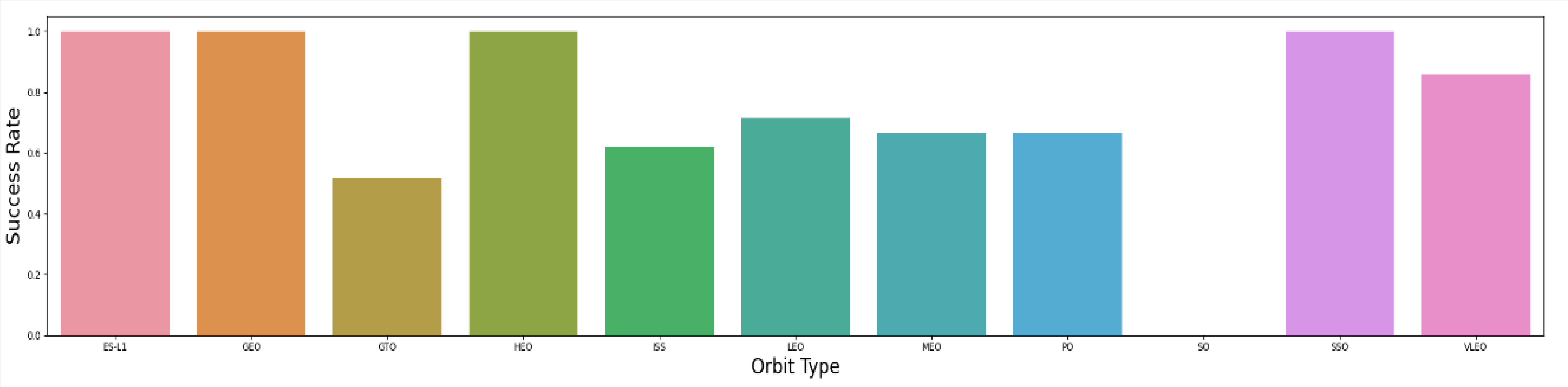
Payload vs. Launch Site

- From the scatter plot of Payload vs. Launch Site, it is observed that greater the payload (greater than 7000) higher the success rate for the rocket. But there’s no clear pattern to take a decision, i.e. if the launch site is dependent on Payload Mass for a successful launch.



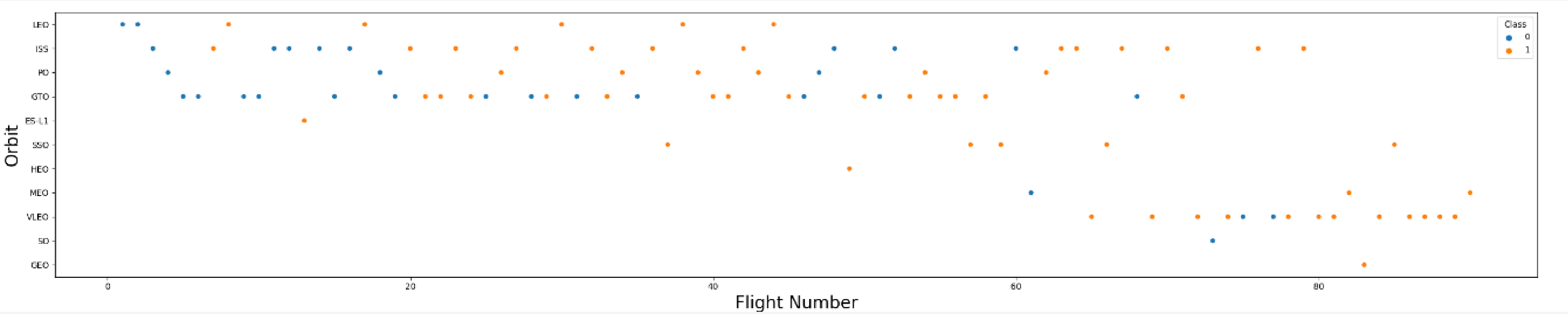
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO and SSO had the highest success rate.



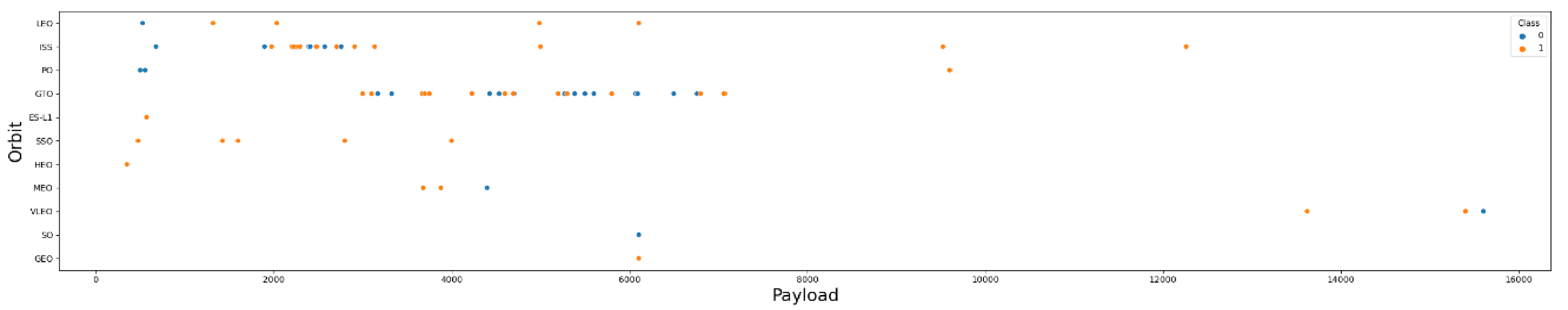
Flight Number vs. Orbit Type

- It is observed that in the LEO orbit success increases with the number of flights.
- On the other hand, there is no relationship between flight number and the GTO orbit.



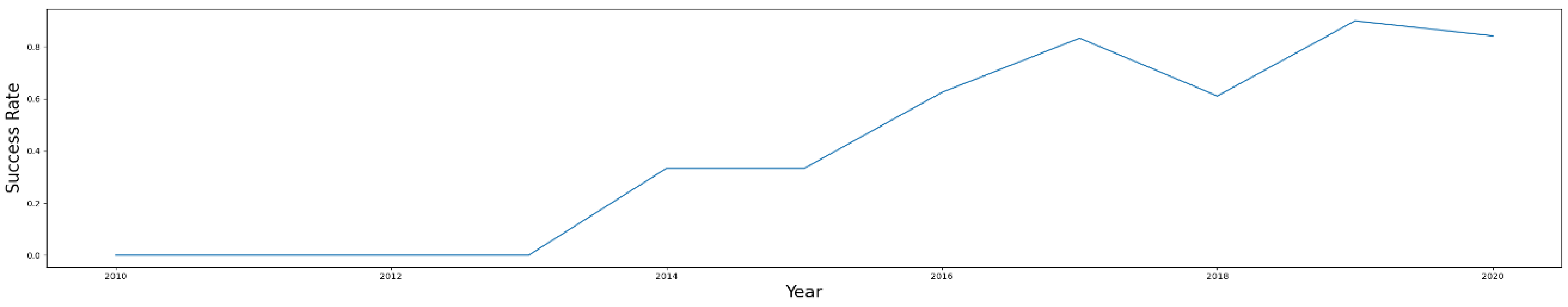
Payload vs. Orbit Type

- It is observed that heavy payloads have positive influence on LEO and ISS orbits.
- On contrast heavy payloads have negative influence on MEO, GTO and VLEO orbits.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

DESCRIPTION:

We used the key word DISTINCT to show only unique values for launch_site column from the table SpaceXTBL.

SQL Query :

```
%sql SELECT DISTINCT("Launch_Site") FROM SPACEXTBL;
 * sqlite:///my_data1.db
Done.
```

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

Launch Site Names Begin with 'CCA'

SQL Query :

```
%sql Select * from SPACEXTBL where "Launch_Site" like 'CCA%' limit 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

DESCRIPTION:

Using keyword ‘Limit 5’ in the query we fetch 5 records from table spacextbl and with condition LIKE keyword with condition ‘CCA%’. The percentage in the end suggests that the Launch_Site name must start with CCA.

Total Payload Mass

SQL Query :

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "Customer"= "NASA (CRS)";

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

SUM("PAYLOAD_MASS__KG_")
45596

DESCRIPTION:

Using the SUM function the total in column Payload_Mass_kg is calculated where the customer value is “NASA (CRS)” from the Customers column from the table SPACEXTBL.

Average Payload Mass by F9 v1.1

SQL Query :

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

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

AVG("PAYLOAD_MASS__KG_")
2928.4

DESCRIPTION:
Using the AVG function the average in column Payload_Mass_kg was calculated. The WHERE clause filters the dataset to only perform calculations on Booster Version "F9 v1.1".

First Successful Ground Landing Date

SQL Query :

```
%sql SELECT MIN(Date) AS "FIRST SUCCESSFUL LANDING DATE " FROM SPACEXTBL WHERE "Landing _Outcome"="Success (ground pad)"  ;
```

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

FIRST SUCCESSFUL LANDING DATE
22-12-2015

DESCRIPTION:
Using the MIN function in column Date it works out the minimum date and WHERE clause allows to perform these calculations only on Landing_Outcome with values "Success (ground pad)".

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query :

```
%sql SELECT "Booster_Version" FROM SPACEXTBL WHERE "Landing _Outcome"="Success (drone ship)" \
and "PAYLOAD_MASS__KG_">4000 AND "PAYLOAD_MASS__KG_"<6000 ;
```

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

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

DESCRIPTION:

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 only to display the appropriate Booster Version.

Total Number of Successful and Failure Mission Outcomes

SQL Query :

```
%sql select "Mission_Outcome",count("Mission_Outcome") from SPACEXTBL GROUP BY "Mission_Outcome" ;
```

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

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

DESCRIPTION:

We used Group by clause to group the landing outcomes for **Whether** Mission Outcome was a success or a failure.

We can see 100 Successful Mission Outcomes and 1 Failure Mission Outcome.

Boosters Carried Maximum Payload

SQL Query :

```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

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

DESCRIPTION:

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

2015 Launch Records

SQL Query :

```
%sql SELECT substr(Date, 4, 2),"Landing _Outcome","Booster_Version","Launch_Site" FROM SPACEXTBL WHERE "Landing _Outcome"="Failure (drone ship)" AND substr(Date,7,4)='2015';
```

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

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

DESCRIPTION:

We used a combinations of the WHERE clause, AND clause and substr functions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names and month number for year 2015.

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

SQL Query :

```
%sql SELECT "Landing _Outcome", COUNT("Landing _Outcome") as COUNT FROM SPACEXTBL WHERE "Date" BETWEEN '04-06-2010' and '20-03-2017'\
    AND "Landing _Outcome" LIKE '%Success%' \
    GROUP BY "Landing _Outcome"  ORDER BY COUNT("Landing _Outcome") DESC
```

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

Landing _Outcome	COUNT
Success	20
Success (drone ship)	8
Success (ground pad)	6

DESCRIPTION:

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.

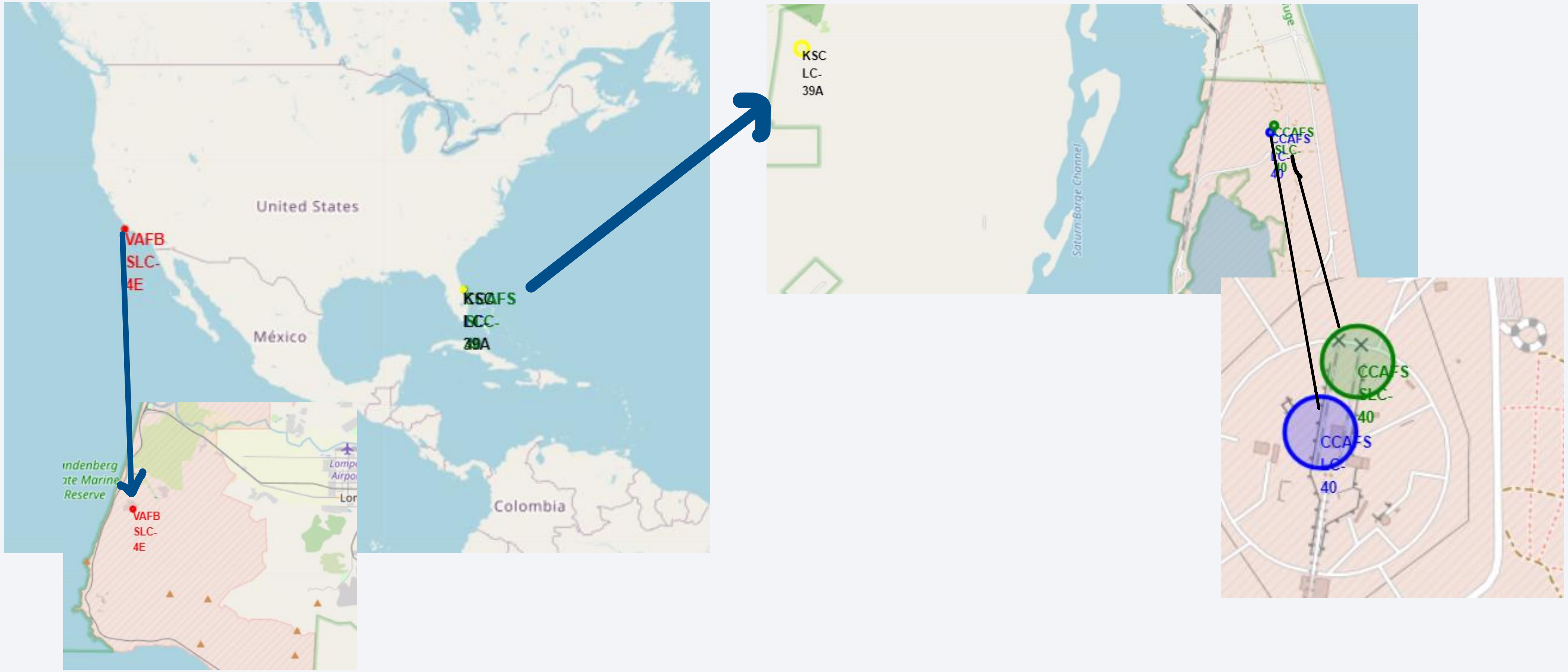
We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

Section 3

Launch Sites Proximities Analysis

All Launch Sites on Global Folium Map

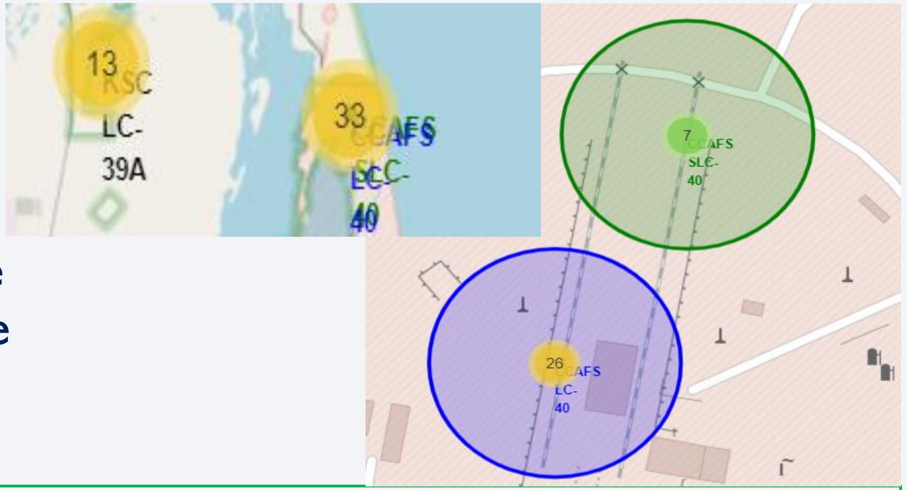
It can be seen that SpaceX launch sites are near the coasts of United States of America i.e. Florida and California region.



Colour Labeled Launch Site Outcomes

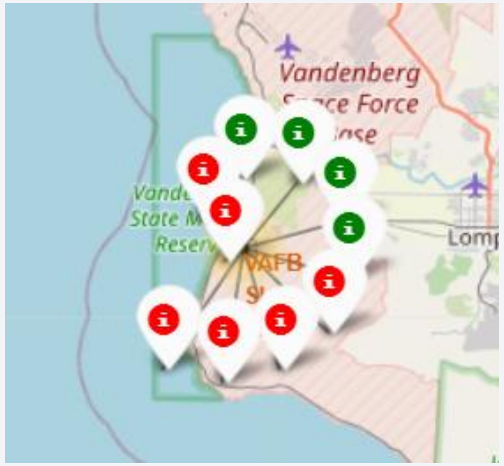
Green marker shows successful launches and **Red marker** shows failed launches.

From these screenshots it can be inferred that KSC LC-39A has the most successful launches.



California Launch Site

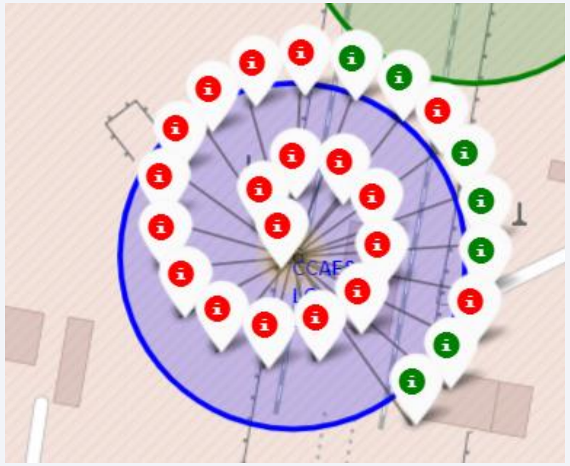
VAFB SLC-4E



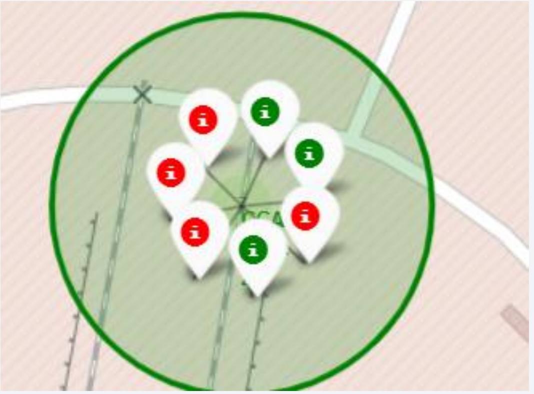
Florida Launch Site

KSC LC-39A

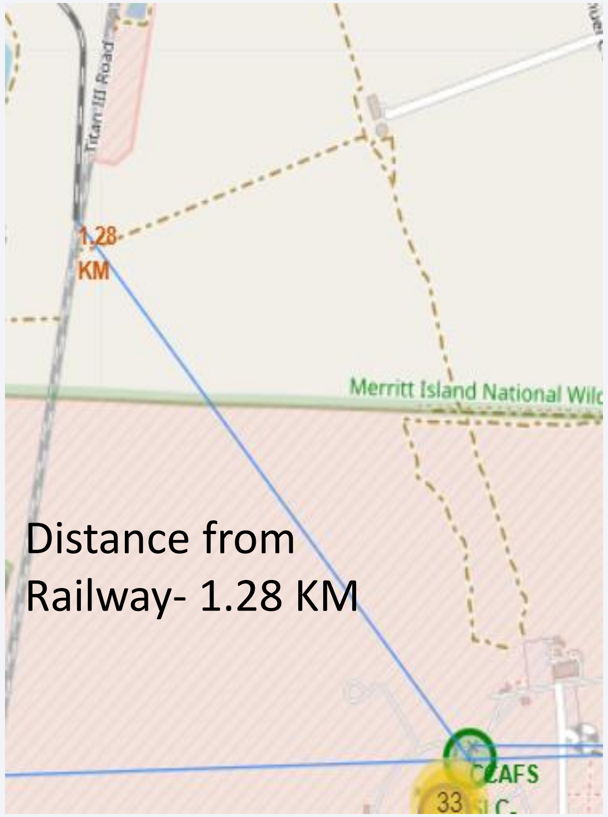
CCAFS LC-40



CCAFS SLC-40

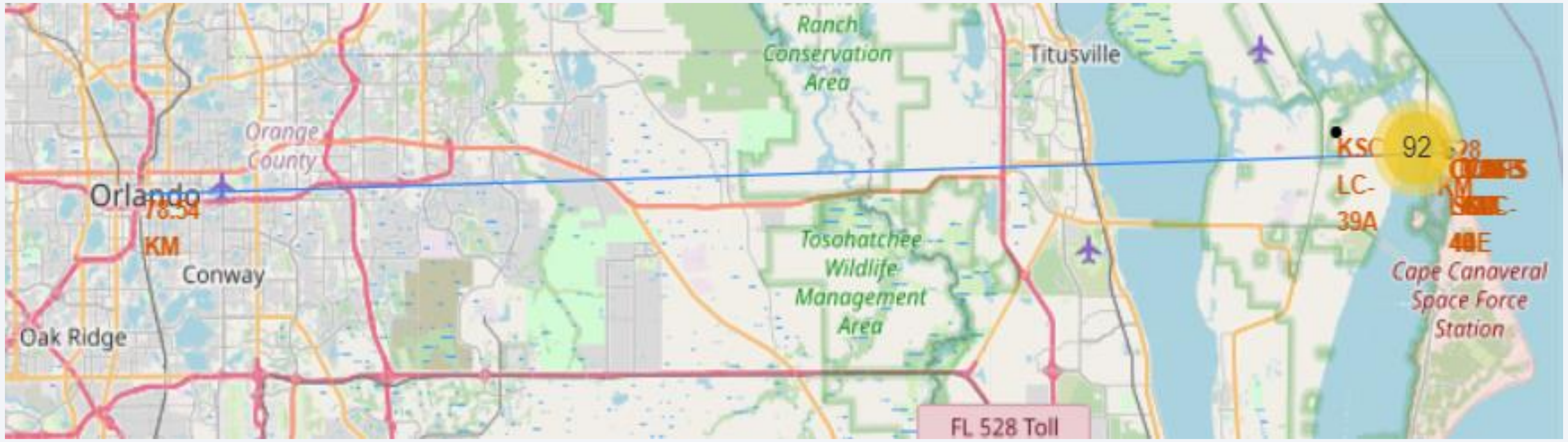


Distance of Launch Site and its Proximities



Distance from Railway- 1.28 KM

Distance from Highway- 0.58 KM, Distance from Coastline point- 0.89 KM



Distance from Closest City- 78.54 KM

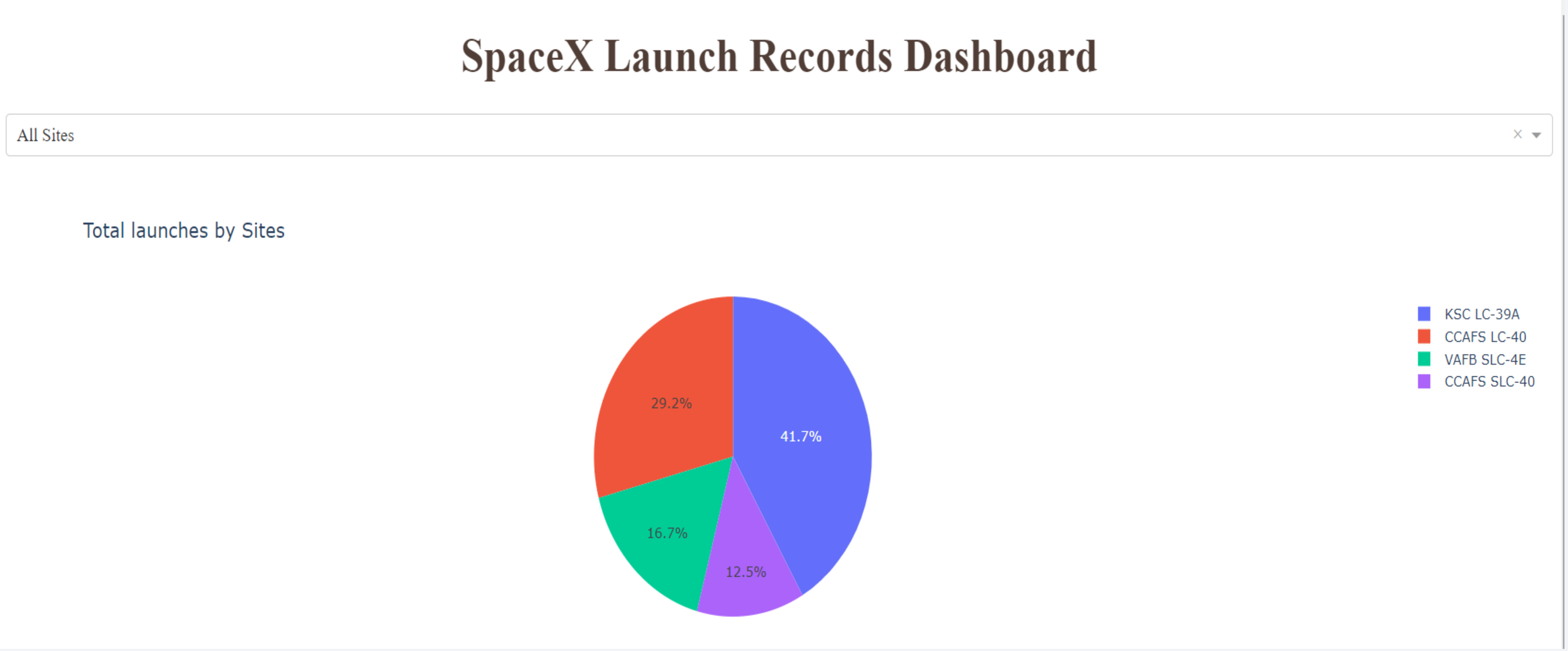
From these images we can infer that -

1) Launch Sites are in close proximities from railways,
2) Launch Sites are in close proximities from coastline,
3) Not always in close proximity to highways and
4) Launch Sites keep certain distance away from cities.

Section 4

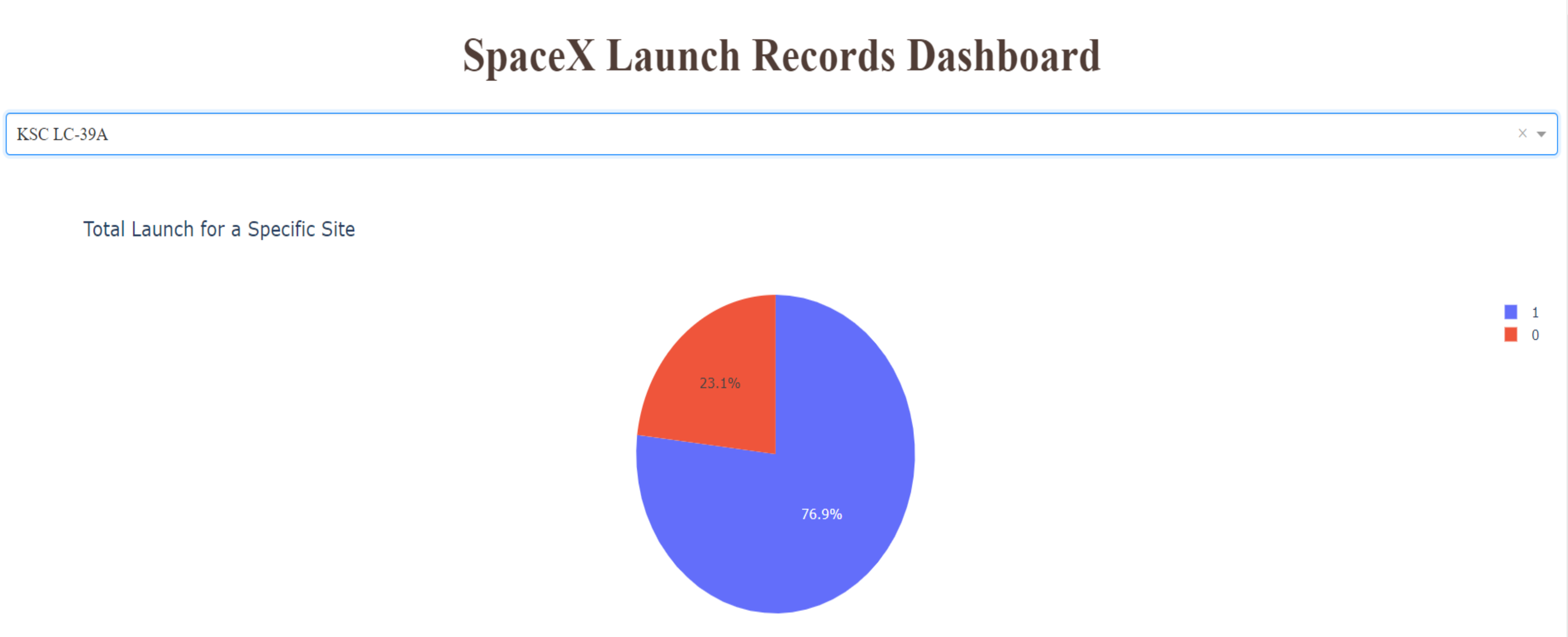
Build a Dashboard with Plotly Dash

Launch Success Count for All Sites



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

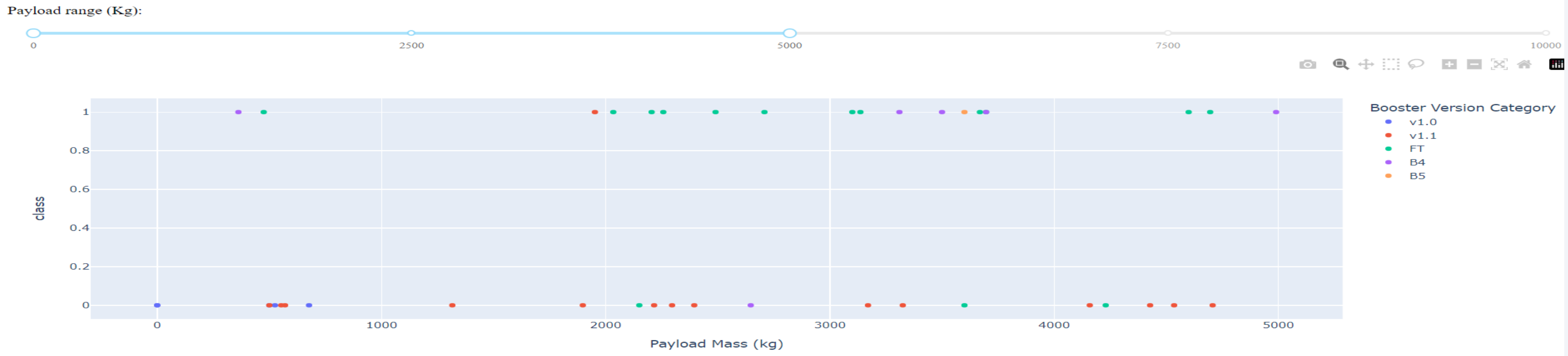
A look at the Highest Success Ratio Launch Site



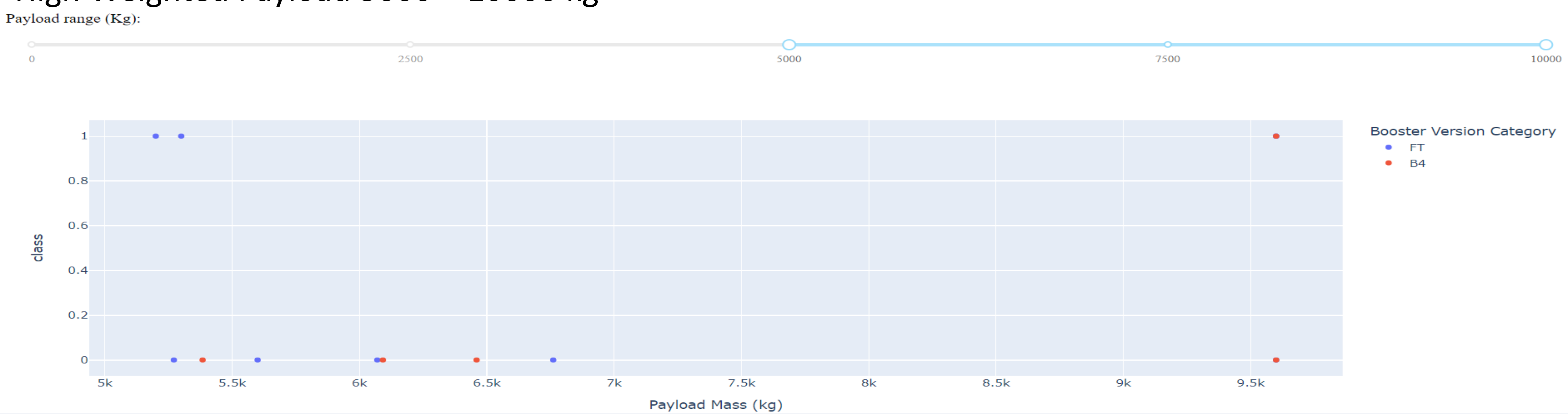
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload vs. Launch Outcome scatter plot for all sites

Low Weighted Payload 0 - 5000 kg



High Weighted Payload 5000 – 10000 kg



Which payload range has the highest launch success rate

- 2000 kg – 6000 kg

Which booster version has the highest launch success rate

- FT

Section 5

Predictive Analysis (Classification)

Classification Accuracy

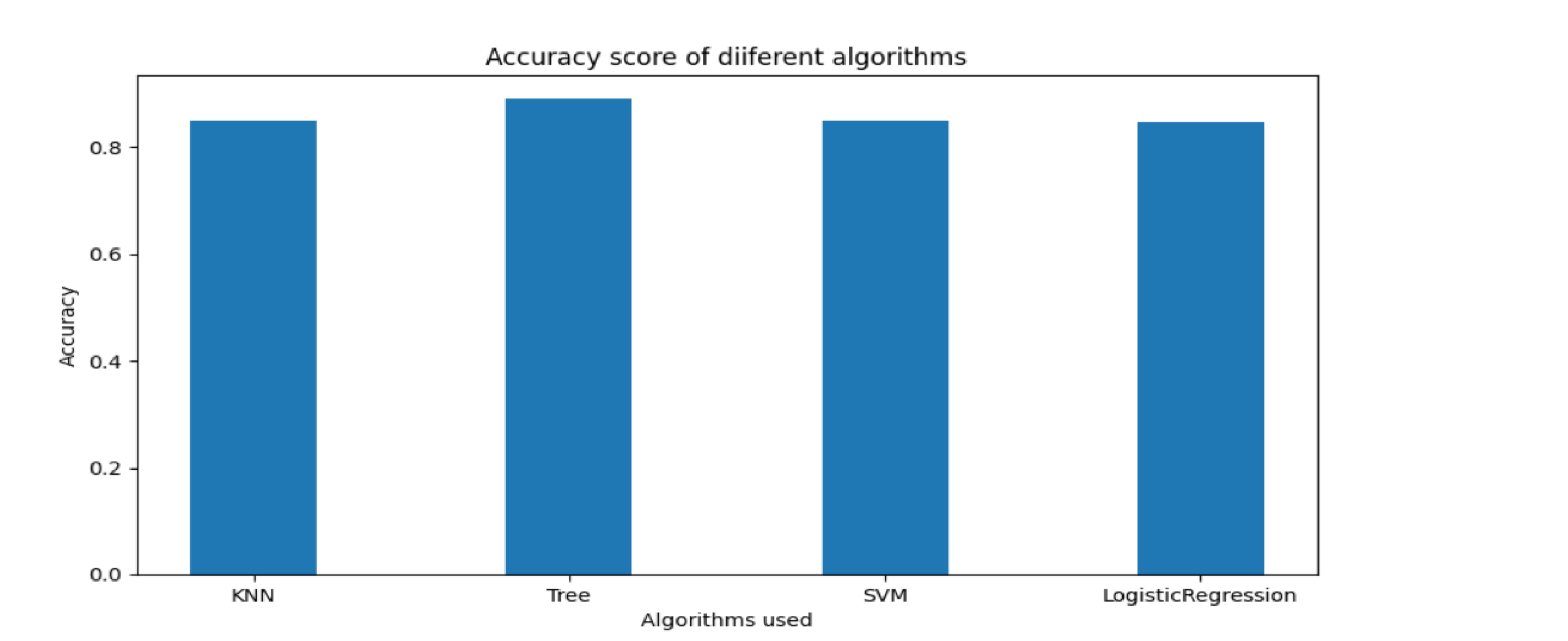
As compared, the accuracy for the training data set is extremely close, but we can see that the decision tree classifier is the model with the highest classification accuracy with a score of 0.9

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'SVM':svm_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
if bestalgorithm == 'SVM':
    print('Best Params is :',svm_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.9
Best Params is : {'criterion': 'gini', 'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 5, 'splitter': 'random'}
```

Classification Accuracy

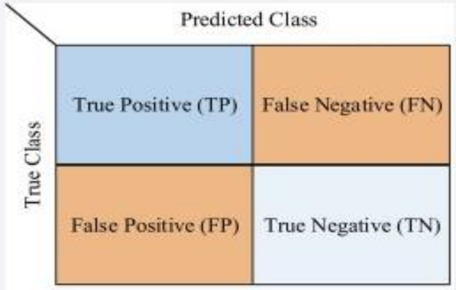
As seen below the accuracy for the training data set is extremely close, but we can see that the decision tree classifier is the model with the highest classification accuracy with a score of 0.9



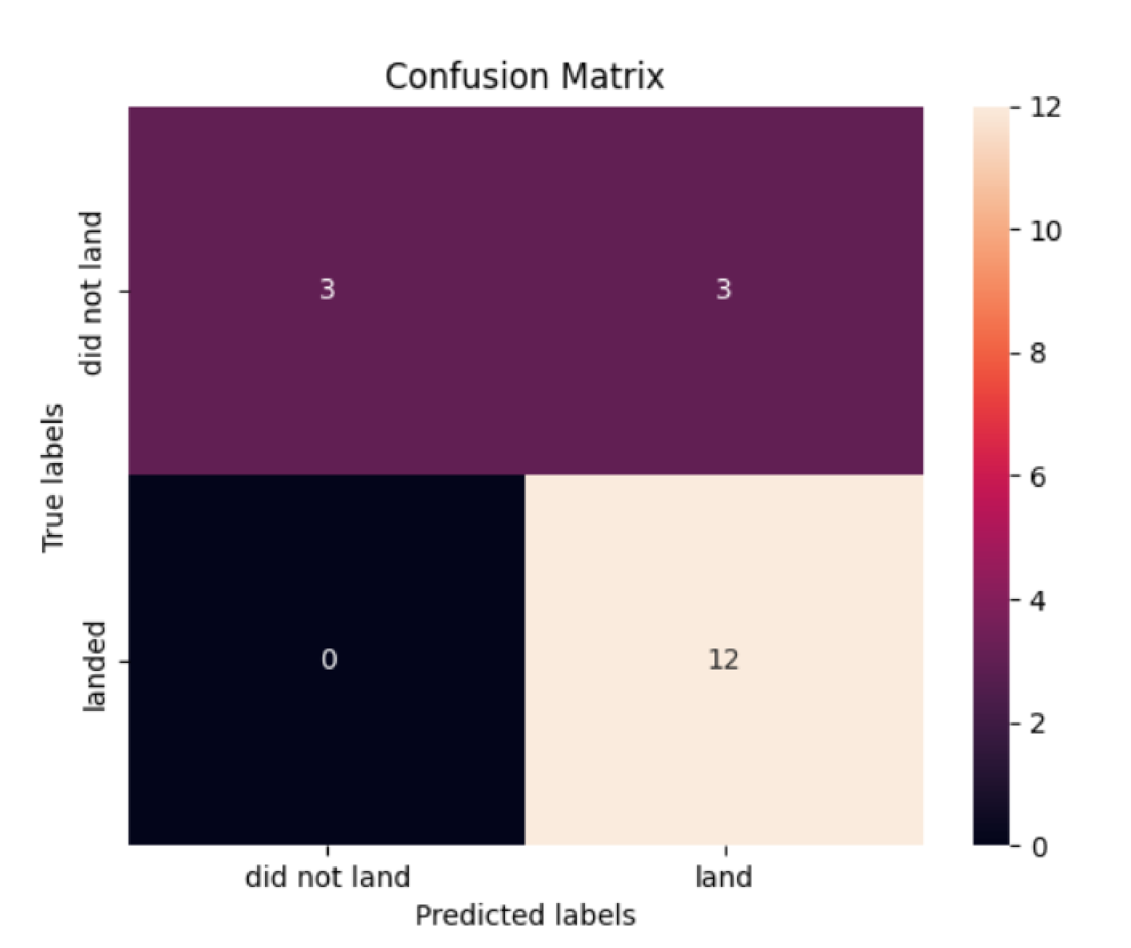
Confusion Matrix

The confusion matrix for the decision tree classifier explains us that the classifier can distinguish between the predicted and actual values.

The major problem in this matrix are the false positives i.e., unsuccessful landing's marked as successful landing by the classifier as that is what leads to error.



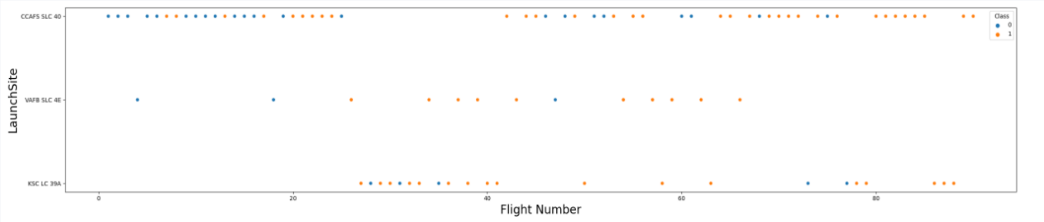
Accuracy: (TP+TN)/Total = (12+3)/18 = 0.83333
Precision: TP/Predicted Yes = 12/15 = 0.8
Prevalence: Actual Yes/Total = 12/18 = 0.6667

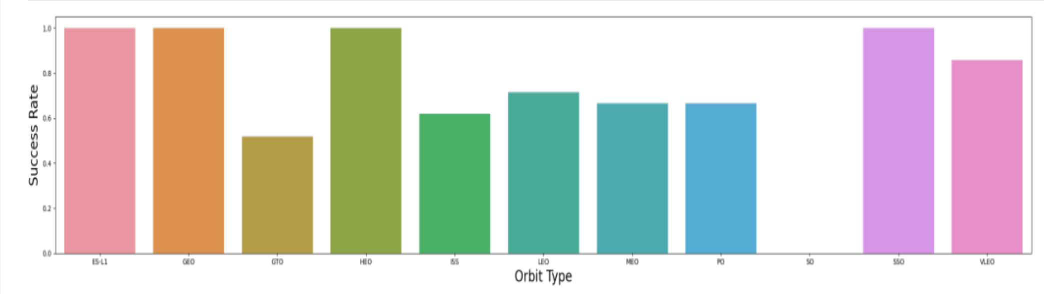


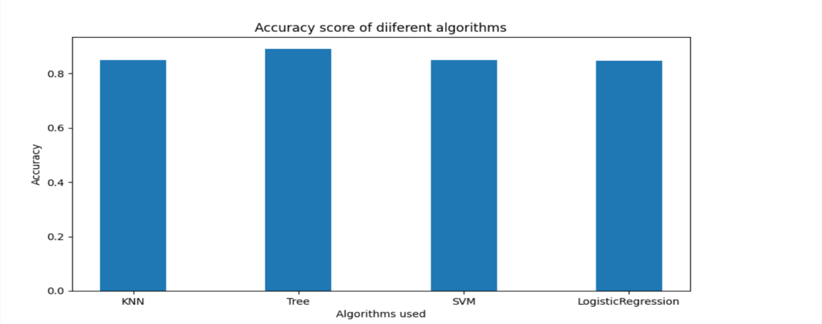
Conclusions

We can conclude that:

- After doing launches for multiple times (number of launches), a launch site's success rate increases.
- Launch success rate has been increasing relatively with time as it reached it's peak in 2020.
- KSC LC-39A had the most successful launches of any sites.
- Payload Mass seems to have negative impact on success.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- The Decision tree classifier is the best machine learning algorithm for this task.







Appendix

GITHUB LINK TO THE REPOSITORY (README) :

- https://github.com/Ritvik-km/SPACEX-Data_Science_Project/blob/d86cb7d1ecd82683953a018218082d7af339f186/README.md

Thank you!