DATA SCIENCE CAPSTONE PROJECT: SPACE Y

Moaz Agha

12-14-2022

OUTLINE

EXECUTIVE SUMMARY

- Data collection via API, SQL, and Web Scraping
- Exploratory data analysis and interactive data visualization were produced
- Reliable Machine Learning models with 83% success rate were generated as predictive algorithms

INTRODUCTION

- The competitor SpaceX launches with a cost of $62 million
- Other providers cost upward of $165 million each
- Much of the savings is because SpaceX can reuse the first stage
- **Problem Statements:**
  - Determining if the first stage would land, thus, estimating the cost of each launch for Space Y
  - The effect of each relationship, of SpaceX rocket launches, on the outcome
  - What could we learn from our competitors?

METHODOLOGY

METHODOLOGY

- Data Collection
  - Using Rest API and web scraping
- Data wrangling
  - Data cleaning of nulls values and irrelevant columns
  - One hot encoding data fields for Machine Learning
- Exploratory data analysis using visualization and SQL
- Interactive visual analytics using Folium and Plotly Dash
- Predictive analysis using classification models

DATA COLLECTION

- SpaceX launch data was obtained through the SpaceX REST API
- The API URL - `spacex_url="https://api.spacexdata.com/v4/launches/past"`
- The URL gives us data about the launches including:
  - The rocket used, payload, launch specifications, landing specifications and outcomes
- Another popular method of collecting data is web scraping:
  - Wiki page Included some HTML tables containing info about Flacon 9 launch records
  - Python BeautifulSoup Package was used to acquire the tables

DATA COLLECTION – SPACEX API

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

response = requests.get(spacex_url)
```

Getting response from API

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

Converting Response to a .json file

```
# Call getBoosterVersion
getBoosterVersion(data)

# Call getLaunchSite
getLaunchSite(data)

# Call getPayloadData
getPayloadData(data)

# Call getCoreData
getCoreData(data)
```

Apply custom functions to clean data

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

Assign list to dictionary then create dataframe

```
data_falcon9 = df.drop(df[df['BoosterVersion'] == "Falcon 1"].index, axis = 0, inplace = True)
data_falcon9 = df.reset_index(drop = True, inplace = True)

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

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

Filter dataframe and export to a .csv file

[Link: Data Collection via API Notebook on GitHub]

DATA COLLECTION – WEB SCRAPING

1. Getting response from HTML

```
html_data = requests.get(static_url).text
```

2. Creating BeautifulSoup Object

```
soup = BeautifulSoup(html_data, 'html5lib')
```

3. Finding tables

```
# Use the find_all function in the BeautifulSoup object
# Assign the result to a list called `html_tables
html_tables = soup.find_all('table')
```

4. Calling columns names

```
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
```

5. Creating dictionary

```
launch_dict= dict.fromkeys(column_names)
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
⋮
```

6. Appending data to keys

7. Converting dictionary to dataframe

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

8. Exporting dataframe to .csv file

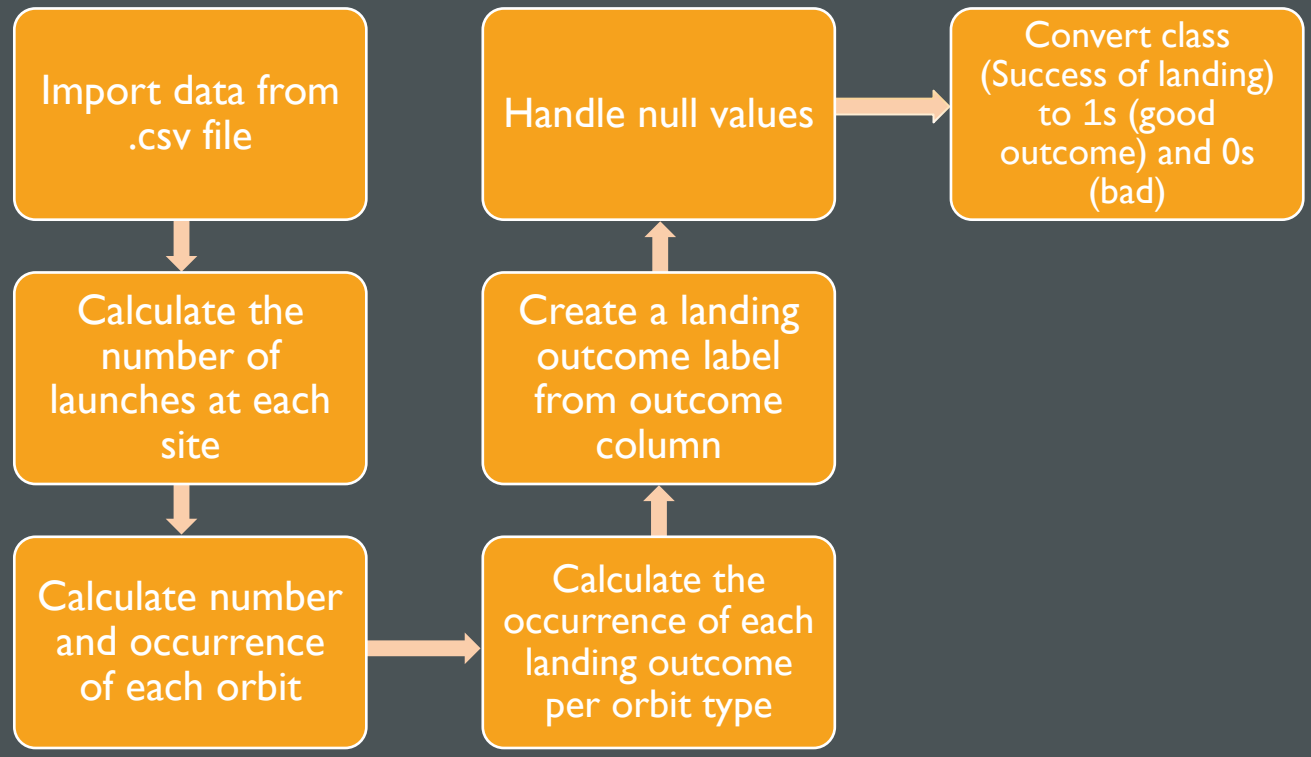
[Link: Data Collection with Web Scraping Notebook on GitHub]

DATA WRANGLING

- Performed exploratory data analysis to find patterns and determine the label for training supervised models
  - There were 8 types of outcomes.
    3 were labelled successful (1s) , 5 were labelled as bad outcome (0s)

```
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
```



Link: Data Wrangling Notebook on GitHub

EDA WITH DATA VISUALIZATION

Categorical Plot — 2 Plots

- 1. Launch outcome based on payload mass vs. flight number
- 2. Launch outcome based on launch site vs. flight number

Scatter Plot — 3 Plots

- 1. Launch outcome based on launch site vs. payload mass
- 2. Launch outcome based on payload mass vs. orbit type
- 3. Launch outcome based on flight number vs. orbit type

Bar Chart — 1 Chart

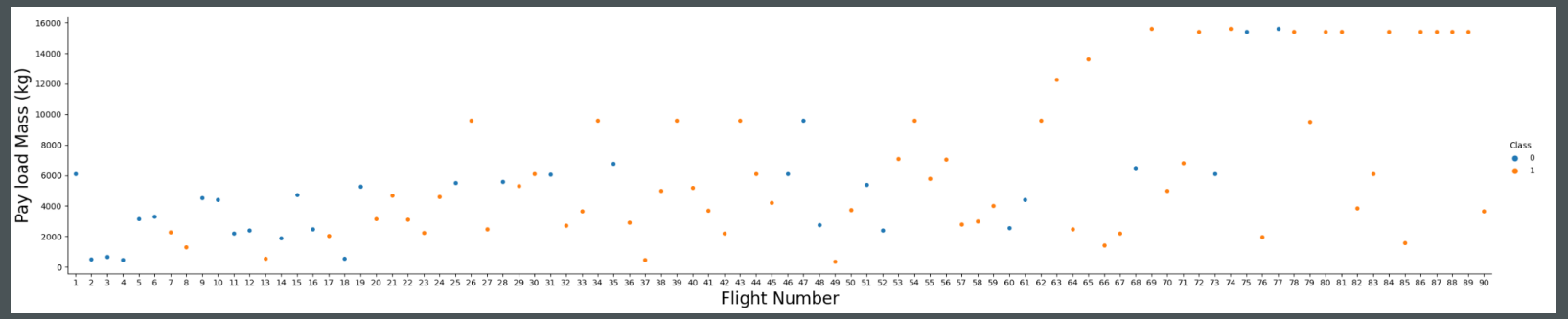
- Relationship between launch success rate of each orbit

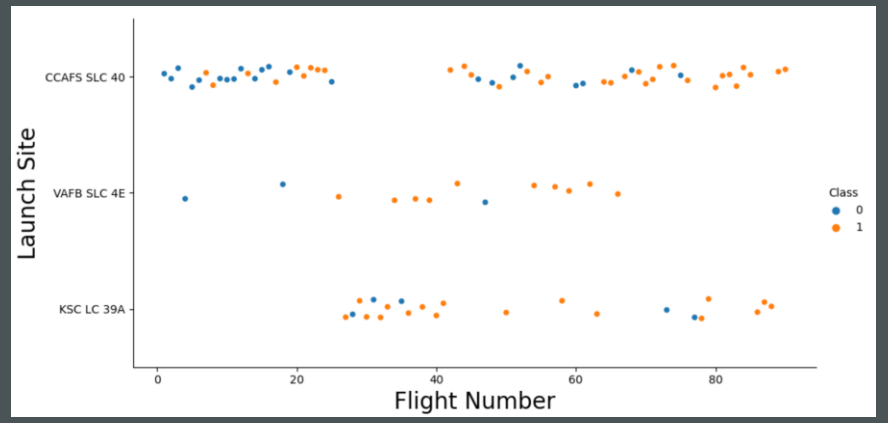
Line Plot — 1 Plot

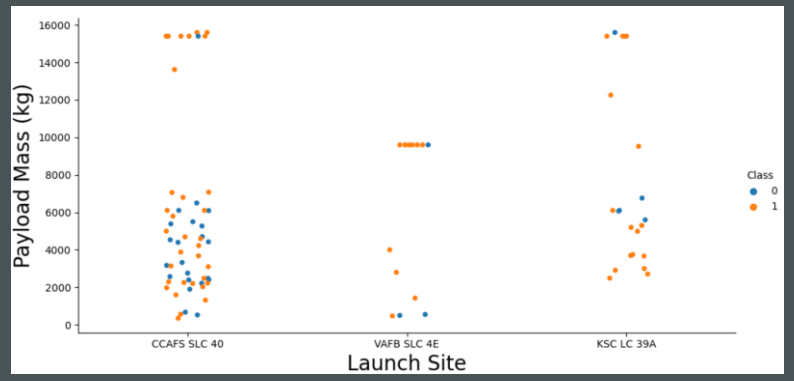
- Launch success yearly trend

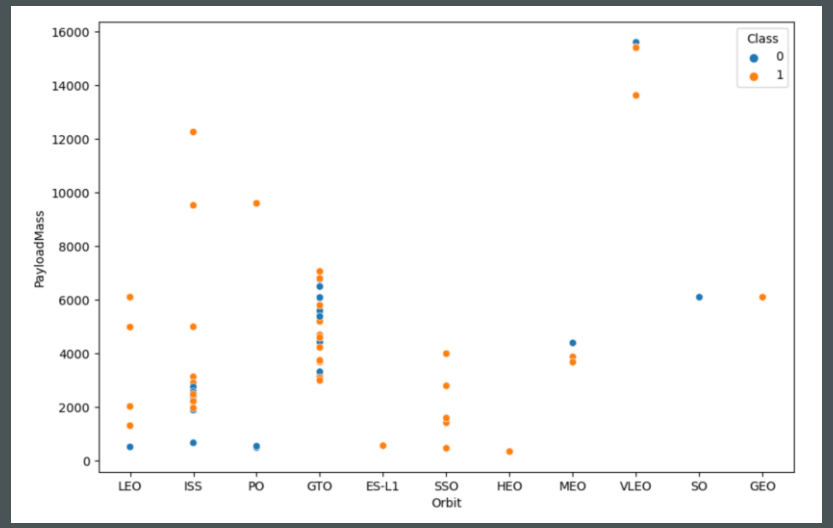
Link: EDA with Data Visualization Notebook on GitHub

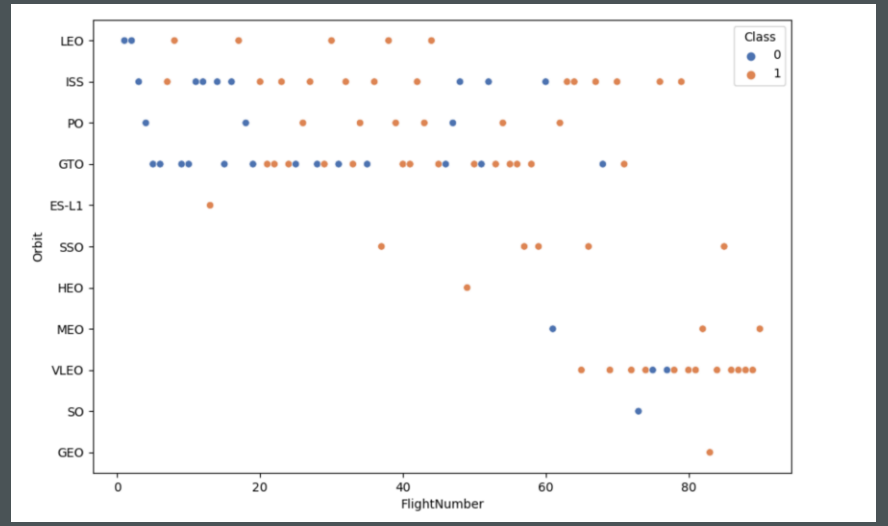
EDA WITH DATA VISUALIZATION: CATEGORICAL & SCATTER PLOT





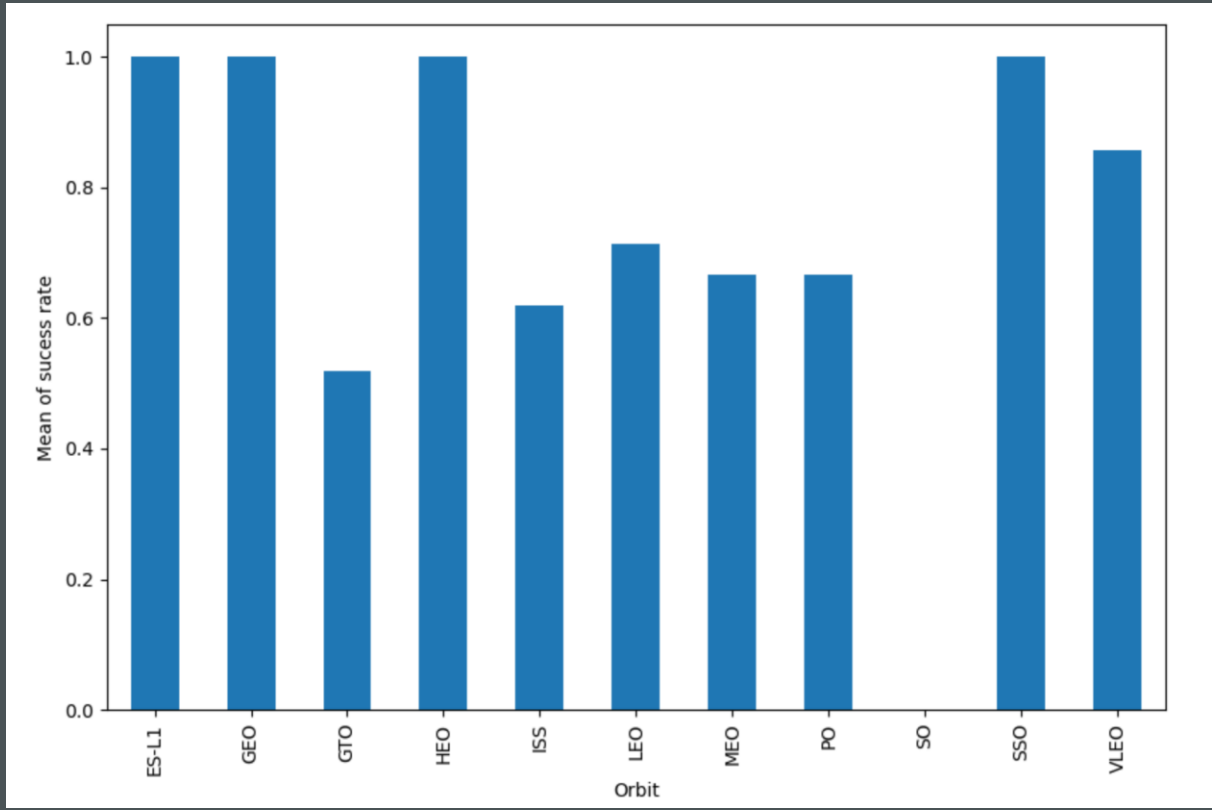


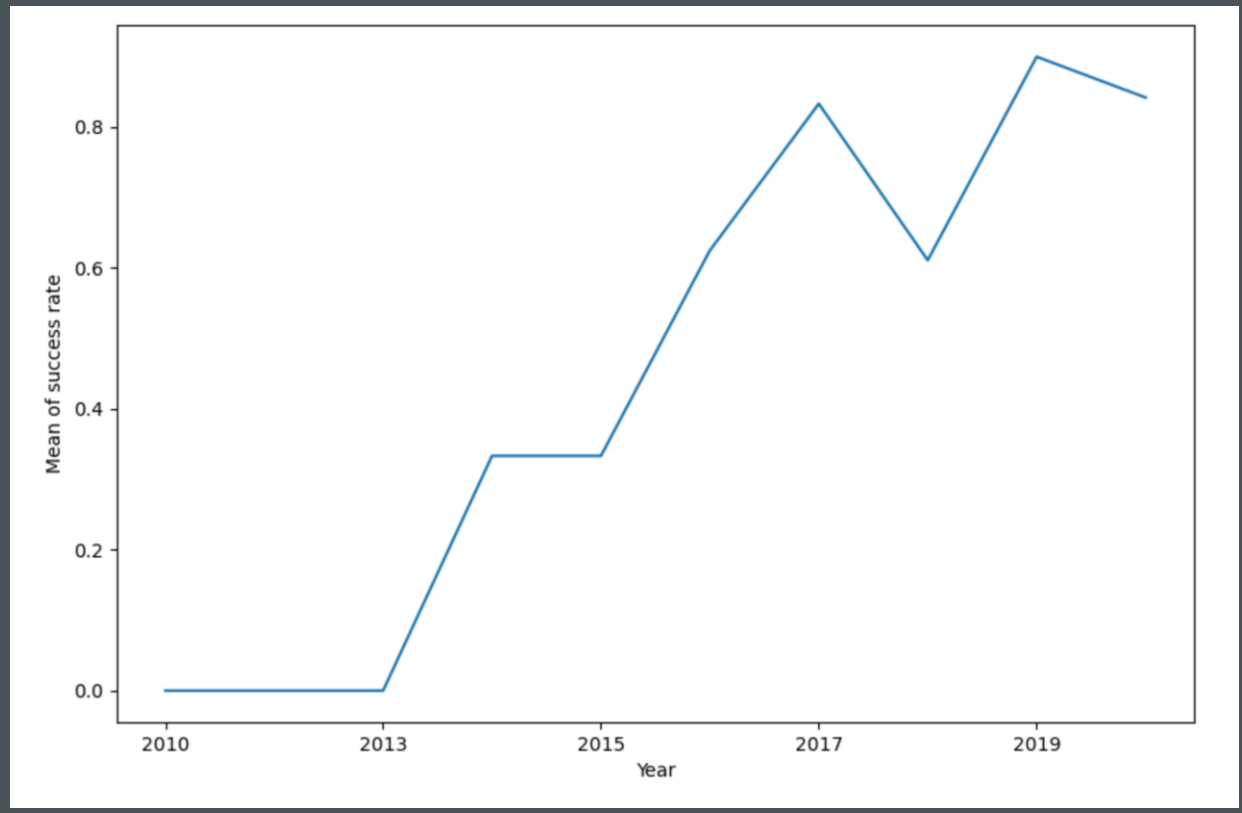




[Link: EDA with Data Visualization Notebook on GitHub]

EDA WITH DATA VISUALIZATION: BAR CHART & LINE PLOT





Link: EDA with Data Visualization Notebook on GitHub

EDA WITH SQL

- IBM's Db2 was used for cloud, fully managing SQL database provided as a service

```
!pip install sqlalchemy==1.3.9
!pip install ibm_db_sa
!pip install python-sql
%load_ext sql

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

- The following SQL queries were used to gather info from the given database:
  - Displaying the names of the unique launch sites in the space mission
  - Displaying multiple records where launch sites begin with the string 'CCA'
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying 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 (Using a subquery)
  - Listing the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
  - Ranking the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order

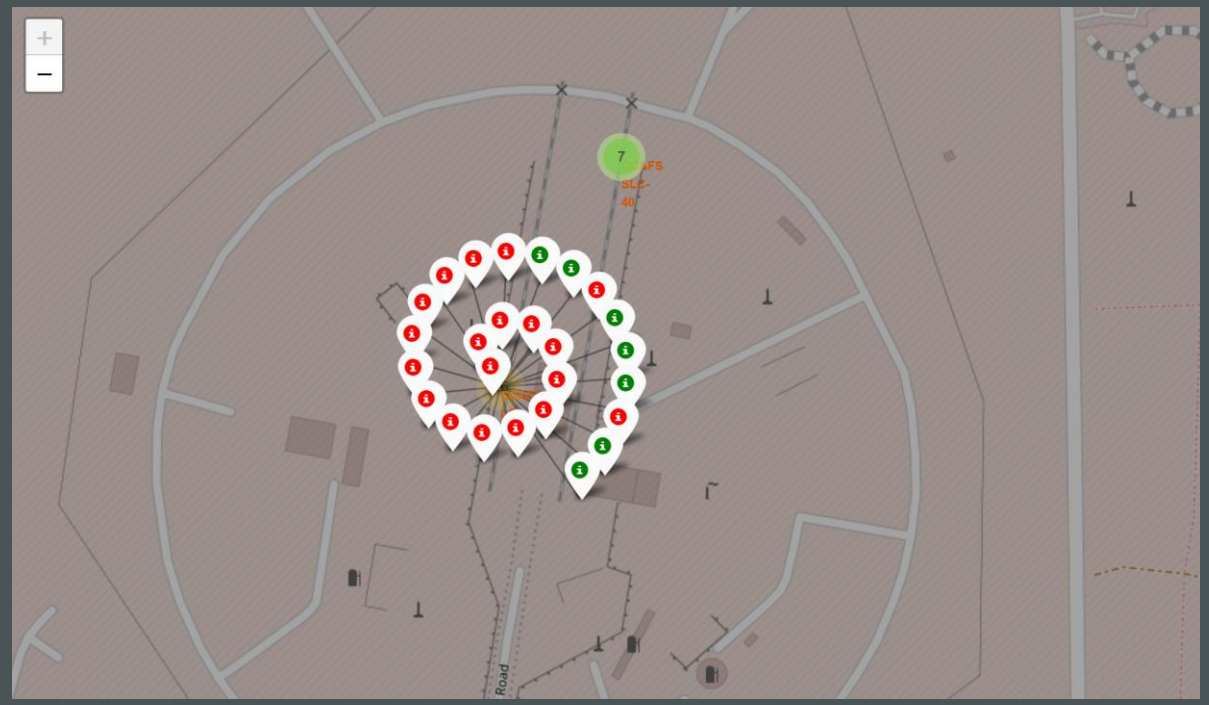
Link: EDA with SQL Notebook on GitHub

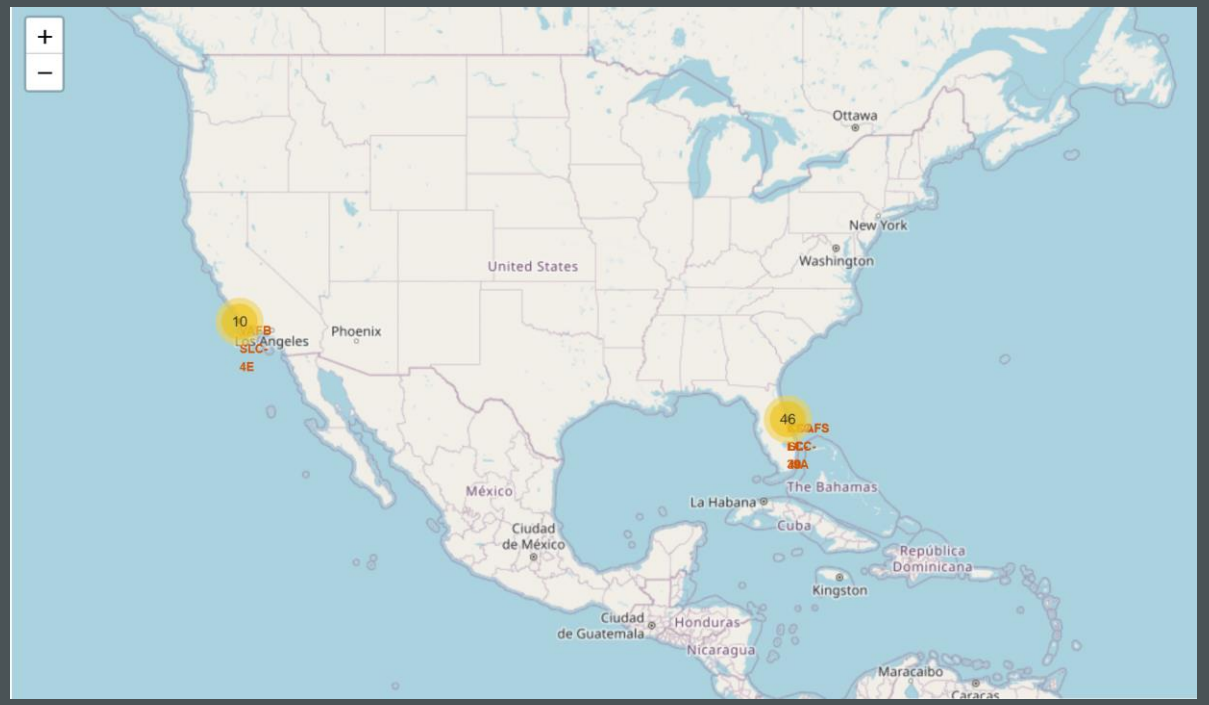
BUILDING AN INTERACTIVE MAP WITH FOLIUM

- Marked all past launch sites on the map, using *folium.circle* and *folium.marker*
  - Using the latitude and longitude coordinates of each site
  - Most of the launch sites were within a close distance of 1.5km from coastline
- Marked the successful and failed launches for each site on the map
  - Easy to visualize the outcome of launches for each site with green and red markers
- Draw a *Polyline* between SLC-40 launch site and its proximities
- Added Folium *MousePosition* to get map coordinates for a mouse over a point on the map

Link: Interactive Visual Analytics with Folium Notebook on GitHub

BUILDING AN INTERACTIVE MAP WITH FOLIUM





[Link: Interactive Visual Analytics with Folium Notebook on GitHub]

BUILD A DASHBOARD WITH PLOTLY DASH

- The dashboard includes:
  - A pie chart showing the total success, for all sites or by certain launch site
  - A slider to allow for payload ranges to be highlighted
  - A scatter plot showing the correlation between payload and success, for all sites or by certain launch site

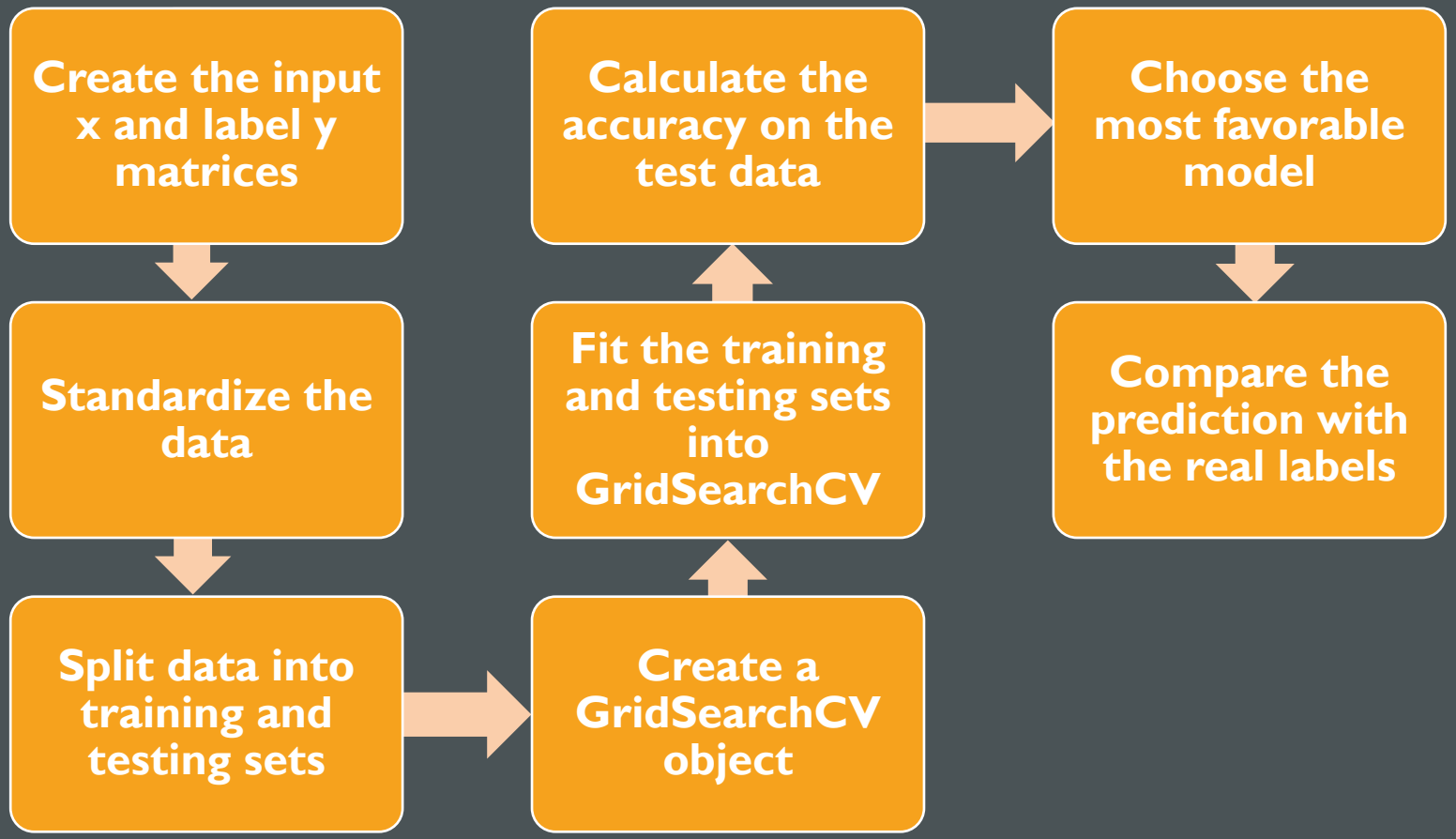
Link: Plotly Dash Python Code on GitHub

PREDICTIVE ANALYSIS (CLASSIFICATION)

- With the help of Scikit-learn, Primary Machine Learning library, predictive analysis was done through the following steps:
  - Loaded data using NumPy and Pandas, transformed the data, and split them into training and testing sets
  - Created a ML pipeline to predict if the 1st stage would land
  - Used GridSearchCV to find the best ML method for the predictions
  - Improved the model using feature engineering and algorithm tuning
  - Found the models with the highest prediction accuracy

Link: Machine Learning Prediction Notebook on GitHub

PREDICTIVE ANALYSIS (CLASSIFICATION)

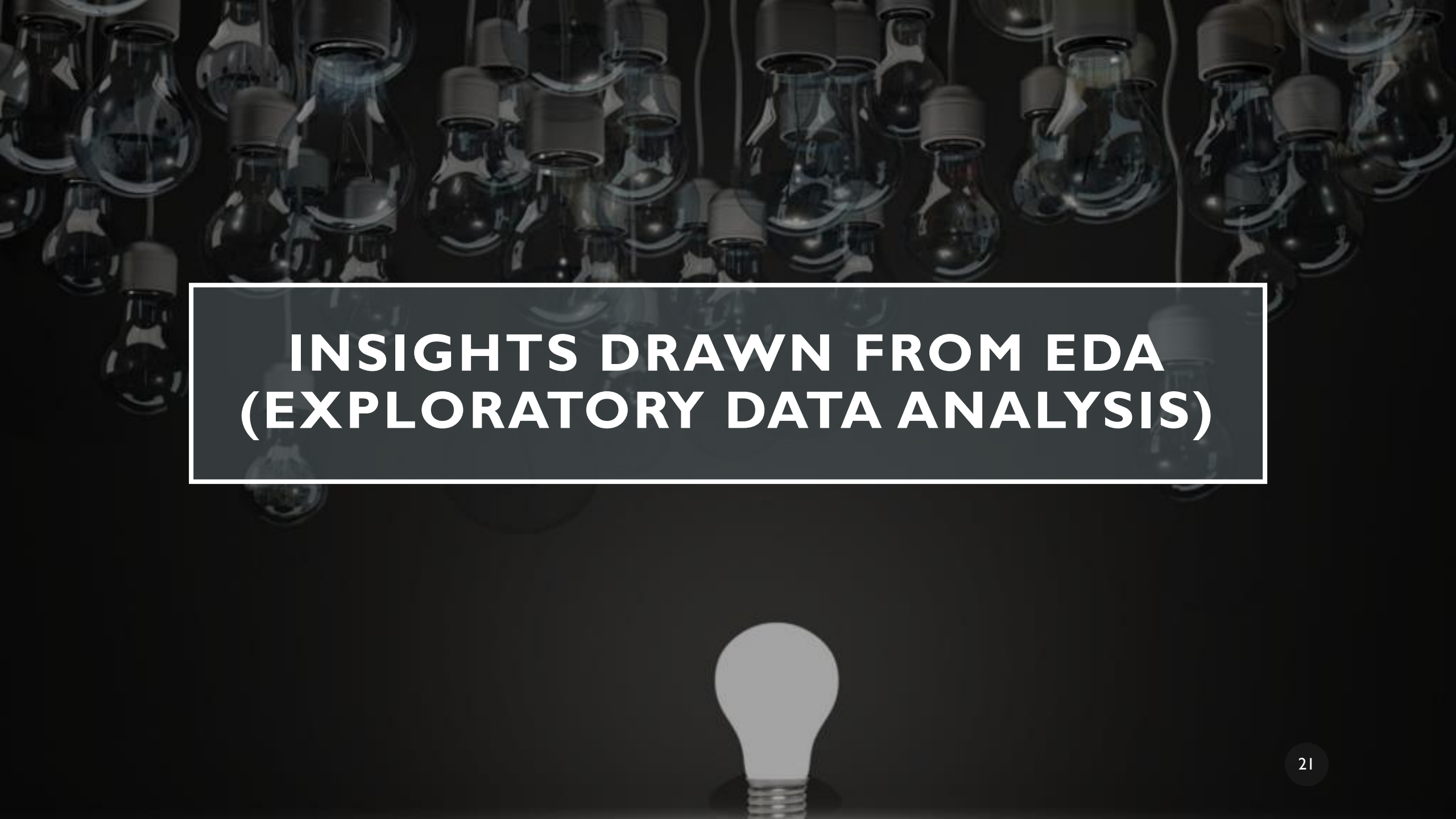


Link: Machine Learning Prediction Notebook on GitHub

RESULTS

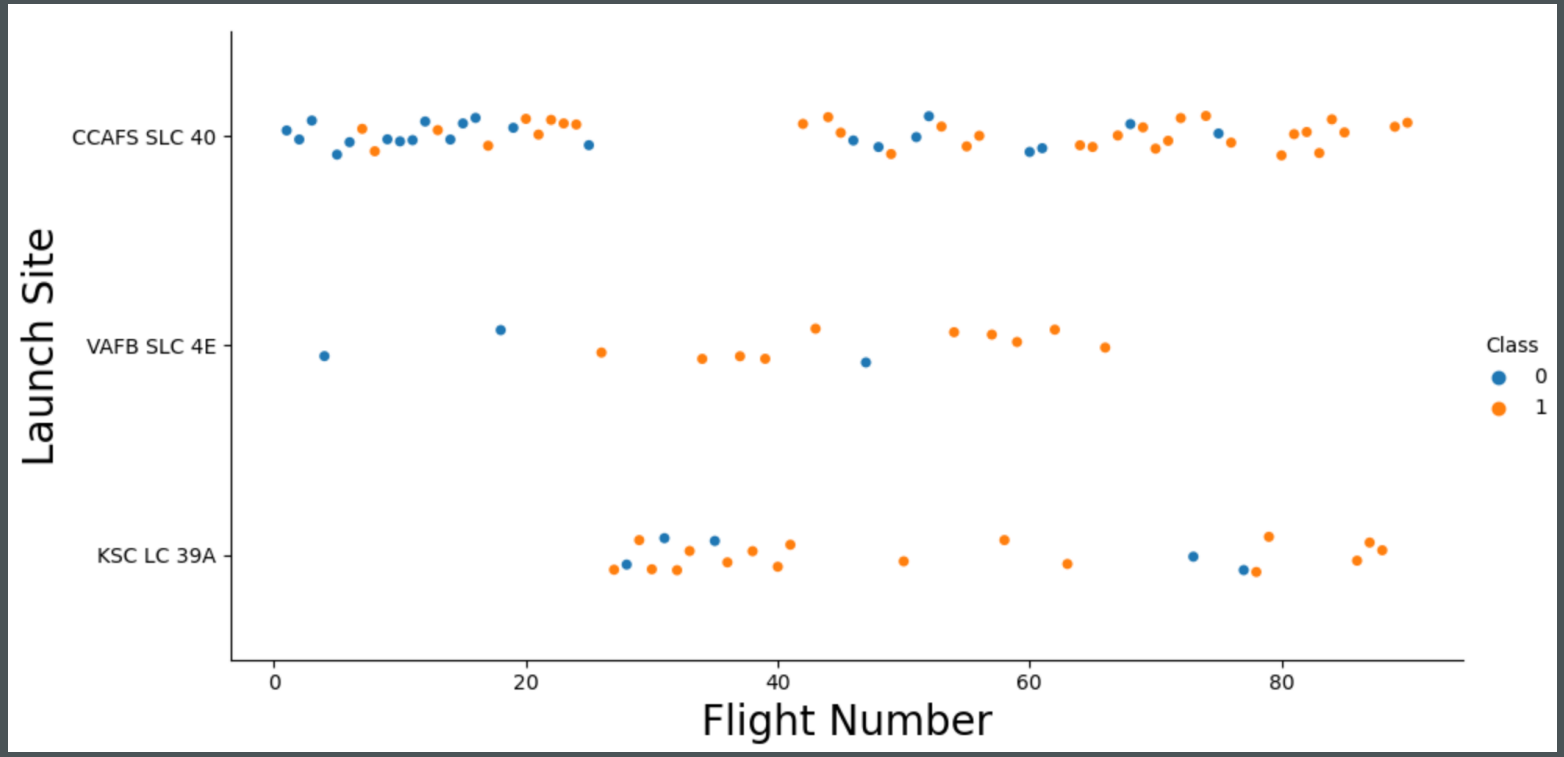
- The SVM, KNN, Logistic Regression, and decision tree models all performed with an accuracy score of 83.3%
- A clear positive progress in the landing outcomes since the year 2015
- Low weighted payloads executed better than the heavier ones
- The EDA shows that successful landing is correlated with flight number
- All the launch tests are located close to the coastline as well as highways and railways
- KSC LC 39A had the most successful launches out of all the sites
- Orbit GEO, HEO, SSO, and ES L1 have the best success rate

INSIGHTS DRAWN FROM EDA (EXPLORATORY DATA ANALYSIS)



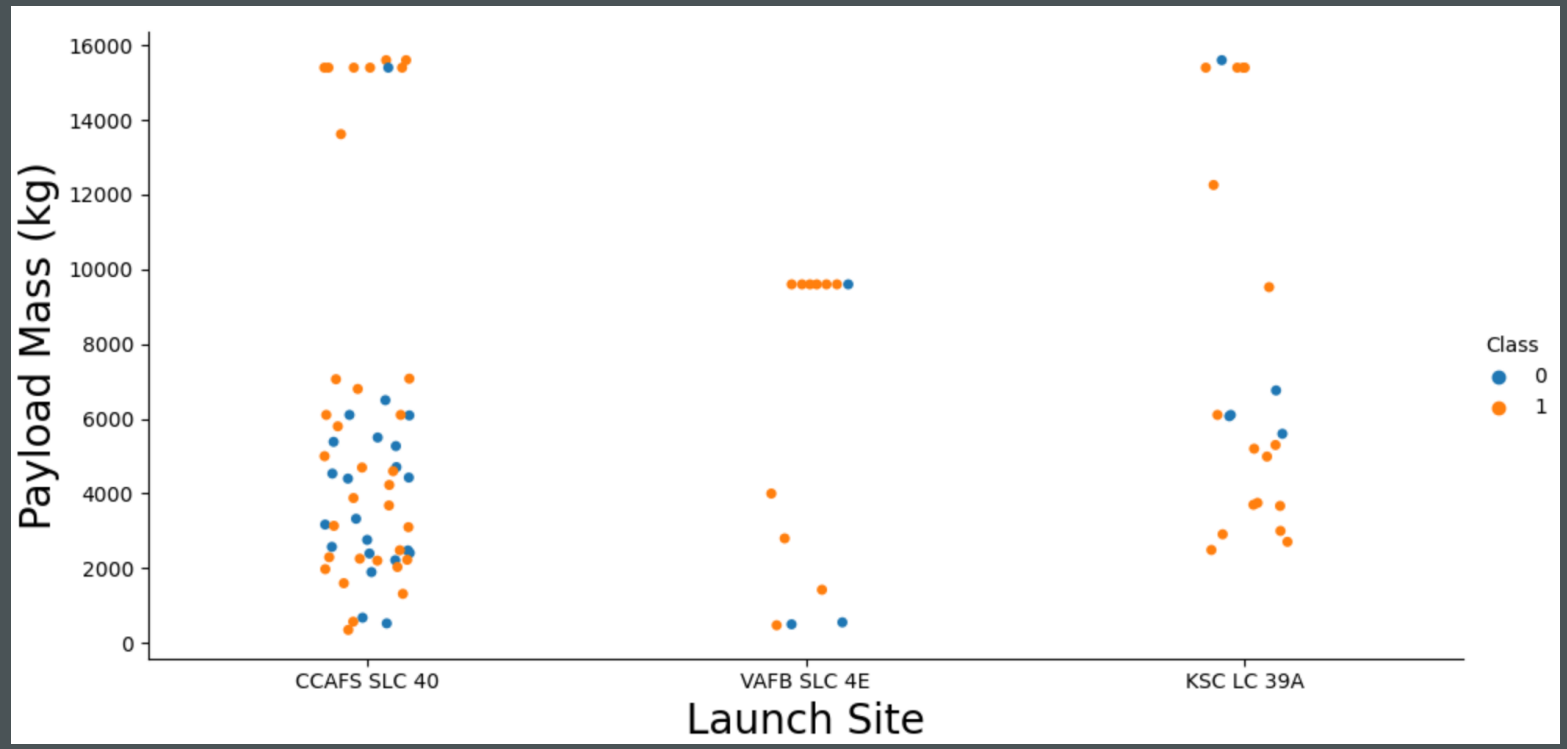
FLIGHT NUMBER VS. LAUNCH SITE

- **CCAFS SLC 40** has the greatest number of launches. While VAFB SLC 4E had the least number of launches
- With higher flight number, around 30 and greater, at each launch site the success rate increases



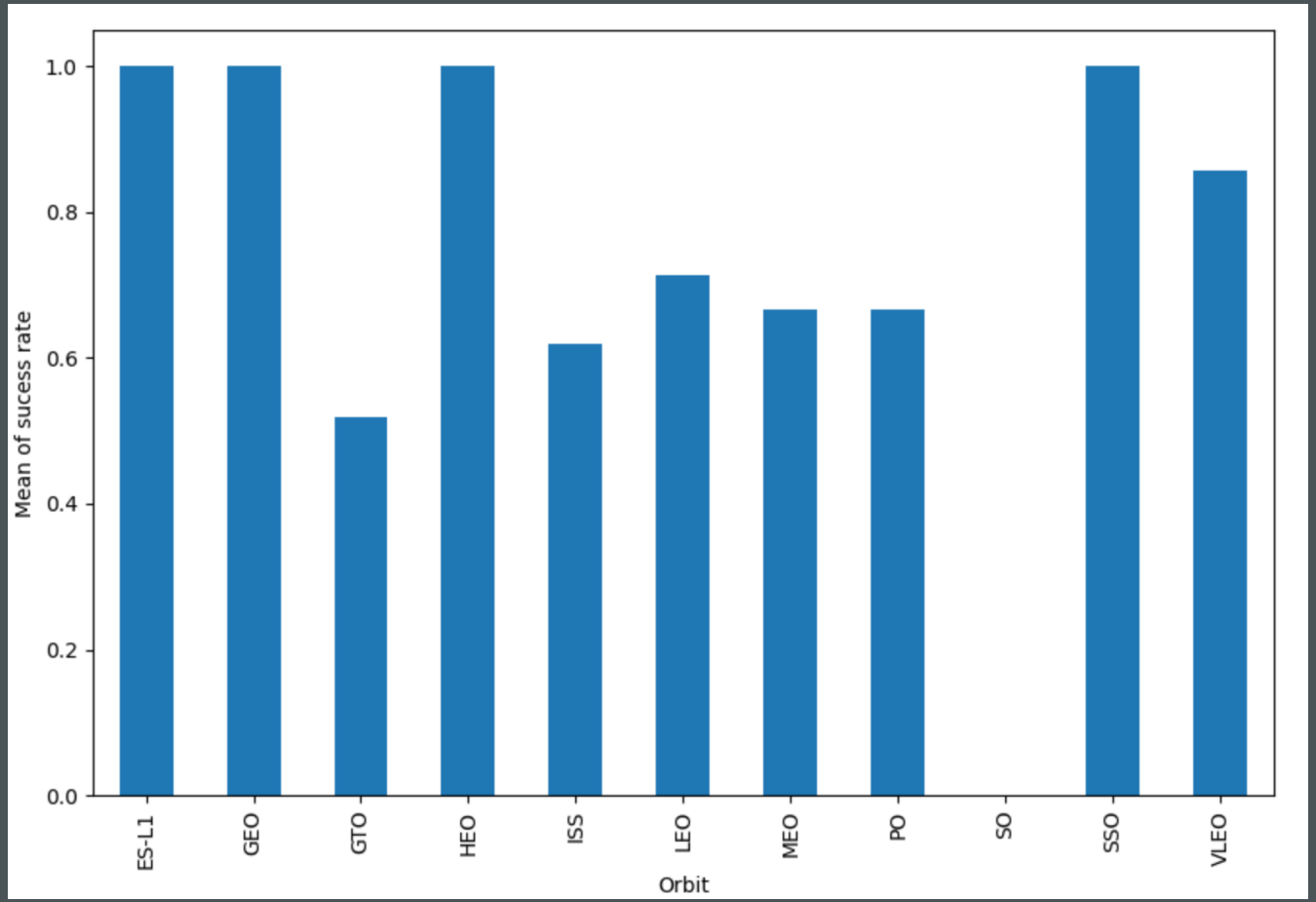
PAYLOAD MASS VS. LAUNCH SITE

- Most of the lower payloads mass launches have been from **CCAFS SLC 40**
- **VAFB SLC 4E** site has no rocket launches for heavy payload mass (greater than 10000 kg)
- Except for **VAFB SLC 4E** site, the greater the payload mass the higher the success rate for the rocket. Yet, there is no clear pattern.



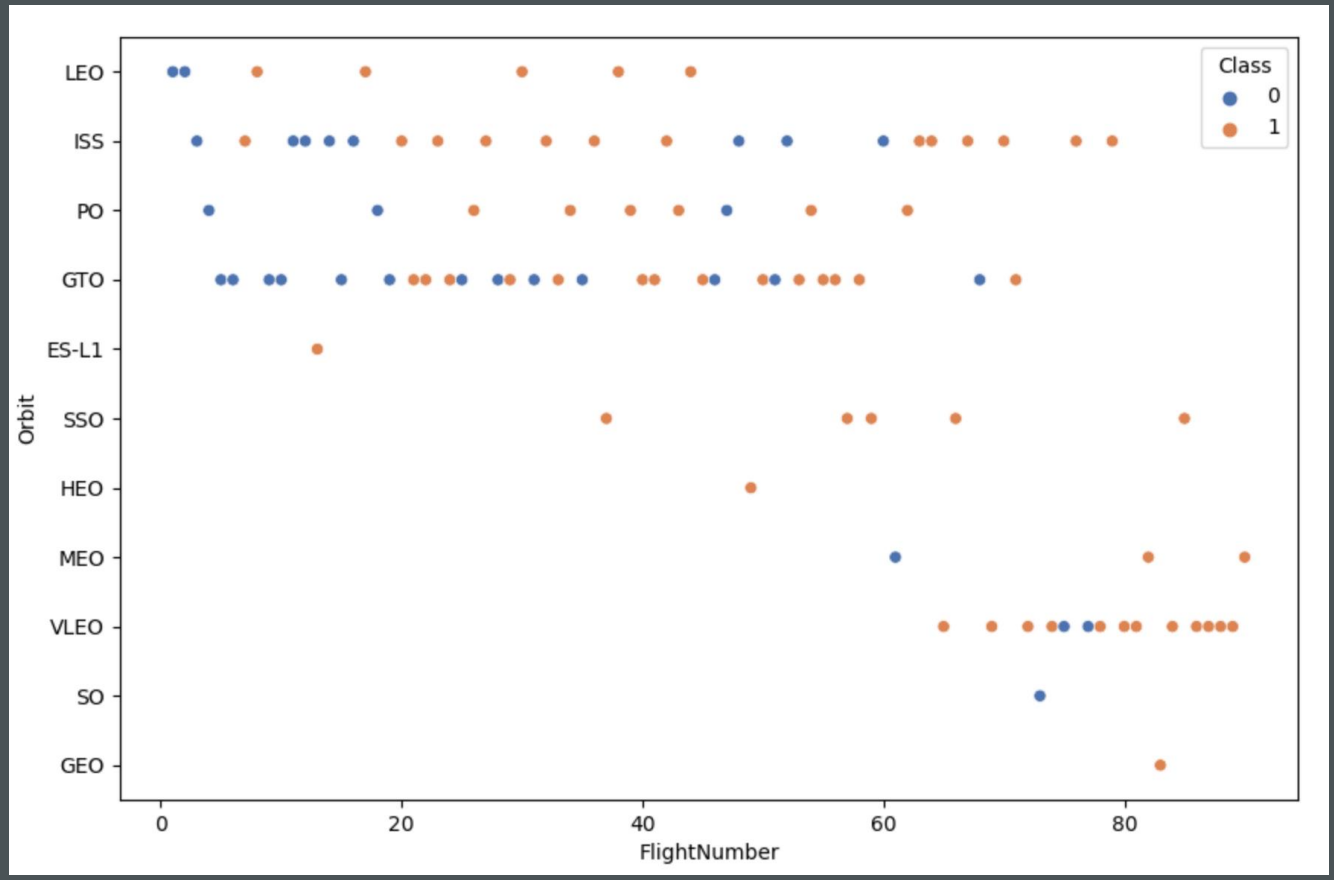
SUCCESS RATE VS. ORBIT TYPE

- Orbit **GEO**, **HEO**, **SSO**, and **ES L1** have the best success rate



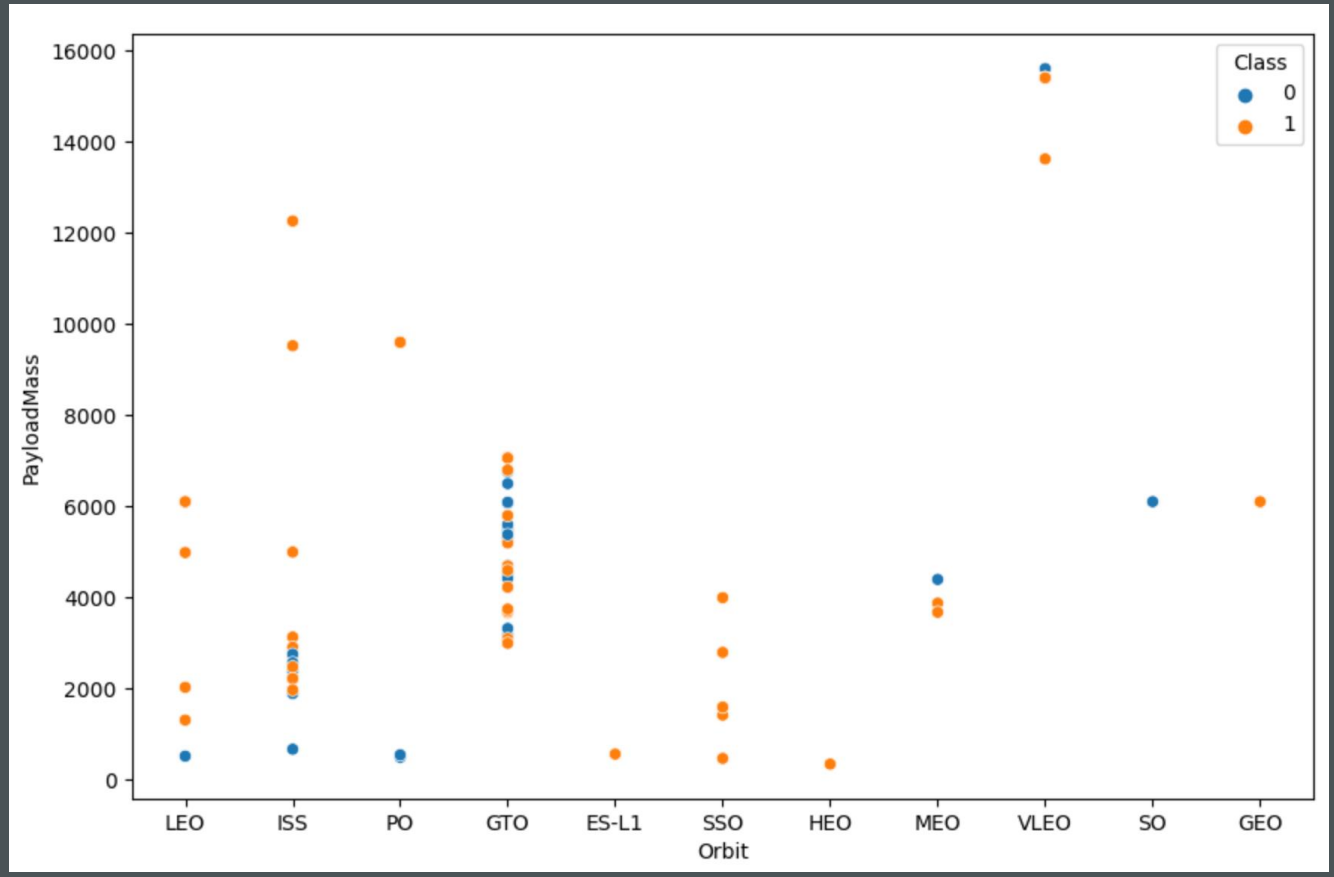
FLIGHT NUMBER VS. ORBIT TYPE

- For orbit **LEO** and **VLEO**, the success rate increases with the number of flights
- There seems to be no correlation between the 2 variables for orbit **GTO** and **ISS**
- Orbit **SSO** seems to be very successful, yet the sample size is small



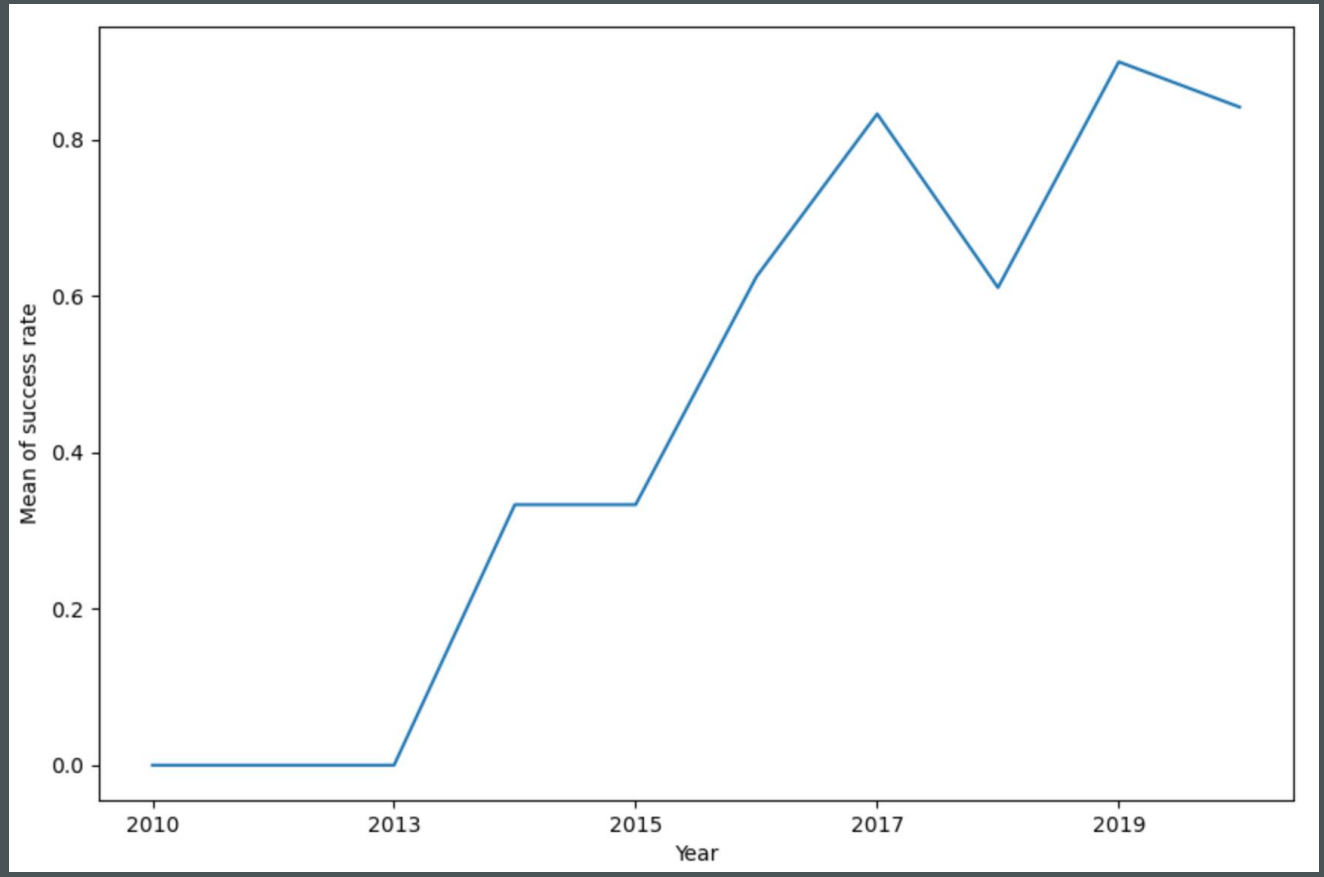
PAYLOAD MASS VS. ORBIT TYPE

- For orbit **LEO**, **PO**, and **ISS,** the success rate increases as the payload mass gets heavier
- There seems to be no correlation between the 2 variables for orbit **GTO**
- Orbit **MEO** and **VLEO** were affected negatively by the heave payload masses



LAUNCH SUCCESS YEARLY TREND

- Launch success rate has increased significantly since **2013**, though there is a slight dip after year **2019**




DECLARE @invoiceDateTemp Datetime
SET @invoiceDateTemp =
INSERT #CTemp
T.couponUpgrade
FROM
OrderDetail AS T INNER JOIN
WHERE
(T.companyCode = @companyCode

EDA WITH SQL

ALL LAUNCH SITE NAMES

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

```
%sql SELECT Distinct LAUNCH_SITE FROM SPACEX
```

- Keyword **DISTINCT** show only unique launches sites from the dataset

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

LAUNCH SITE NAMES BEGIN WITH 'CCA'

- Condition keyword **LIKE** with the percentage sign in 'CCA%', indicating the name must start with CCA
- Keyword **LIMIT 5** fetch 5 records from the table

Display 5 records where launch sites begin with the string '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

ALL LAUNCH SITE NAMES

- Function **SUM** calculates total in the column it is used in
- **WHERE** clause filters the data to fetch customers by name 'NASA (CRS)'

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

```
%%sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass FROM SPACEX
WHERE CUSTOMER LIKE 'NASA (CRS)'
```

total_payload_mass
45596

AVERAGE PAYLOAD MASS BY F9 V1.1

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

```
%%sql SELECT AVG(PAYLOAD_MASS__KG_) as average_payload_mass FROM SPACEX
WHERE BOOSTER_VERSION LIKE 'F9 v1.1%'
```

- Function **AVG** calculates the average of the column it is used in

average_payload_mass
2534

FIRST SUCCESSFUL GROUND LANDING DATE

List the date when the first successful landing outcome in ground pad was acheived.

```
%%sql SELECT min(DATE) FROM SPACEX
WHERE LANDING__OUTCOME = 'Success (ground pad)'
```

- Function **MIN** works out the minimum value in the column it is used in

1
2015-12-22

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 & 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%%sql SELECT BOOSTER_VERSION FROM SPACEX
WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING__OUTCOME='Success (drone ship)'
```

- **AND** logical operator specifies additional filter conditions

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

TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES

- Function **COUNT** returns the number of values in the column it is used in
- Condition keyword **LIKE** was used twice, one time for each condition of success and failure

List the total number of successful and failure mission outcomes

```
%%sql SELECT COUNT(MISSION_OUTCOME) as total_number_of_successful_and_failure_missions FROM SPACEX
WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
```

total_number_of_successful_and_failure_missions
101

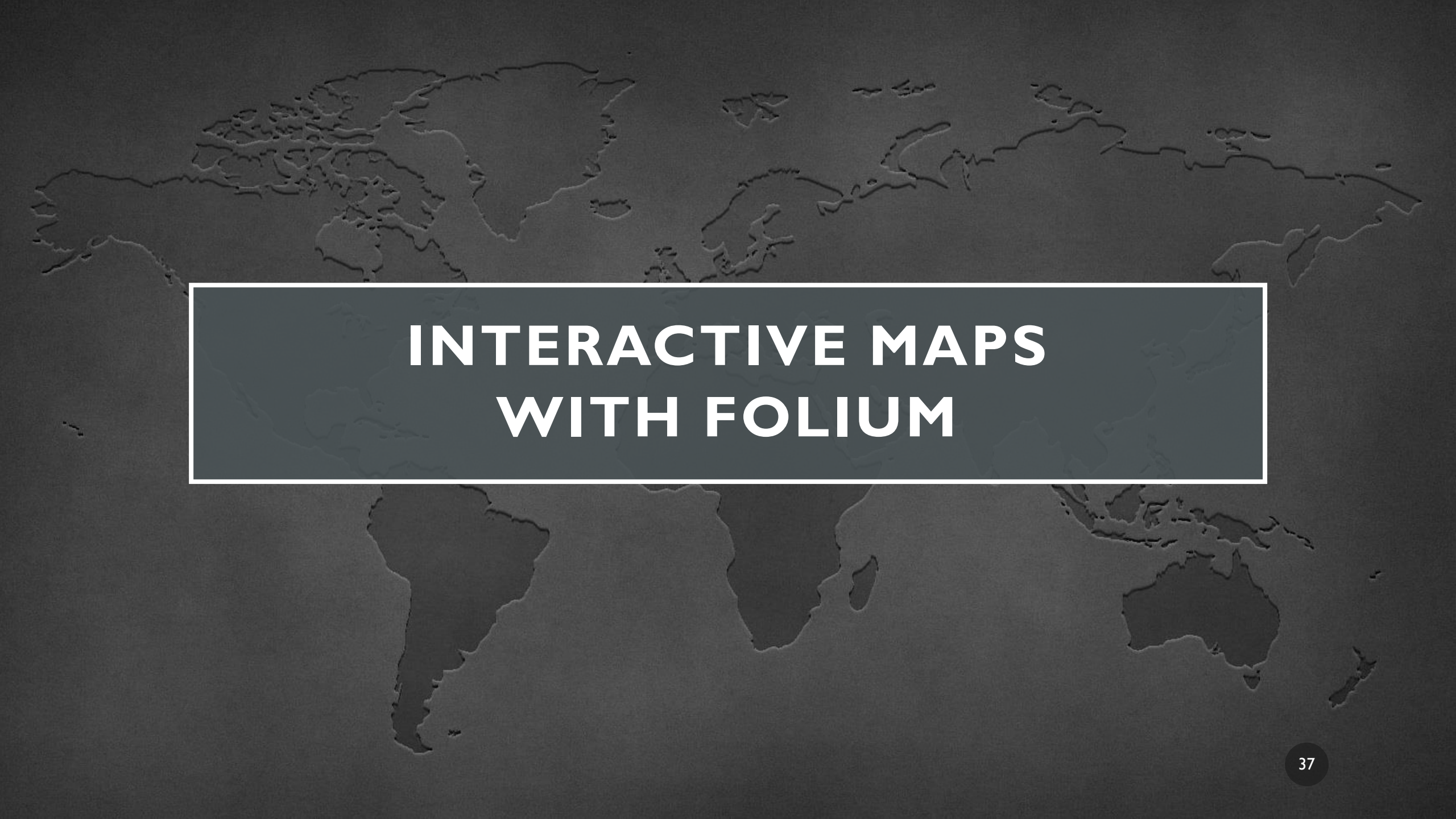
BOOSTERS CARRIED MAXIMUM PAYLOAD

- A query that appears inside another query is called a **Subquery**, which was used here

List the names of the booster_versions which have carried the maximum payload mass using a subquery

```
%%sql SELECT BOOSTER_VERSION FROM SPACEX
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX)
```

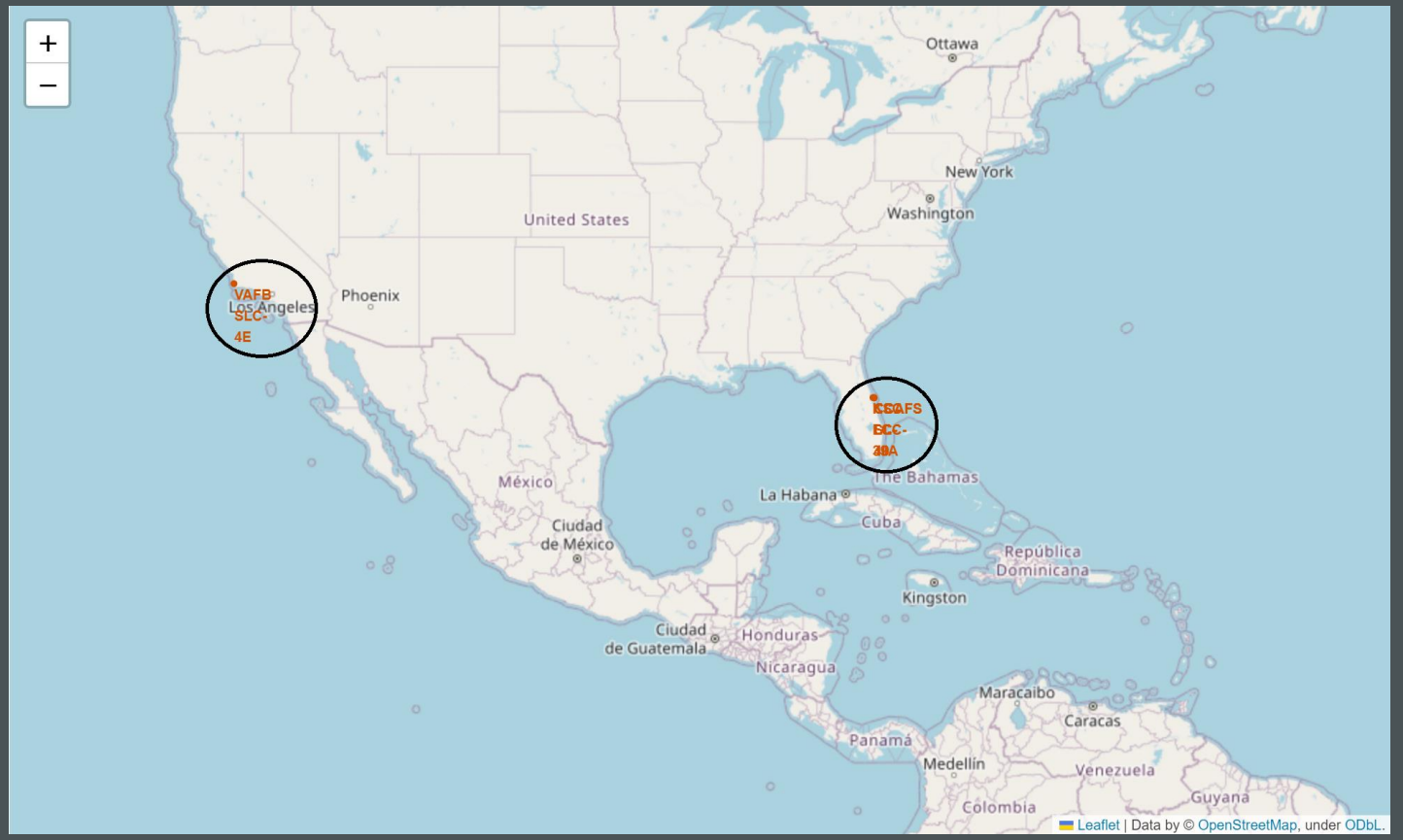
booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7



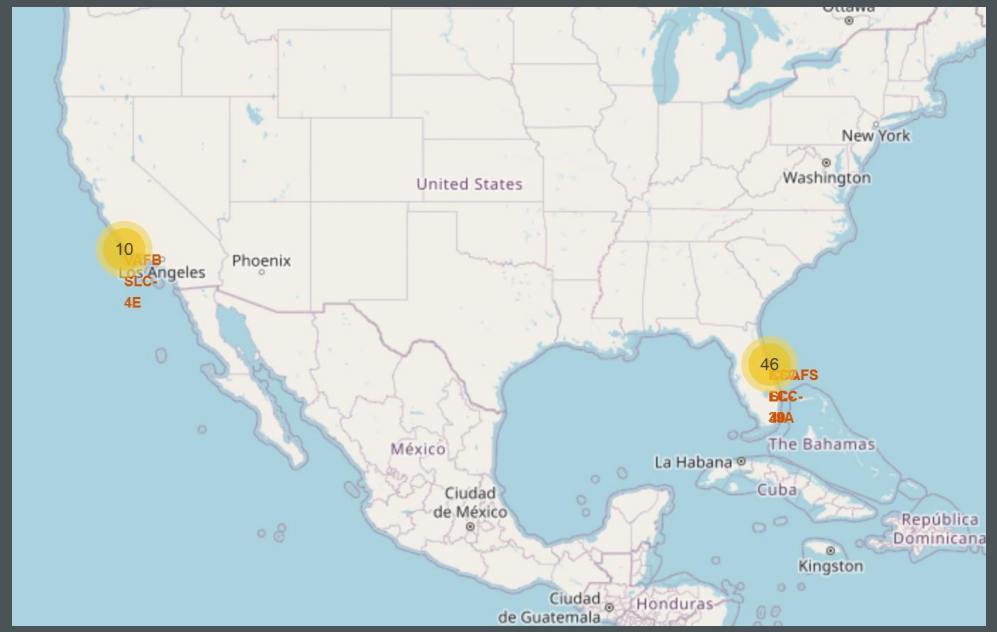
INTERACTIVE MAPS WITH FOLIUM

LAUNCH SITE LOCATIONS

- SpaceX launch sites are near the equator line and U.S.A coasts (Florida & California)



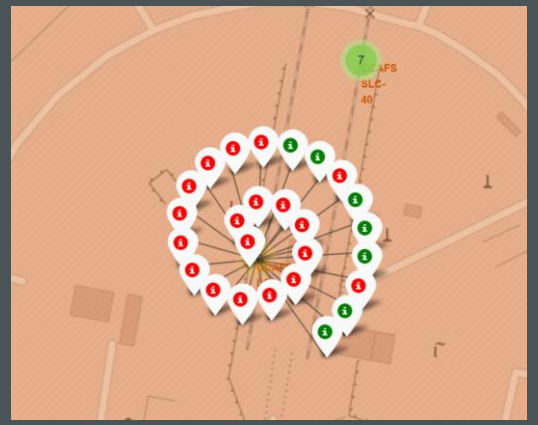
SUCCESS RATE OF ROCKET LAUNCHES (COLOR LABELED)



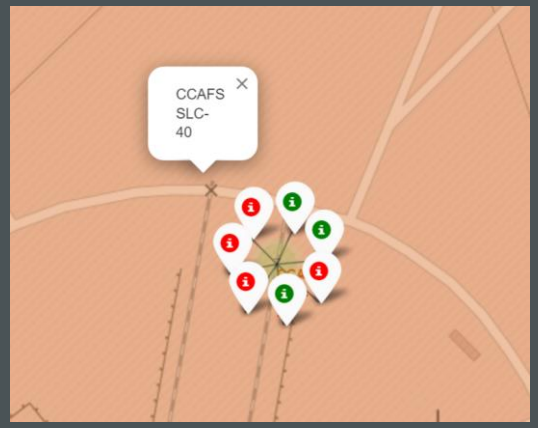
- Green markers show successful launches, while red markers show failures
- **KSC LC-39A** has the highest success rate of rocket launches

VAFB SLC-4E

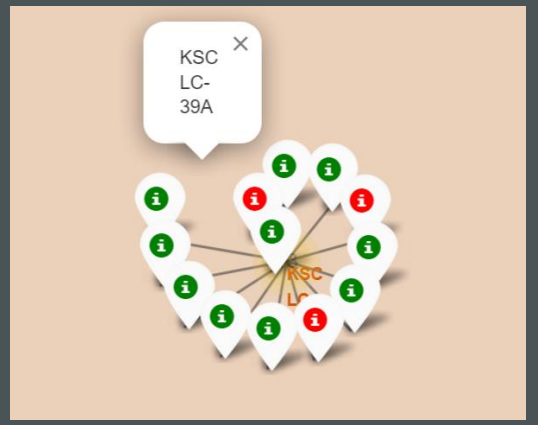
CCAFS LC-40



CCAFS SLC-40

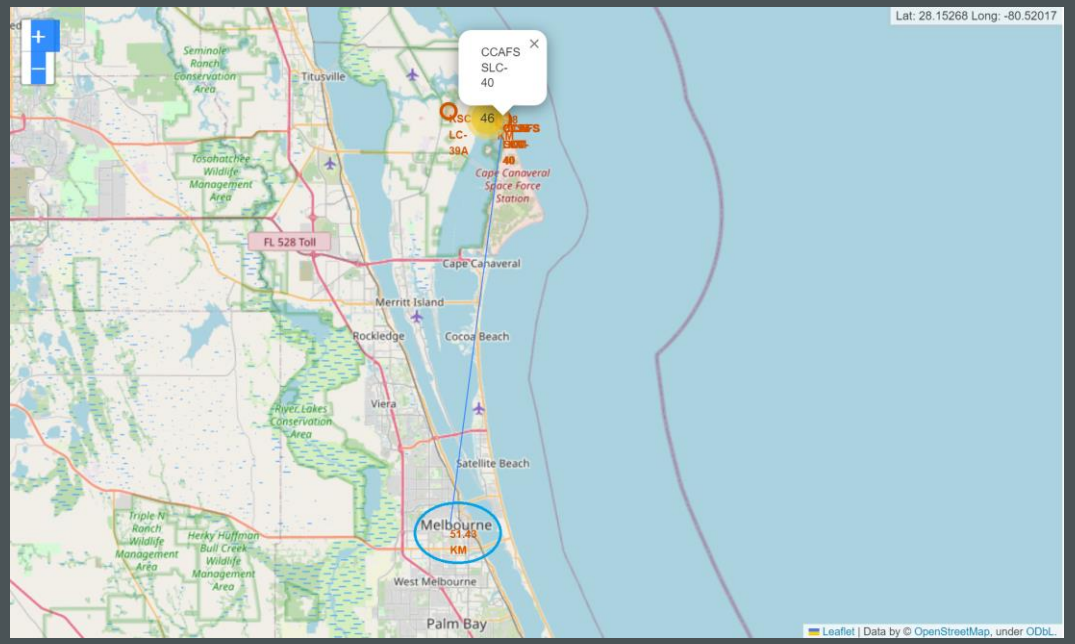


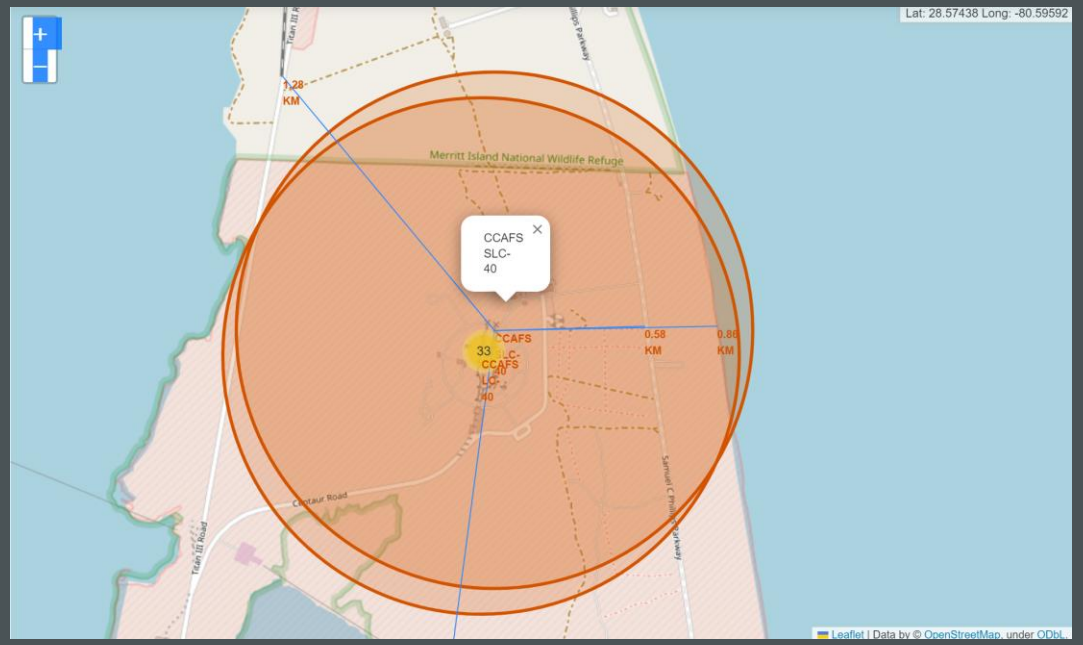
KSC LC-39A



LAUNCH SITE DISTANCE TO LANDMARKS

- Taking site **CCAFS SLC-40** as an example, it is **over 50KM** far from the nearest city, Melbourne
- The site is **less than 1KM** far from coastline, which helps with the rocket landing on water bodies
- The site is **less than 1KM** highway, and a **bit over 1KM** far from the nearest railway.
  Thus, providing convenient transportation required for launches

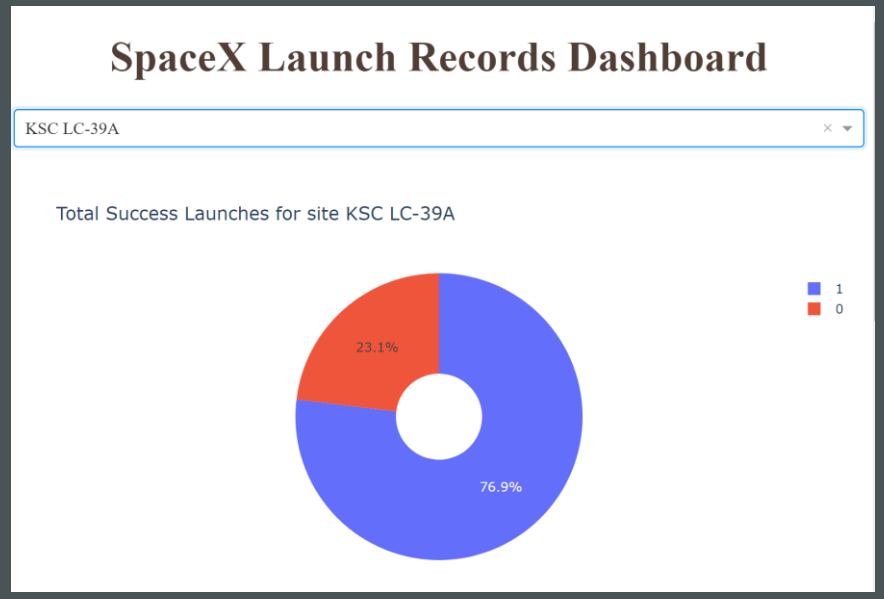


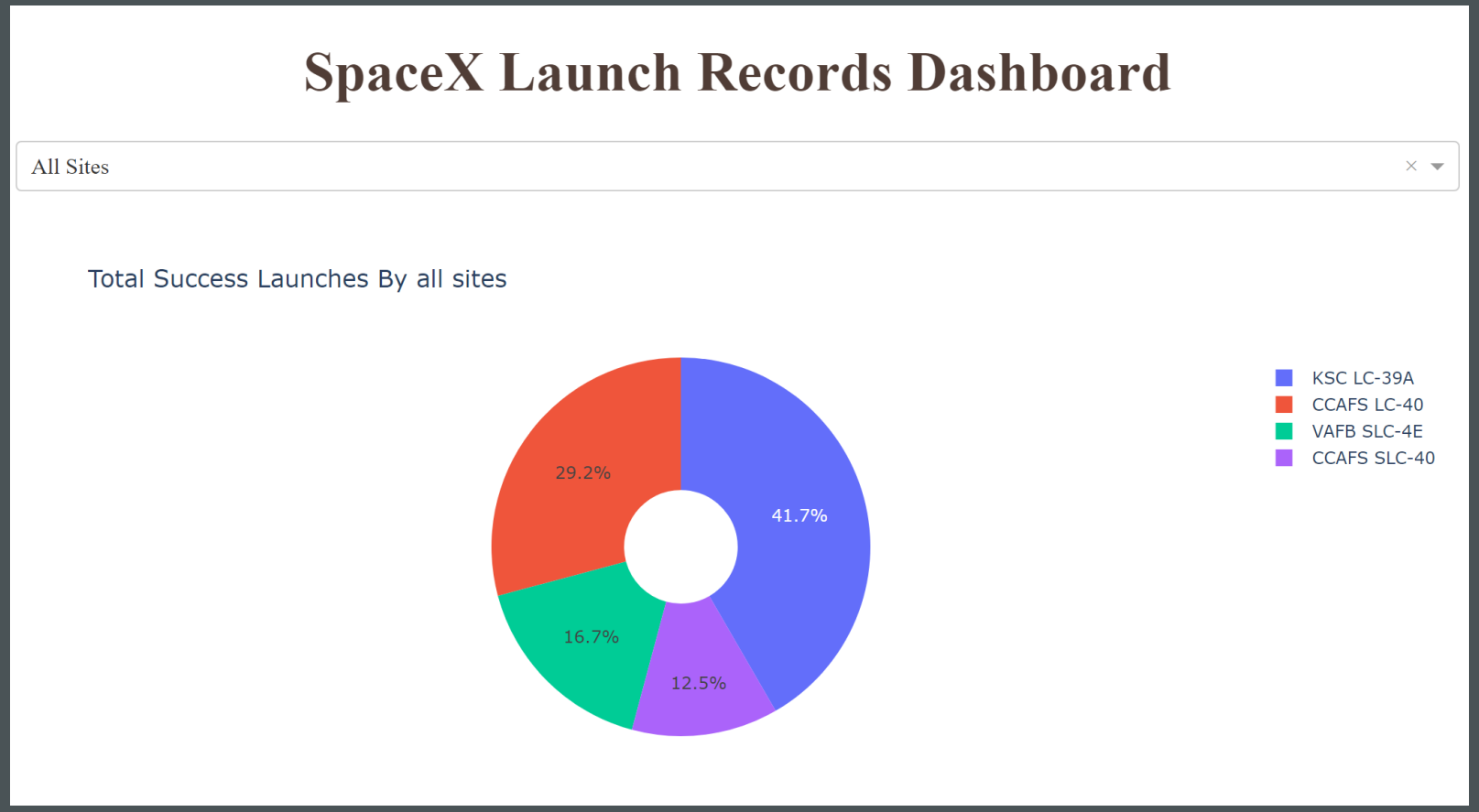


BUILD A DASHBOARD WITH PLOTLY DASH

TOTAL SUCCESS LAUNCHES BY ALL SITES

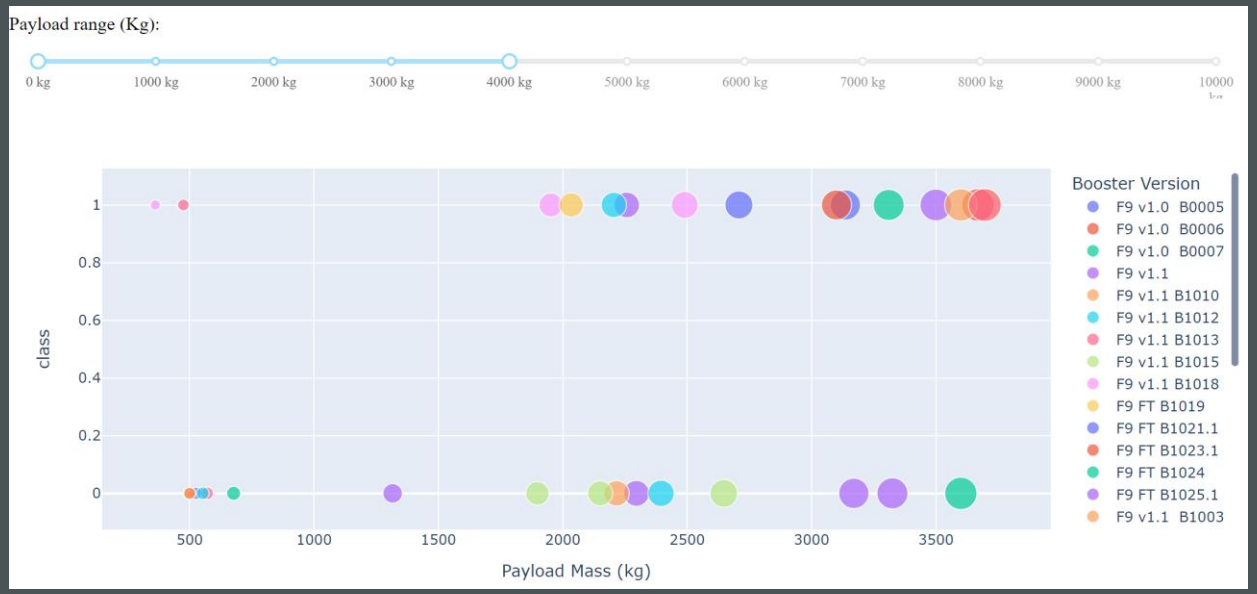
- Site **KSC LC-39A** has the highest success rate of all sites





TOTAL SUCCESS LAUNCHES BY ALL SITES

Low weighted payload **(0kg – 4000kg)**



High weighted payload **(4000kg – 10000kg)**

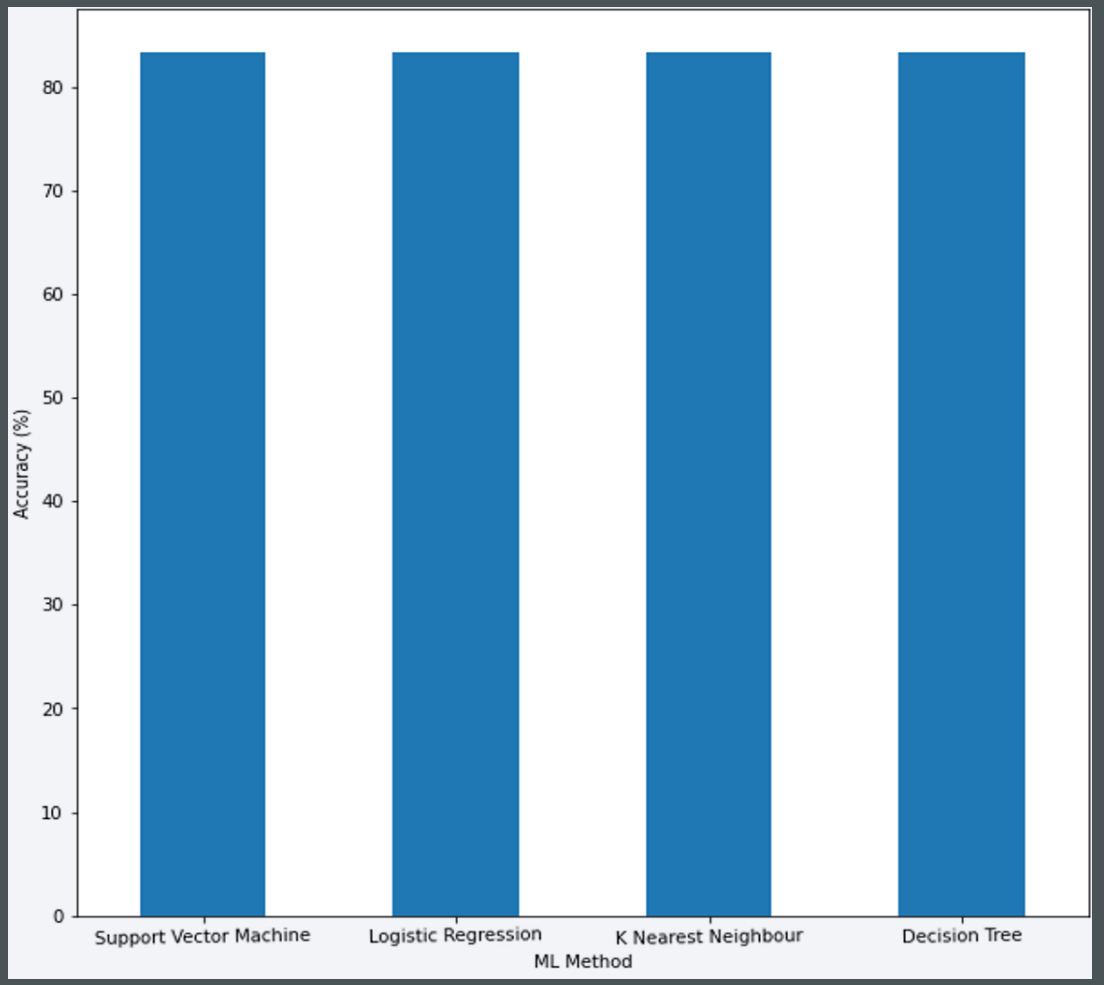
- Success rate for low weighted payloads is better than the heave weighted
- Payload range between 2000kg and 4000kg has the highest success rate

PREDICTIVE ANALYSIS (CLASSIFICATION)

CLASSIFICATION ACCURACY

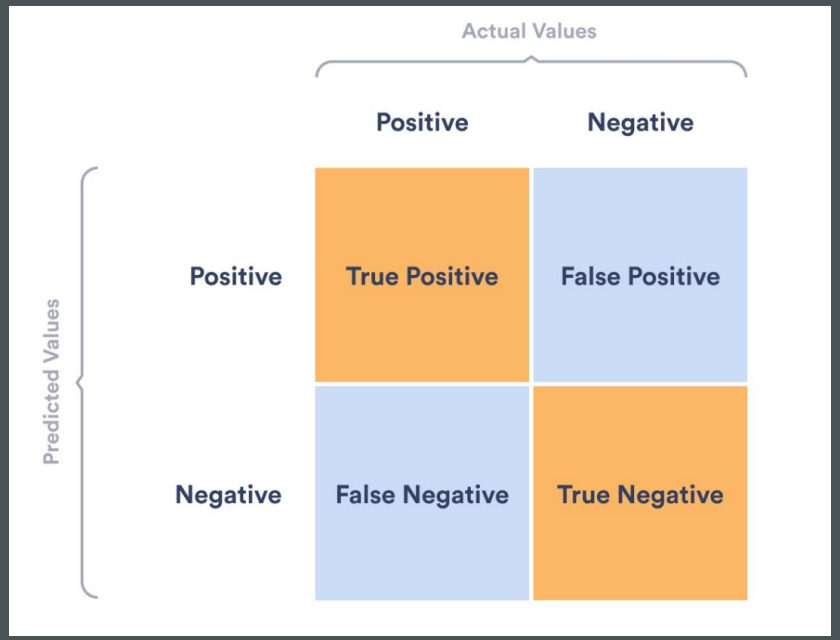
- All the models have performed with the same accuracy
- Decision tree model was selected for the classification

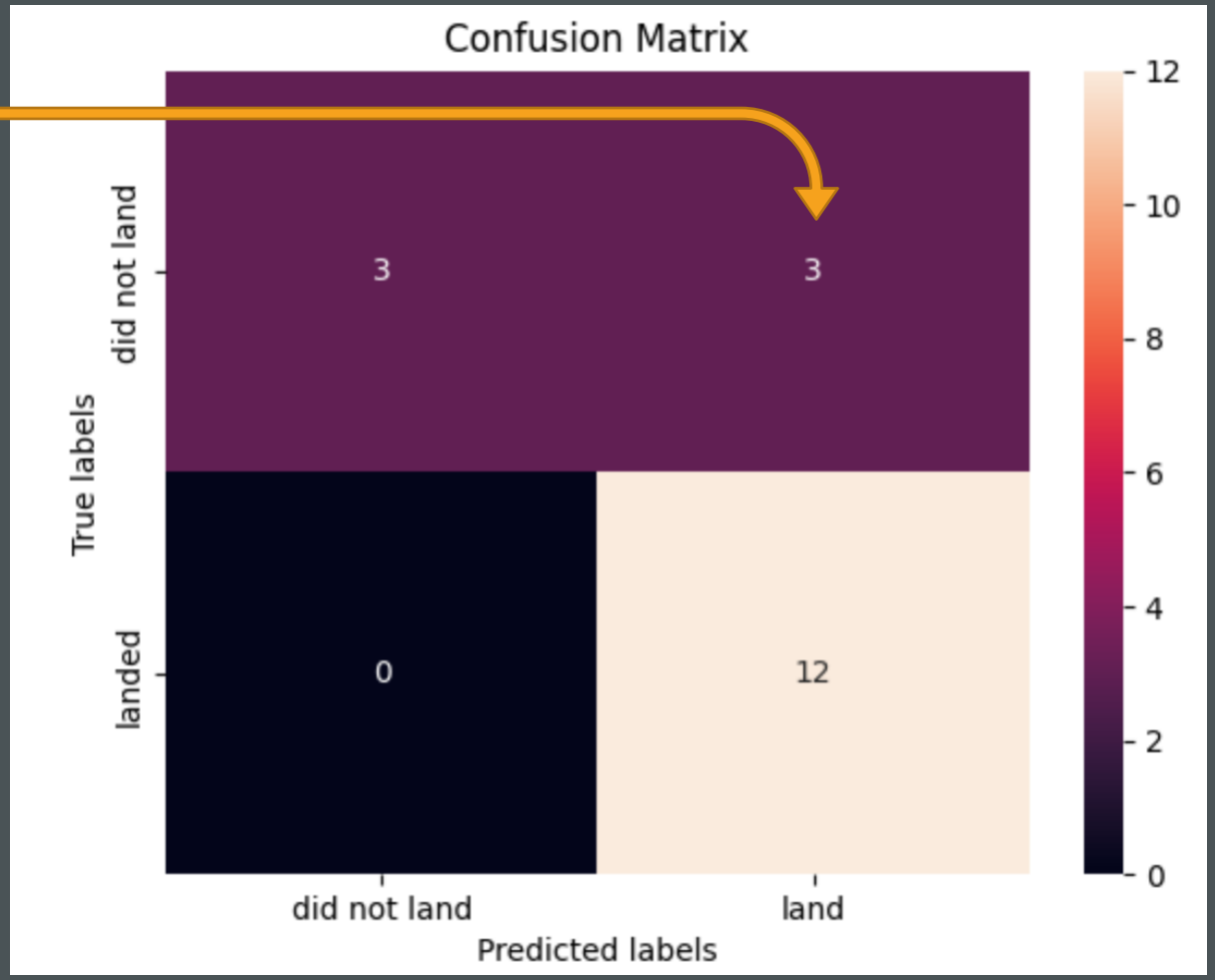
	ML Method	Accuracy Score (%)
0	Support Vector Machine	83.333333
1	Logistic Regression	83.333333
2	K Nearest Neighbour	83.333333
3	Decision Tree	83.333333



CONFUSION MATRIX

- All the models have the same confusion matrix
- The model failed to predict 3 labels





CONCLUSIONS

- For successful rocket launches of SpaceY, we compared and analyzed the data of the competitor SpaceX. The following crucial points were found:
  - All launch sites are close to the coasts, highways and railways. And away from cities
  - The landing success increased with flight number
  - Year 2015 onwards, the success rate of rocket launches improved significantly
  - Site **KSC LC-39A** has the highest launch success rate
  - Orbits **ES-L1, GEO, HEO, SSO, VLEO** has the highest success rate
  - Low weighted payloads (less than 4000kg) performed better than the heavier payloads

The data was used to train a Machine Learning model to predict landing outcome with 83.3% accuracy. This helps saving SpaceY money by reusing the first stage of the rockets.


SPACEY
THANK YOU