IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nhlamuselo Decesion Ngoveni
08 June 2023

Outline

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

Executive Summary

The aim of the project is to predict the successful landing for the Falcon 9. SpaceX was used as source of data including
Wikipedia through web scraping. The wrangling of data was done and EDA was done to mark target features. Insights were carried out through the use of visualisation and Maps. A dashboard was created and machine learning models were experimented to find out landing results.

Introduction

The purpose of the project is to find out the successful landing of Falcon 9. SpaceX launches their Falcon 9 at a cheaper price compare to other companies. Hence, if we can find out that it can land on first stage then we will determine the price. As such, the information which led to landing can be used by other companies to reduce cost.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Wikipedia through web scraping
  - SpaceX API
- 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

The data was collected using two different method, SpaceX Rest API and Wikipedia (Web Scraping). The data collected from web scraping was converted into data frames

Data Collection – SpaceX API

1. The was requested in .json format

```
[9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_space
```

2. The requested data was normalised to data frame

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

3. The normalised data was passed through dictionary and filtered to Falcon 9 only

```
[80]: # Hint data['BoosterVersion']!='Falcon 1'
      data_falcon9 = data[data.BoosterVersion == 'Falcon 9']
      data_falcon9
```

4. Then Falcon 9 data was saved to CSV

We can now export it to a CSV for the next section,but to make the answers consistent, in the next lab we will provide data in a pre-selected date range.

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


GitHub

Data Collection - Scraping

1. Data is requested from Wikipedia page through URL link that contains data using beautiful soap

```
[11]: # Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all("table")
print(html_tables)
```

2. Add column names

```
[14]: print(column_names)
```

3. Create a data frame by passing the launch HTML tables

```
[29]: headings = []
for key,values in dict(launch_dict).items():
    if key not in headings:
        headings.append(key)
    if values is None:
        del launch_dict[key]

def pad_dict_list(dict_list, padel):
    lmax = 0
    for lname in dict_list.keys():
        lmax = max(lmax, len(dict_list[lname]))
    for lname in dict_list.keys():
        ll = len(dict_list[lname])
        if  ll < lmax:
            dict_list[lname] += [padel] * (lmax - ll)
    return dict_list

pad_dict_list(launch_dict,0)

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

4. Save data to csv

We can now export it to a CSV for the next section, but to make the answers consistent and in case you have difficulties finishing this lab.

Following labs will be using a provided dataset to make each lab independent.

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

[GitHum](#)

Data Wrangling

GitHub

1. Import libraries and load space X dataset, from last section.

```
[3]: from js import fetch
     import io

     URL = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv'
     resp = await fetch(URL)
     dataset_part_1_csv = io.BytesIO((await resp.arrayBuffer()).to_py())
```

Load Space X dataset, from last section.

```
[4]: df=pd.read_csv(dataset_part_1_csv)
     df.head(10)
```

2. Calculate the number of launches on each side

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

3. Calculate the number and occurrence of mission outcome per orbit type

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

4. Create a landing outcome label from outcome columns to determine success rate and save to csv

```
[20]: df["Class"].mean()

[20]: 0.6666666666666666
```

We can now export it to a CSV for the next section,but to make the answers consistent, in the next lab we will provide data in a pre-selected date range.

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

EDA with Data Visualization

- Scatter plot - to determine the relationship between the flight number, launch sites, play load,and orbit.
- Bar Chart - To show large and small changes orbit success through mean vs orbit
- Line graph - to track changes of success rate trend overtime

[GitHub]

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)
- List the date when the first succesful landing outcome in ground pad was acheived.
- 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
- Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

[GitHub]

Build an Interactive Map with Folium

1: Mark all launch sites on a map

2: Mark the success/failed launches for each site on the map

3: Calculate the distances between a launch site to its proximities

You can draw a line betwee a launch site to its closest city, railway, highway, etc. to find the their coordinates on the map first

GitHub

Build a Dashboard with Plotly Dash

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

Predictive Analysis (Classification)

- Through Analysing all models, tree has the highest percentage accuracy of 88%.

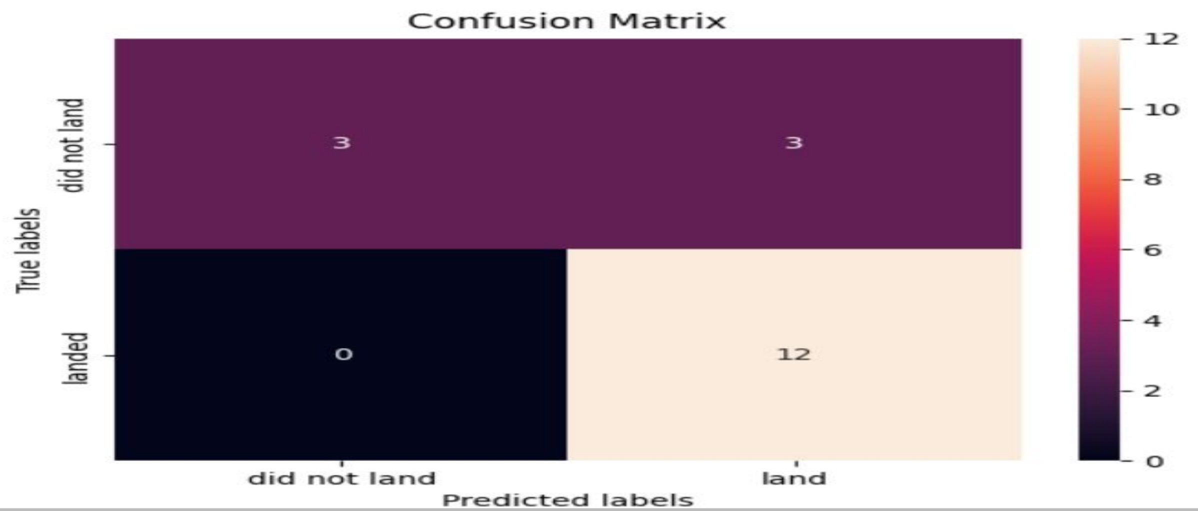
```
[30]: print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
      print("accuracy :",tree_cv.best_score_)

tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'random'}
accuracy : 0.8892857142857145
```

It went on and achieve score of 83%

```
[31]: tree_cv.score(X_test, Y_test)
[31]: 0.8333333333333334
```

We can plot the confusion matrix

```
[32]: yhat = tree_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```



Results

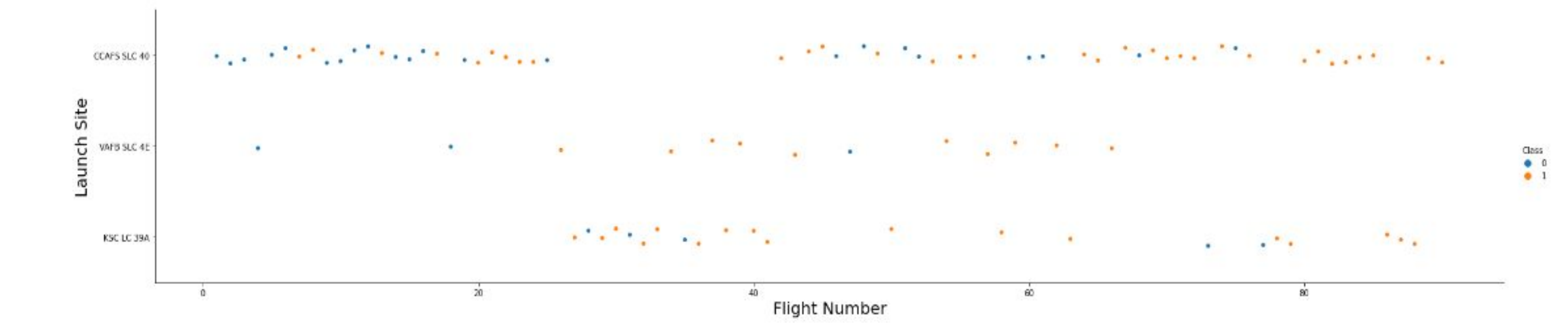
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

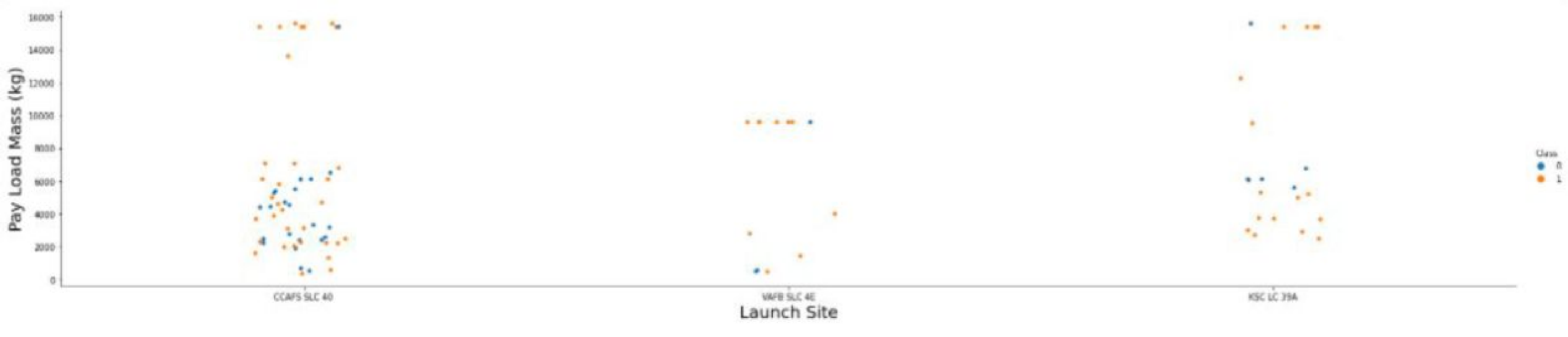
- Scatter plot of Flight Number vs. Launch Site



- From the Scatter plot, CCAFS-SLC-40 had the highest launches whereas, VAFB-SLC-4E site had the lowest..

Payload vs. Launch Site

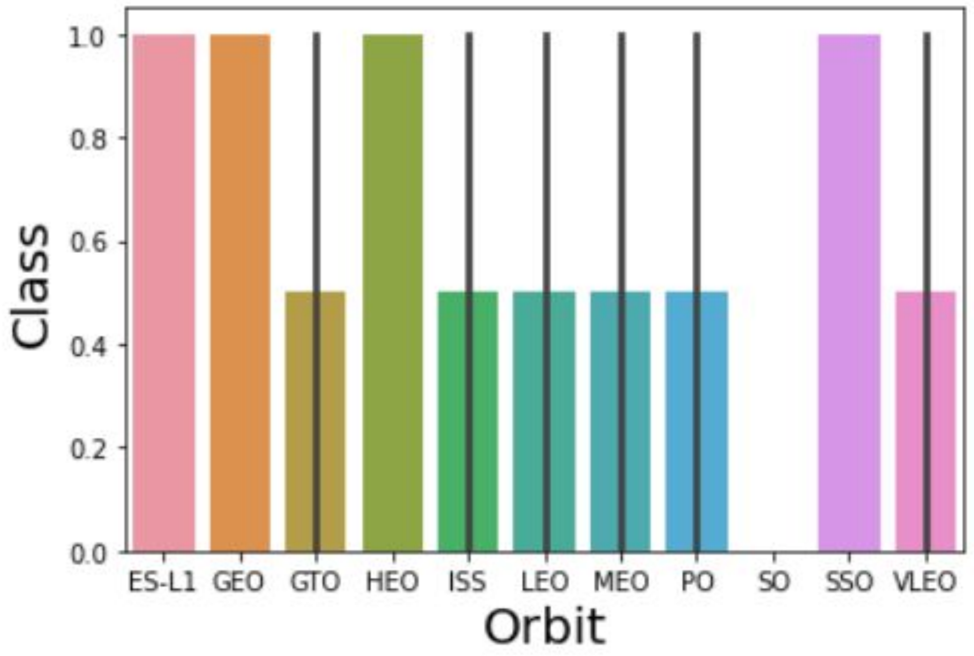
- Scatter plot of Payload vs. Launch Site



- From the Scatter plot, CCAFS-SLC-40 had many precise lower Payload Launches and fewer precise high Payload Lauches whereas, VAFB-SLC-4E has average precise Pay Load Mass and Lower not precise Pay Load.

Success Rate vs. Orbit Type

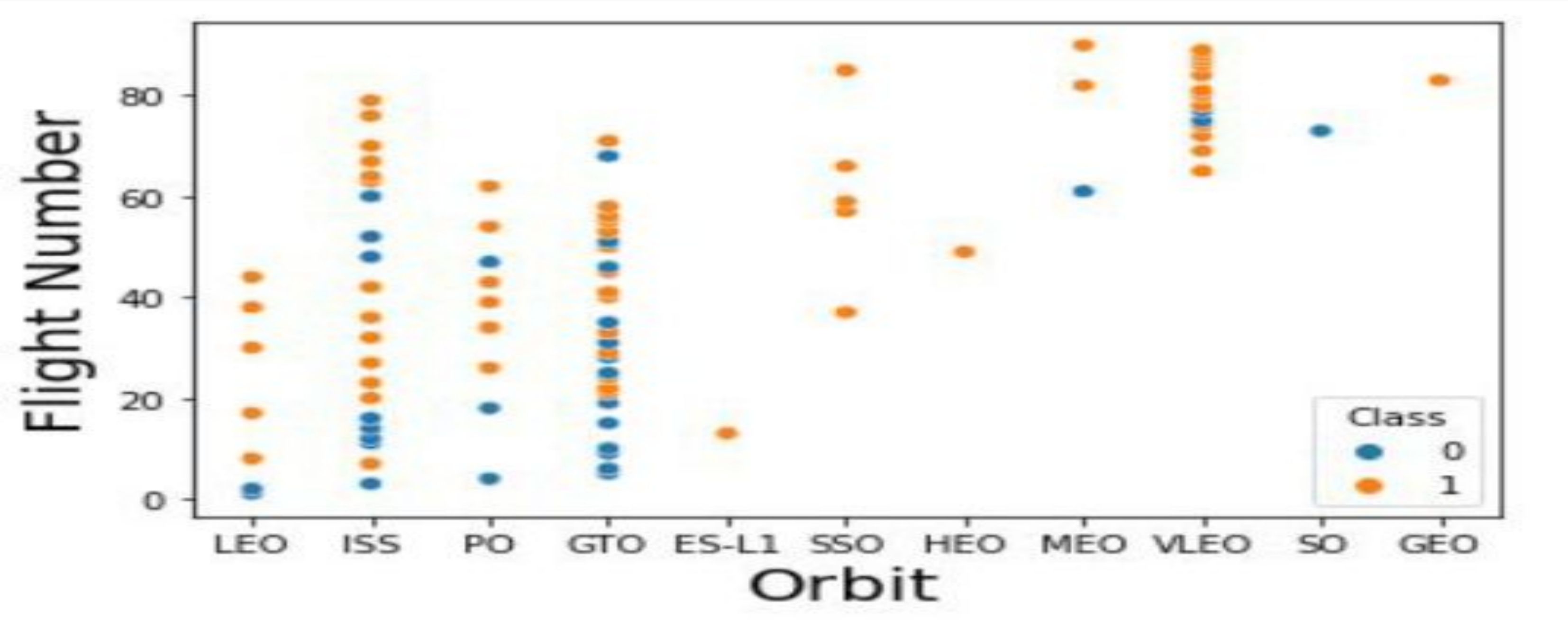
- Bar chart for the success rate of each orbit type



- From the bar chart, ESL1, GEO and SSO show high success rate of orbit type.

Flight Number vs. Orbit Type

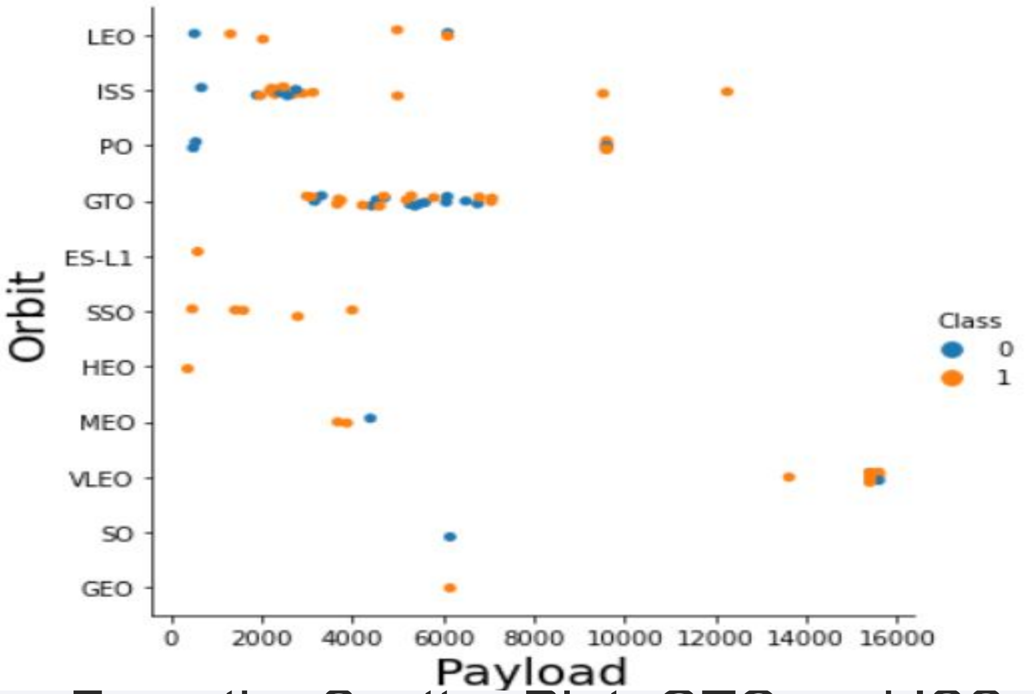
- Scatter point of Flight number vs. Orbit type



- From the scatter plot, ISS, GT, PO and VLEO has the highest flight number that complete the orbit. SSO and VLEO Display high success rate.

Payload vs. Orbit Type

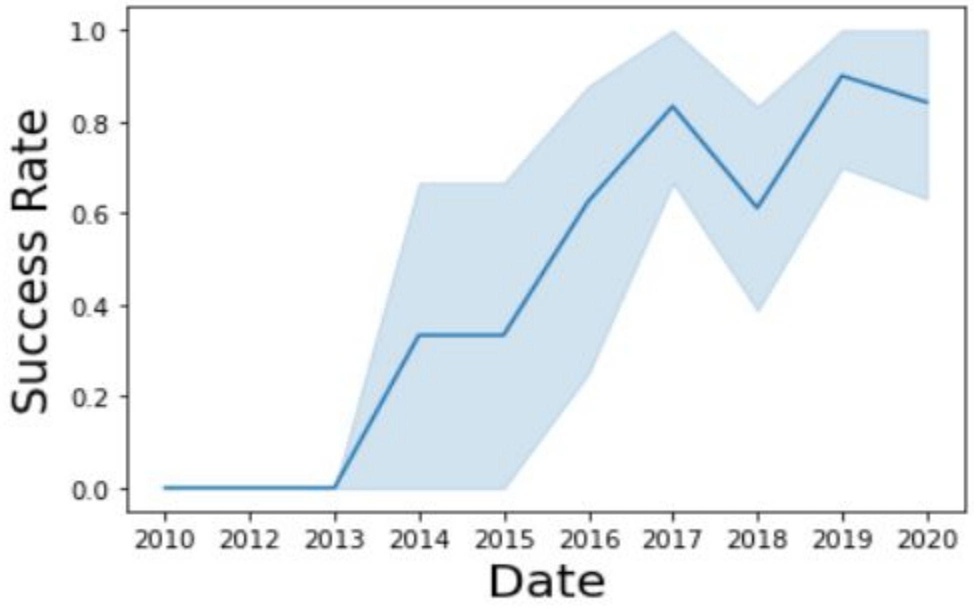
- Scatter point of payload vs. orbit type



- From the Scatter Plot, GTO and ISS shows most precise high payload.

Launch Success Yearly Trend

- Line chart of yearly average success rate



-

- From the line chart, the average launching success rate increased over time from 2010 t0 2020

All Launch Site Names

- Names of the unique launch sites

Out[39]:

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

the performed query shows the unique that performed the launch according to SpaceX data.

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

In [40]:
```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

Out[40]:

time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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)
07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- the performed query shows 5 records where launch sites begin with `CCA` according to SpaceX data.

Total Payload Mass

- Calculations of the total payload carried by boosters from NASA

```
In [41]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

Out[41]:

1
45596

- the performed query shows Calculations of the total payload carried by boosters from NASA according to SpaceX data.

Average Payload Mass by F9 v1.1

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

```
In [42]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

Out[42]:

1
2928.400000

- the performed query shows Calculations of the average payload mass carried by booster version F9 v1.1 according to SpaceX data.

First Successful Ground Landing Date

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

In [43]:
```
%sql select min(DATE) from SPACEXTBL where Landing__Outcome = 'Success (ground pad)'
```

Out[43]:

1
2015-12-22

-
-
-
- Present your query result with a short explanation here

Successful Drone Ship Landing with Payload between 4000 and 6000

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

In [44]:
```
%sql select BOOSTER_VERSION from SPACEXTBL where Landing__Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_
```

Out[44]:

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

Total Number of Successful and Failure Mission Outcomes

- Calculations of the total number of successful and failure mission outcomes

```
In [45]: %sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure

Out[45]: 1
         100
```

- the performed query shows Calculations of the total number of successful and failure mission outcomes according to SpaceX data.

Boosters Carried Maximum Payload

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

```
In [46]:  %sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTB
```

Out[46]:

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

- the performed query shows List of the names of the booster which have carried the maximum payload mass according to SpaceX data.

2015 Launch Records

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

In [182...

```
%%sql
select booster_version,launch_site,landing__outcome,DATE from space where landing__outcome LIKE '%drone%' AND DAT
```

Out[182...

booster_version	launch_site	landing__outcome	DATE
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)	10-01-2015
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)	14-04-2015
F9 v1.1 B1018	CCAFS LC-40	Precluded (drone ship)	28-06-2015

- the performed query shows List of the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 according to SpaceX data.

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

- Ranked 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 [48]:  %sql select * from SPACEXTBL where Landing__Outcome like 'Success%' and (DATE between '2010-06-04' and '2017-03-2
```

Out[48]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__o
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success
2016-07-18	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success

Section 3

Launch Sites Proximities Analysis

Launch Sites Locations Analysis with Folium

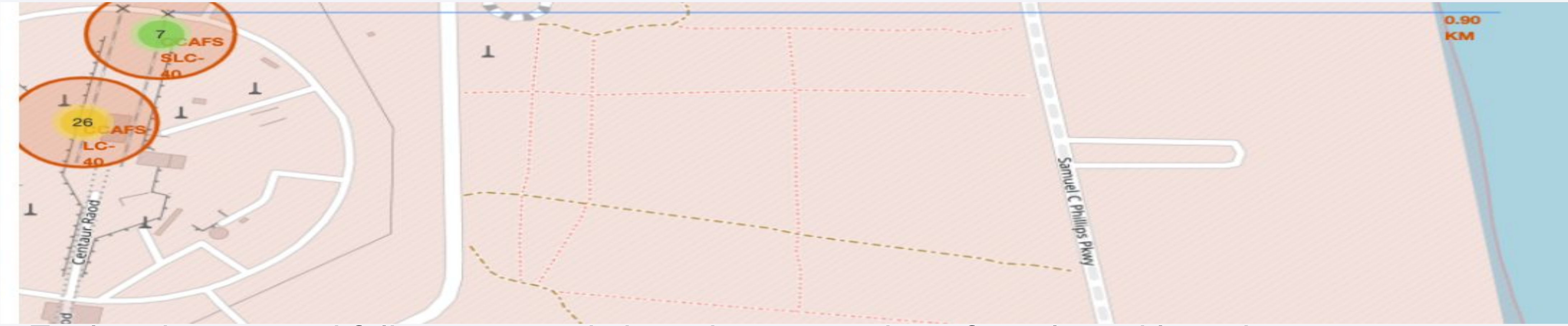
- Explored generated folium map that make a proper screenshot to include all launch sites' location markers on a global map

- The screenshot gives us an understanding of lanches success and failures from different sites

Launch Sites Locations Analysis with Folium

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

Launch Sites Locations Analysis with Folium



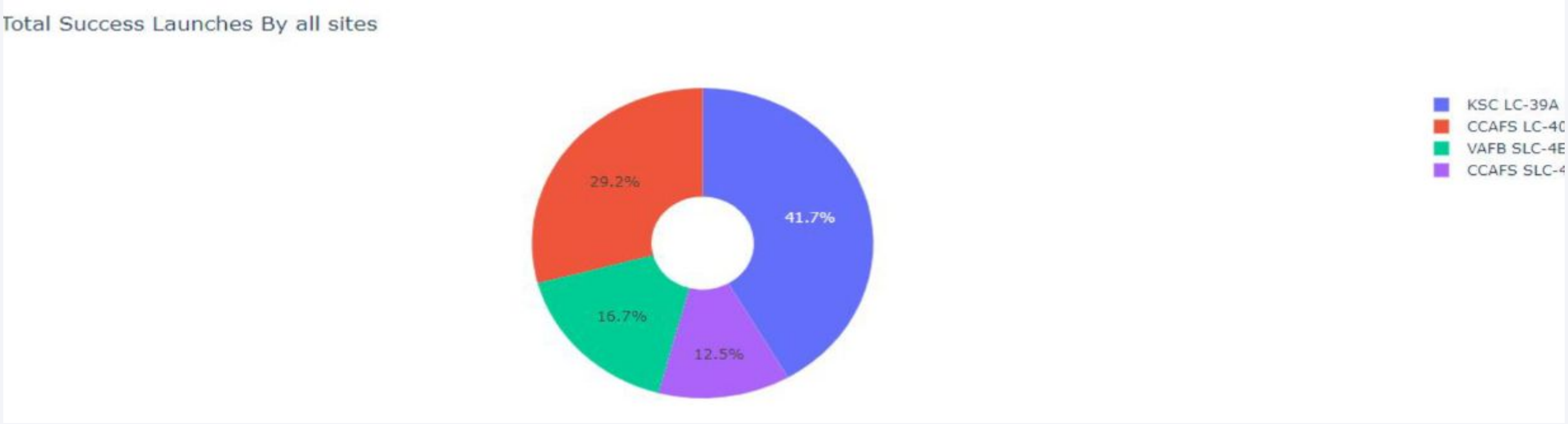
- Explored generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- By drawing a line between a launch site to its closest city, railway, highway, etc. to we can find the their coordinates on the map first

Section 4

Build a Dashboard with Plotly Dash

Build a Dashboard Application with Plotly Dash

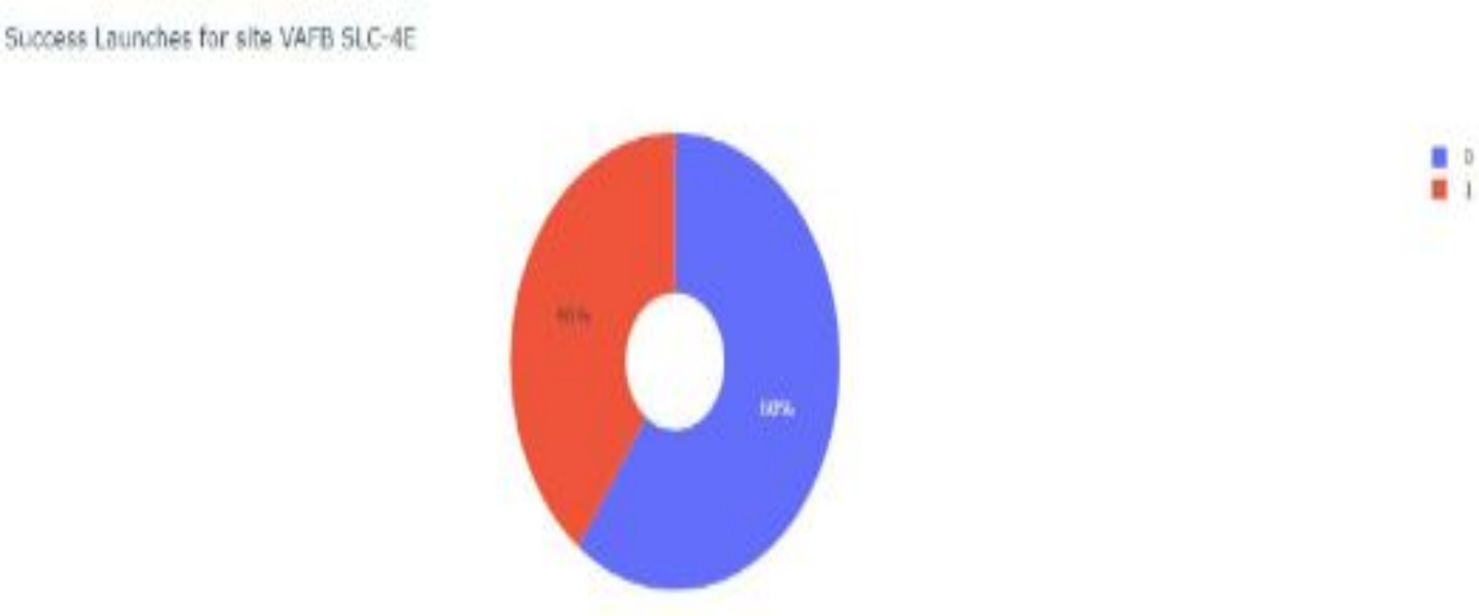
- The screenshot of launch success count for all sites, in a piechart



The pie chart shows launch success count for all sites.KSC LC-39A has the highest count.

Build a Dashboard Application with Plotly Dash

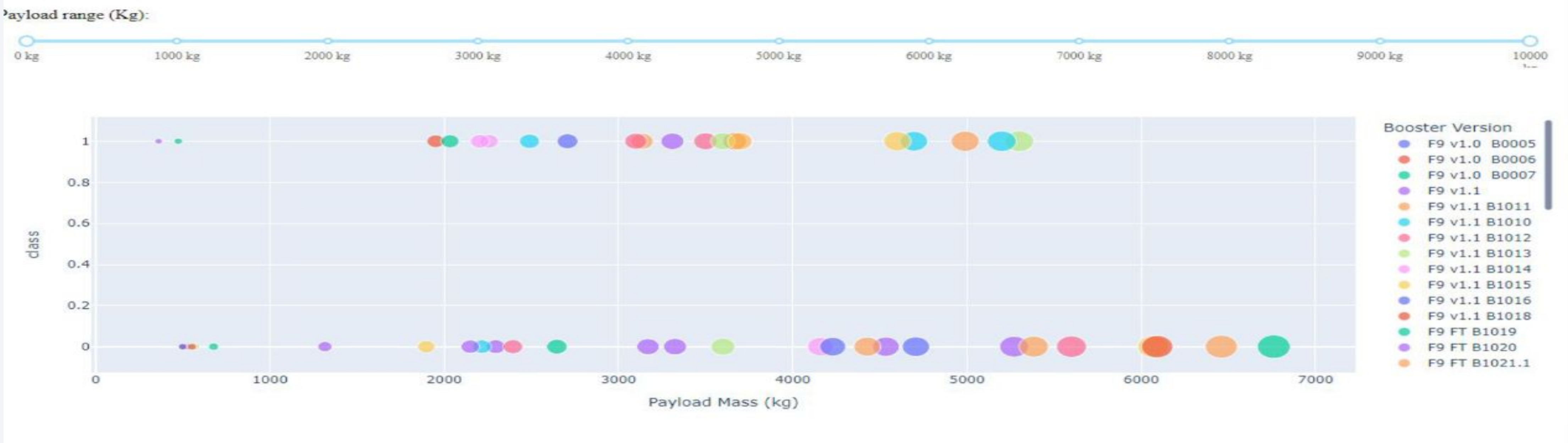
- The screenshot of the piechart for the launch site with highest launch success ratio



- The pie chart shows launchsite with highest launch success ratio

Build a Dashboard Application with Plotly Dash

- The screenshot show the Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider



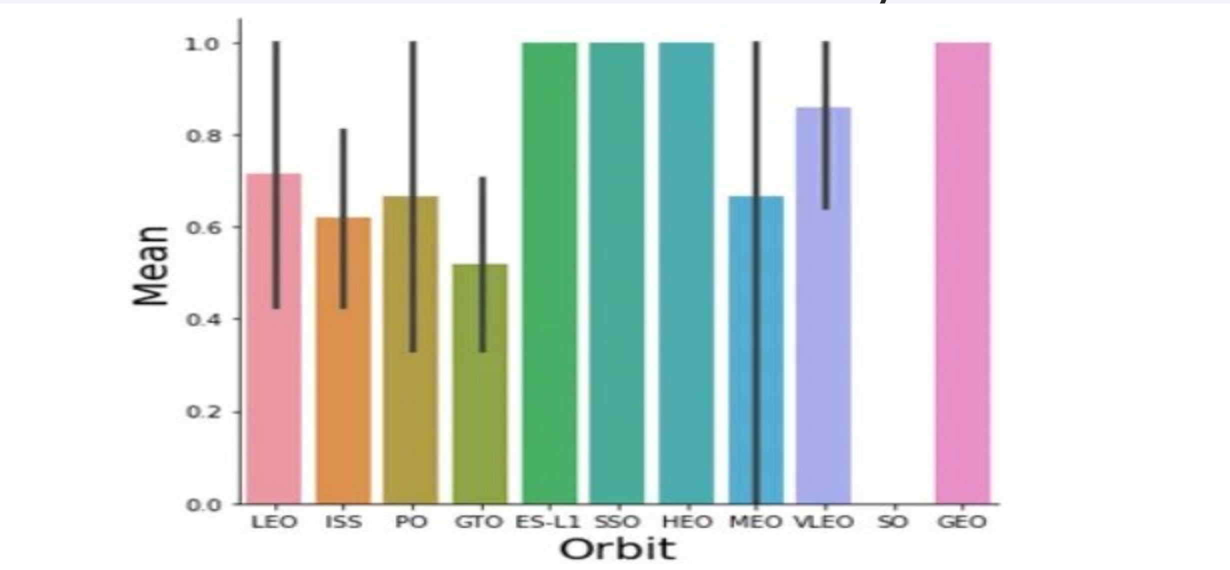
- The important elements and findings that can be shown in this for variables such as which payload range or booster version have the largest success rate is that, using the slider we can determine sites that were successful and those that failed on booster version.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

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



- The model that works the best is by perform Exploratory Data Analysis

Confusion Matrix

- The model allows to predict if the Falcon 9 first stage will land successfully by performing exploratory data analysis.

Conclusions

Through different data analysis that has been done through different models, companies can be able to find the best way that works for the them in other to reduce cost. The performed testing allows them to make an informed decision to save money spent rocket launches.

Appendix

- [Github]

Thank you!