IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Sandeep P Vangala>
<01/14/2023>

Outline

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

Executive Summary

- Summary of methodologies
  - ❖ Data collection via different proceedings (API-Interface, Web Scraping, API Scraping)
  - ❖ Data wrangling
  - ❖ Exploratory data analyzing with SQL and visualization
  - ❖ Interactive visual analytics with Folium and Dashboard
  - ❖ Predictive Analysis for different classification model by using machine learning
- Summary of all results
  - ❖ Exploratory data analysis for the results of the different classification models
  - ❖ Interactive analytics demo in screenshots
  - ❖ Results of Predictive analysis

Introduction

- Project background and context

  When compared to other suppliers, Space X offers their Falcon 9 rocket launches at a low cost of $62 million. Other vendors charge launch fees of more than $165 million. The first stage's reuse is what accounts for the significant cost difference. The ability to forecast whether or not a rocket's first stage can land again and be reused is therefore of significant importance to companies other than Space X. Therefore, the stated objective of this project is to build a machine learning pipeline with various classifiers in order to be able to forecast whether the first stage will successfully land.

- Problems you want to find answers

  Are there specific elements that affect the likelihood of a successful first stage landing? Is there a relationship between the several elements that make up a successful land maneuver?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data were collected by using SpaceX-API und web scraping from Wikipedia page
- Perform data wrangling
  - Data were cleaned from irrelevant information and creating one-hot-encoding data fields for machine learning
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Data were visualized by different kinds of plots for searching the pattern.
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Describe how data sets were collected.
  - ✓ Requesting the SpaceX-API
    - ❖ Decoding the response content, coming in json-format, and turn into a Pandas dataframe
    - ❖ Cleaning dataframe from uninteresting columns and multiple rows
    - ❖ Filter according to desired parameters
    - ❖ Checking the dataset for missing data and filling missing data up where required
- You need to present your data collection process use key phrases and flowcharts
  - ✓ Transforming the received html-table into a Pandas DataFrame

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

i. Getting response from SpaceX-API
ii. Converting response to json-file and transferring into a Data-Frame
iii. Clean up the data using prepared functions
iv. Creating a final dataset with the columns of interest
v. Filtering dataset for selected Booster Version

- GitHub URL: https://github.com/vangalasandeep/IBMCapstone/blob/master/Data%20Collection%20API%20Lab.ipynb

1

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

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

2

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

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

3

```
# Call getBoosterVersion
getBoosterVersion(data)

# Call getLaunchSite
getLaunchSite(data)

# Call getPayloadData
getPayloadData(data)

# Call getCoreData
getCoreData(data)
```

4

```
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

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df1[df1['BoosterVersion']!='Falcon 1']
```

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

1. Requesting Falcon9 Launchdata from Wiki-page
2. Creating Beautiful-soup object from HTML response
3. Extract all column/variable names from the HTML table header
4. Create a data frame by parsing the launch HTML tables
5. Exporting data to CSV-file

- https://github.com/vangalasandeep/IBMCapstone/blob/master/jupyter-labs-webscraping.ipynb

1.
```
[4]:
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_
# assign the response to a object
html_data=requests.get(static_url).text
```

2.
```
[21]:
# Use BeautifulSoup() to create a BeautifulSoup
soup = BeautifulSoup(html_data,'html.parser')
```

3.
```
# Assign the result to a list calle
html_tables=soup.find_all('table')
```

4.
```
[49]:
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders 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:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict['Flight No.'].append(flight_number)
            #print(flight_number)
```

5.
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

Exploratory Data Analysis and Determine Training Labels

Calculate the number of launches on each site

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

Calculate the number and occurrence of each orbit

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

Calculate the number and occurence of mission outcome per orbit type

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

Create a landing outcome label from Outcome column

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise

landing_class = np.where(df['Outcome'].isin(bad_outcomes),0,1)
print(landing_class)
```

Exporting data to CSV-file

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

- https://github.com/vangalasandeep/IBMCapstone/blob/master/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Different chart types and dependencies were plotted

- Payload-Mass vs. Flight Number / Launch site vs. Flight Number / Launch site vs. Payload-Mass
- orbit vs. success rate / Flight number vs. orbit / Payload Mass vs. orbit
- Launch success yearly trend

The aim is to find patterns and dependencies in the plots, which can then be used to train the machine learning engine. The different chart types have different advantages for this.

- scatterplots are useful to show relationships between variables
- Bar charts are suitable for comparing the ratio of a variable in discrete classes with one another, if necessary, grouping them as well
- Line plots show the progression of variables over time

- GitHub URL: https://github.com/vangalasandeep/IBMCapstone/blob/master/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first succesful landing outcome in ground pad was acheived
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- 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
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order

- GitHub URL: https://github.com/vangalasandeep/IBMCapstone/blob/master/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- Objects mentioned in the interactive Map:
- Every launch site is indicated on a map.
  - Latitude and longitude coordinates are used to mark NASA Johnson Space Center as a point on the map, and it is also given a text label.
  - Each X-space launch location is represented as a point, given a text label, and its latitude and longitude are shown on a map. Consequently, indicating the location and distance from the equator
- The launch outcome for each start at each launch point is illustrated and color-coded.
  - Green markers indicate successful launches, while red markers indicate unsuccessful ones, providing a fast snapshot of the success percentage each launch site.
- Measure the separations between a launch location and its environs.
  - Insertion of connecting lines from a chosen launch point (CCAFS SLC-40) to the coastline and close-by infrastructure like the closest railroad, road, or city. Establish the respective distances and enter them.

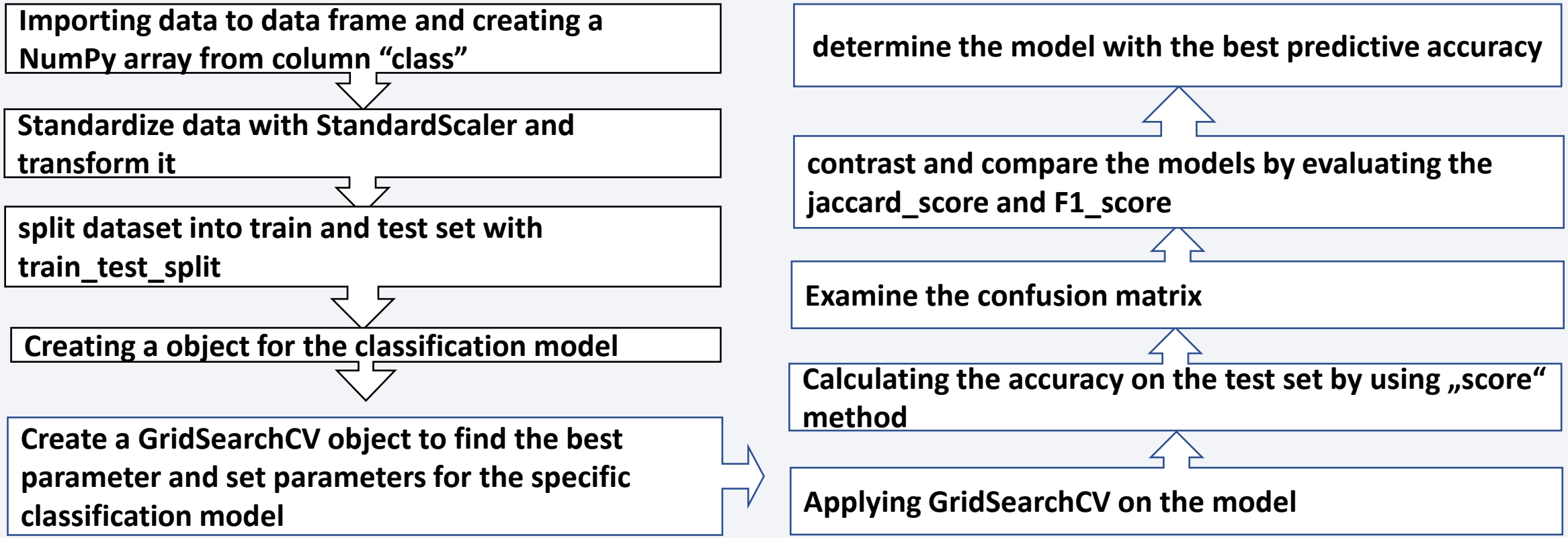
- GitHub URL: https://github.com/vangalasandeep/IBMCapstone/blob/master/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- List of all launch sites in a drop-down menu
  - Adding a dropdown menu to allow users to choose potential launch sites
- Making a pie chart showing successful and unsuccessful launches (for all sites combined or for a single site)
  - A pie chart for all sites was included. Depending on choices, the overall number of successful launches against failed launches for the sites is aggregated or specific to a single site.
- Include a payload range slider.
  - Add a Range Slider to Payload
- Selection Payload mass vs. success rate is plotted in a scatterplot that is customizable for the various booster models.
  - Include a scatterplot to demonstrate the relationship between payload and launch success or failure rate.

GitHub URL: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_3/lab_theia_plotly_dash.md.html

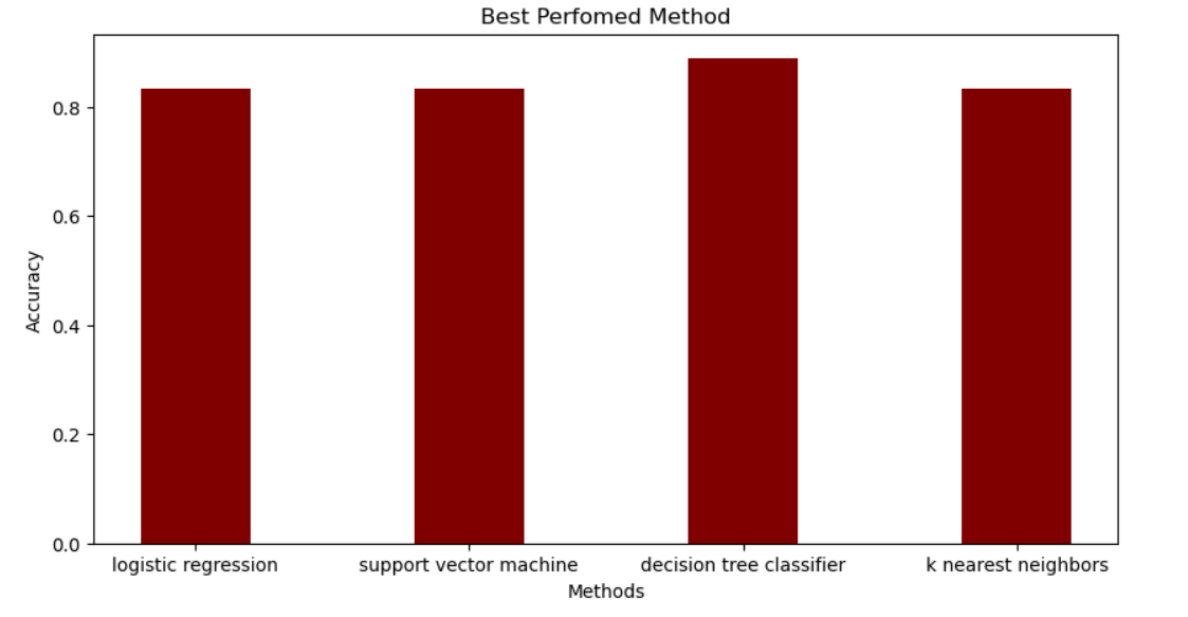
Predictive Analysis (Classification)

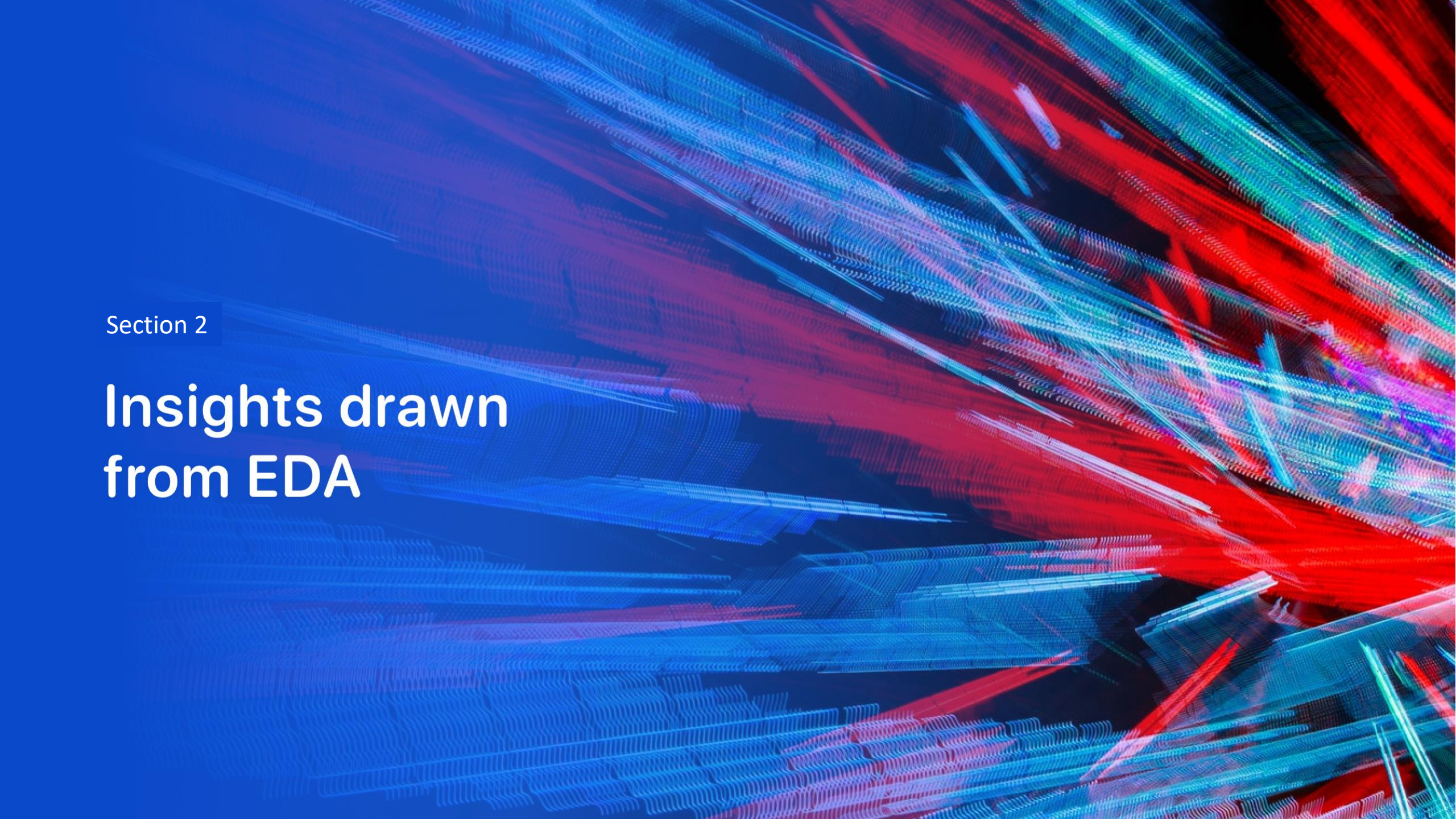


GitHUB URL:
https://github.com/vangalasandeep/IBMCapstone/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

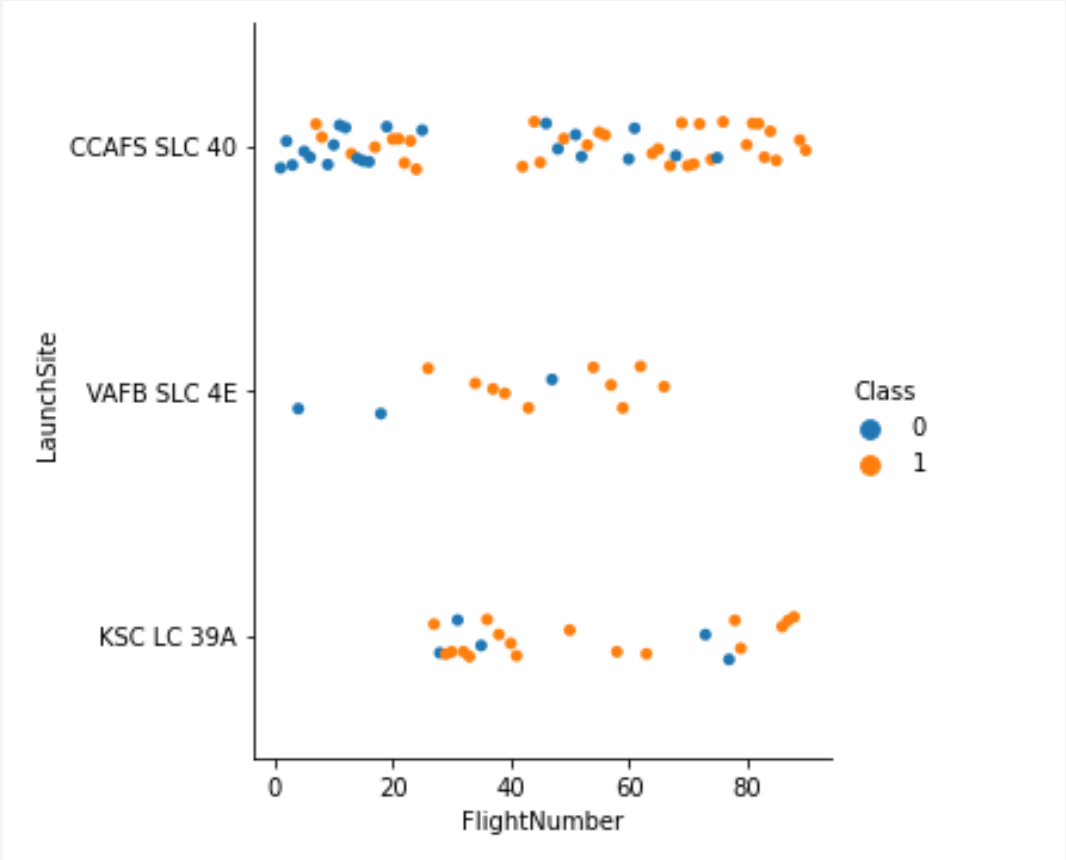
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results




Section 2
Insights drawn from EDA

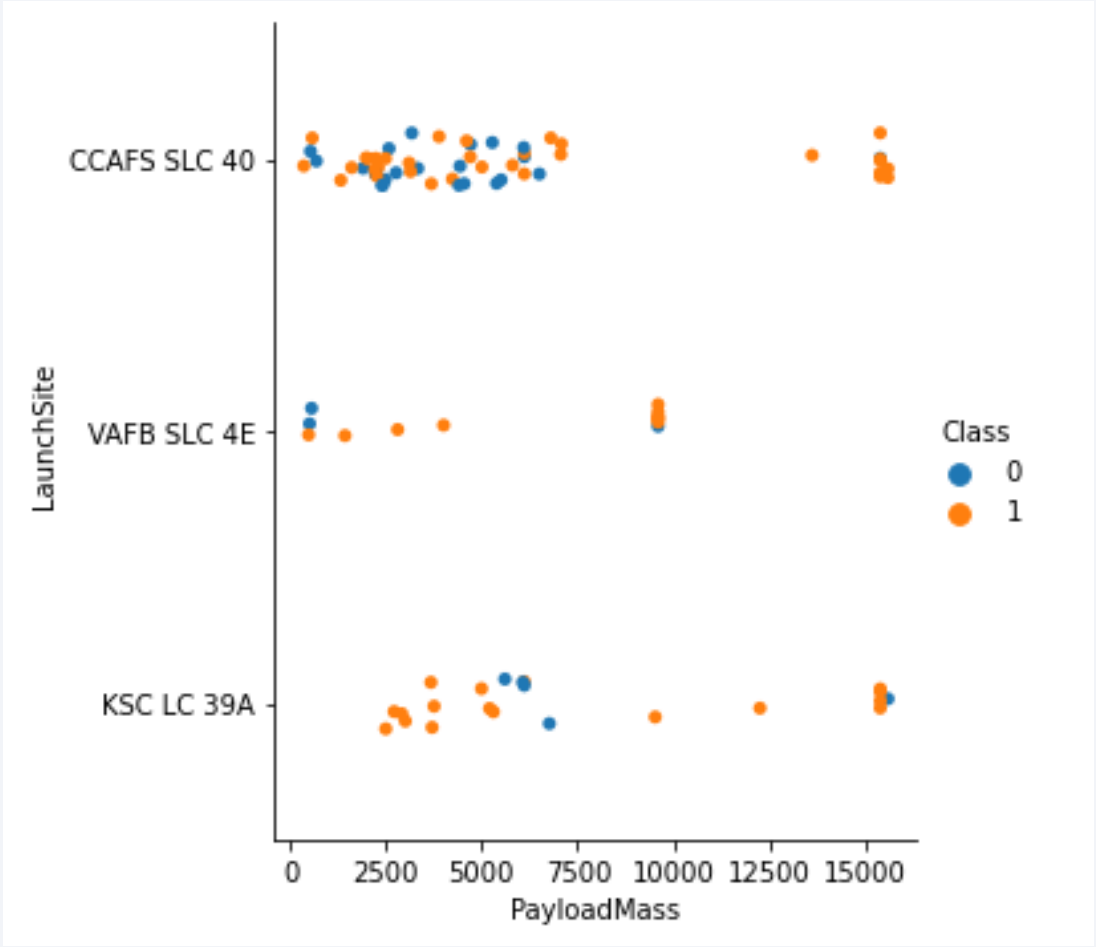
Flight Number vs. Launch Site

- The CCAFS SLC 40 launch site hosted the initial launch attempts.
- The chances of success grew as the initial attempts developed
- VAFB SLC 4E and KSC LC 39A have the highest success rates for starting positions.



Payload vs. Launch Site

- For a large payload, every launch site displays a high success rate.
- The VAFB SLC 4E launch site appears inadequate for launches with heavy payloads.

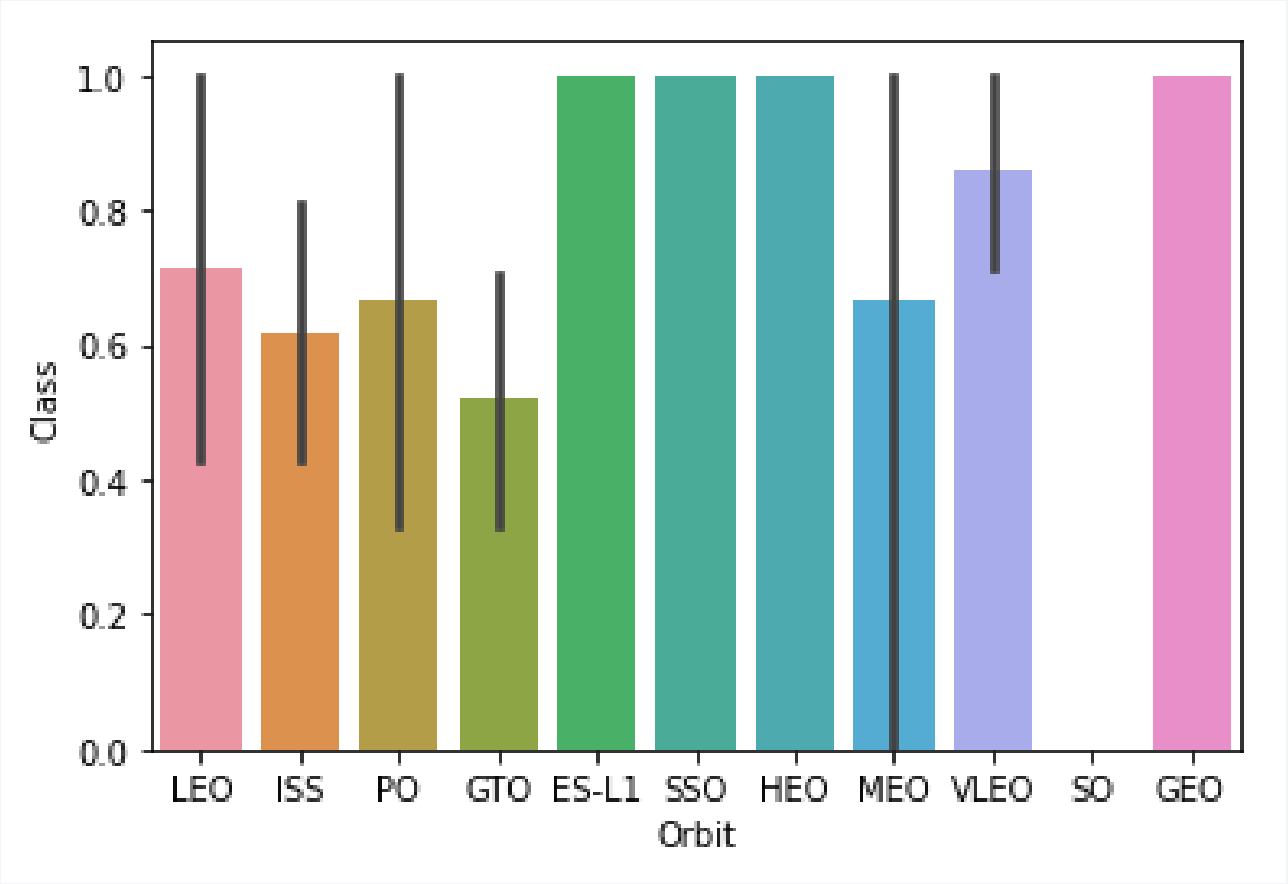


Success Rate vs. Orbit Type

The orbits ES-L1, SSO, HEO, VLEO, and GEO have success rates of more than 80%.

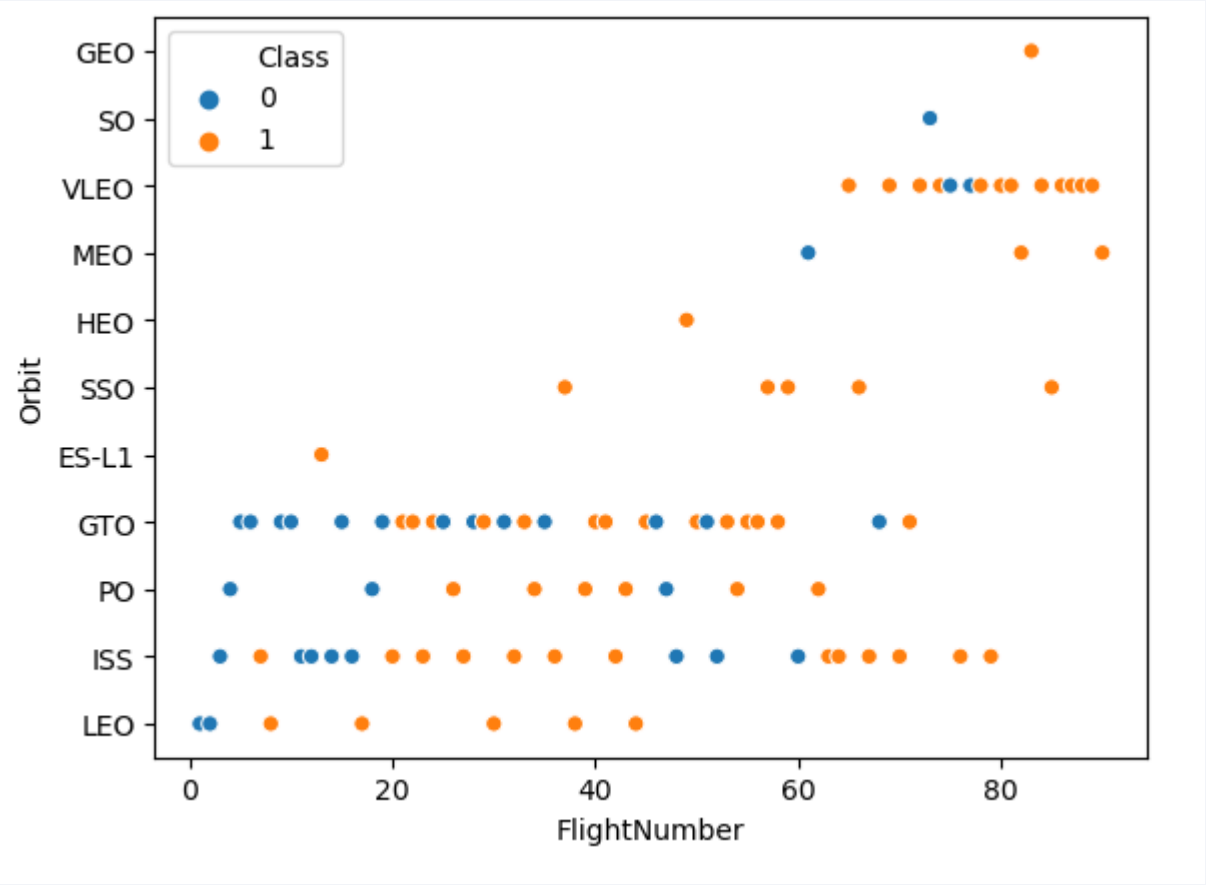
LEO, ISS, PO, GTO, and MEO orbits have success rates between 50% - 80%.

Without successful launches, orbit cannot exist.



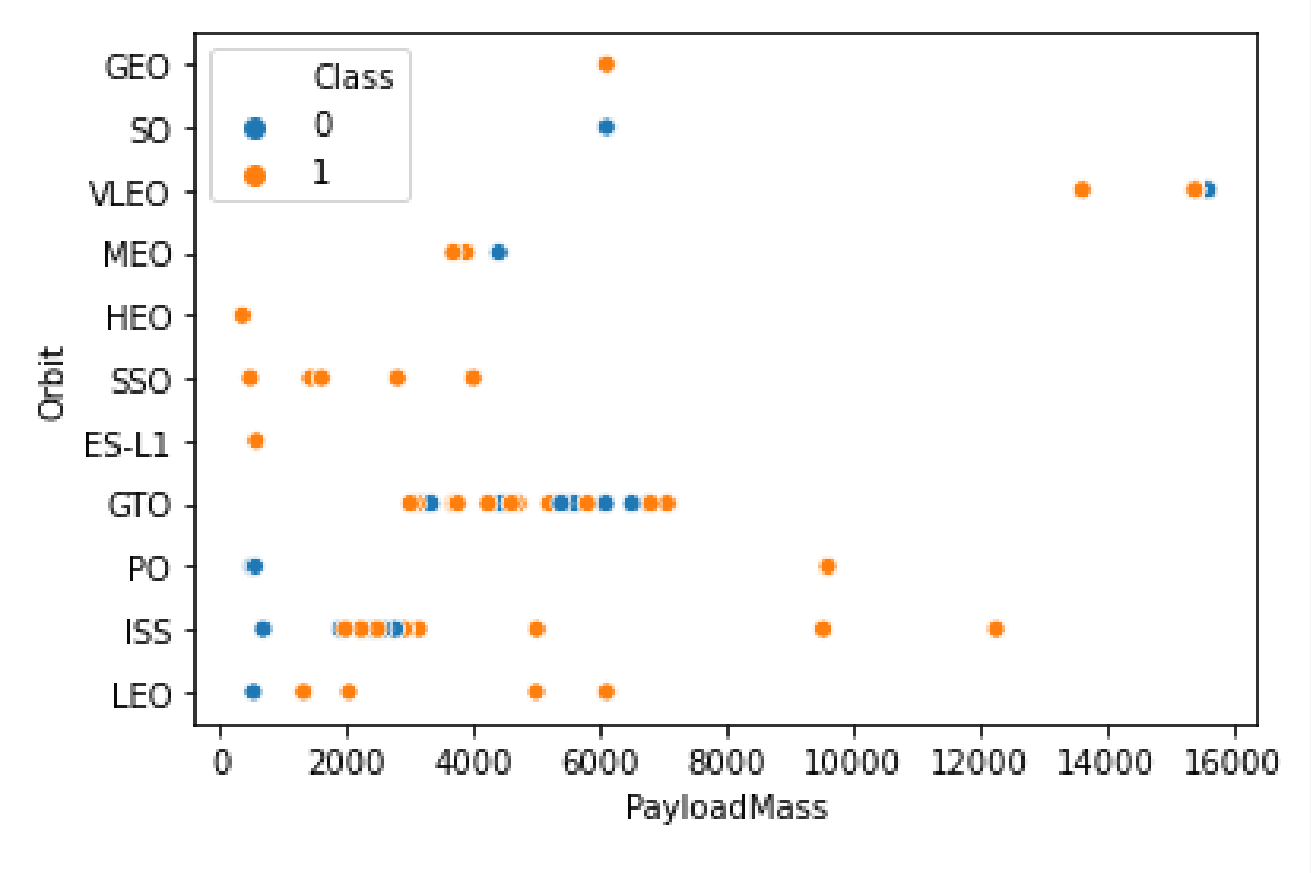
Flight Number vs. Orbit Type

- The majority of launches, including the initial ones, were in low orbit up to the GTO, where the likelihood of success rose as the number of launches rise.
- The majority of the orbital VLEO testing were productive.



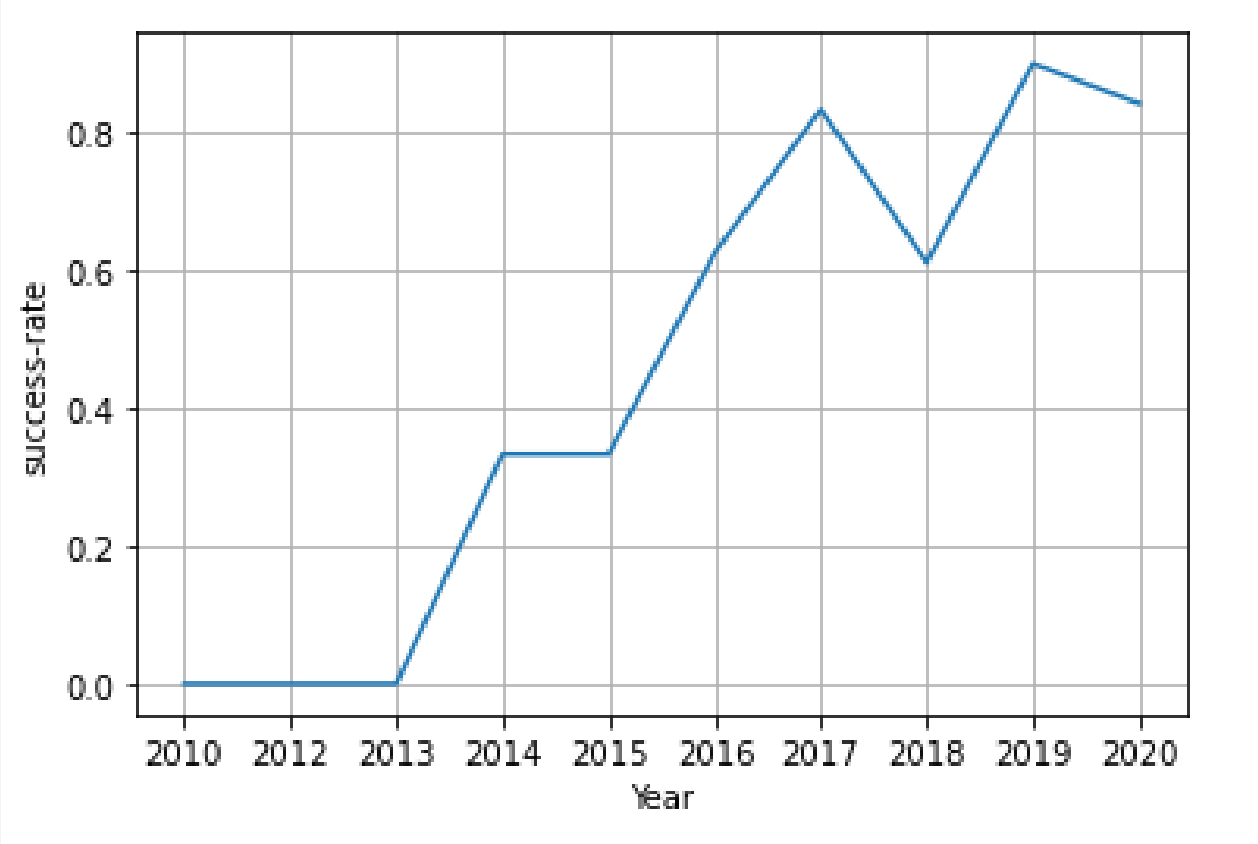
Payload vs. Orbit Type

- Apparently, payload and orbit have no underlying relationship.
- Low payloads and low orbits indicate a low success rate, but this could also have been the result of the early launch attempts.
- The most launches occurred in the orbits GTO and ISS, and ISS also included the most diverse array of payloads.



Launch Success Yearly Trend

- There were no successful launch attempts between 2010 and 2013.
- The likelihood of success increased dramatically in the years that followed.
- The most likely year to succeed (more than 90%) was 2019.



All Launch Site Names

- Find the names of the unique launch sites

```
%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

- Query with the keyword "DISTINCT" returns all unique launch sites.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

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

```
%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

- The query result shows a selection of 5 entries in the “launch_site" column which start with "CCA“.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER LIKE 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

- The result returns the total payload of all boosters launched by NASA.

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1%'
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

- Results gives the average payload of the launches with the booster version "F9 v1.1"

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

```
%%sql SELECT MIN("Date")
FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE '%Success%'
```

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

MIN("Date")
01-05-2017

- Query returned the first successful landing on 01/05/2017

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" LIKE 'Success (drone ship)'
       AND
       "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000;
```

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

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

- Result lists the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
%%sql SELECT
    sum(CASE WHEN "Mission_Outcome" LIKE 'Success%' THEN 1 ELSE 0 END) AS 'Success',
    sum(CASE WHEN "Mission_Outcome" LIKE 'Failure%' THEN 1 ELSE 0 END) AS 'Failure'
    from SPACEXTBL
```

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

Success	Failure
100	1

- Results display the total number of successful and failed mission outcomes. It states that the failed are minimal.

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

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

In [32]:
```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

Out[32]:

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

- Query returns the names of the booster versions that carried the maximum payload, a total of 12 booster versions

2015 Launch Records

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

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

In [17]:
```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE DATE LIKE '%2015' AND LANDING_OUTCOME = 'Failure (drone ship)';
```

```
   ibm_db_sa://xkl40378:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
 * sqlite:///my_data1.db
Done.
```

Out[17]:

Booster_Version	Launch_Site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

- For all launches in 2015 that resulted in a failed landing in a drone ship, the combined query gives the names of the booster versions, the launch location, and the landing outcome.

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

Task 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

In [77]:
```
%sql SELECT 'Success' as landing_outcome, Count(*) FROM SPACEXTBL Where LANDING_OUTCOME like 'Succ%'\
and substr(date,7)||substr(date,4,2)||substr(date,1,2) \
between '20100604' and '20170320'\
union \
SELECT 'failure' as landing_outcome, Count(*) FROM SPACEXTBL Where  LANDING_OUTCOME like 'Failu%'\
and substr(date,7)||substr(date,4,2)||substr(date,1,2) \
between '20100604' and '20170320'
```

```
   ibm_db_sa://xkl40378:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
 * sqlite:///my_data1.db
Done.
```

Out[77]:

landing_outcome	Count(*)
Success	8
failure	7

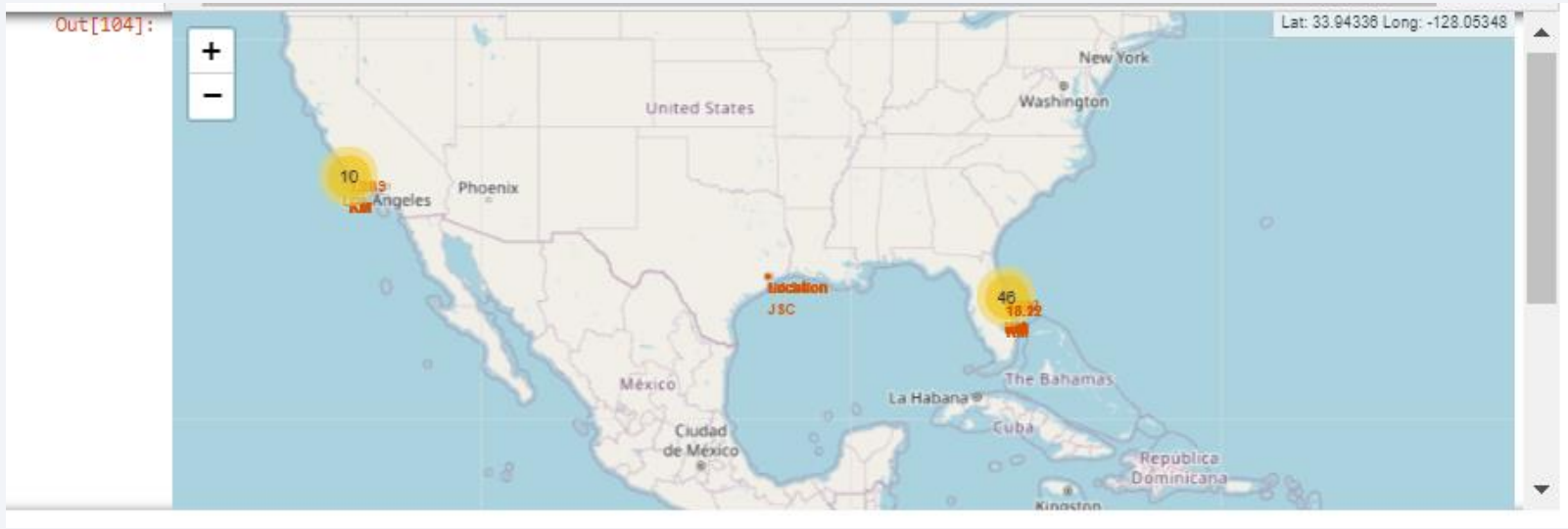
- There are 8 successful mission and 7 failed mission for the specified duration.

Section 3

Launch Sites Proximities Analysis

Interactive launch sites Map with Folium

- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map



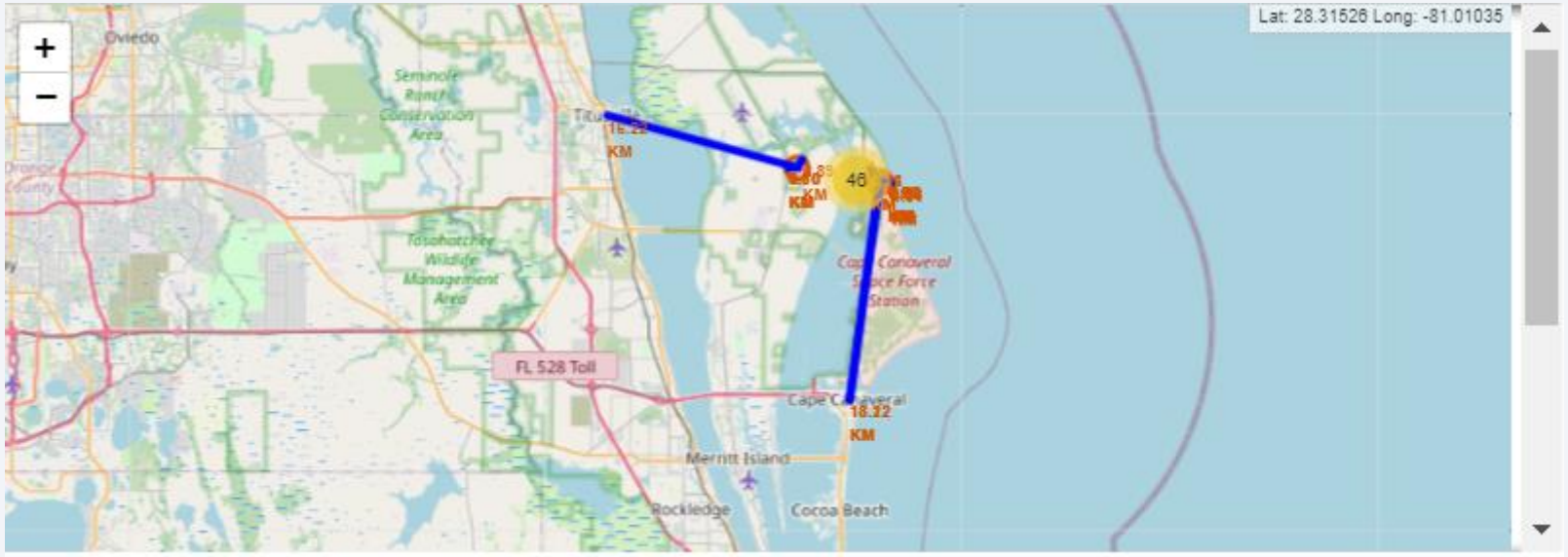
- There a 3 launch sites across North America located in 3 different states.
  - California, Texas and Florida. Among the 3 launch sites, Florida Launch site has more launches compared to California & Texas.

Color-labeled launch outcomes with Folium

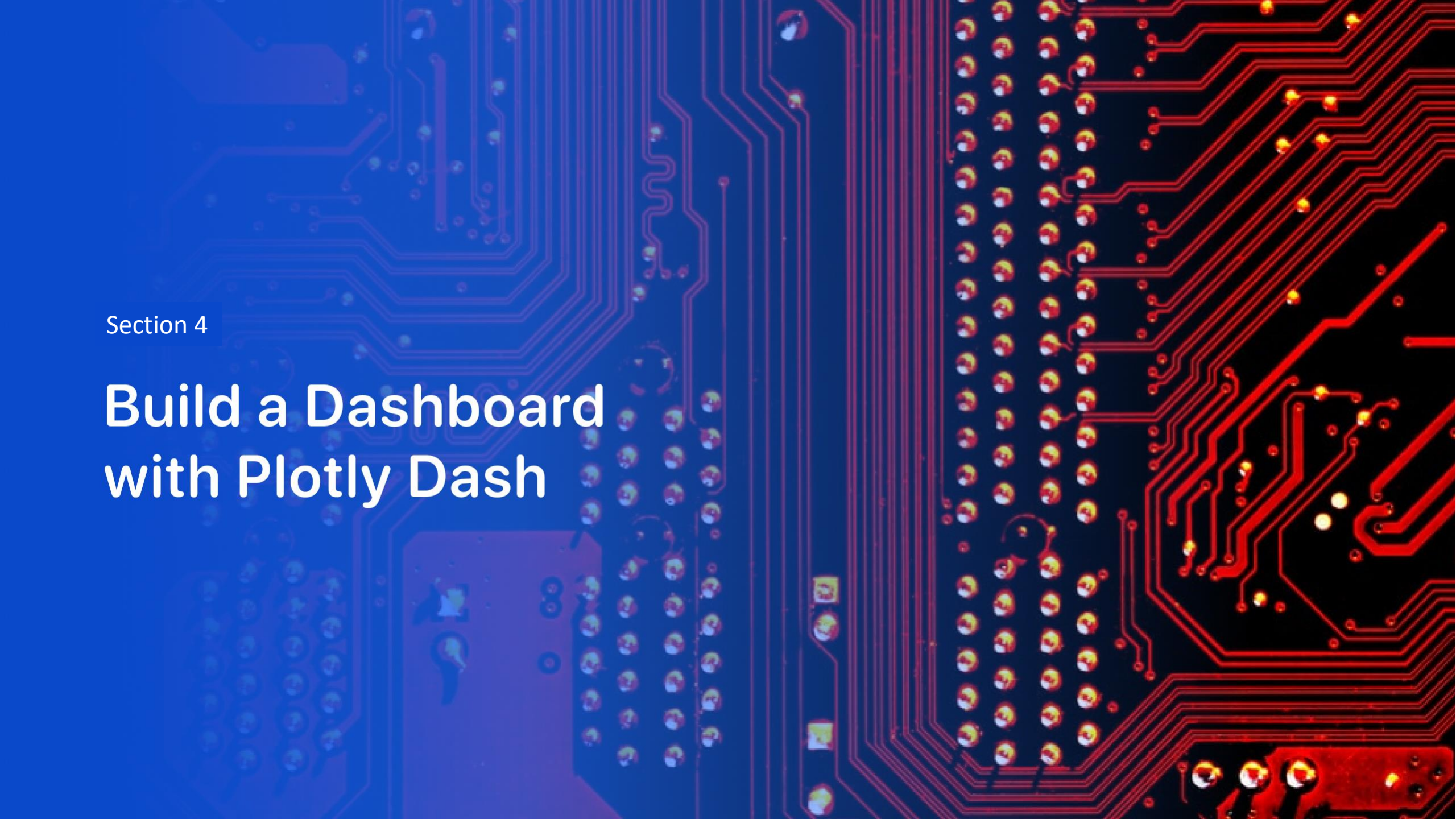
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map

- California Launch site has 4 successful launches and 6 failed launches.

Launch site to its proximities with Folium

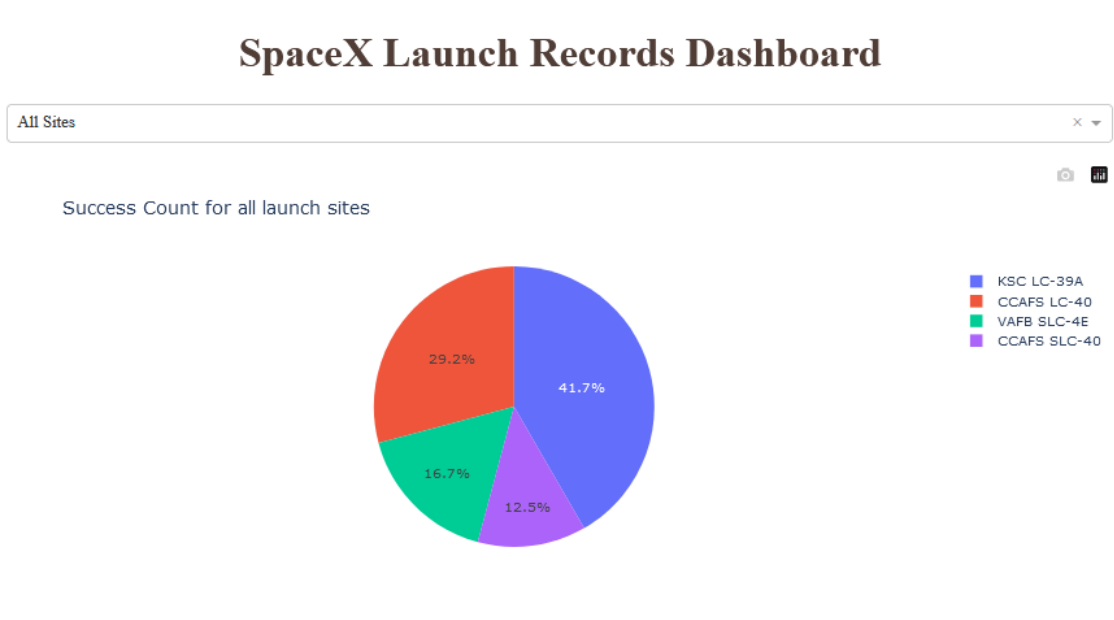


- Proximity for one of Florida Launch site is 0.9 km from coastline


Section 4
Build a Dashboard with Plotly Dash

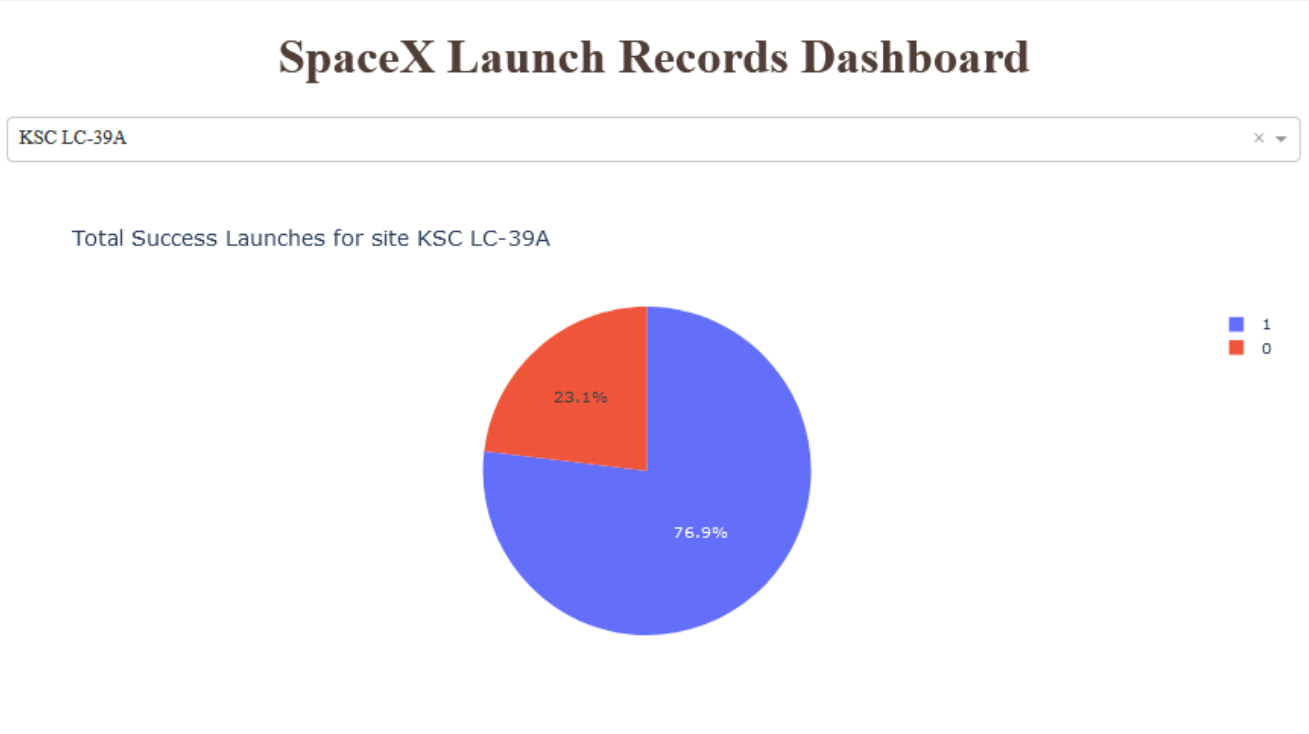
Start success overview for all start sites

- Show the screenshot of launch success count for all sites, in a pie-chart



- The graphic shows that most successful launch attempts took place at the launch site "KSC LC-39A". In total, more than 2/3 of the total successful missions ran at the two launch sites "KSC LC-39A" and "CCAFS LC-40".

Total Success launches for Site KSC LC-39A

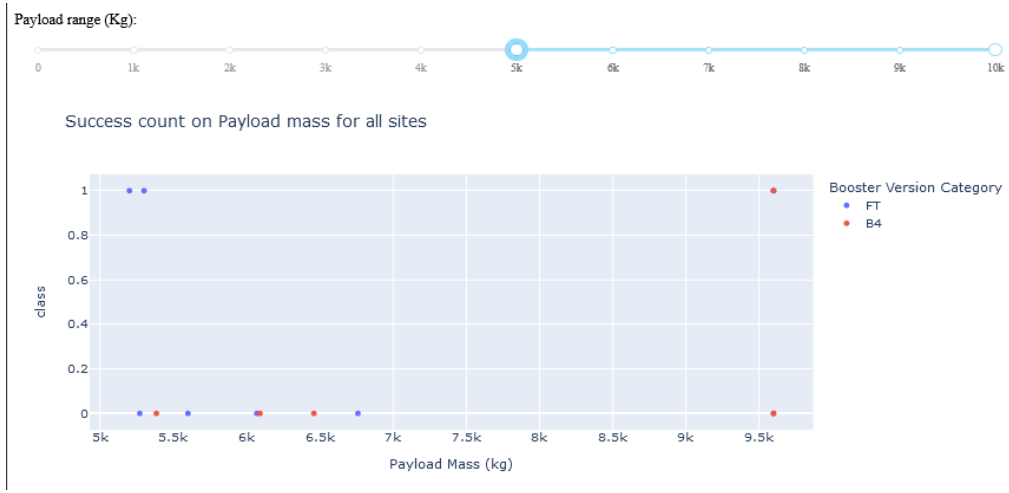


- 76.9% of the missions carried out at the "KSC LC-39A" launch site were successful, while 23.1% were unsuccessful.

Launch success at different payload masses

- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.



Higher range of payload mass

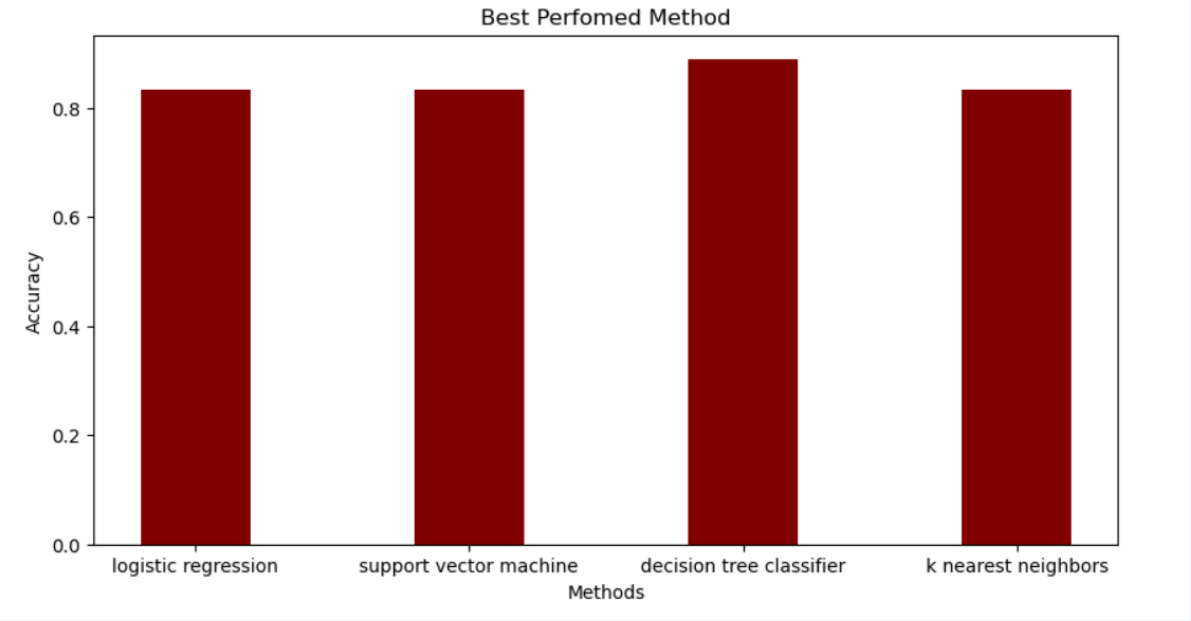
Lower range of payload mass

Section 5

Predictive Analysis (Classification)

Classification Accuracy

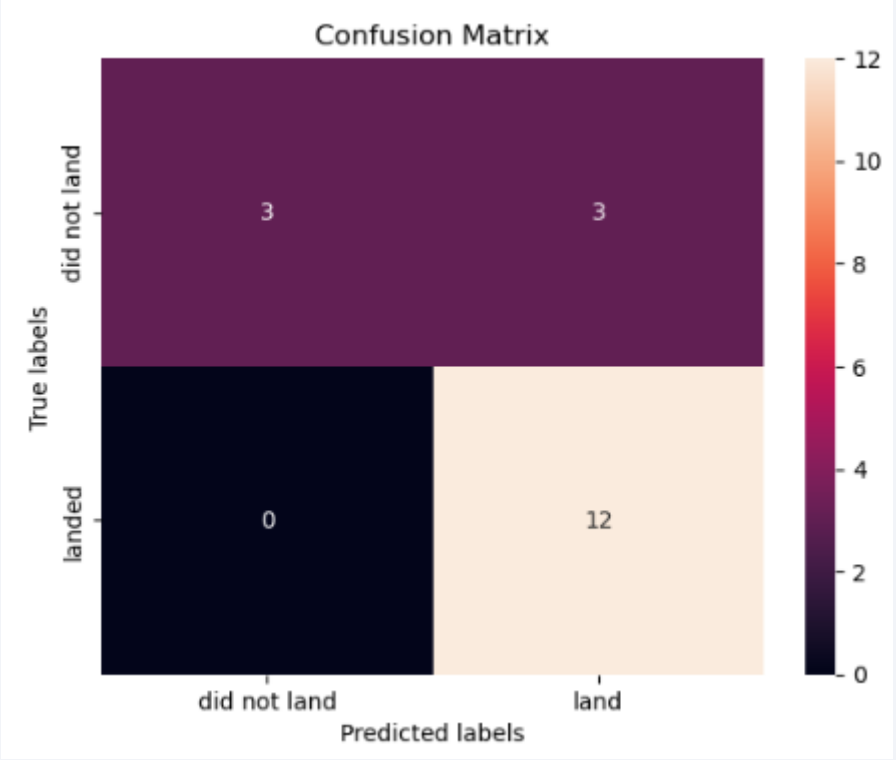
- Visualize the built model accuracy for all built classification models, in a bar chart



- Decision Tress has the highest classification accuracy

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation
- 12 True positives
- Zero True Negatives
- False positives & false negatives are 3



Conclusions

- It is observed that Decision tree has the highest classification accuracy with accuracy score of 88.8%.
- There are total of 99 launches in the data set provided
- Florida launch site has more launches and successful launches compared to the other two launch sites.
- Total payload mass of 45,596 Kg's carried by boosters launched by NASA.
- 2010 to 2013 saw lower success rates (perhaps because they were also used to adjust and gain experience), but from 2013 to 2020, success rates steadily increased.

Appendix

Github url for Capstone Project:

https://github.com/vangalasandeep/IBMCapstone

Thank you!