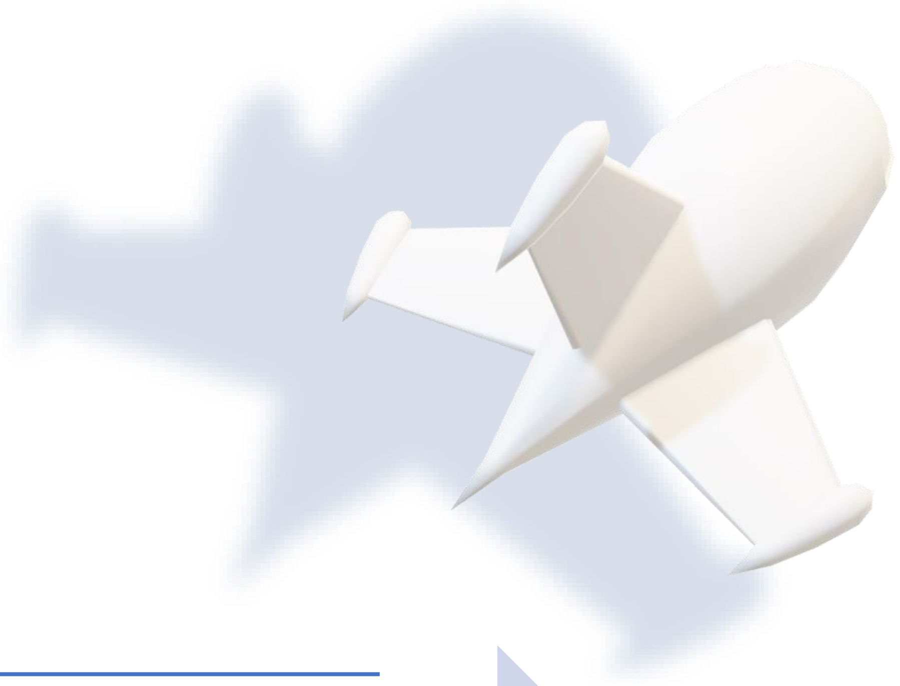
Applied Data Science Capstone

Anush Debnath

4th May 2023



Outline

Executive Summary → Introduction → Methodology → Results → Conclusion → Appendix

Executive Summary

Summary of methodologies

- Data Collection with API
- Data Collection with Web Scraping
- Exploratory Data Analysis Using SQL and Visualization
- Interactive Visual Analytics with Folium
- Building an Interactive Dashboard with Plotly Dash
- Prediction Analysis With Classification Models

Summary of all results

- Data Analysis along with Interactive Visualizations
- Model Prediction Analysis

Introduction

Project background and context

Prediction of the Falcon 9 first stage will land successfully. SpaceX 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 SpaceX can reuse the first stage. 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 SpaceX for a rocket launch.

Problems you want to find answers

Factors affecting the landing of rocket

How to achieve the best result

SECTION 1

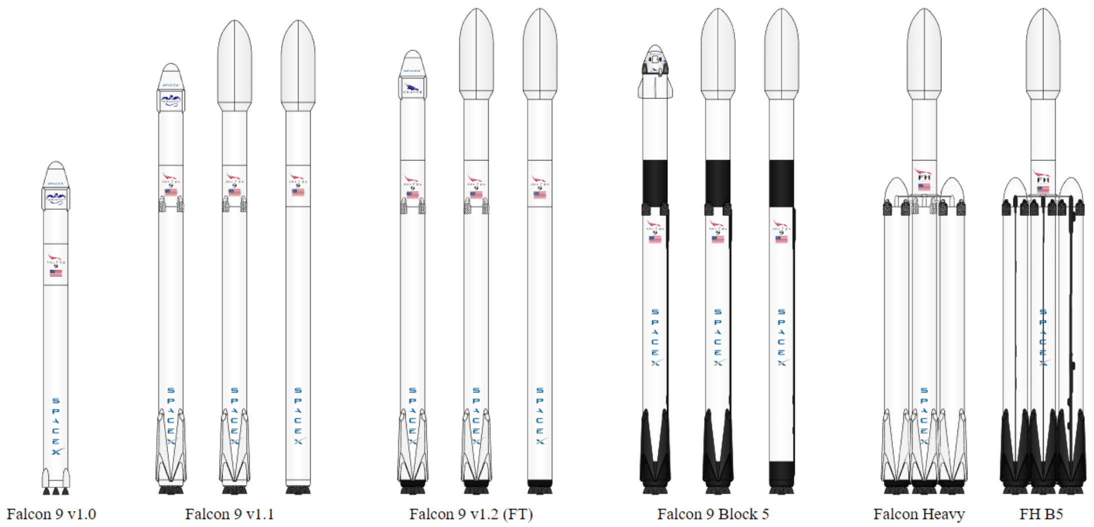
METHODOLOGY

Executive Summary

- **Data collection methodology:** From SpaceX Rest API, Web Scraping from Wikipedia
- **Perform data wrangling:** Transforming Data for Machine Learning Models
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Showing Patterns between Data with the help of Graphs**
- **Perform interactive visual analytics using Folium and Plotly Dash**
- **Perform predictive analysis using classification models:** Performing Train test split then fitting the train data to the model.

Data Collection

Data collection is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes.



Data Collection – SpaceX API

Getting Data from API Or by web Scrapping

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

Make Dataframe from it

```
jlist = requests.get(static_json_url).json()
df2 = pd.json_normalize(jlist)
df2.head()
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

Then, we need to create a Pandas data frame from the dictionary launch_dict.

```
# Create a data from launch_dict
data_falcon9 = pd.DataFrame(launch_dict)
```

Filter the Dataframe

```
# Hint data['BoosterVersion']!='Falcon 1'

# data_falcon9['BoosterVersion']!='Falcon 9' returns true for all rows except 'Falcon 9' and running drop, drops those rows.
data_falcon9.drop(data_falcon9[data_falcon9['BoosterVersion']!='Falcon 9'].index, inplace = True)
```

Now that we have removed some values we should reset the FlgihtNumber column

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

```
# Calculate the mean value of PayloadMass column
avg_payload_mass = data_falcon9["PayloadMass"].astype("float").mean(axis=0)
# Replace the np.nan values with its mean value
data_falcon9["PayloadMass"].replace(np.nan, avg_payload_mass, inplace=True)
```

You should see the number of missing values of the `PayLoadMass` change to zero.

```
data_falcon9.isnull().sum()
```

Save the Dataframe in CSV, JSON, XML etc. format

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40

https://github.com/Anushdebnath03

Data Collection - Scraping

Getting Response from Web page

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
data  = requests.get(static_url).text
```

Creating BeautifulSoup Object

```
soup = BeautifulSoup(data,'html.parser')
```

Finding tables

```
html_tables=soup.find_all("table")
html_tables
```

Getting column names

```
column_names = []

# Apply find_all() function with `th` elemen
# Iterate each th element and apply the prov
# Append the Non-empty column name (`if name
ths = first_launch_table.find_all('th')
for th in ths:
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date
1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	F9 v1.0B0003.1	Failure	4 June 2010
2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010
3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt	22 May 2012
4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success	F9 v1.0B0006.1	No attempt	8 October 2012
5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success	F9 v1.0B0007.1	No attempt	1 March 2013

Creation of dictionary and appending data to Keys

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
```

Converting dictionary to dataframe

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

https://github.com/Anushdebnath03

Dataframe to Csv file

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

Data Wrangling

Data wrangling is the process of converting raw data into a usable form. It may also be called data munging or data remediation. You'll typically go through the data wrangling process prior to conducting any data analysis in order to ensure your data is reliable and complete.

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	L
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	

Calculate the number of launches on each site

```
df['LaunchSite'].value_counts()

CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

Calculate the number and occurrence of each orbit

```
df['Orbit'].value_counts()
```

Calculate the number and occurence of mission outcome per orbit type

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

Create a landing outcome label from Outcome column

```
df['Class'] = df['Outcome'].apply(lambda landing_class: 0 if landing_class in bad_outcomes else 1)
df[['Class']].head(8)
```

Export dataframe into csv file

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

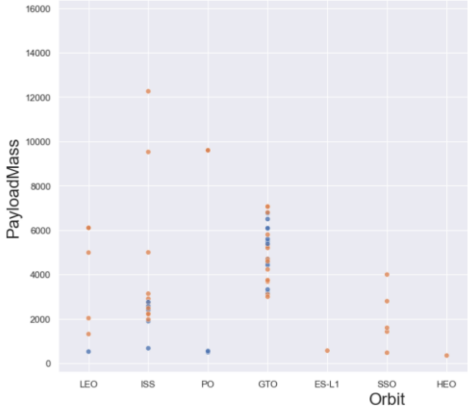
https://github.com/Anushdebnath03

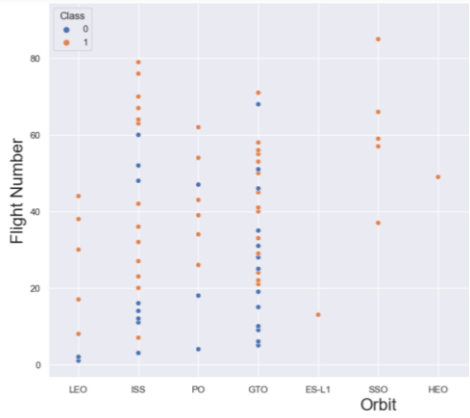
EDA with Data Visualization

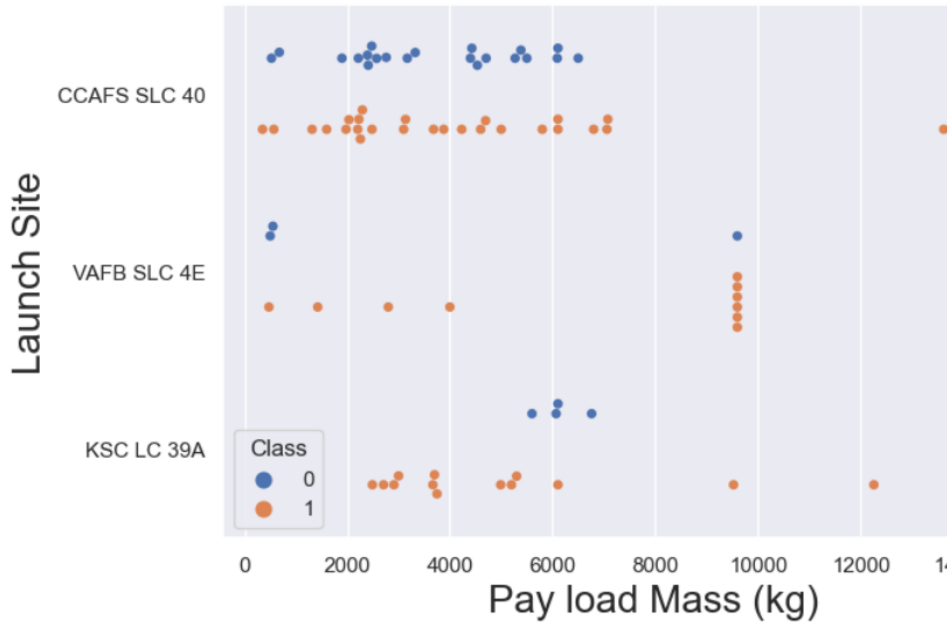
Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

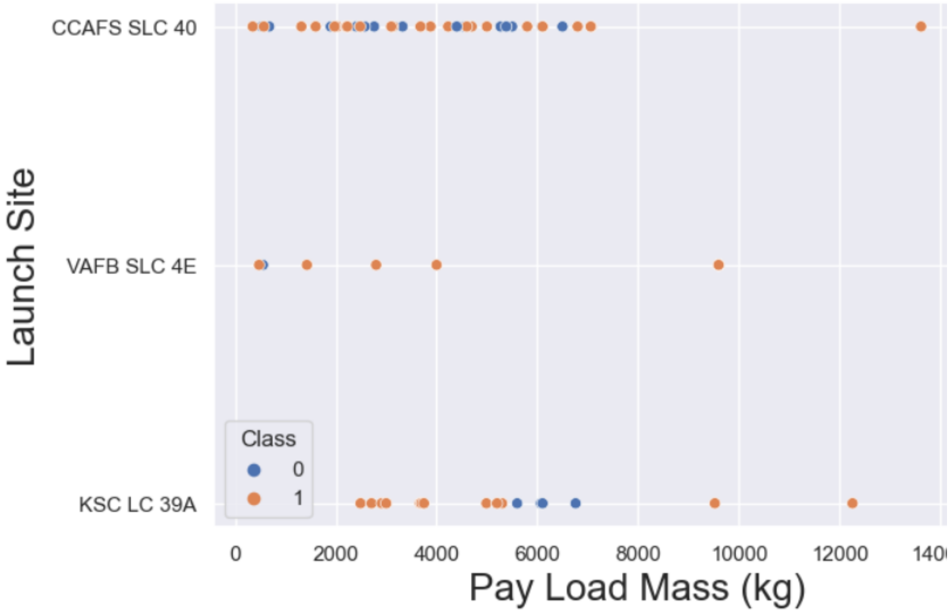
- Scattered Plot: Visualize the relationship between Flight Number and Launch Site, Payload and Launch Site, Success rate of each orbit type, Flight Number and Orbit type, Payload and Orbit type.

We can determine both the probability of successful landing









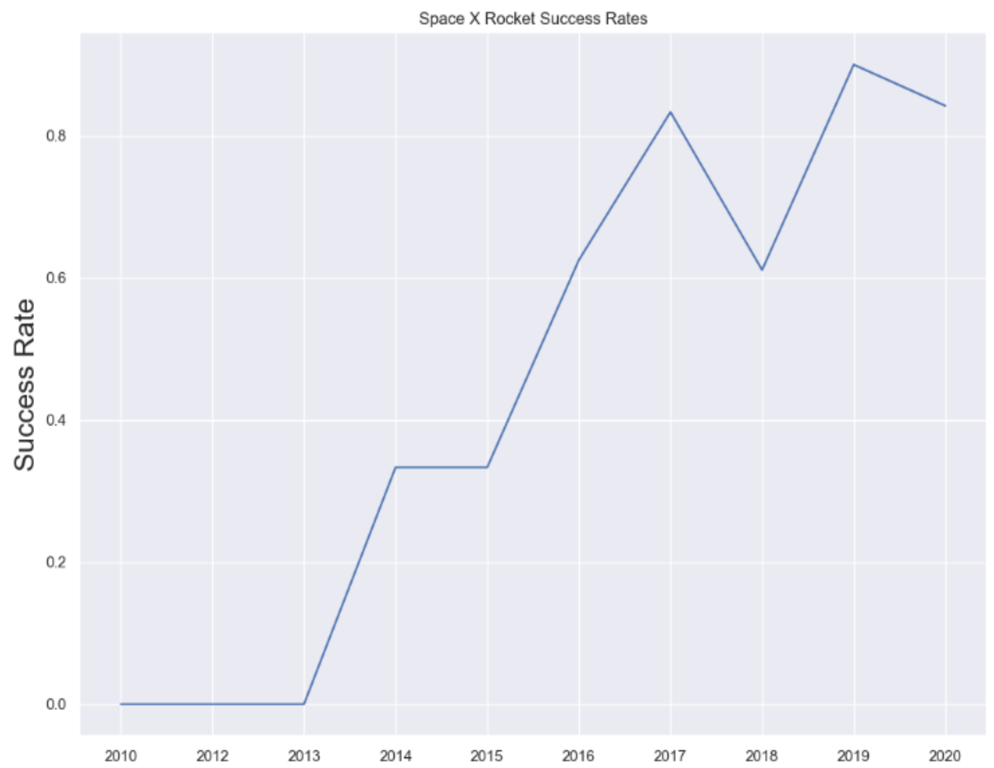
Line Graph:

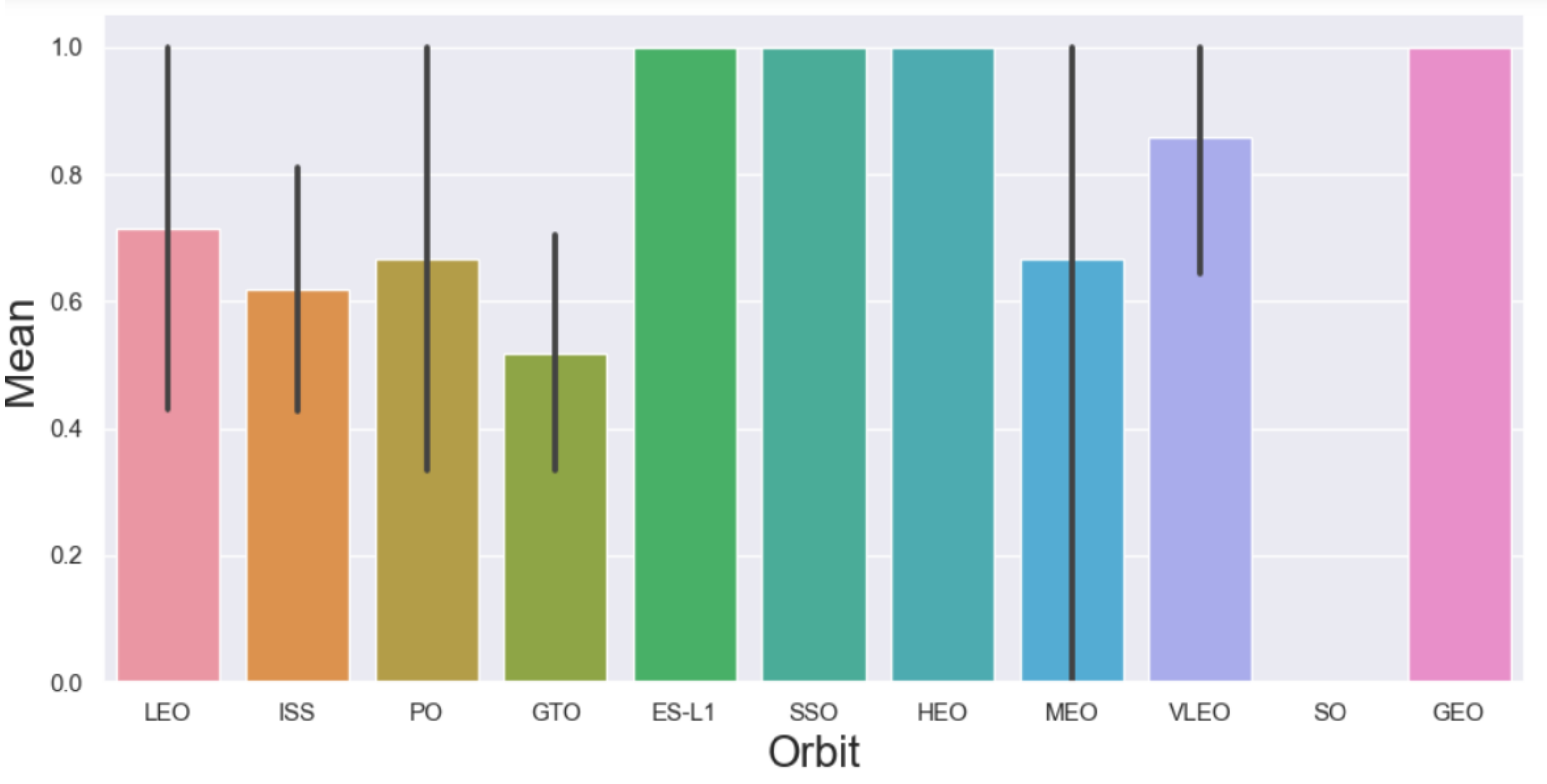
Landing Success rate

Bar graph:

success rate of each orbit type

With the help of bar Graph we can easily determine which orbits have the highest probability of success





https://github.com/Anushdebnath03

EDA with SQL

While most databases focus on the management of structured and relational datasets, SQL Server is also capable of handling multiple data types, including non-relational and unstructured data.
Here we use IBM's Db2 service which is the database to run your mission-critical workloads

Steps to Link your DB2 in Jupyuter notebook

Import necessary packages

```
!pip install sqlalchemy==1.3.9
!pip install ibm_db_sa
!pip install ipython-sql
```

```
%load_ext sql
```

```
%sql ibm_db_sa://my-username:my-password@my-hostname:my-port/my-db-name?security=SSL
```

```
%sql <query>
```

Task Performed

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 failed landing_outcomes in drone ship, their booster versions, and launch site names for the in year 2015
10. 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 powerful Python library that helps you create several types of Leaflet maps. By default, Folium creates a map in a separate HTML file. Since Folium results are interactive, this library is very useful for dashboard building

Circle marker

function to circle the coordinates
folium.CircleMarker().add_to(m)

Marker Cluster Object

cluster multiple maps, simplifies the view
folium.Marker().add_to(m)

Map Maker

To create a base map, simply pass your starting coordinates to Folium
m=folium.Map()

Icon Maker

Function to make Icons
folium.Marker(location=[0,20], icon=folium.Icon().add_to(m)

Poly Line

folium can show linear elements on a map using PolyLine
trail_coordinates = [“”coordinates……..””]
folium.PolyLine(trail_coordinates, tooltip="Coast").add_to(m)

Task Performed

Task 1: Mark all launch sites on a map
Task 2: Mark the success/failed launches for each site on the map
Task 3: Calculate the distances between a launch site to its proximities

Build a Dashboard with Plotly Dash

Scatter Graph
It shows the relationship between Success rate
Pie Chart
It shows the percentage of success in relation to launch site

Task Performed

TASK 1: Add a Launch Site Drop-down Input Component
Task 2: Add a callback function to render success-pie-chart based on selected site dropdown
TASK 3: Add a Range Slider to Select Payload
TASK 4: Add a callback function to render the success-payload-scatter-chart scatter plot

Dash and its components

import dash
import dash_html_components as html
import dash_core_components as dcc
from dash.dependencies import Input, Output

Pandas

import pandas as pd

Pie Chart

px.pie()

Plotly

import plotly.express as px

Dropdown

dcc.Dropdown()

Rangeslider

dcc.RangeSlider()

Scatter Chart

px.scatter()

https://github.com/Anushdebnath03

Predictive Analysis (Classification)

Loading the dataframe

- Create a NumPy array from the column Class in data
- Use the function train_test_split to split the data X and Y into training and test data
- Create a GridSearchCV object

Fitting in the models

- Checking Accuracy
- Getting best hyperparameter for each algorithum
- Plotting confusion matrix

Evaluation

- Finding Best Performing Classification model

https://github.com/Anushdebnath03

Results

Exploratory data analysis results

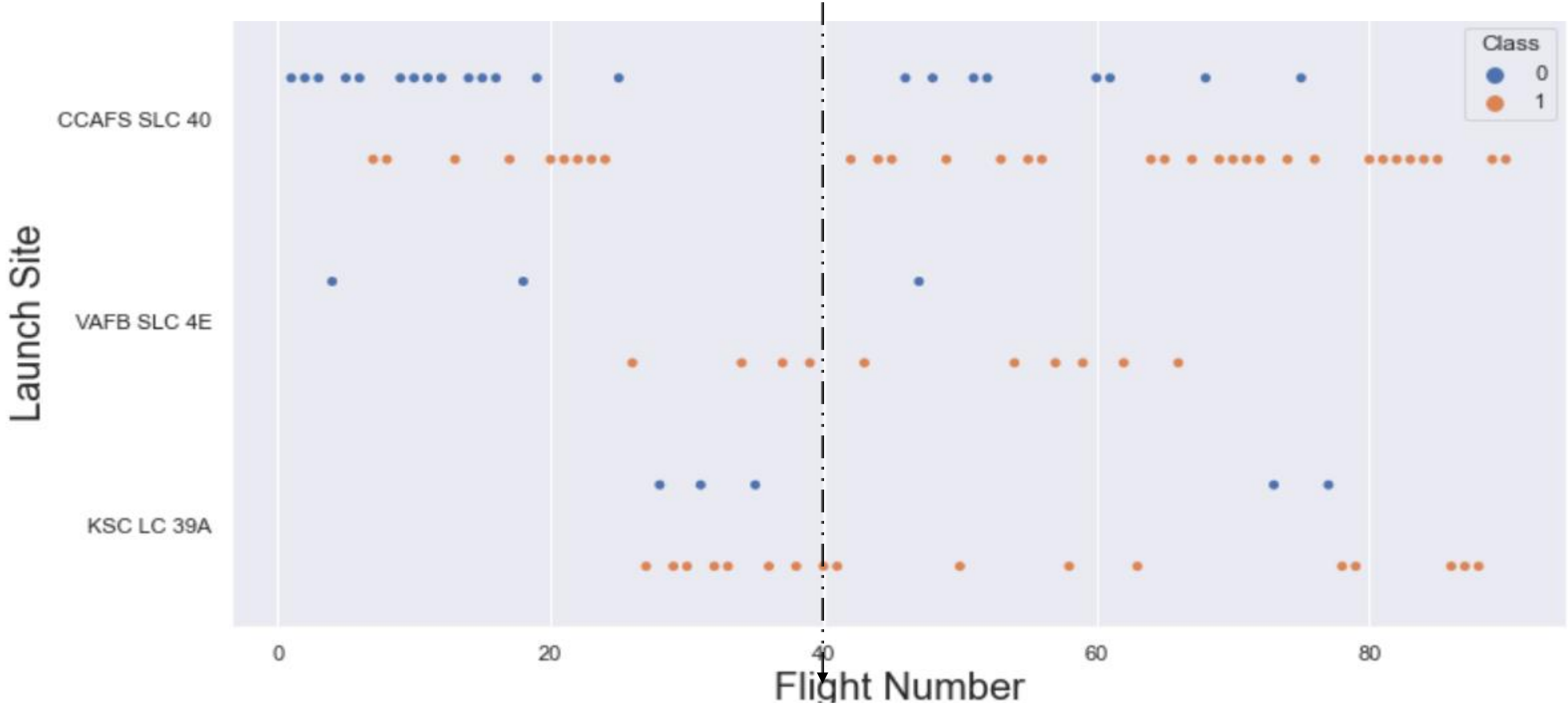
Interactive analytics demo

Prediction analysis report

SECTION 2

INSIGHTS DRAWN FROM EDA

Flight Number vs. Launch Site



The success rate for the rocket is increasing with higher flight number(grater then 40)

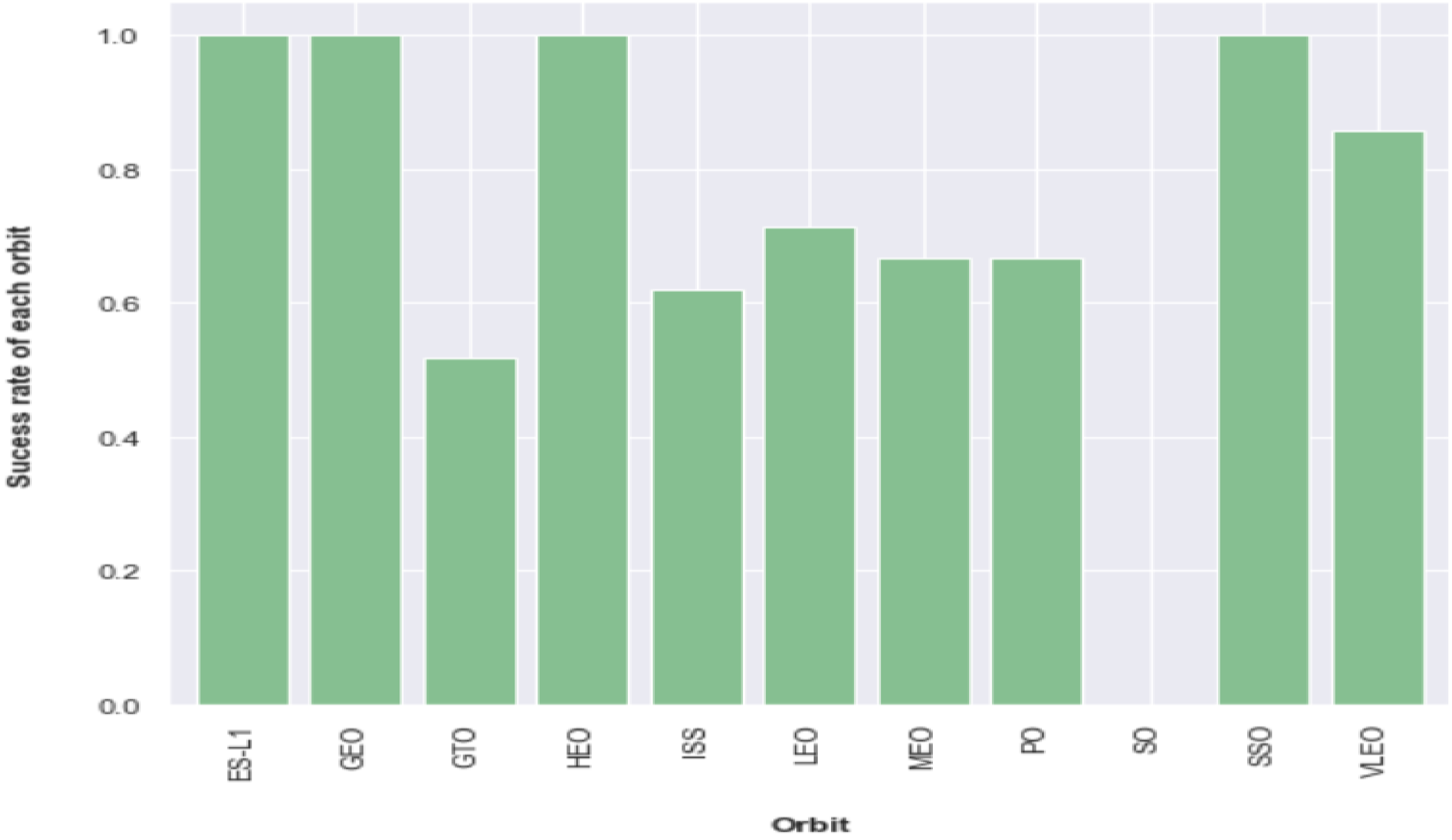
https://github.com/Anushdebnath03

Payload vs. Launch Site

The greater the payload mass(greater then 9000kg) the higher the success rate for the rocket

https://github.com/Anushdebnath03

Success Rate vs. Orbit Type

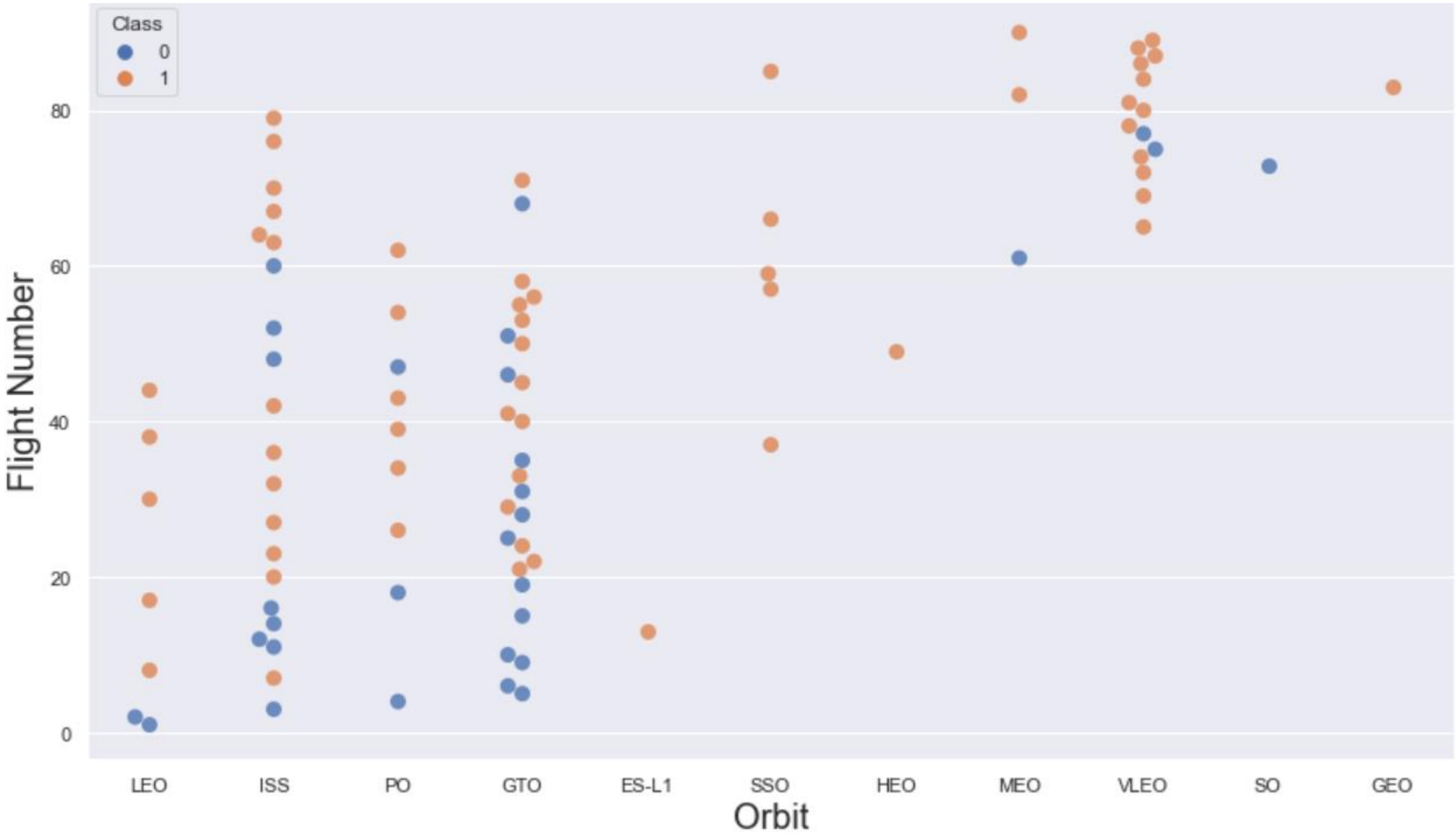


ES-L1, GEO, HEO, SSO has highest Success rates. *SO has poorest*.

https://github.com/Anushdebnath03

Flight Number vs. Orbit Type

The LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



https://github.com/Anushdebnath03

Payload vs. Orbit Type

You should observe that Heavy payloads have a negative influence on GTO, MEO, VLEO orbits and positive on Polar LEO, ISS orbits.

https://github.com/Anushdebnath03

Launch Success Yearly Trend

you can observe that the success rate since 2013 kept increasing till 2020

https://github.com/Anushdebnath03

All Launch Site Names

%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'

%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

Total Payload Mass

%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

https://github.com/Anushdebnath03

Average Payload Mass by 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

%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

%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

https://github.com/Anushdebnath03

Total Number of Successful

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

and Failure Mission Outcomes

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

Successful Mission
100

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);
```

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

https://github.com/Anushdebnath03

2015 Launch Records

%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

%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

%sql SELECT COUNT(LANDING__OUTCOME) AS "Rank success count between 2010-06-04 and 2017-03-20" FROM SPACEX \
WHERE LANDING__OUTCOME LIKE '%Success%'
AND DATE > '2010-06-04' AND DATE < '2017-03-20'
;

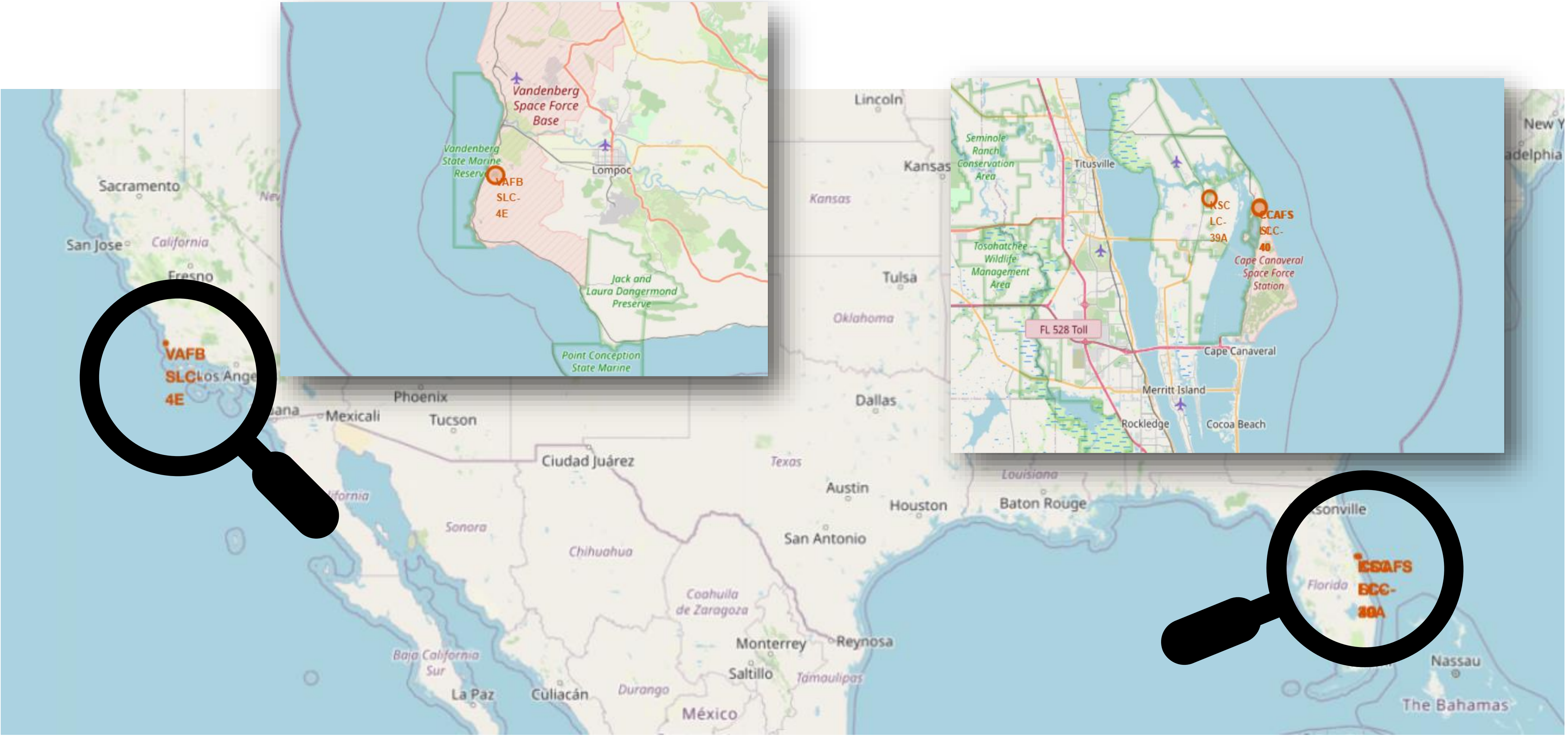
Rank success count between 2010-06-04 and 2017-03-20
8

https://github.com/Anushdebnath03

SECTION 3

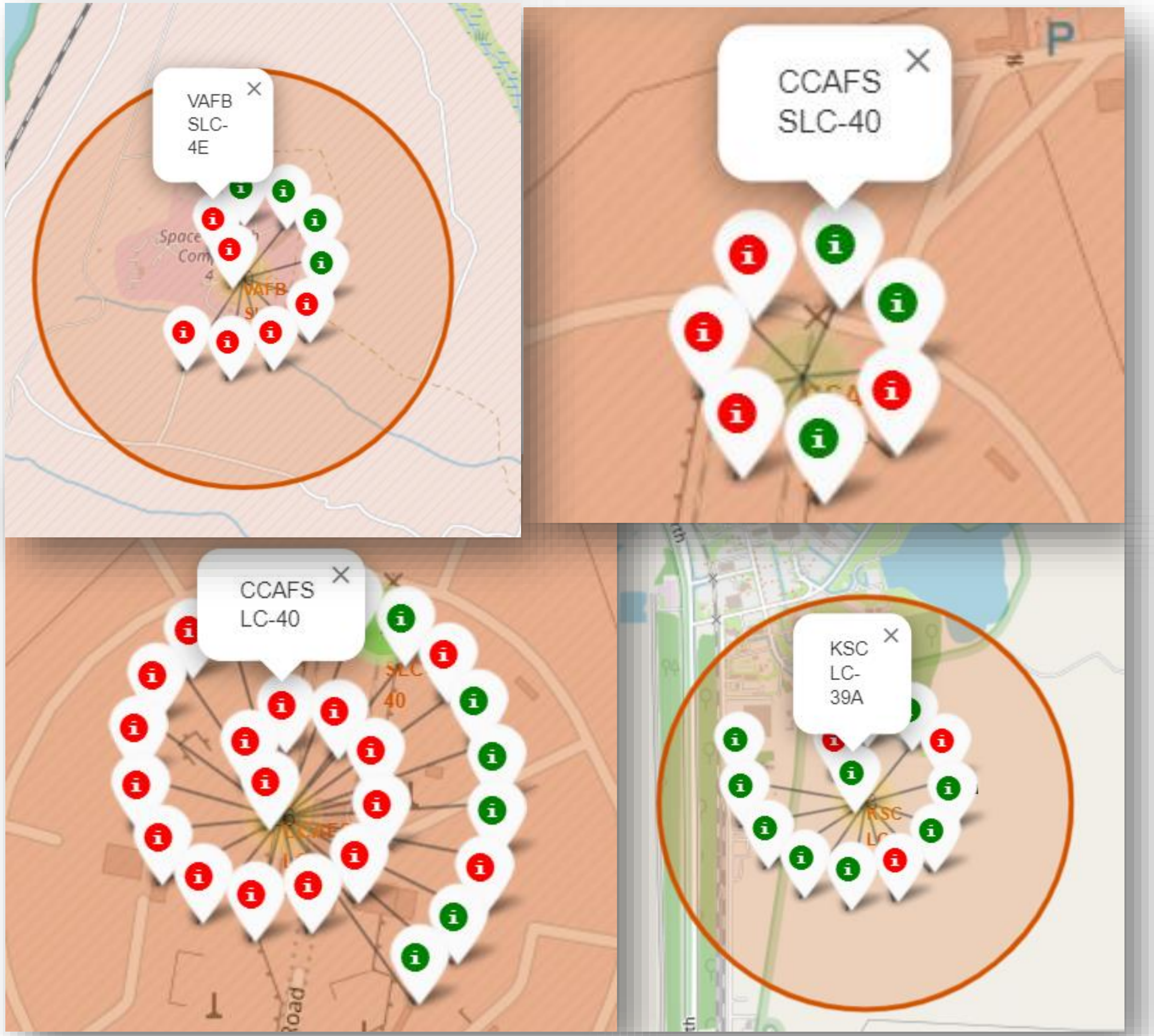
LAUNCH SITES PROXINITIES ANALYSIS

All Launch Sites on a Map



https://github.com/Anushdebnath03

Coloured Label Landing Records



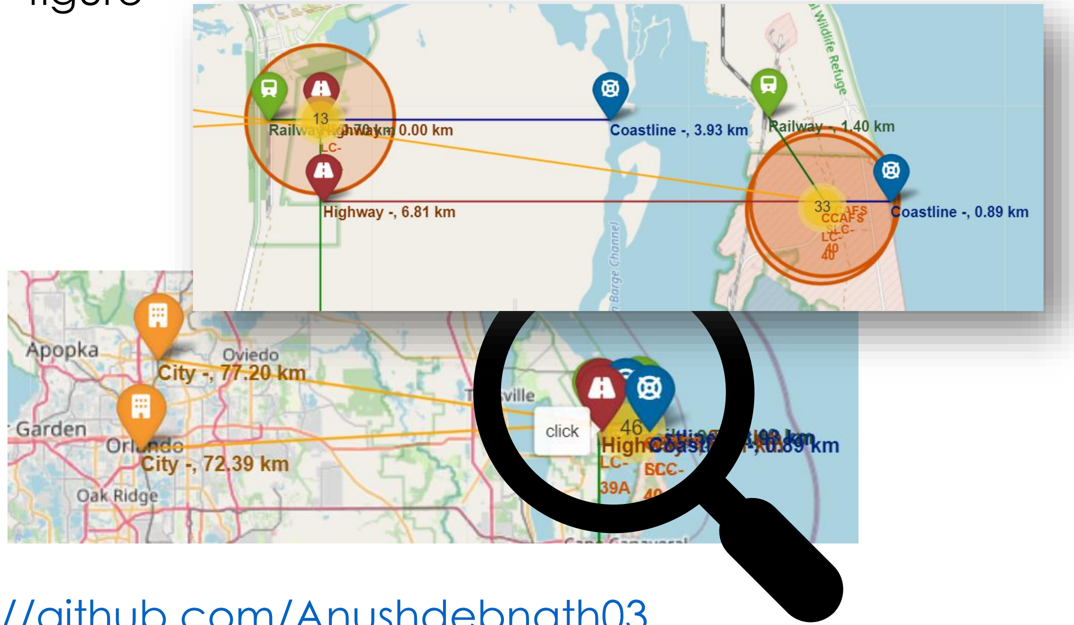
Successful Landing

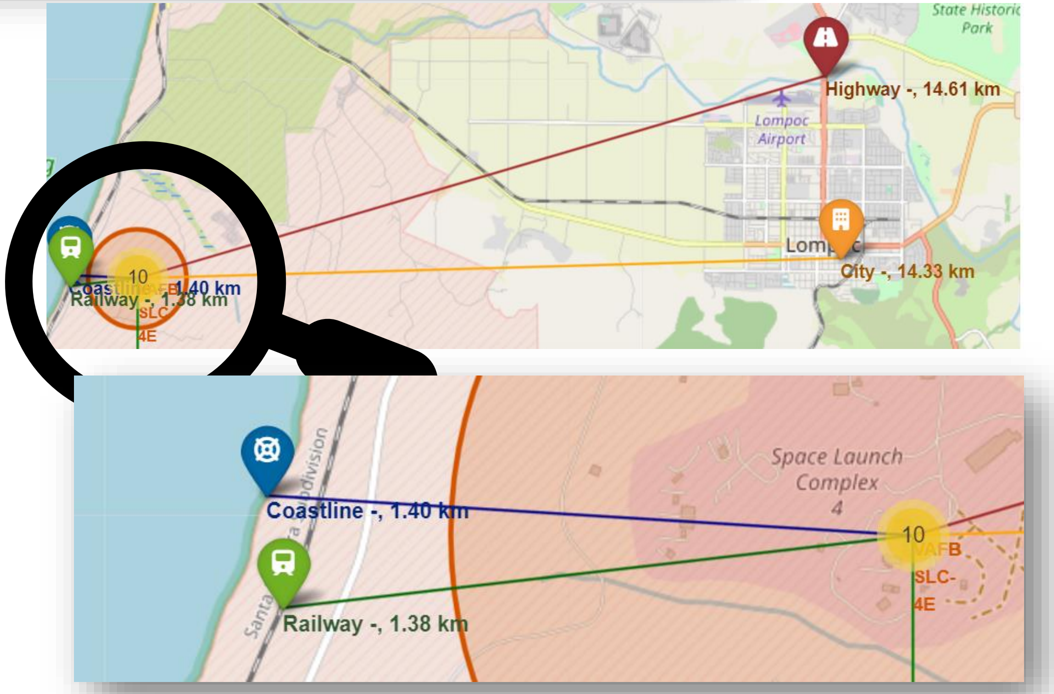
Unsuccessful Landing

https://github.com/Anushdebnath03

Distances between a launch site to its proximities

Distance from equator is greater than 3000 km for all the sites
Further Launch site data is shown in the figure





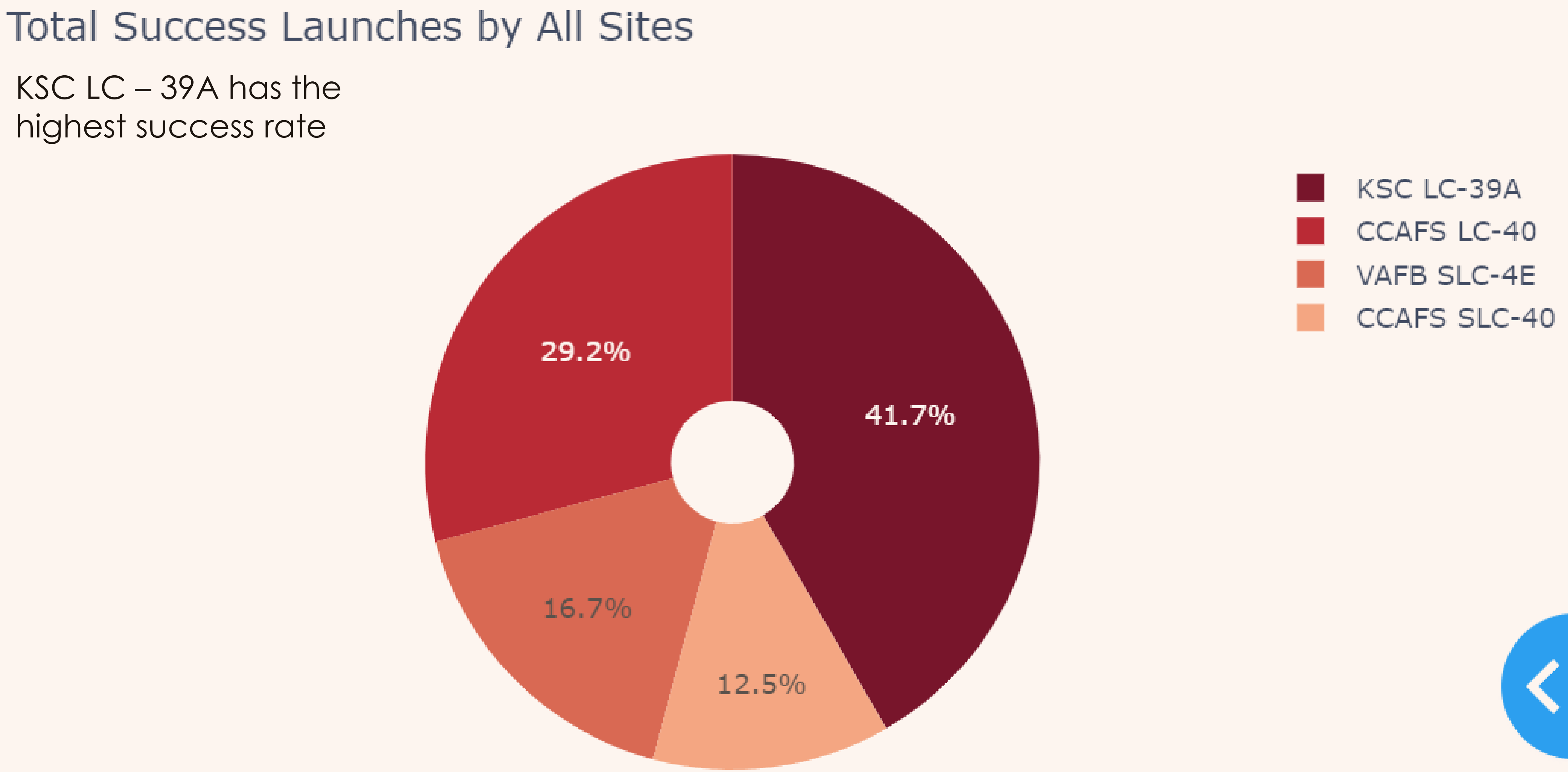
https://github.com/Anushdebnath03

SECTION 4

BUILDING A DASHBOARD WITH PLOTLY DASH

Total Success Launched by All Sites

KSC LC – 39A has the highest success rate



Correlation between Payload and Success of all sites

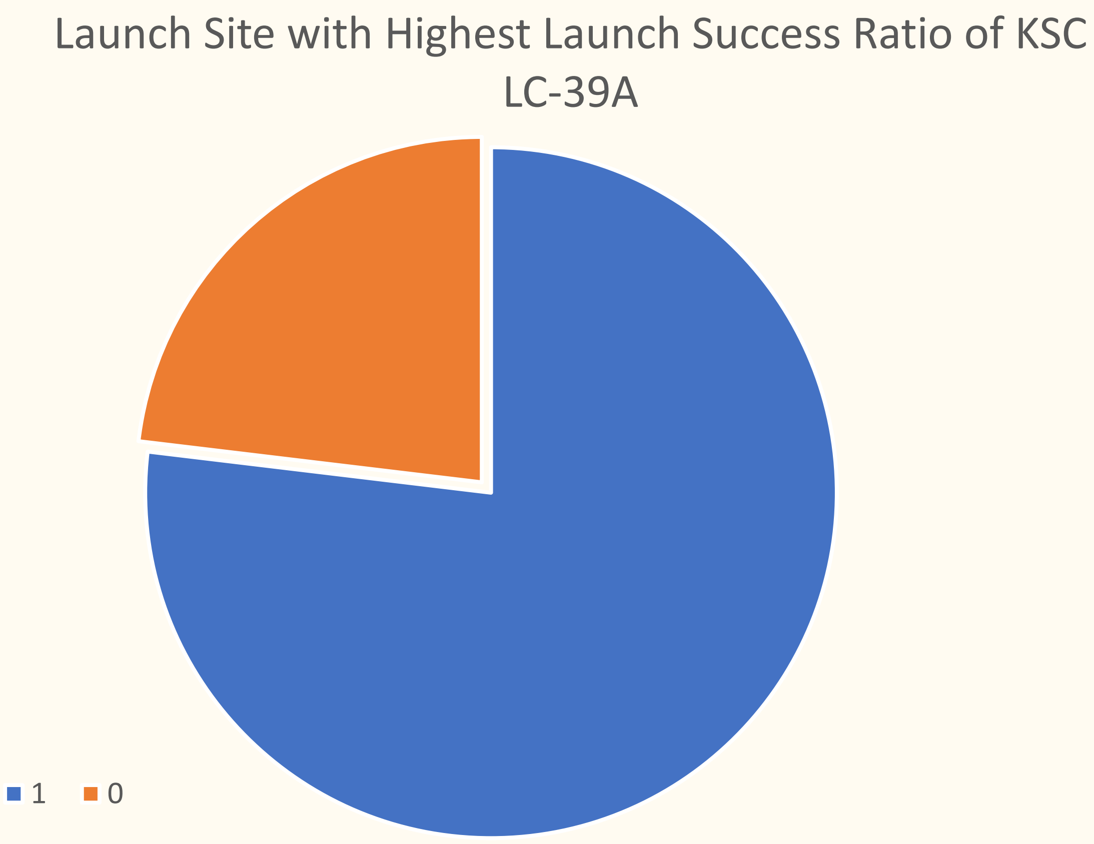
Launch Site with Highest Launch Success Ratio

Analysis

KSC LC-39A achieved 76.9% success rate and 23.1% failure rate

Highest success rate

payload range: 2000-10000 kg
F9 Booster: FT

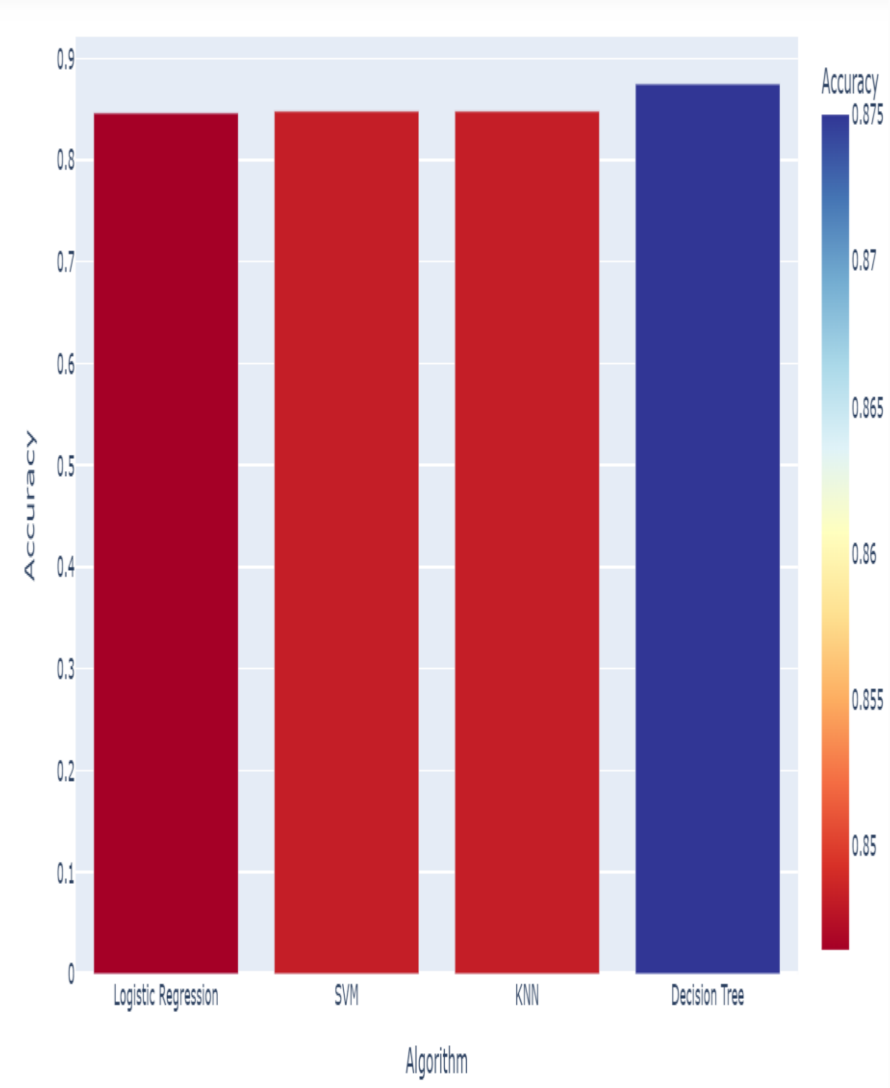


SECTION 5

PREDICTIVE ANALYSIS (CLASSIFICATION)

Classification Accuracy

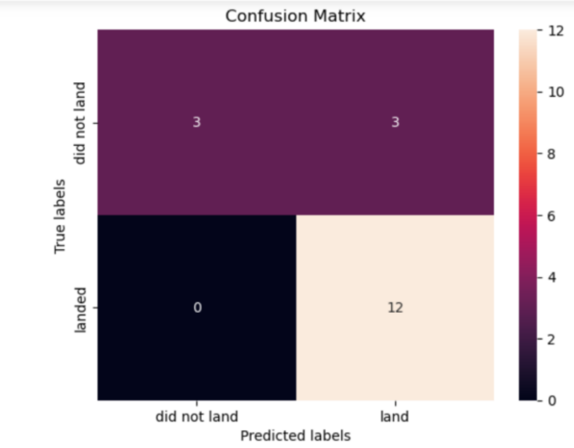
Models	Accuracy on Test Data	Best Parameters	Accuracy
Logistic Regression(LR)	0.8333	'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'	**0.8464**
Support Vector Machine (SVM)	0.8333	'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'	**0.8482**
Decision Tree Classifier(DT)	0.8333	'criterion': 'gini', 'max_depth': 14, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 5, 'splitter': 'random'	**0.875**
K Nearest Neighbors(KNN)	0.8333	'algorithm': 'auto', 'n_neighbors': 10, 'p': 1	**0.8482**



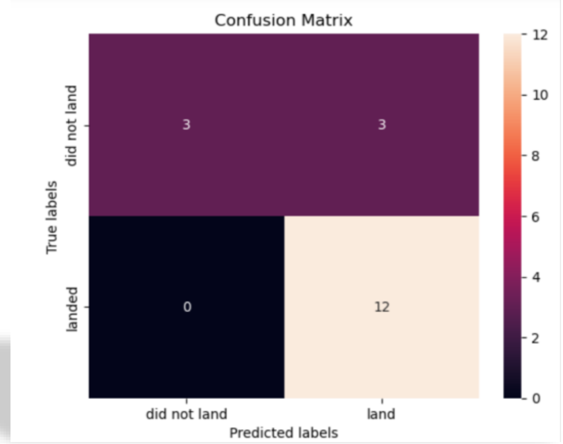
Confusion Matrix

	Predicte d No	Predicte d Yes	
Actual No	True Negative TN=3	False Positive FP=3	6
Actual Yes	False Negative TN=0	True Positive TP=12	12
	3	15	Total =18

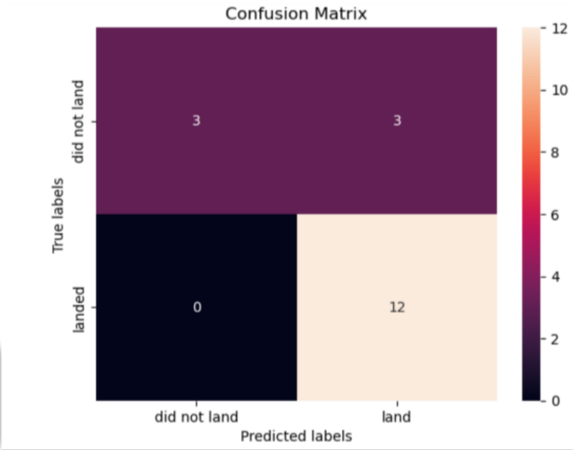
Accuracy: (TP+TN)/TOTAL = (12+3)/18 = 0.83333
Misclassification Rate: (FP+FN)/TOTAL = (3+0)/18 = 0.1667
True Positive Rate: TP/ACTUAL YES = 12/12 = 1
False Positive Rate: FP/ACTUAL NO = 3/6 = 0.5
True Negative Rate: TN/ACTUAL NO = 3/6 = 0.5
Precision: TP/PREDICTED NO = 12/15 = O.8
Prevalence: ACTUAL YES/TOTAL = 12/18 = 0.6667



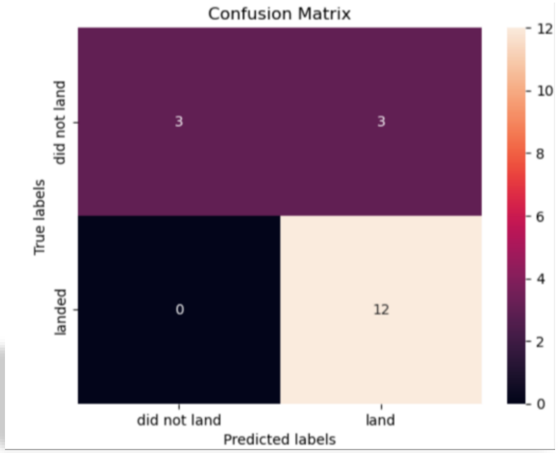
Logistic Regression(LR)



Support Vector Machine(SVM)



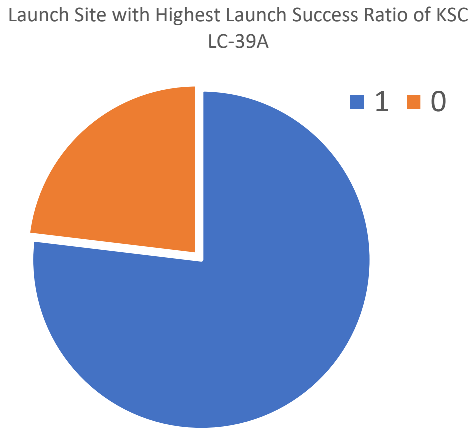
Decision Tree Classifier(DT)



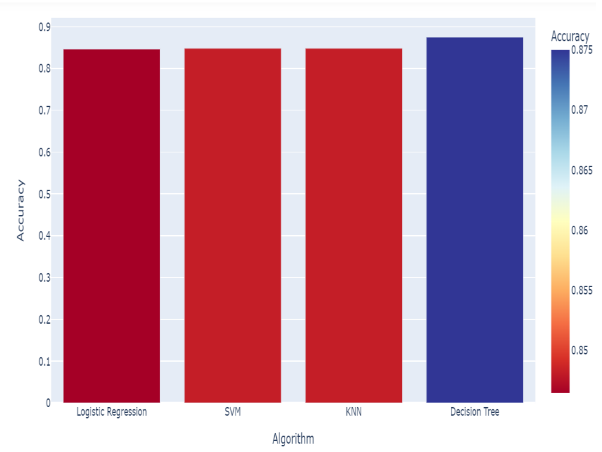
K Nearest Neighbors(KNN)

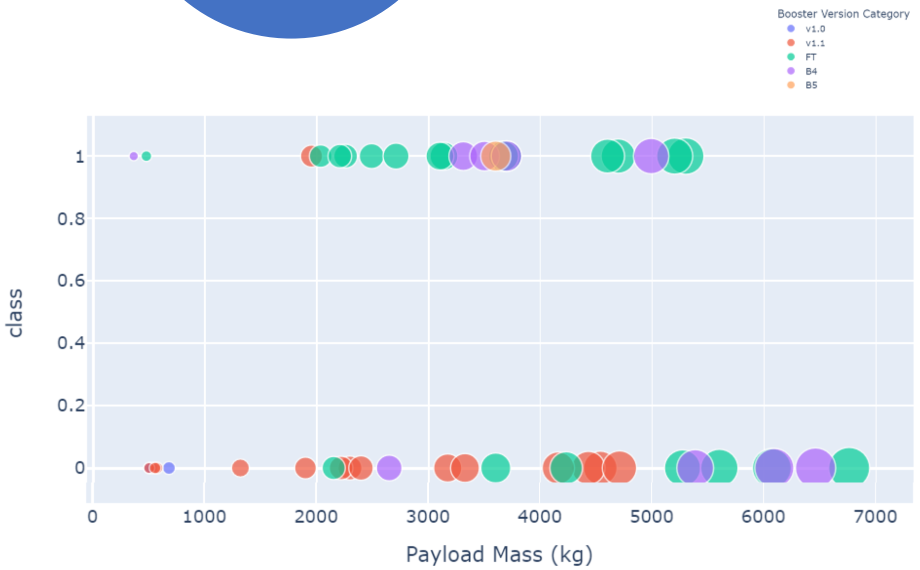
Conclusions

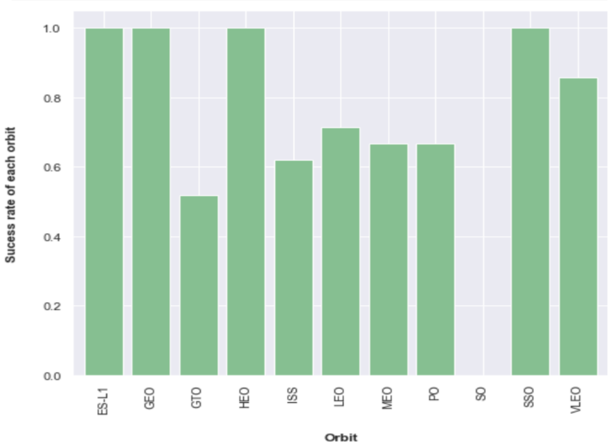
- For this dataset, Decision Tree Classifier Algorithm provides the best accuracy.
- KSC LC-39A achieved 76.9% success rate Highest success rate payload range: 2000-10000 kg, F9 Booster: FT but increasing payload mass seems to have negative impact on success.
- ES-L1, GEO, HEO, SSO Orbits have the highest Success rate.
- Success rates for SpaceX launches has been increasing relatively with time and it looks like soon they will reach the required target.



	Accuracy
Logistic Regression	0.846429
SVM	0.848214
KNN	0.848214
Decision Tree	0.875000







Appendix

Interactive Plotly

Folium Measure Control Plugin Tool

Folium Custom Title Layers with labels

IBM Cognos Visualization Tool

Basic Decision Tree Construction

THANK YOU