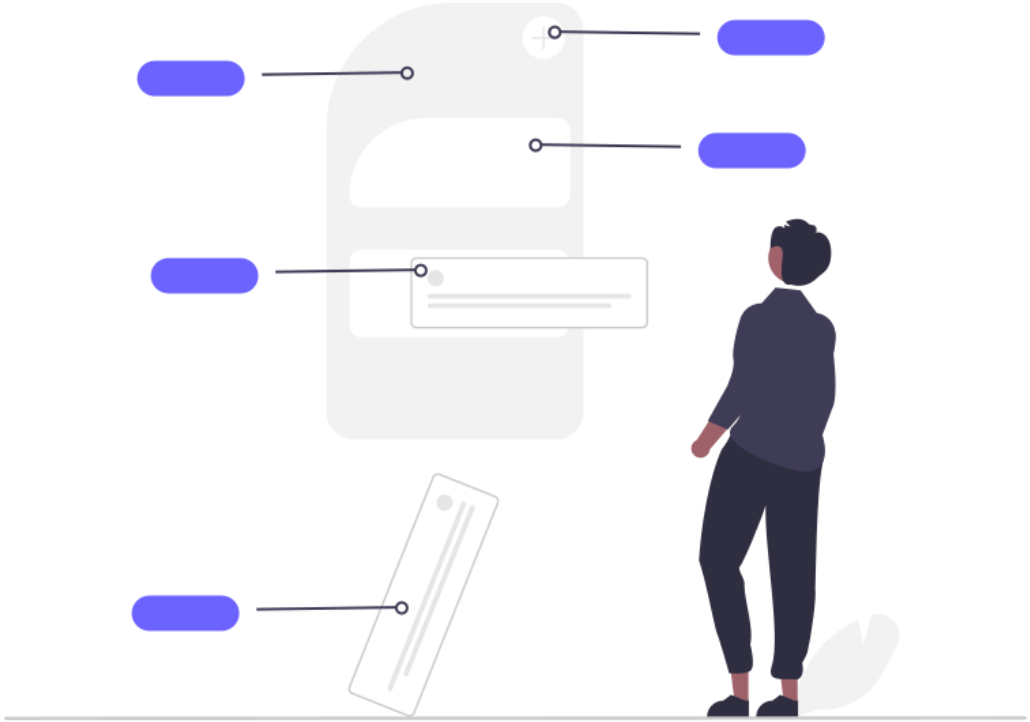
IBM Data Science Capstone Project – SpaceX

Ceren Pajanoja
16 Jan 2022

OUTLINE

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



EXECUTIVE SUMMARY

- **Summary of methodologies**
  - Data collection
  - Data Wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Machine Learning models for predictive analysis
- **Summary of all results**
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

INTRODUCTION

- **Project background and context**

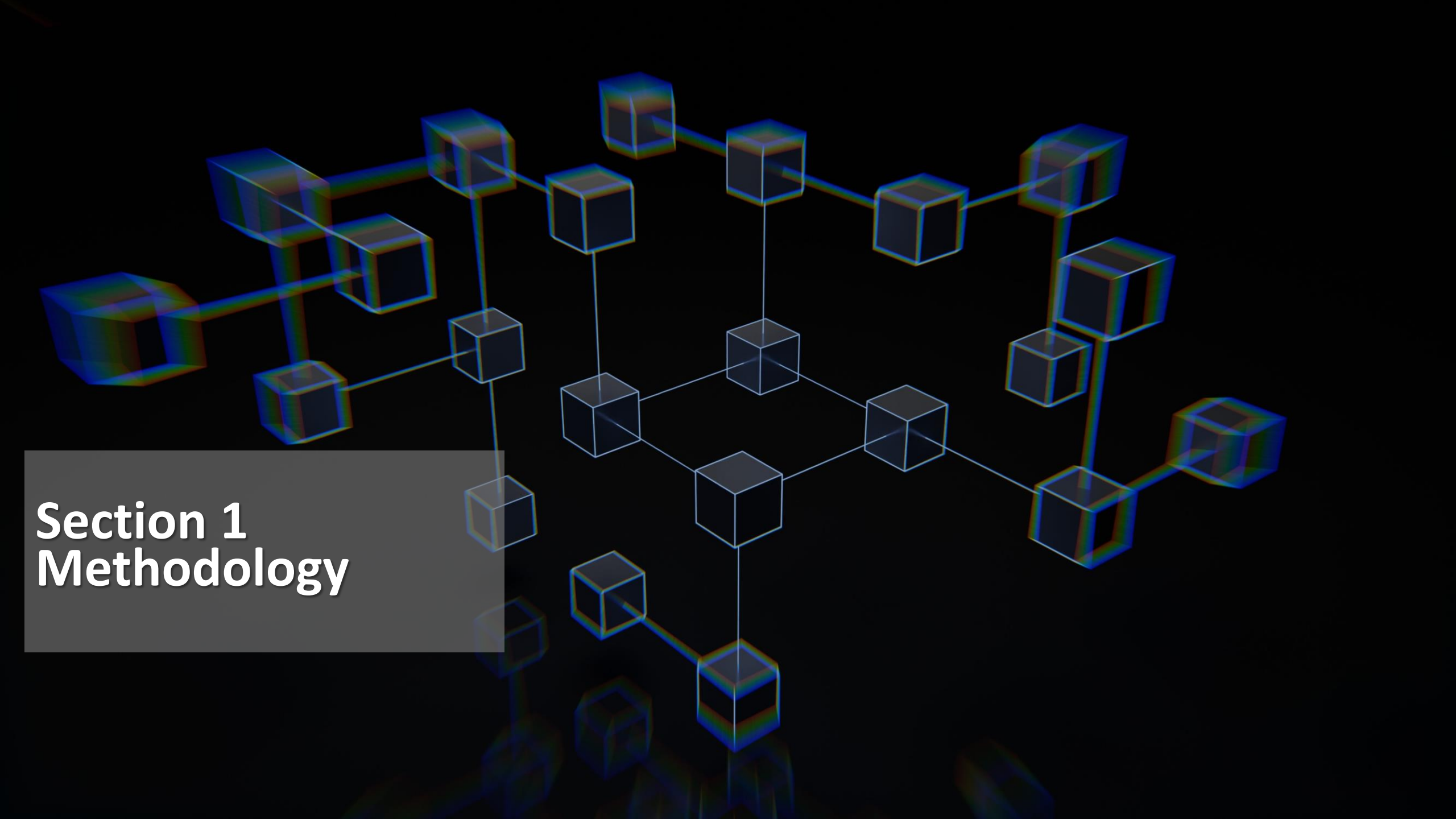
We predicted if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine 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 SpaceX for a rocket launch.

- **Problems you want to find answers**

What factors influence launch success rate? (such as payload mass, orbit type, location of launch site etc)

How can we determine the optimal launch site location based on the information on existing launch site locations?

What are the optimum conditions for predicting successful rocket landing rate?



Section 1
Methodology

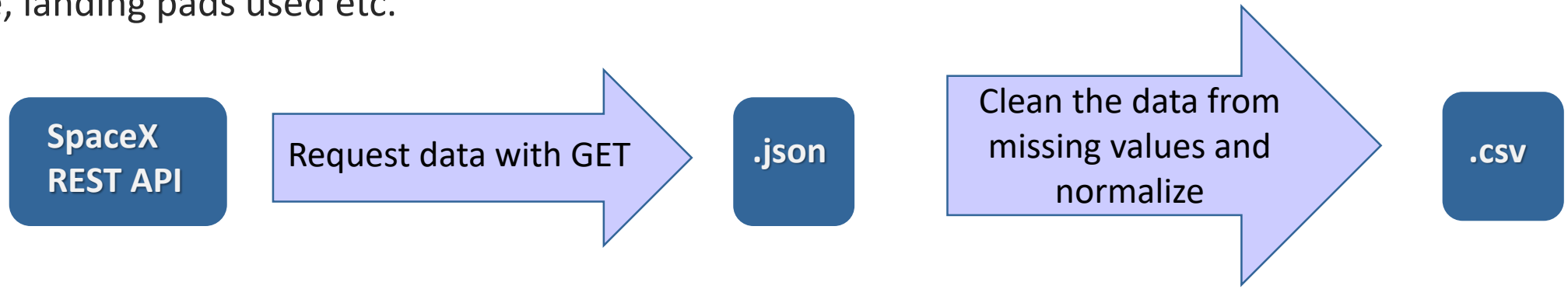
METHODOLOGY

Executive Summary

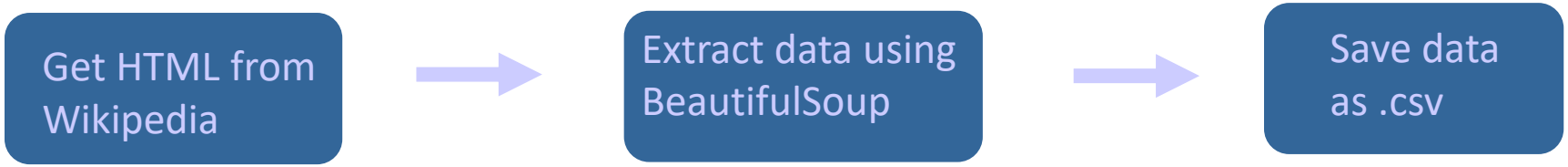
- Data collection methodology:
  - SpaceX REST API
  - (Web Scrapping) from Wikipedia
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

DATA COLLECTION

Launch data is gathered data from SpaceX REST API, and with this API we obtained information such as rocket, payload mass, launch site used, outcome of landing, type of landing, number of flights from the core, landing pads used etc.



We used information from Wikipedia page titled ”List of Falcon 9 and Falcon Heavy Launches” via web scraping (using BeautifulSoup) to collect historical launch records.



Data Collection – SpaceX API

1) Get Response from API

Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

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

2) Converting Response to .json file

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

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

3) Apply custom functions to clean data

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

4) Assign list to dictionary, and then create dataframe

```
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}
```

5) Filter dataframe and save as .csv file format

```
data_falcon9 = new_df[new_df['BoosterVersion']=='Falcon 9']
```

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

Github Link:
https://github.com/CerenPaja/IBM_Data_Science_Project/blob/master/Data%20Collection%20API.ipynb

Data Collection – Scraping

1) Get Response from HTML

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
response = requests.get(static_url)
```

2) Create BeautifulSoup Object

```
soup = BeautifulSoup(response.content, 'html.parser')
```

3) Find Tables and get column names

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header
() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0
`) into a list called column_names
cells = first_launch_table.find_all('th')


#print(cells)
for row in cells:
    #print(row)
    name=extract_column_from_header(row)
    #print(name)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

4) Create dictionary

```
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.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

5) Append data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainr
owheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to
launch a number
        if rows.th:
            if rows.th.string:
```

6) Convert dictionary to dataframe and save as .csv file

```
#df=pd.DataFrame(launch_dict)
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

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

Github Link:
https://github.com/CerenPaja/IBM_Data_Science_Project/blob/master/Data%20Collection%20with%20Web%20Scraping.ipynb

Data Wrangling

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

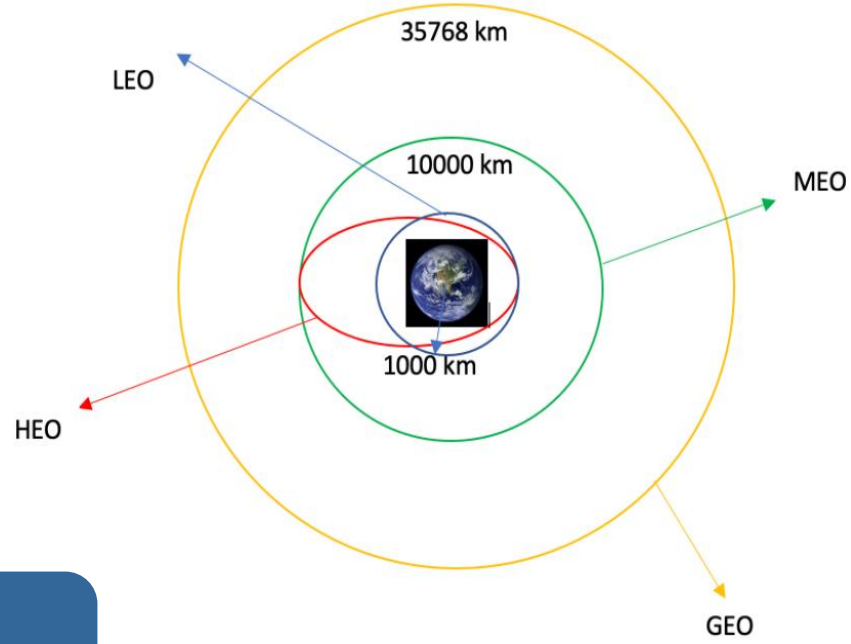
We performed exploratory data analysis and determined the training labels;

- 1 == successful
- 0 == unsuccessful

We calculated the number of launches at each site, and the number and occurrence of each orbits.

We created landing outcome label from outcome column → Exported the results to csv

Here are some common orbit types SpaceX uses;



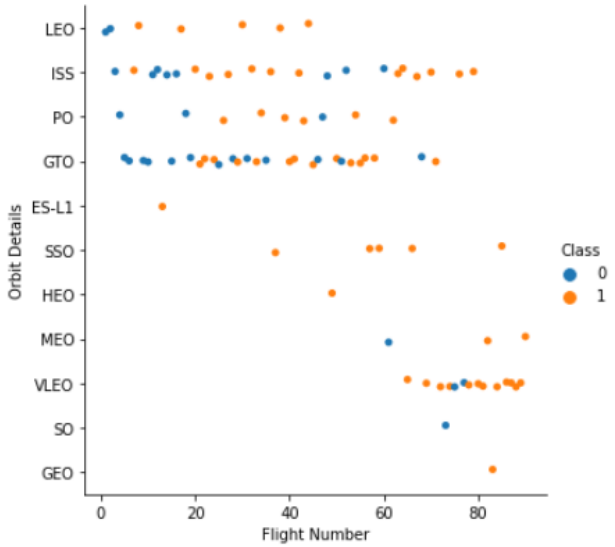
Github Link:
https://github.com/CerenPaja/IBM_Data_Science_Project/blob/master/EDA.ipynb

EDA with Data Visualization

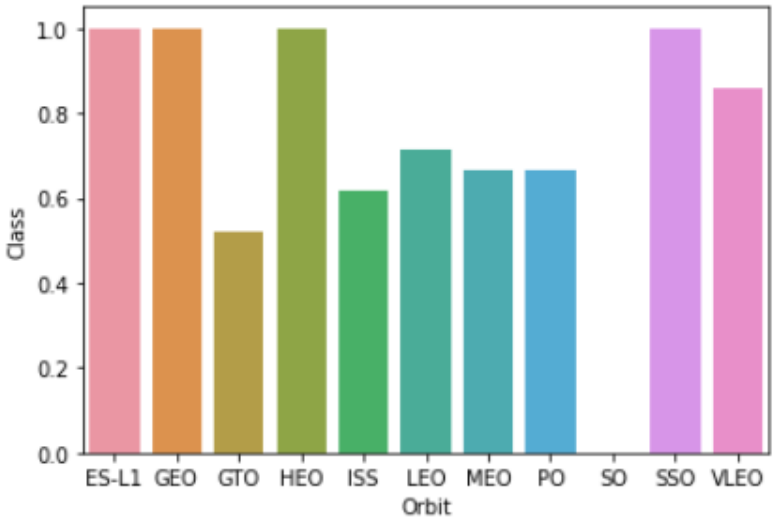
We explored the data by visualizing the relationship between;

Flight number vs Payload mass
Flight number vs Launch site
Payload vs Launch site
Orbit vs Flight number
Payload vs Orbit type
Orbit vs Payload Mass

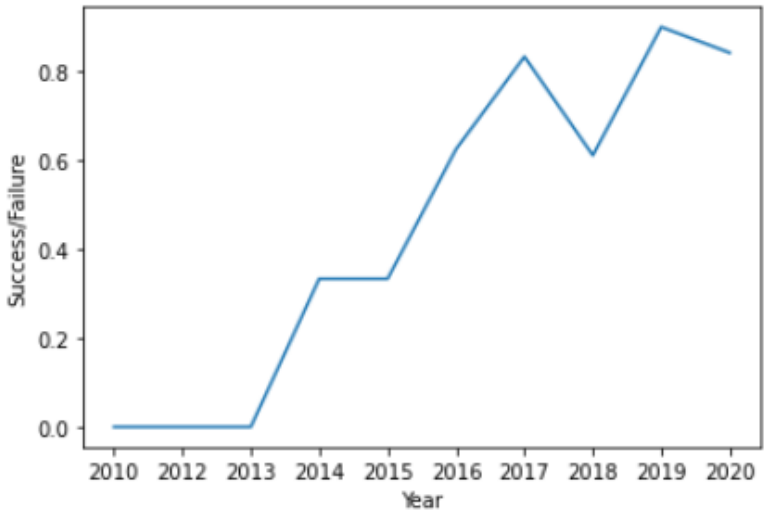
via Scatter Plots



Mean Vs Orbit via **Bar Plot**



Success Rate vs Yeas via **Line Plot**



Scatter Plots ➔ show relationship between 2 variables

Bar Plots ➔ show comparison of data between groups

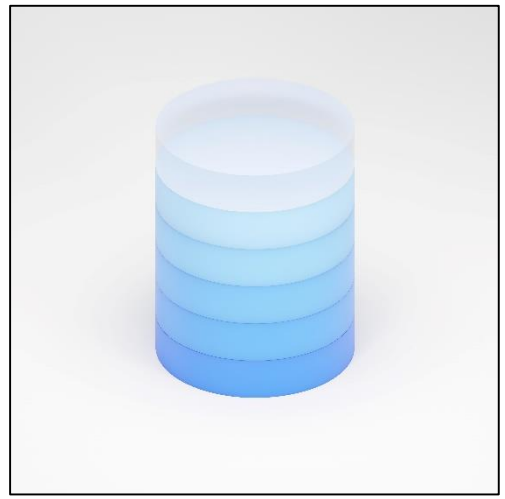
Line Plots ➔ show data variables and trends

Github Link:
https://github.com/CerenPaja/IBM_Data_Science_Project/blob/master/EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

We usedSQL to get insight from the data. We wrote queries to understand SpaceX data:

- Displaying the names of unique launch sites in the space mission.
- Displaying 5 records where launch sites begin with the string 'CCA'
- Displaying the total payload mass carried by boosters 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. Use a subquery
- Listing the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Ranking 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



Github Link:
https://github.com/CerenPaja/IBM_Data_Science_Project/blob/master/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

We created an interactive map to visualize Launch Data;

We marked all launch sites, and added map objects around each launch site with Circle Marker.

We assigned the launch_outcomes (failure or success) to class 0 for failure, and 1 for success. For this we have used color-labeled marker clusters; Green and Red markers on the map (MarkerCluster())

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

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

Github Link:
https://github.com/CerenPaja/IBM_Data_Science_Project/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

We built an interactive dashboard with Plotly dash

We plotted **pie charts** showing the total launches by a certain site/ all sites;

- *Display relative proportions of multiple classes of data*
- *Size of each circle is made proportional to the total quantity it represents*

We plotted **scatter graph** showing the relationship with Outcome and Payload Mass (Kg) for the different booster version;

- *Showing relationship between 2 variables*
- *Displays non-linear pattern*
- *Range of data such as max or min value can be determined*

Github Link:
https://github.com/CerenPaja/IBM_Data_Science_Project/blob/master/dashboard.py

Predictive Analysis (Classification)

BUILDING MODEL:

1. Load the data using numpy and pandas
2. Transform Data
3. Split data into training and test data sets
4. Built different machine learning models
5. Set parameters and algorithms using GridSearchCV
6. Fit datasets into GridSearchCv objects and train the data

IMPROVING MODEL:

1. Check accuracy for each model
2. Tune different hyperparameters for each algorithms
3. Plot confusion matrix

EVALUATING MODEL:

1. Feature Engineering
2. Algorithm Tuning

FINDINGBEST PERFORMING CLASSIFICATION MODEL:

1. Model with best accuracy score is the best performing model
2. All scores for different algorithms are listed in the notebook

Github Link:
https://github.com/CerenPaja/IBM_Data_Science_Project/blob/master/Machine%20Learning%20Prediction.ipynb

Results

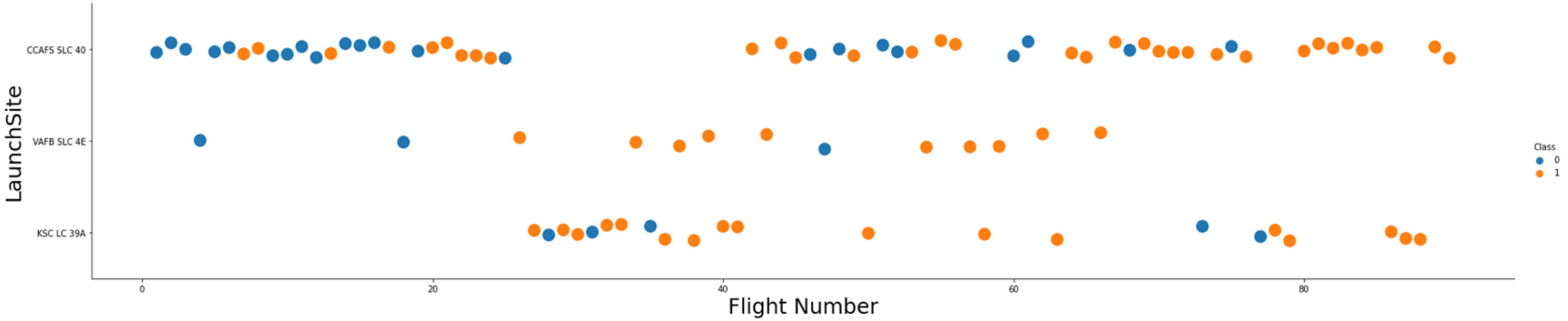
Exploratory data analysis results

Interactive analytics demo in screenshots

Predictive analysis results

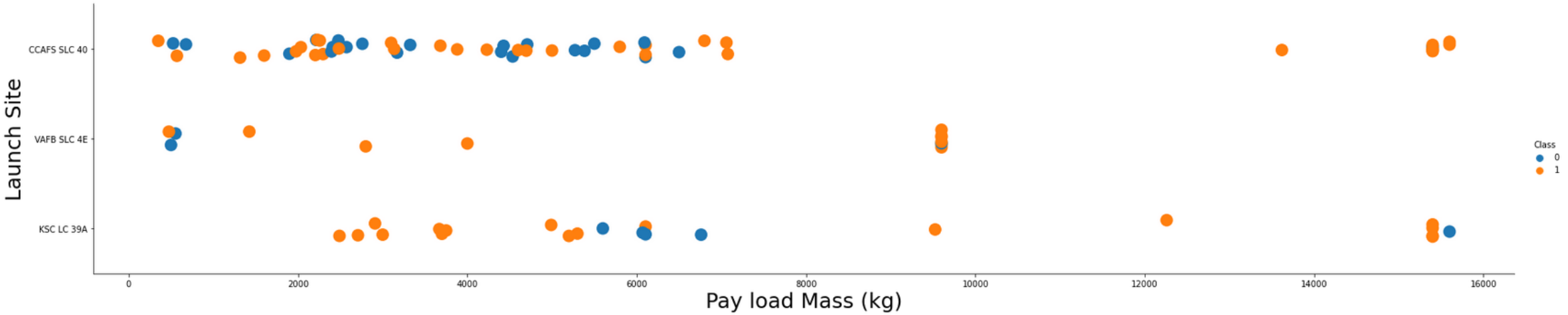
Section 2
Insights Drawn from EDA

Flight Number vs. Launch Site



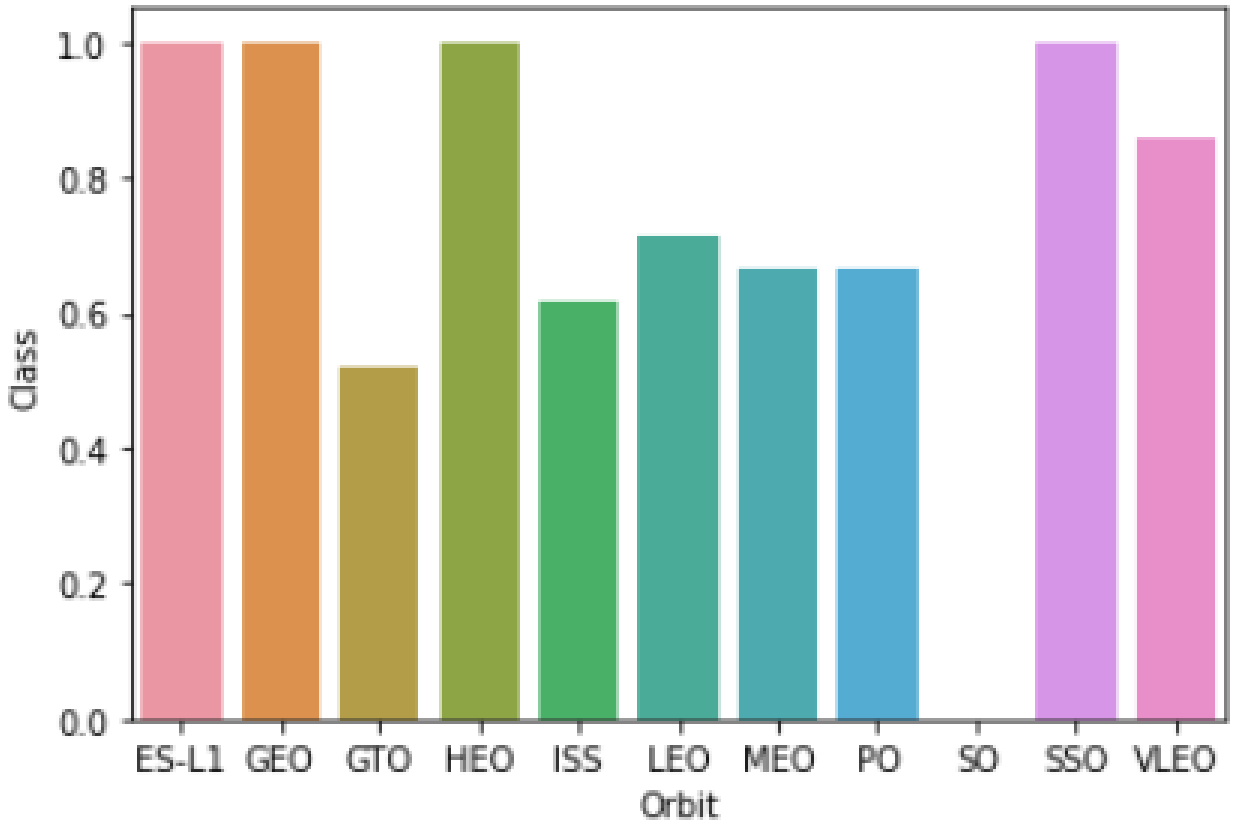
The higher the flight numbers for launch site CCAFS SLC 40 the higher success rate for the rocket

Payload vs. Launch Site



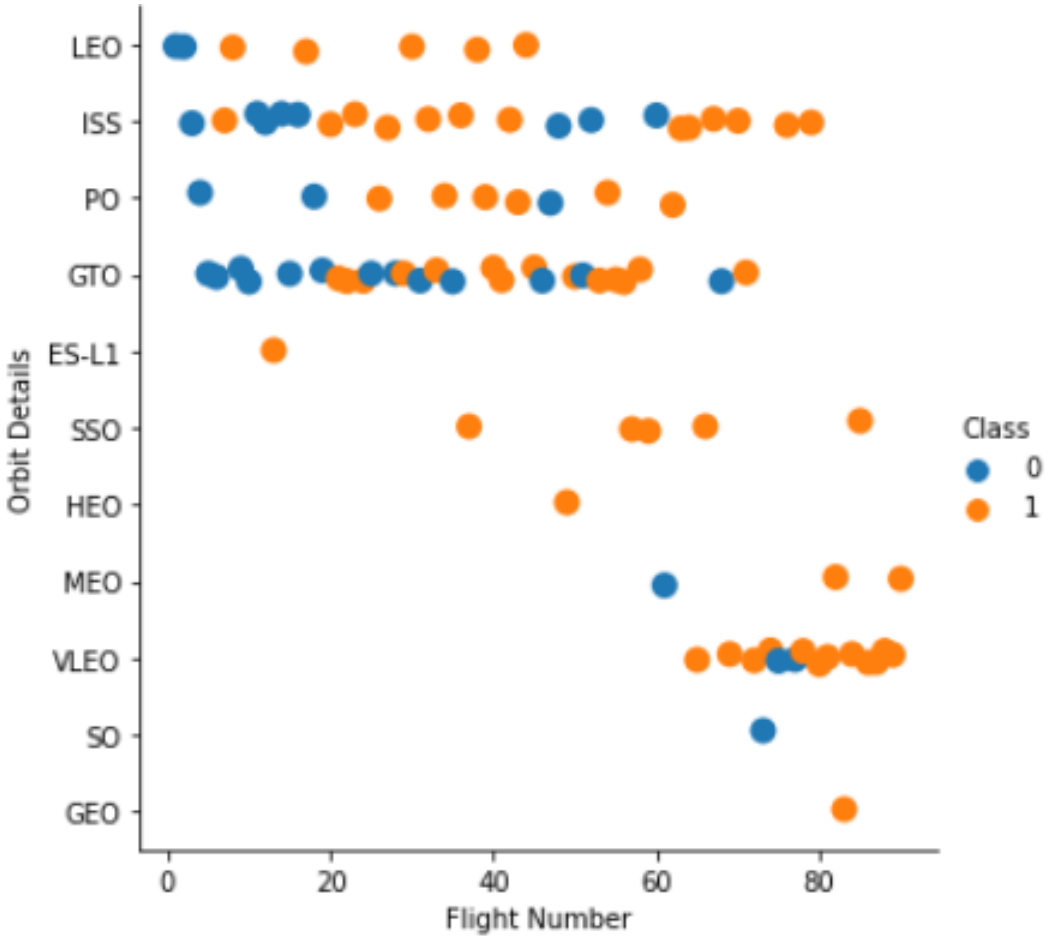
The greater the payload mass for launch site CCAFS SLC 40 the higher success rate for the rocket

Success Rate vs. Orbit Type



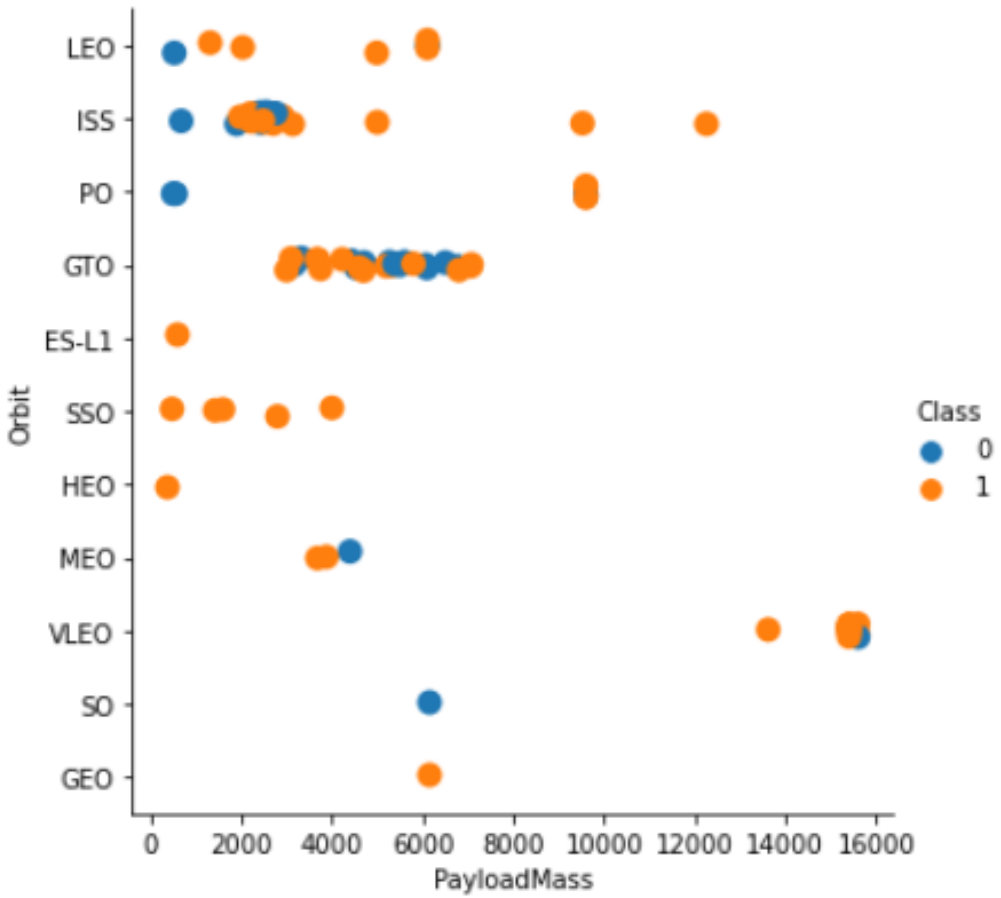
Ee can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

Flight Number vs. Orbit Type



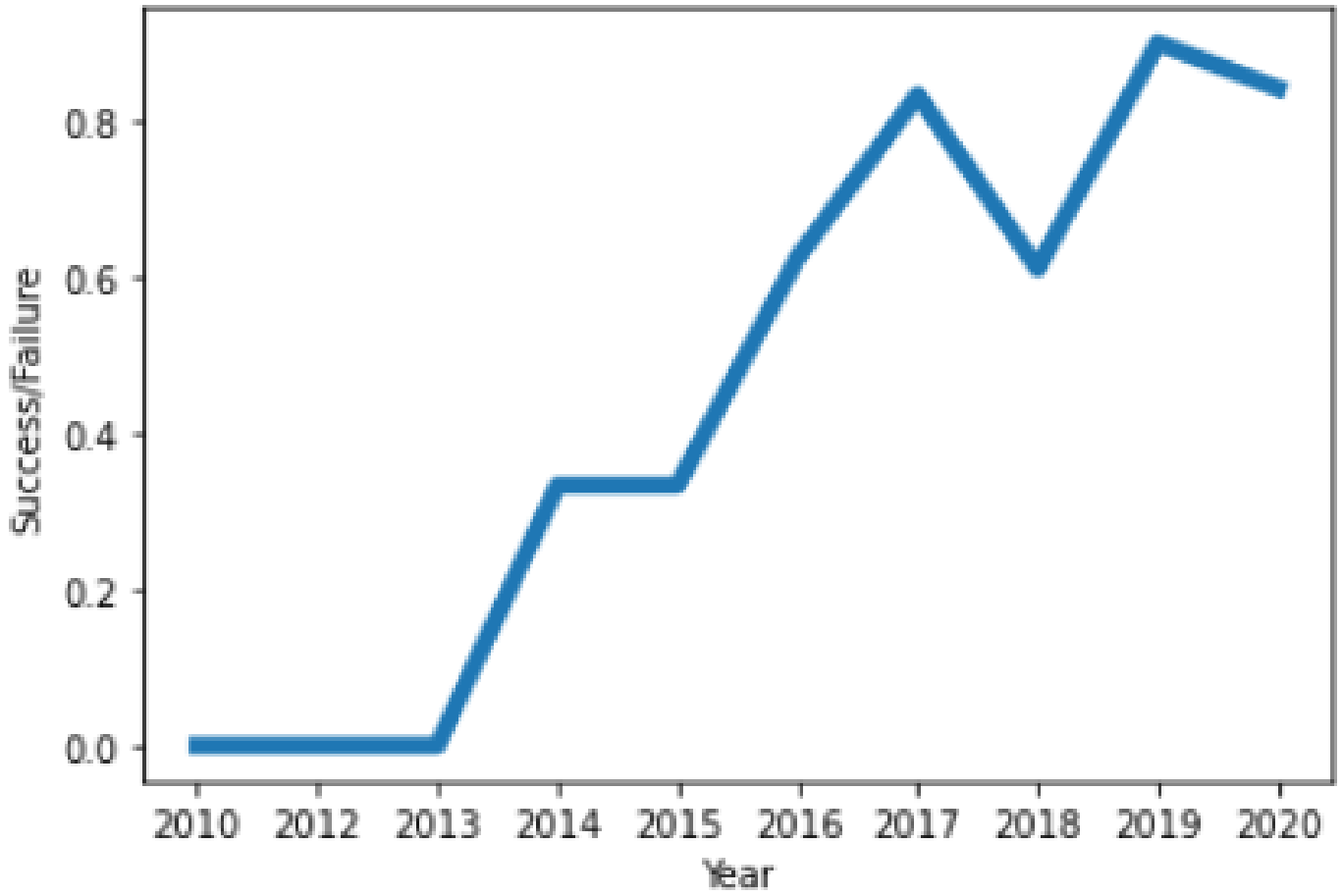
For LEO orbit success is related to the number of flights, whereas in GTO orbit, there is no relationship between flight number and the orbit.

Payload vs. Orbit Type



We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits, whereas they have a negative influence on MEO

Launch Success Yearly Trend



We can observe that success rate since 2013 kept on increasing till 2020.

All Launch Site Names

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

```
%sql select Unique(LAUNCH_SITE) from SPACEXTBL;
```

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

We used the key word Unique to show only unique launch sites from the SpaceX data.

Launch Site Names Begin with 'CCA'

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

```
%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT
5;
```

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

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

Total Payload Mass

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

```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
```

payloadmass
45596

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

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 payloadmass from SPACEXTBL WHERE Booster_Version = 'F9 v
1.1';
```

payloadmass
2928

The average payload mass carried by booster version F9 v1.1 is calculated as 2928.4

First Successful Ground Landing Date

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

Hint:Use min function

```
%sql select min(DATE) from SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

1
2015-12-22

We observed that the dates of the first successful landing outcome on ground pad was December 22nd 2015

Successful Drone Ship Landing with Payload between 4000 and 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 SPACEXTBL where LANDING__OUTCOME='Success (drone ship)' and PA
YLOAD_MASS__KG_ BETWEEN 4000 and 6000
```

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

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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME LIKE '%Success%'
```

1
100

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME LIKE '%Failure%'
```

1
1

We used '%' to filter for WHERE MISSION_OUTCOME was a success or a failure.

Boosters Carried Maximum Payload

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

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

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

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

2015 Launch Records

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

```
%sql SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL  where MISSI
ON_OUTCOME LIKE '%Failure%' AND EXTRACT(YEAR FROM DATE)='2015' ;
```

1	mission_outcome	booster_version	launch_site
6	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40

We used a combinations of the WHERE, LIKE and AND conditions to filter for failed landing outcomes for year 2015

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)) between the date 2010-06-04 and 2017-03-20, in descending order

```
%sql SELECT LANDING__OUTCOME,count LANDING__OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY count(LANDING__OUTCOME)DESC;
```

landing__outcome	landing__outcome_1
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

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.

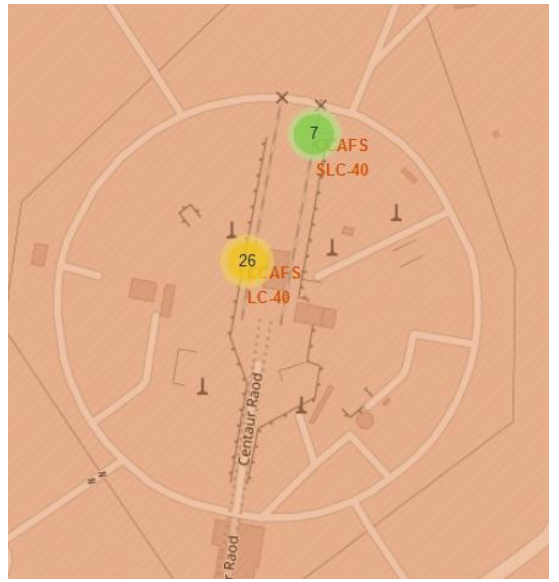
Section 3
Launch Sites Proximities Analysis

All launch sites global map markers

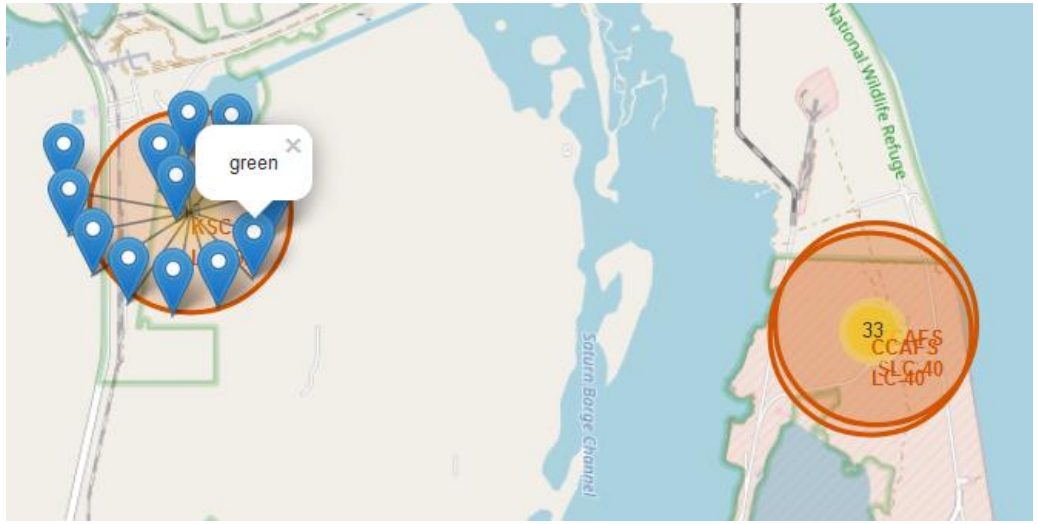
We can see launch sites in Florida and California

Markers showing launch sites with color labels

We can see launch sites in California Red indicates failed launch, Green indicates successful launch

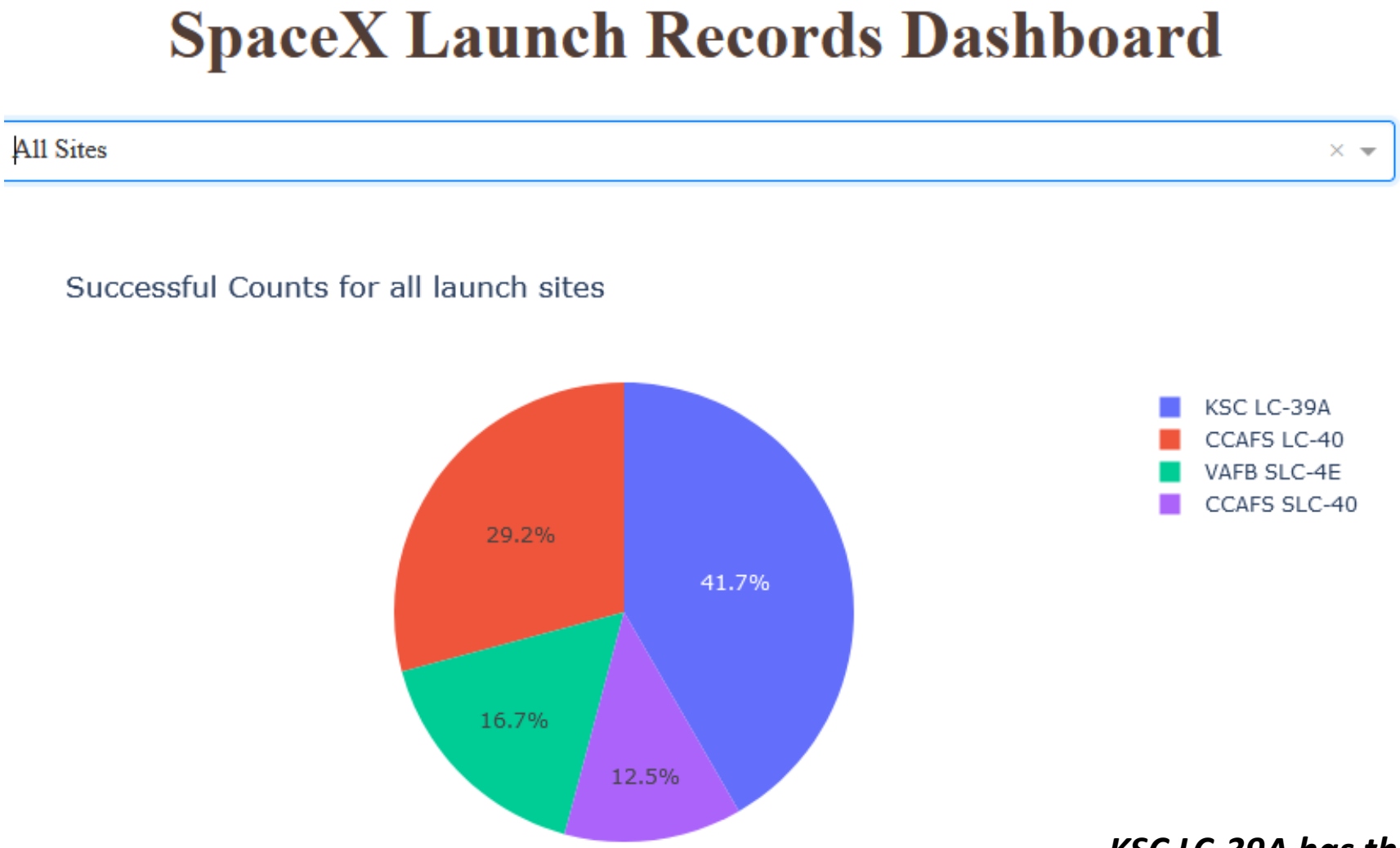


We can see launch sites in Florida



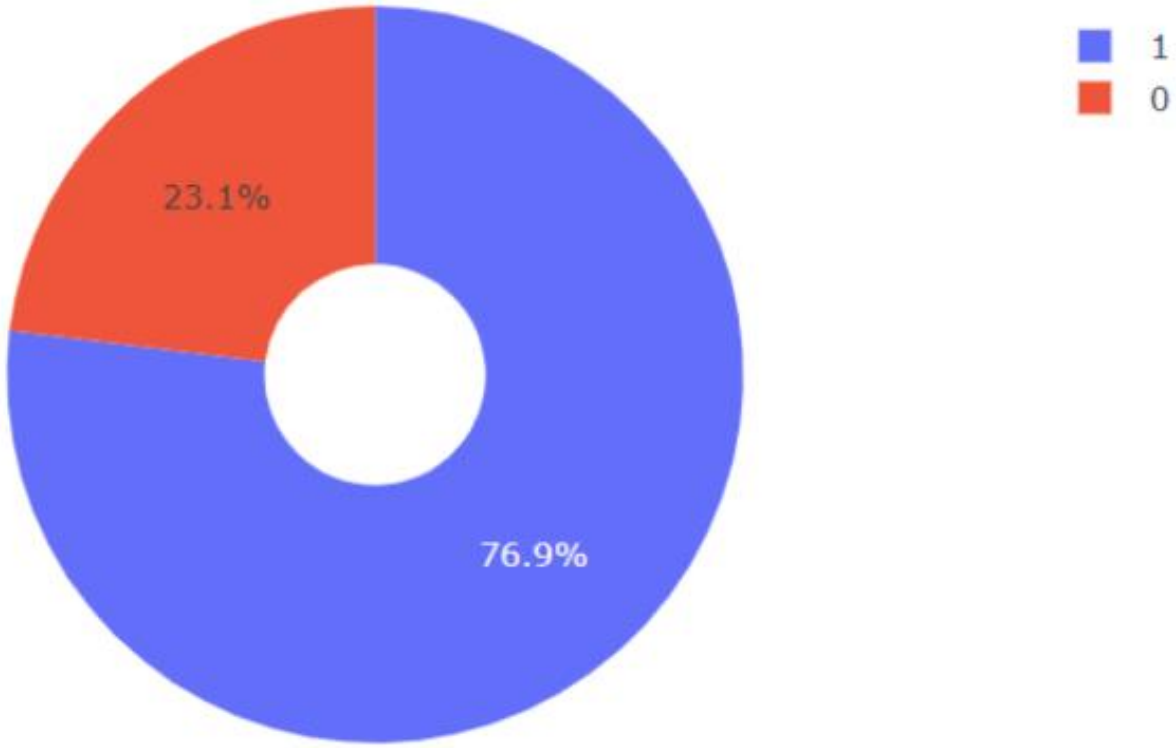
Section 4
Build a Dashboard with Plotly Dash

Pie chart showing success percentage for all launch sites



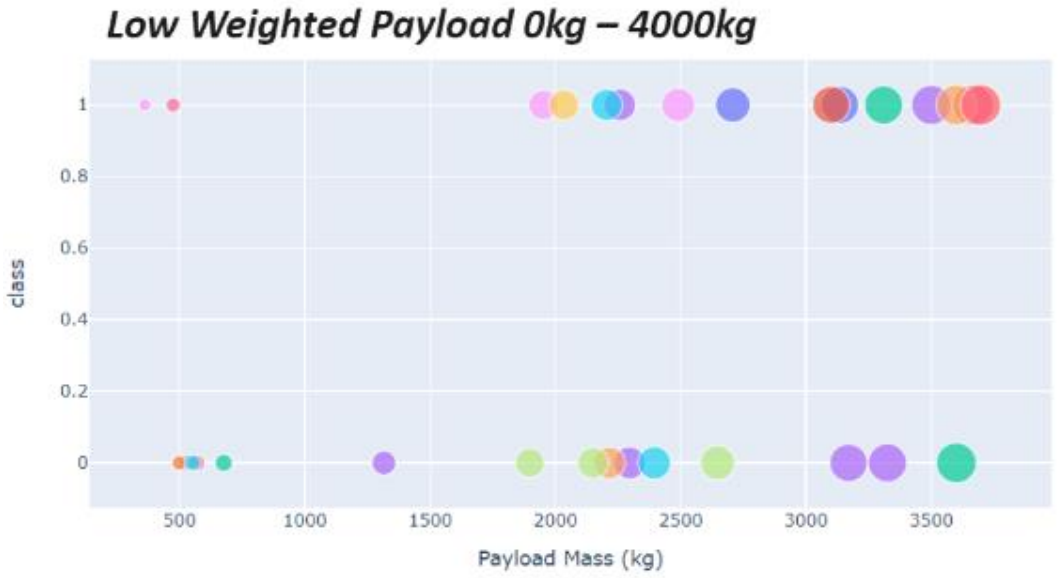
KSC LC-39A has the most successful launches

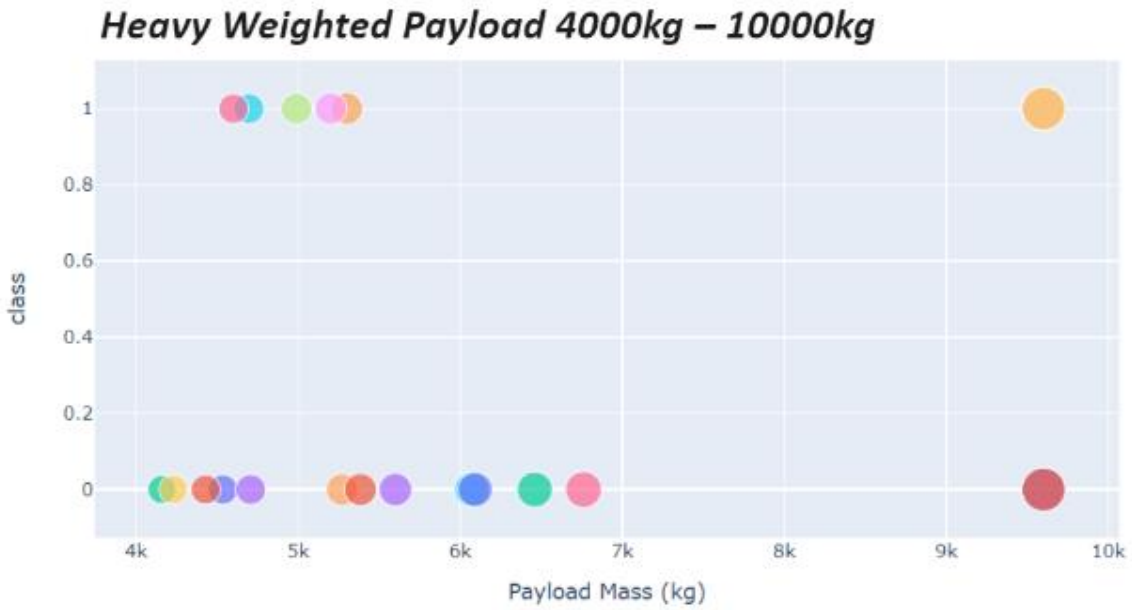
Pie chart showing launch site with the highest launch success ratio



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

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider





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

Section 5
Predictive Analysis (Classification)

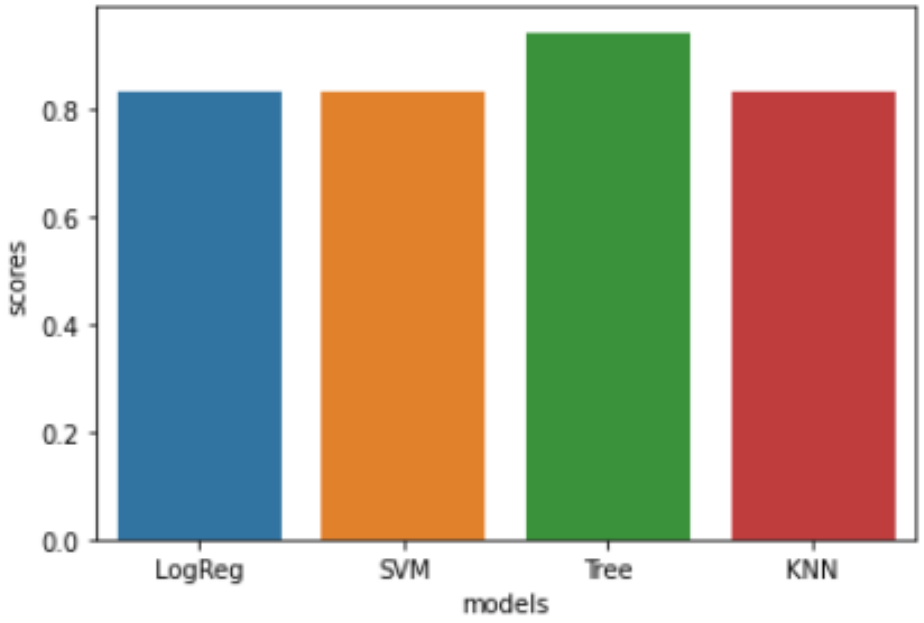
Classification Accuracy

Find the method performs best:

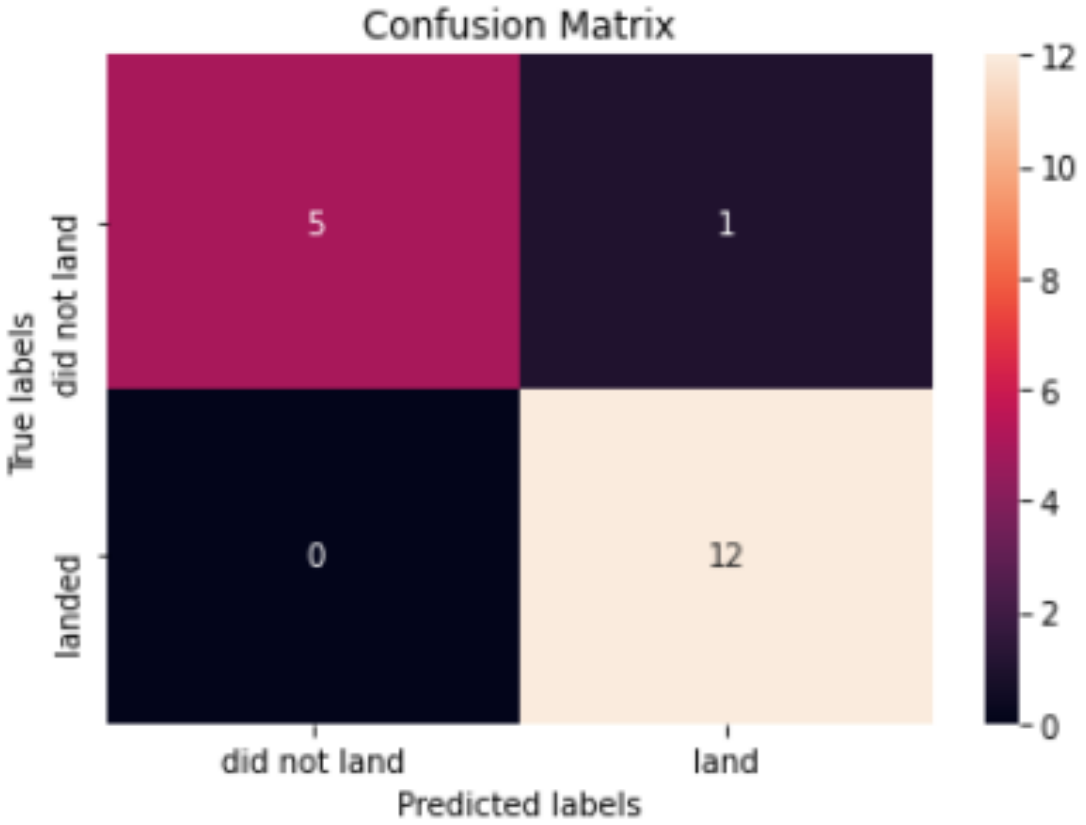
```
scores = [lr_score,svm_score,tree_score,knn_score]
print(scores)
print(scores.index(max(scores)))
```

```
[0.8333333333333334, 0.8333333333333334, 0.9444444444444444, 0.8333333333333334]
2
```

The decision tree classifier is the model with the highest classification accuracy



Confusion Matrix



The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.

Conclusions

Our conclusions;

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.


Thank You !