Winning Space Race with Data Science

Mohammed Yassine Labib
12/09/2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with Data Visualization
  - EDA with SQL
  - Using Folium to build an interactive map
  - Using Plotly Dash to build the Dashboard
  - Building the predictive models (classification)
- Summary of all results
  - EDA results
  - Interactive analytics
  - Predictive analysis

Introduction

- Project background and context

In this project we had to predict if the SpaceX's Falcon 9 first stage 'boosters landing' will be successful. SpaceX advertises Falcon 9 rocket launches on its website, with a very low cost, 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.

- Problems to find answers to

Along this data science journey, data had to be collected from various sources and improved in terms of quality by performing data wrangling. The exploration of the processed data using various techniques such as SQL querying, Python scripting helped in gaining further insights into the data by applying statistical analysis and data visualization to distinguish the key features to study and drill down into finer levels of detail by splitting the data into groups defined by categorical variables or factors. Finally, predictive models we built, evaluated, refined and benchmarked in order to discover and share more exciting insights.

Section 1

Methodology

Methodology

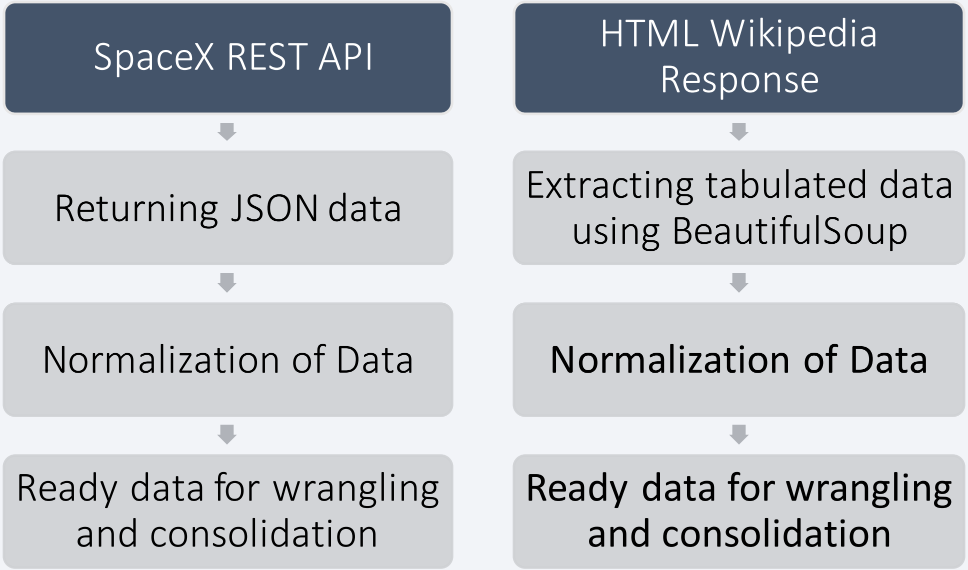
Executive Summary

- Data collection methodology:
    - SpaceX Rest API
    - BeautifulSoup web scrapping
- Perform data wrangling
    - Formating Data and using One Hot Encoding to prepare the data for Machine Learning techniques
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - LR,KNN,SVM,DT models were built and evaluated for the best classifier

Data Collection

- Two main sources of data were used in the collection phase:
- SpaceX REST API: launch data
  - Mainly available through api.spacexdata.com/v4
  - Provides accurate launch data such as: Booster/rocket information, payload/payload mass, launch information (location, DateTime, Orbit..), landing information, and operations outcomes.

  - BeautifulSoup was also used to scrap for more data through the web.



Data Collection – SpaceX API

- Data collection with SpaceX REST API

- GitHub URL : [IBM-CapstoneProjectRepo/jupyter-labs-spacex-data-collection-api.ipynb at main · helpyassine/IBM-CapstoneProjectRepo · GitHub]

Getting API Response

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

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

Converting Response to JSON file

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

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

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

Retrieve and clean the data

Prepare a tabulated format using dictionary to 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}
```

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

Refine the final dataframe and export it as a CSV file.

```
#export data_falcon9 to csv
data_falcon9.to_csv('dataset_part_1.csv', index=False)
data_falcon9.shape
```

Data Collection - Scraping

- Web scraping process
- GitHub URL: [IBM-CapstoneProjectRepo/jupyter-labs-webscraping.ipynb at main · helpyassine/IBM-CapstoneProjectRepo · GitHub]()

Getting HTML Response

Creating BeautifulSoup Object

Locate the tables

Get column names and rows

Clean the data

Prepare a tabulated format using dictionary to dataframe

Refine the final dataframe and export it as a CSV file.

```
# use requests.get() method with the provided static_url
# to get the HTML content of the page
# assign the response to a object
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a res
# assign the object to a variable
soup = BeautifulSoup(response.text, 'html.parser')
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainro
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to l
        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 `Flig
            launch_dict['Flight No.'].append(flight_number)
            #print(flight_number)
            datatimelist=date_time(row[0])
```

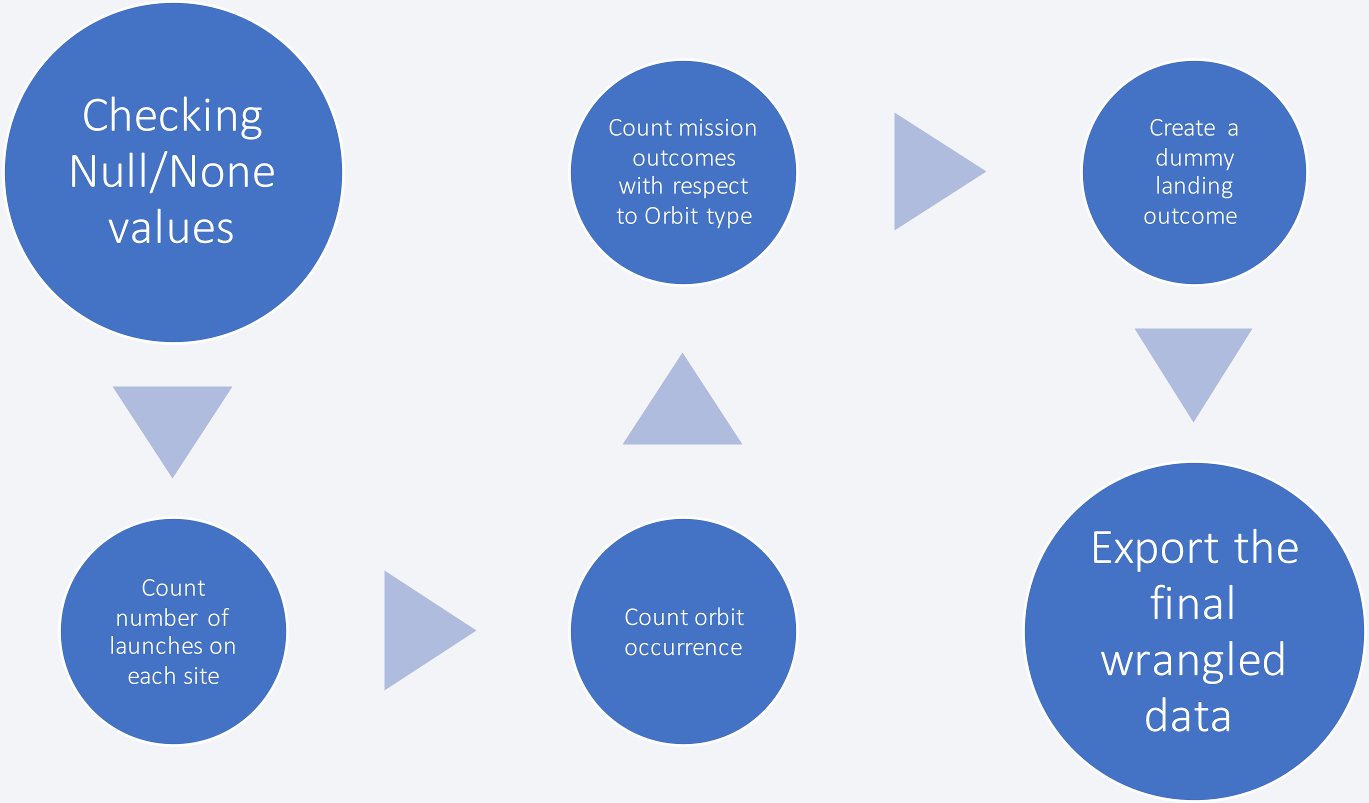
```
# faced problems creating a df out of the dictonary so using orient and tran

df = pd.DataFrame.from_dict(launch_dict, orient='index')
df = df.transpose()
```

```
df.head(20)
```

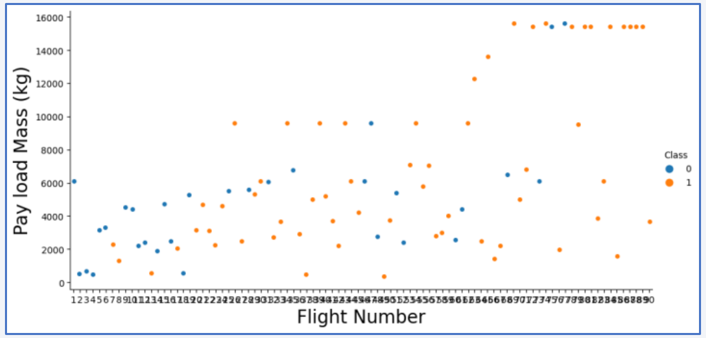
	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Ve Bo
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	v1.0B(
1	2	CCAFS	Dragon	0	LEO	NASA (COTS)\nNRO	Success[9]	v1.0B(
2	3	CCAFS	Dragon	525 kg	LEO	NASA (COTS)	Success[20]	v1.0B(
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA (CRS)	Success	v1.0B(
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA (CRS)	Success	v1.0B(
5	6	VAFB	CASSIOPE	500 kg	Polar orbit	MDA	Success[30]	v1.1

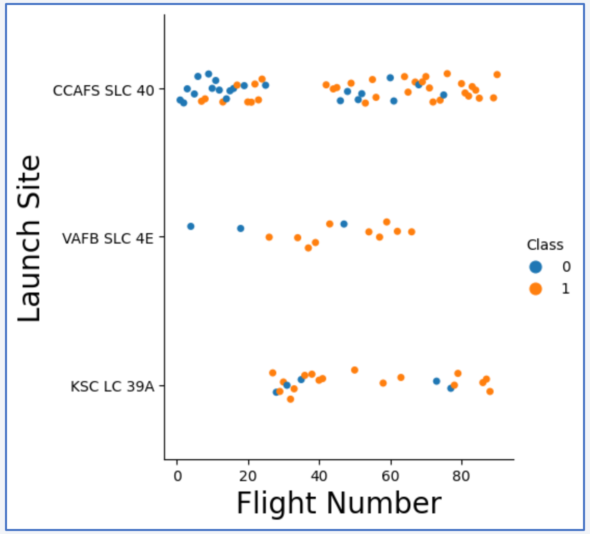
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

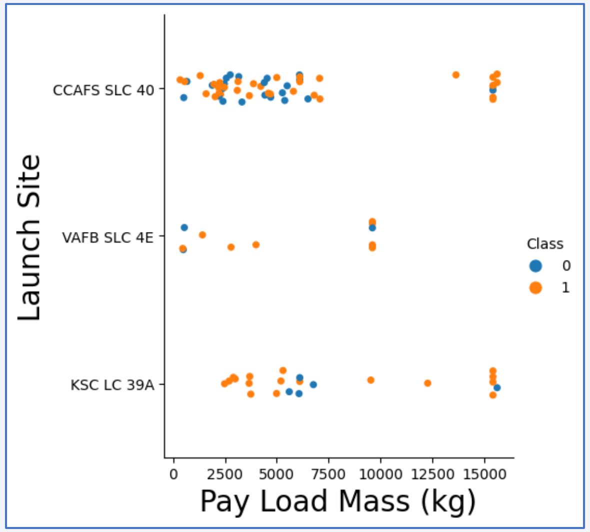
Data Wrangling

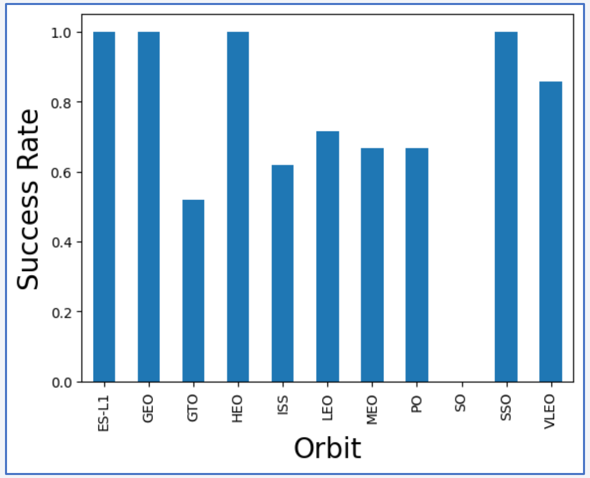


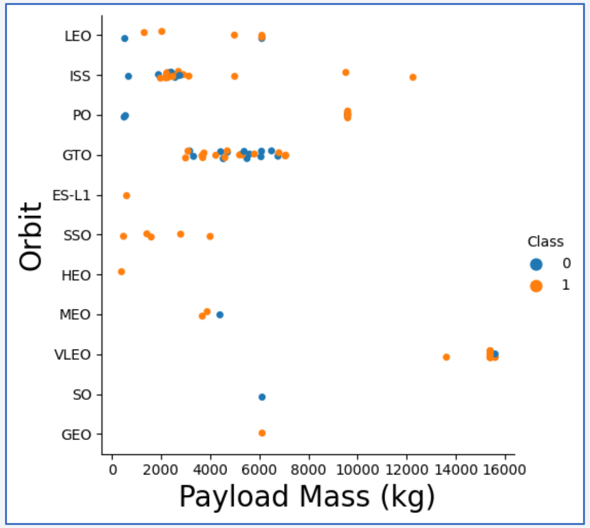
EDA with Data Visualization

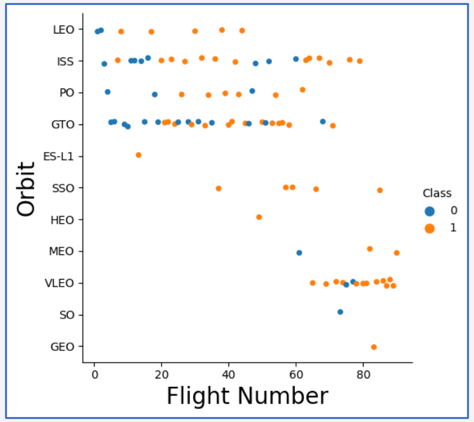


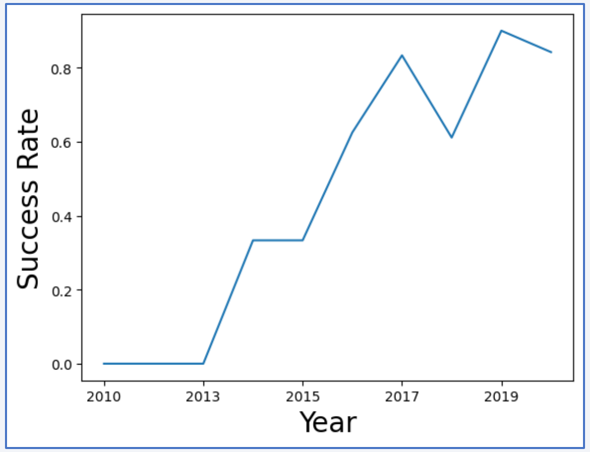








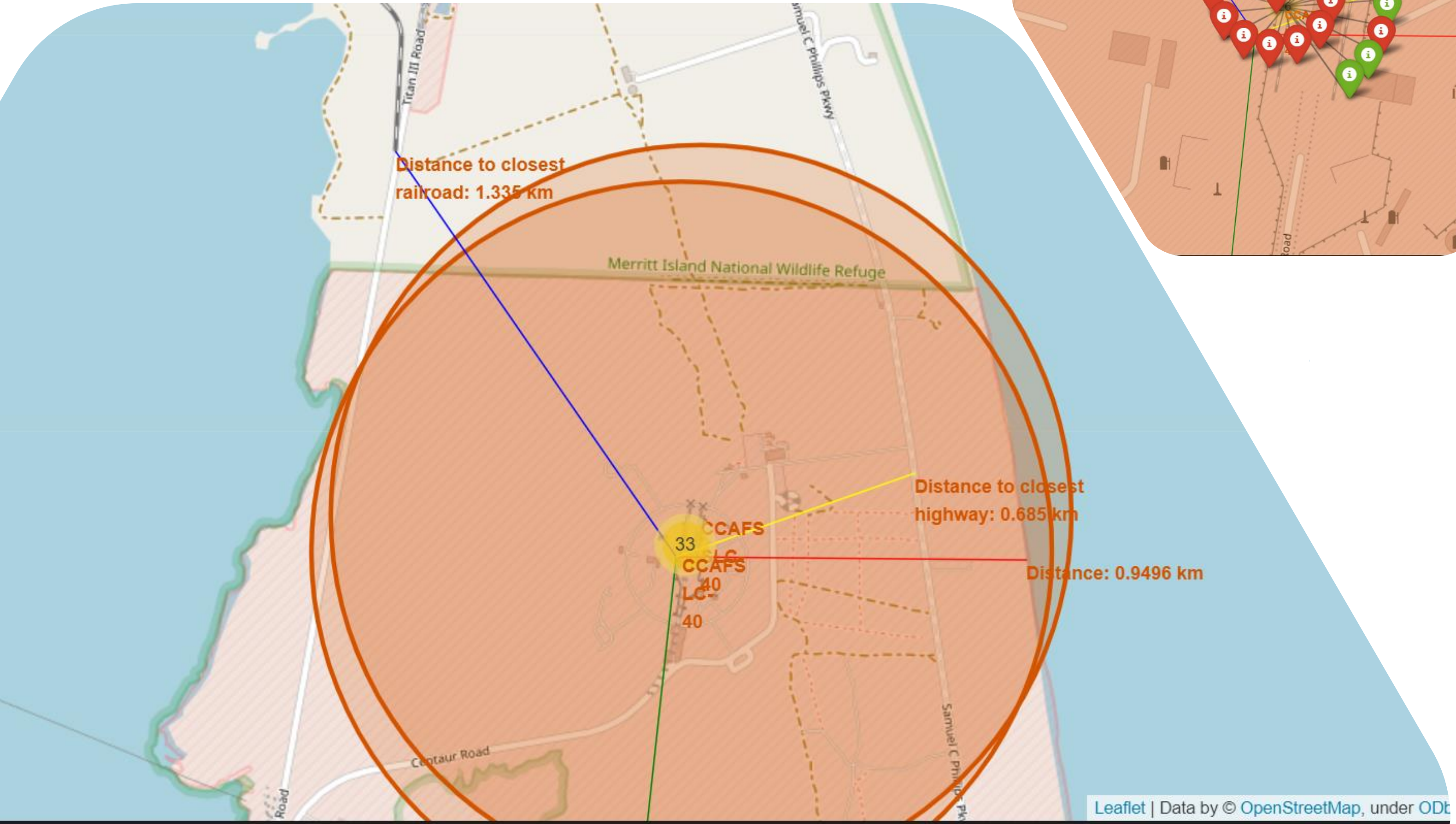




- GitHub URL: [IBM-CapstoneProjectRepo/jupyter-labs-eda-dataviz.ipynb at main · helpyassine/IBM-CapstoneProjectRepo · GitHub]

EDA with SQL

- SQL queries performed:
  - Displaying the names of the 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 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 URL: IBM-CapstoneProjectRepo/jupyter-labs-eda-sql-coursera.ipynb at main · helpyassine/IBM-CapstoneProjectRepo · GitHub

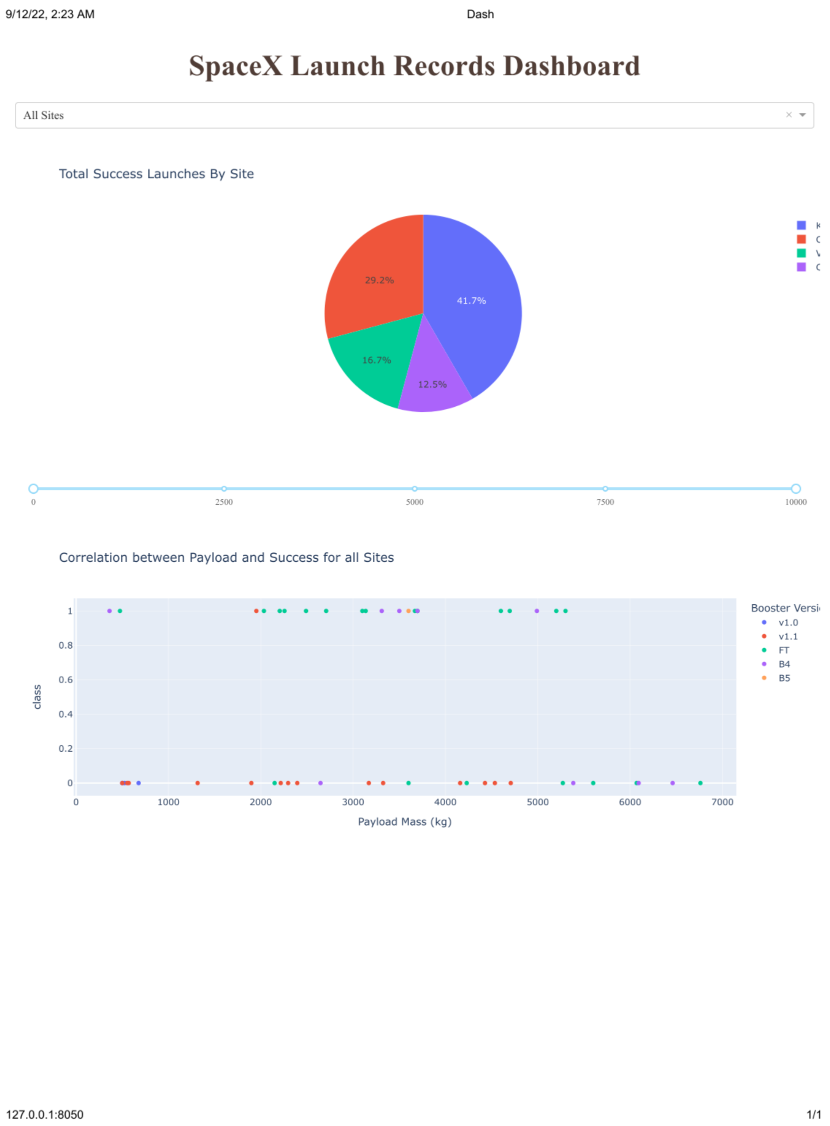


Build an Interactive Map with Folium

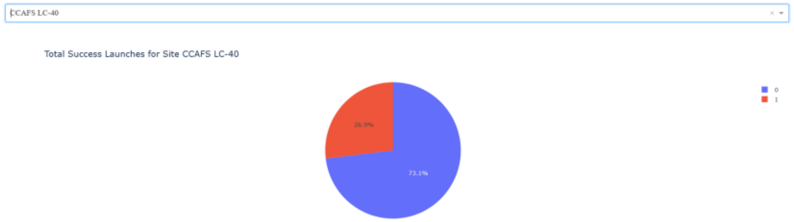
- Using Folium helped in mapping GIS information and also get answers to many questions such as where are the launch sites located and are they close to the equator. Also helped visualizing the launch Class, measure the distances across multiple important points to answers related questions.

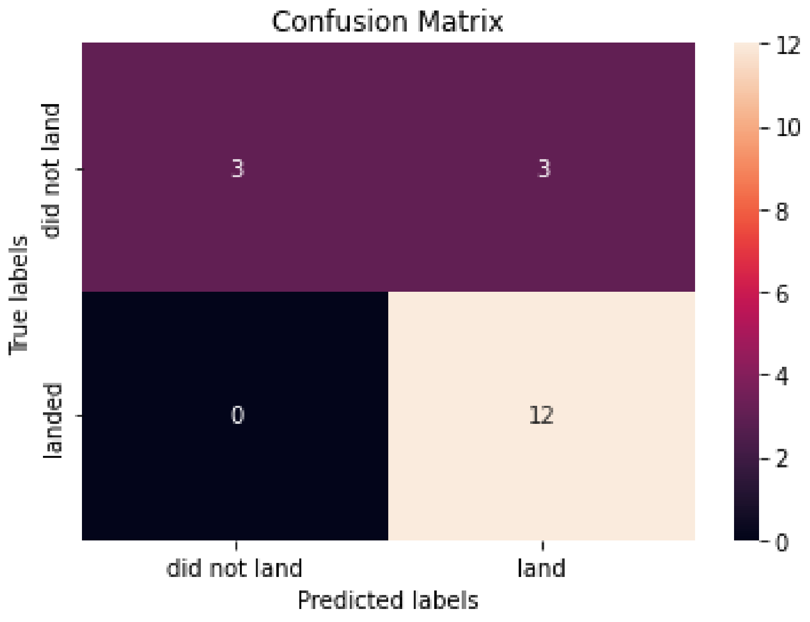
- GitHub URL:[IBM-CapstoneProjectRepo/lab_jupyter_launch_site_location.ipynb at main · helpyassine/IBM-CapstoneProjectRepo · GitHub]

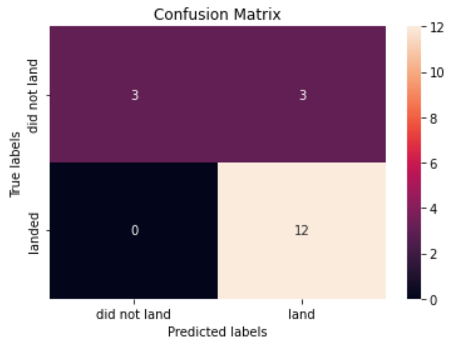
Build a Dashboard with Plotly Dash

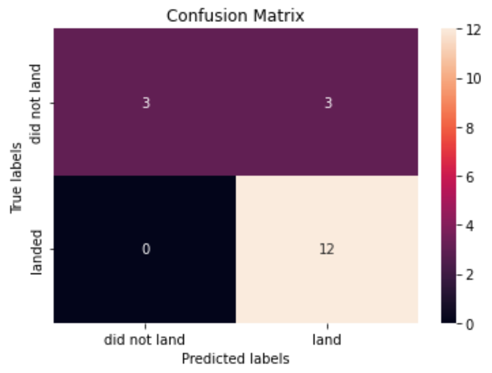


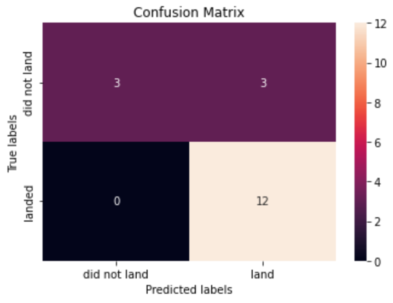
- In the dashboard, a dropdown list was added to enable Launch Site selection, in addition to a pie chart to show the total successful launches count for all sites.
- Moreover, a scatter chart to show the correlation between payload and launch success controlled by a slider to select payload range were added.
- GitHub URL:[IBM-CapstoneProjectRepo/spacex_dash_app.py at main · helpyassine/IBM-CapstoneProjectRepo · GitHub]

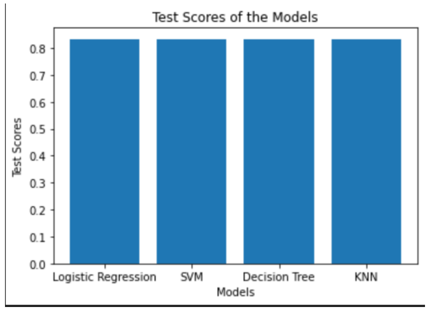












Predictive Analysis (Classification)

- The SVM, KNN, and logistic Regression models were developed and fited to achieve the highest accuracy based on multiple parameters. An accuracy of 83.33% across all the machine learning models.
- GitHub URL: IBM-CapstoneProjectRepo/SpaceX_Machine Learning Prediction_Part_5.ipynb at main · helpyassine/IBM-CapstoneProjectRepo · GitHub

Results

- Low weighed payloads perform better than the heavier ones
- The launch success rates of SpaceX are directly proportional to time as they get experience the failure probability decrease.
- The SVM, KNN and LR models are the best in terms of predicting an accurate result.
- KSC LC 39A can be said is the best location with the highest success launch rate.
- GEO, HEO, SSO and ES L1 orbits have a strong correlation with the missions success rates.

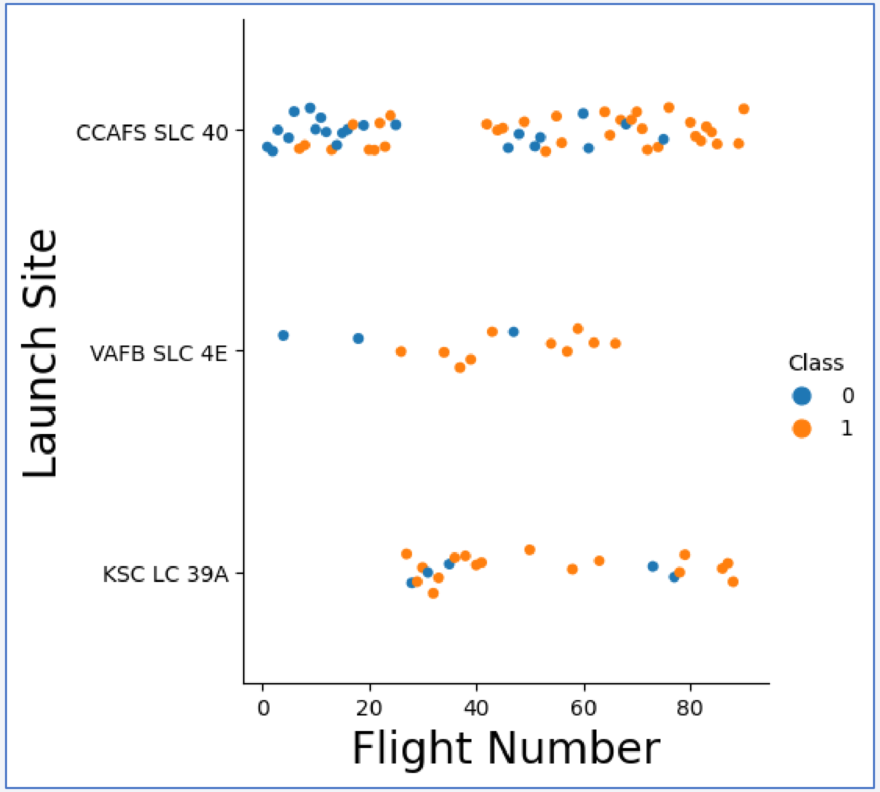
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

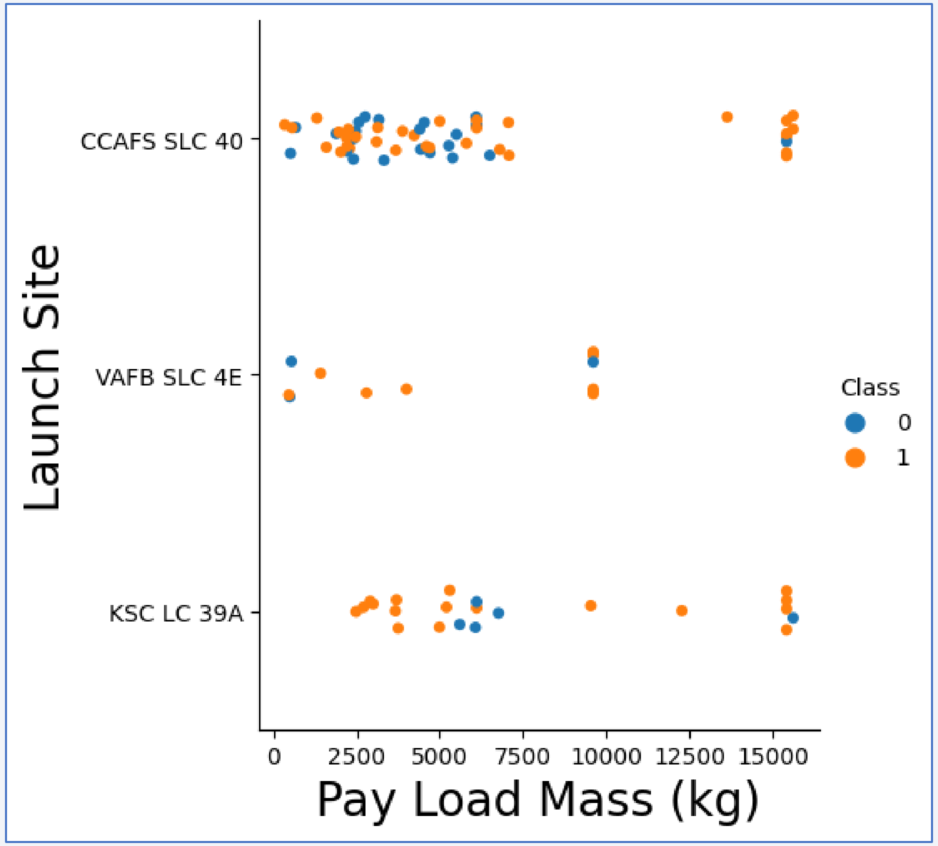
- Scatter plot of Flight Number vs. Launch Site

- We can clearly see that launches from CCAAFS SLC 40 are significantly higher than launches from other locations.



Payload vs. Launch Site

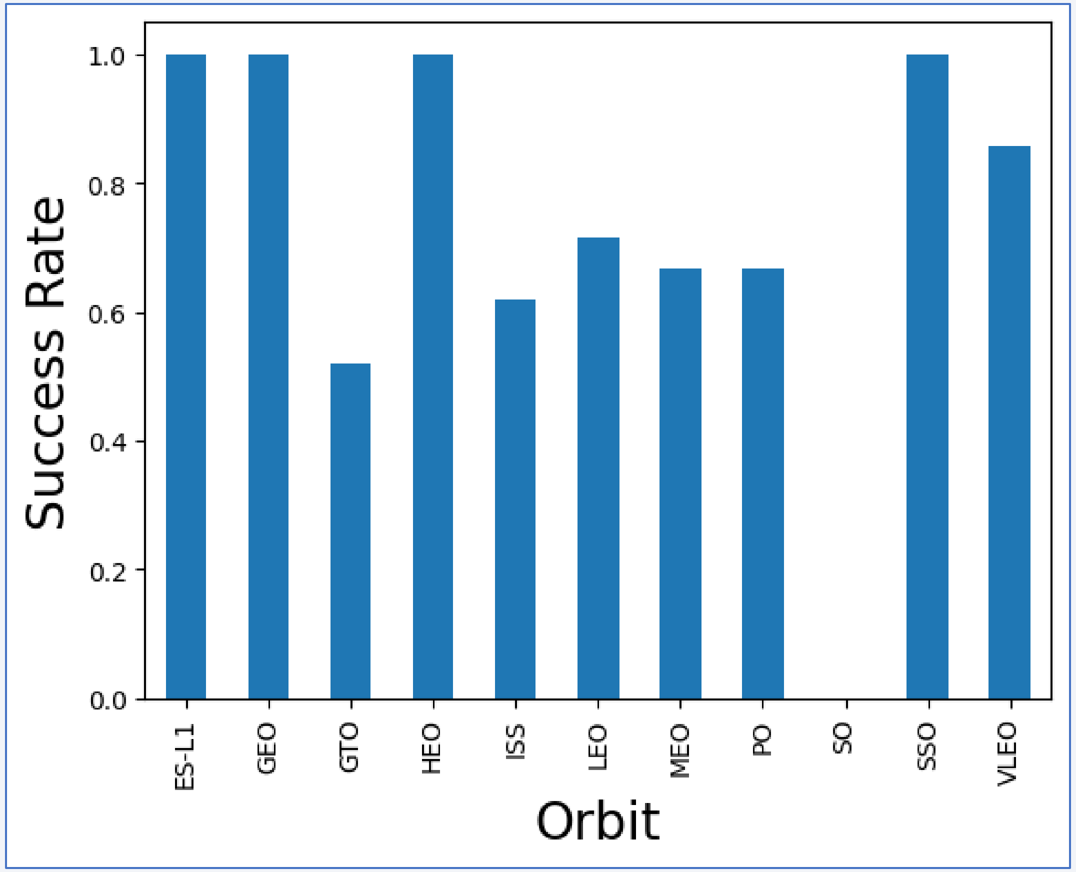
- Scatter plot of Payload vs. Launch Site
- We can conclude that Payloads with lower mass were launches through the CCAFS SLC 40 launch site.



Success Rate vs. Orbit Type

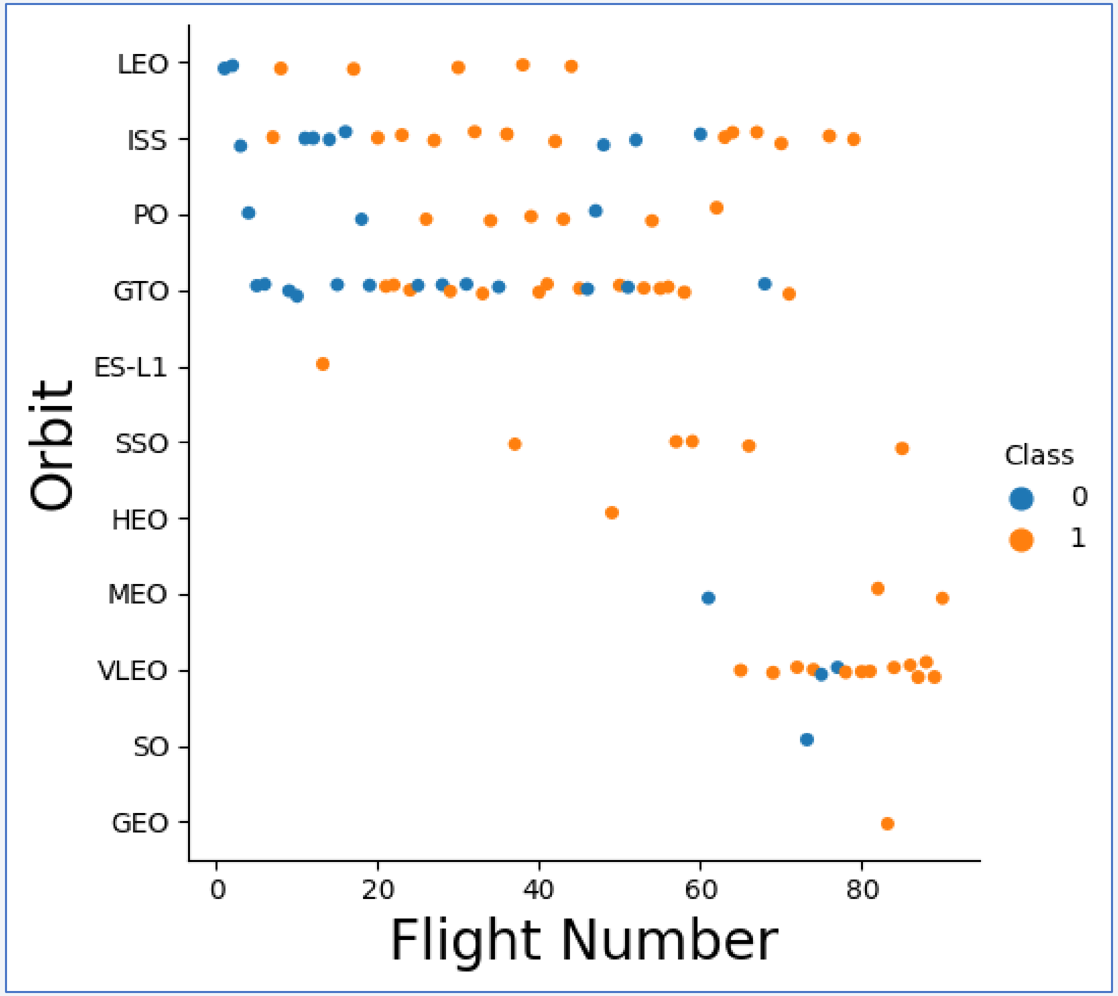
- Bar chart for the success rate of each orbit type

- We can see clearly that ES-L1, GEO, HEO and SSO are the orbits with high success.



Flight Number vs. Orbit Type

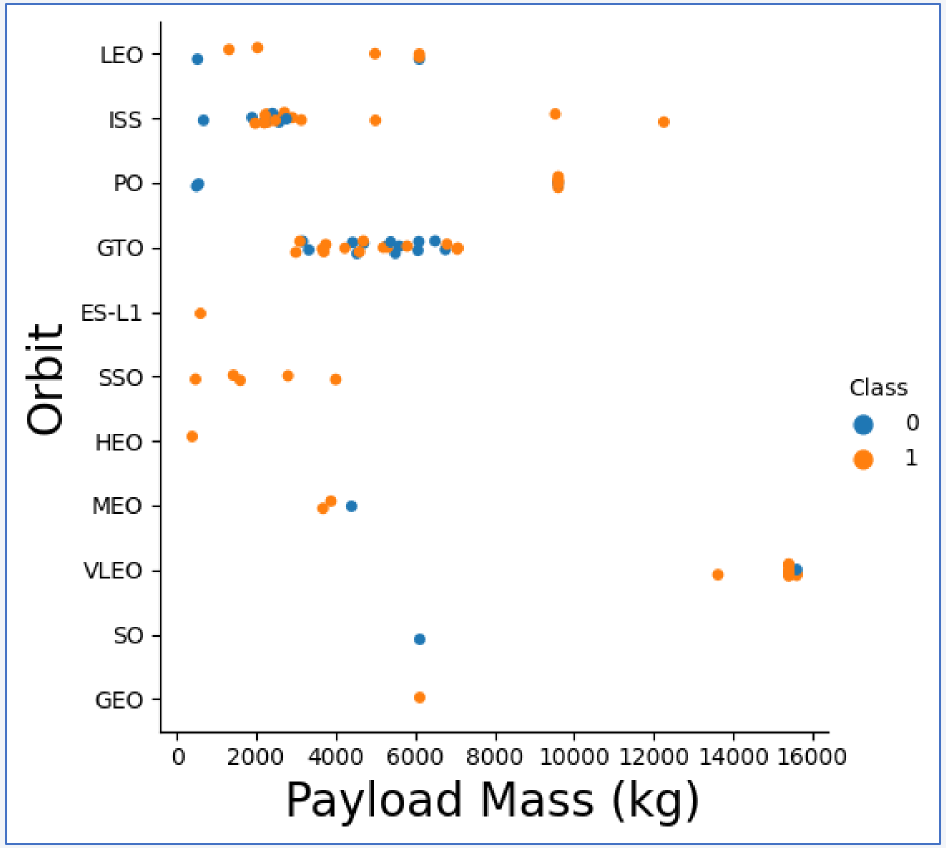
- Scatter point of Flight number vs. Orbit type
- There is a shift to VLEO in the recent launches.



Payload vs. Orbit Type

- Scatter point of payload vs. orbit type

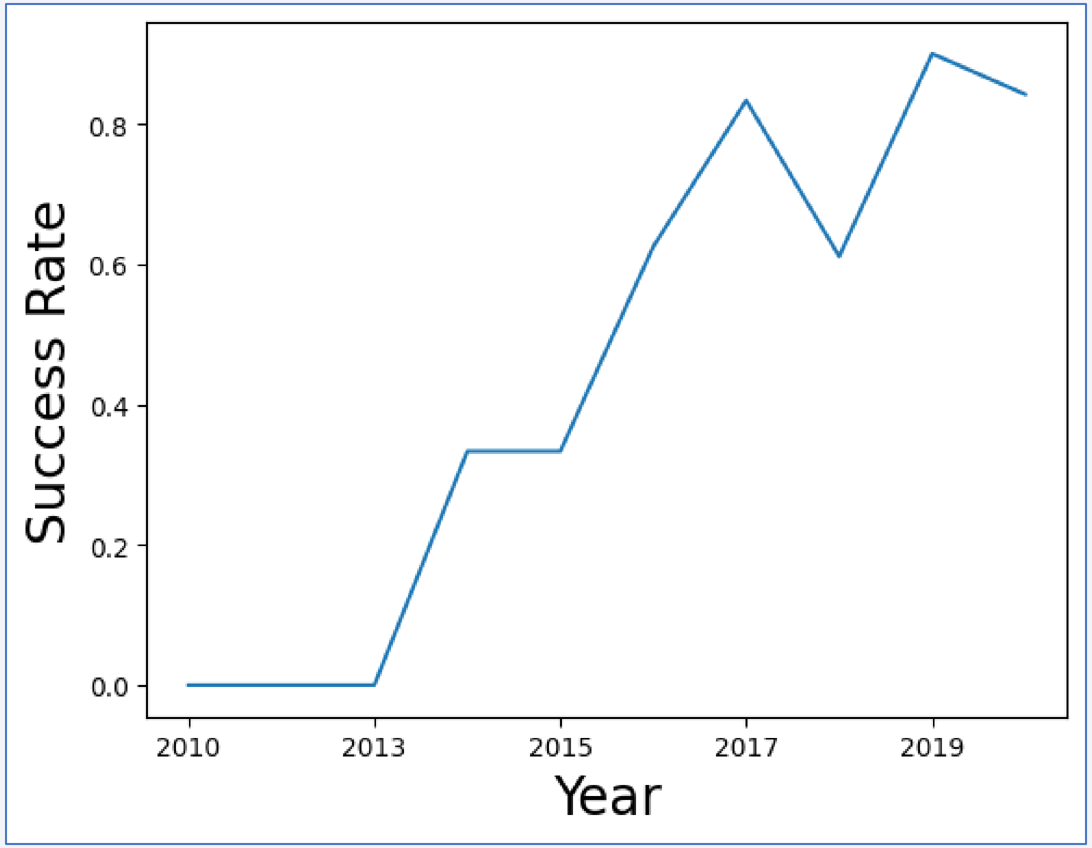
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are present.



Launch Success Yearly Trend

- Line chart of yearly average success rate

- We can observe that the sucess rate since 2013 kept increasing till 2020



All Launch Site Names

- Finding the names of the unique launch sites

```
Query = "select DISTINCT LAUNCH_SITE from SPACEXTBL"
```

Out[16]:

	LAUNCH_SITE
0	CCAFS LC-40
1	CCAFS SLC-40
2	KSC LC-39A
3	VAFB SLC-4E

Launch Site Names Begin with 'CCA'

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

```
Query = "select * from SPACEXTBL where LAUNCH_SITE LIKE 'CCA%'"
results = pd.read_sql(Query, pconn)
results.head()
```

	DATE	TIME__UTC_	BOOSTER_VERSION	LAUNCH_SITE	PAYLOAD	PAYLOAD_MASS__KG_	ORBIT	CUSTOMER	MISSION_OUTCOME	LANDING__OUTCOME
0	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
5	2013-12-03	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

Total Payload Mass

- Calculating the total payload carried by boosters from NASA

```
 Query = "select sum(PAYLOAD_MASS__KG_) AS Total_Payload_NASA from SPACEXTBL where CUSTOMER = 'NASA (CRS)'"
```

Out[41]:

	TOTAL_PAYLOAD_NASA
0	45596

Average Payload Mass by F9 v1.1

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

```
Query = "select avg(PAYLOAD_MASS__KG_) AS Average_Payload_F9v11 from SPACEXTBL where BOOSTER_VERSION LIKE 'F9 v1.1%'"
```

Out[44]:

	AVERAGE_PAYLOAD_F9V11
0	2534

First Successful Ground Landing Date

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

Query = "select DATE from SPACEXTBL Where LANDING__OUTCOME LIKE 'Success (ground pad)%' ORDER BY DATE ASC LIMIT 1"

Out[49]:

	DATE
0	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
Query = "select BOOSTER_VERSION,LANDING__OUTCOME,
PAYLOAD_MASS__KG_ from SPACEXTBL \

Where PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000  \

AND LANDING__OUTCOME LIKE 'Success (drone ship)%'"
```

Out[52]:

	BOOSTER_VERSION	LANDING__OUTCOME	PAYLOAD_MASS__KG_
0	F9 FT B1022	Success (drone ship)	4696
1	F9 FT B1026	Success (drone ship)	4600
2	F9 FT B1021.2	Success (drone ship)	5300
3	F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

- Calculating the total number of successful and failure mission outcomes

```
Query = "select count(case when LANDING__OUTCOME LIKE '%Success%'
then 1 end) as Success, \
                 count(case when LANDING__OUTCOME LIKE '%Failure%'
then 1 end) as Failure \
    from SPACEXTBL"
```

Out[59]:

	SUCCESS	FAILURE
0	61	10

Boosters Carried Maximum Payload

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

```
Query = "select BOOSTER_VERSION, PAYLOAD_MASS__KG_ from SPACEXTBL \

Where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)"
```

Out[61]:

	BOOSTER_VERSION	PAYLOAD_MASS__KG_
0	F9 B5 B1048.4	15600
1	F9 B5 B1049.4	15600
2	F9 B5 B1051.3	15600
3	F9 B5 B1056.4	15600
4	F9 B5 B1048.5	15600
5	F9 B5 B1051.4	15600
6	F9 B5 B1049.5	15600
7	F9 B5 B1060.2	15600
8	F9 B5 B1058.3	15600
9	F9 B5 B1051.6	15600
10	F9 B5 B1060.3	15600
11	F9 B5 B1049.7	15600

2015 Launch Records

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

```
Query = "select LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
from SPACEXTBL \

       Where LANDING__OUTCOME LIKE 'Failure (drone ship)%' and
Year(Date) = 2015"
```

Out[65]:

	LANDING__OUTCOME	BOOSTER_VERSION	LAUNCH_SITE
0	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
1	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- 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

```
Query = "select  DATE, \
                 count(case when LANDING__OUTCOME LIKE
'%Success%' then 1 end) as Success, \
                 count(case when LANDING__OUTCOME LIKE
'%Failure%' then 1 end) as Failure \
                 from SPACEXTBL \
                 WHERE DATE BETWEEN '2010-06-04' and '2017-
03-20' \
                 group by DATE \
                 order by DATE ASC"
```

Out[68]:

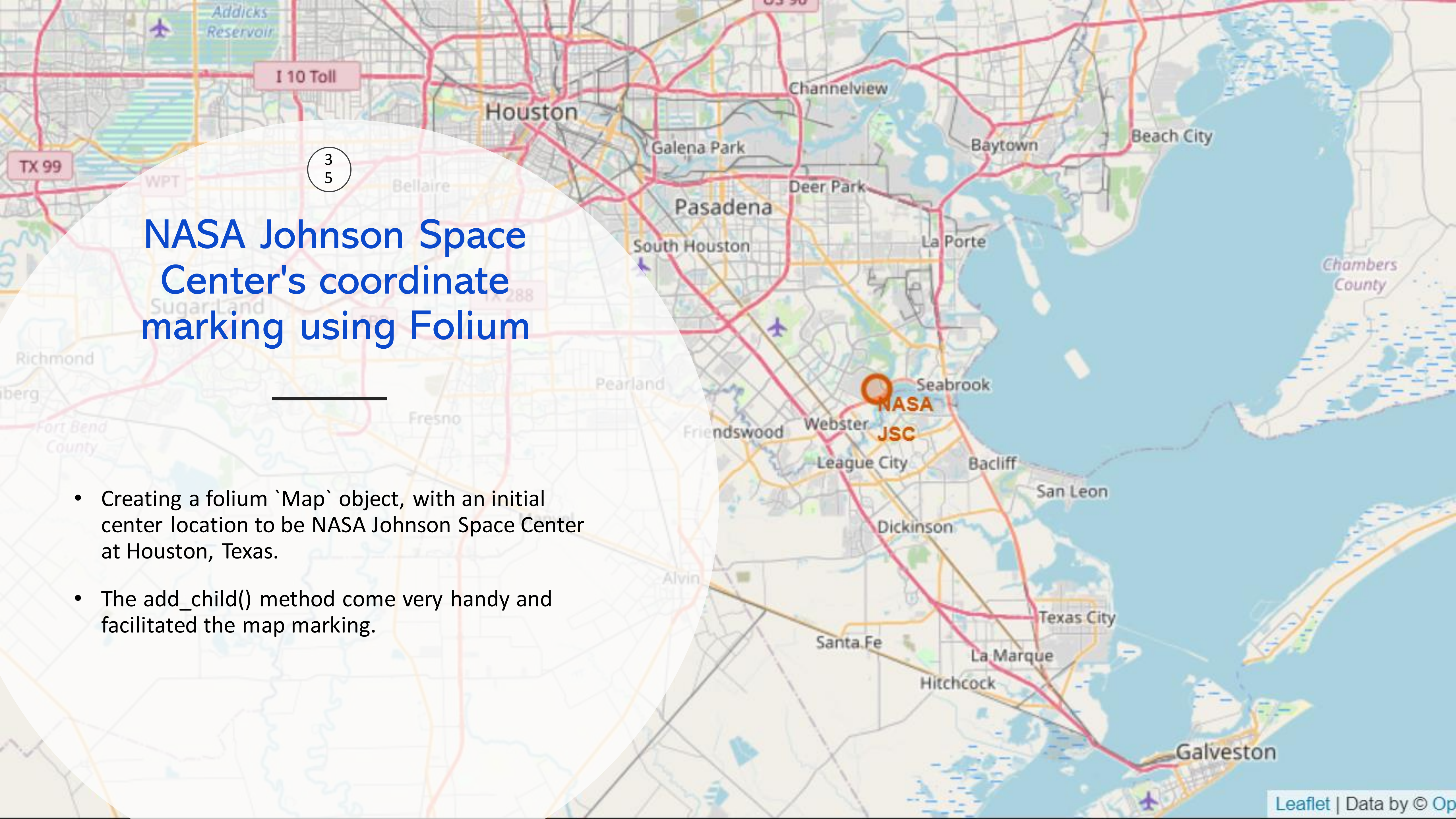
	DATE	SUCCESS	FAILURE
0	2010-06-04	0	1
1	2010-12-08	0	1
2	2012-05-22	0	0
3	2012-10-08	0	0
4	2013-03-01	0	0
5	2013-09-29	0	0
6	2013-12-03	0	0
7	2014-01-06	0	0
8	2014-04-18	0	0
9	2014-07-14	0	0
10	2014-08-05	0	0
11	2014-09-07	0	0
12	2014-09-21	0	0
13	2015-01-10	0	1
14	2015-02-11	0	0
15	2015-03-02	0	0
16	2015-04-14	0	1
17	2015-04-27	0	0
18	2015-06-28	0	0
19	2015-12-22	1	0
20	2016-01-17	0	1
21	2016-03-04	0	1
22	2016-04-08	1	0
23	2016-05-06	1	0
24	2016-05-27	1	0
25	2016-06-15	0	1
26	2016-07-18	1	0
27	2016-08-14	1	0
28	2017-01-14	1	0
29	2017-02-19	1	0
30	2017-03-16	0	0

Section 3

Launch Sites Proximities Analysis

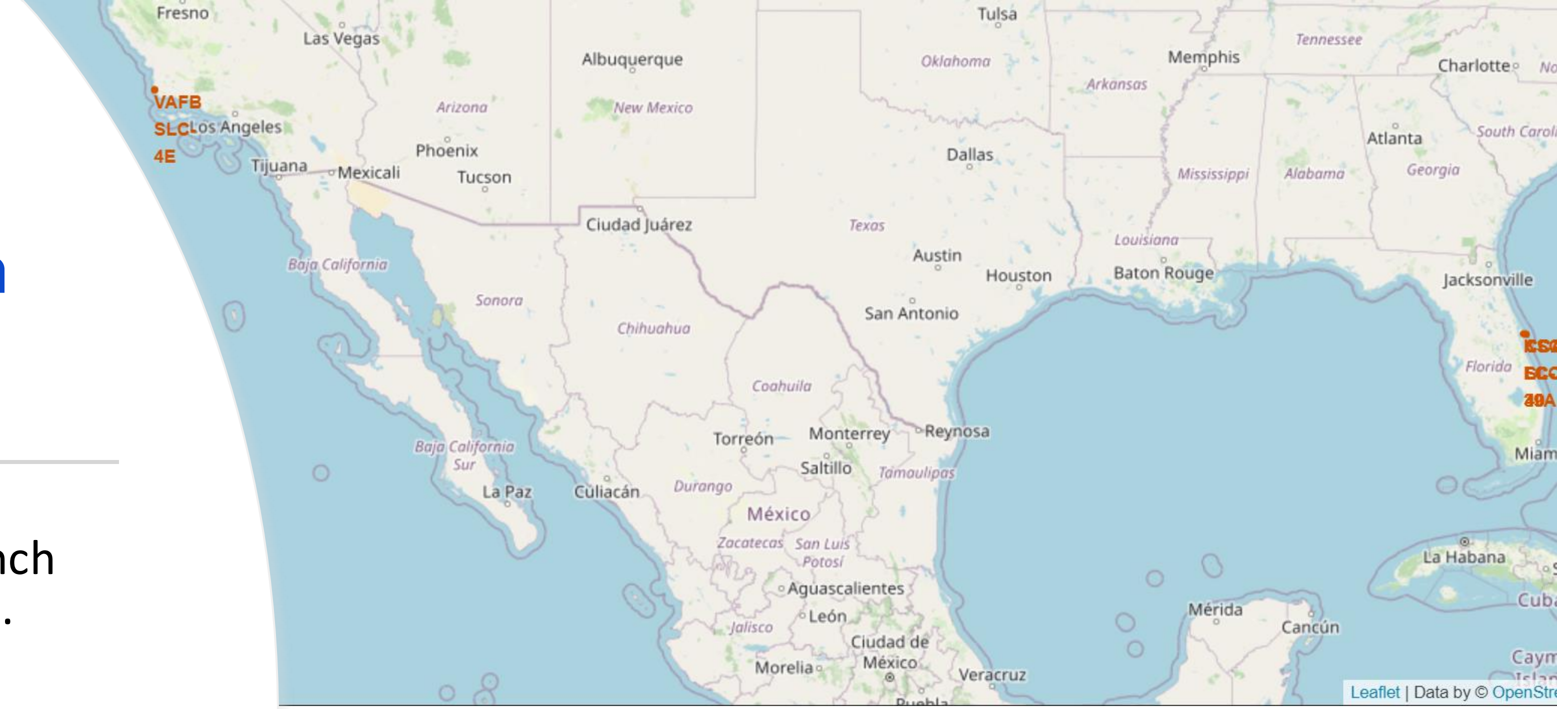
NASA Johnson Space Center's coordinate marking using Folium

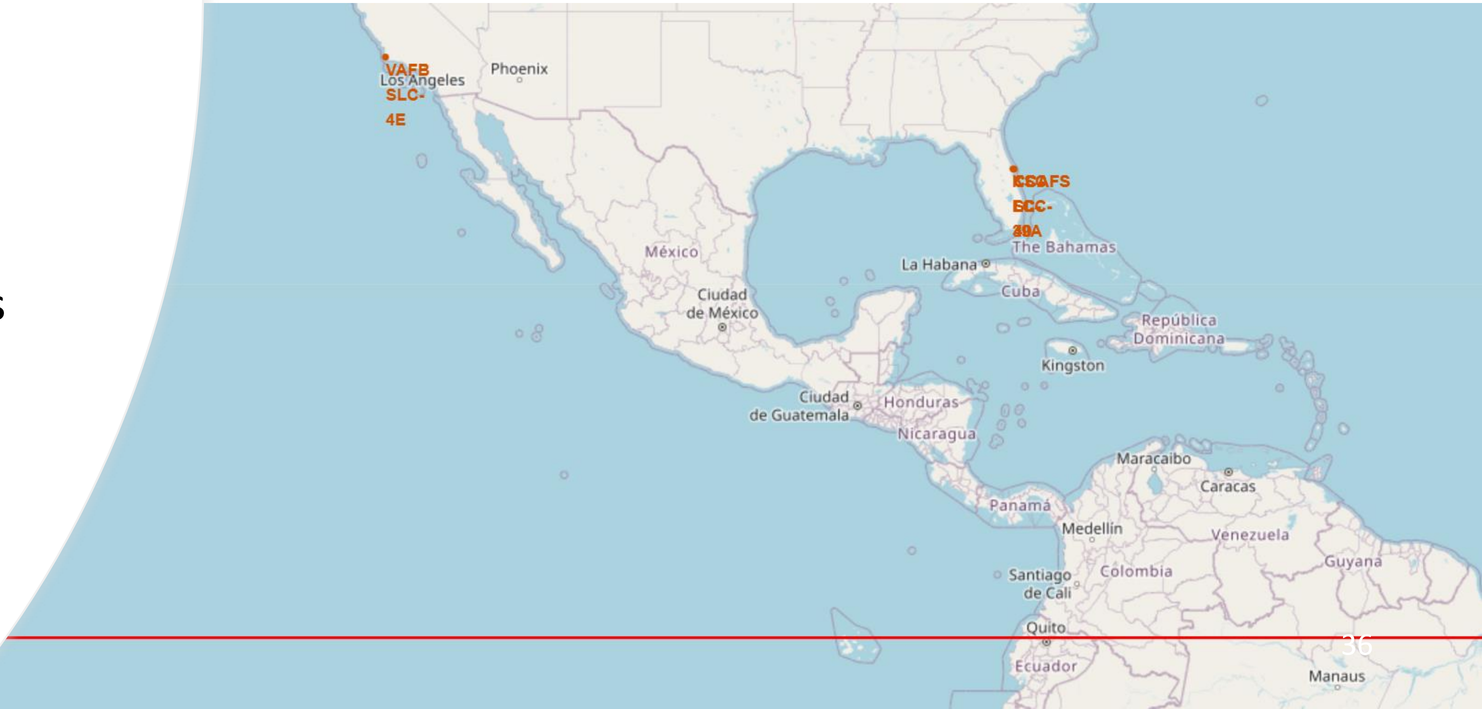
- Creating a folium `Map` object, with an initial center location to be NASA Johnson Space Center at Houston, Texas.
- The add_child() method come very handy and facilitated the map marking.



Marking the Launch sites on the map using Folium

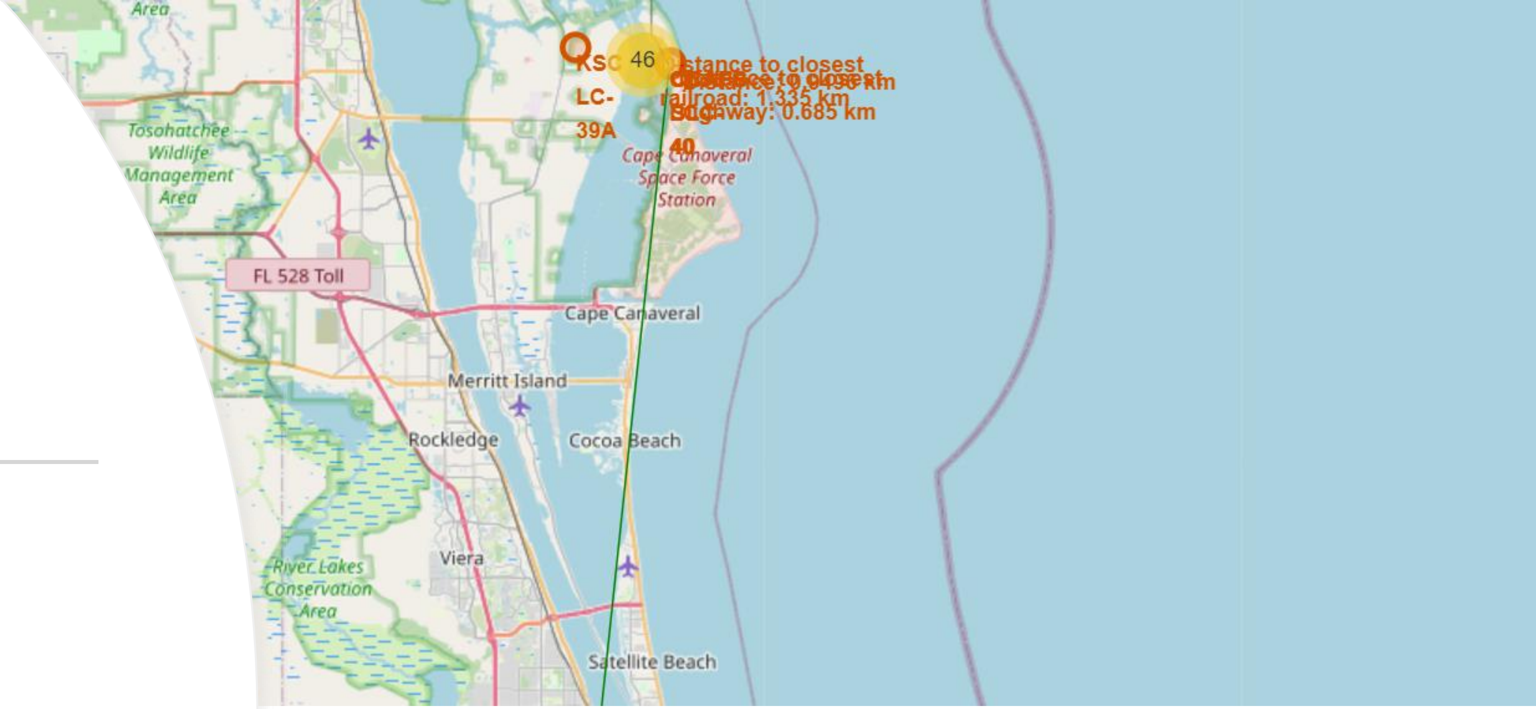
- In this step we marked the map with launch site nodes to clearly see they geolocation. And also to answer some interesting questions on their proximity to the equatorial line and to coasts.
- Using the polyline on the map to distinguish the equatorial line we can see that even if the launch locations are a bit far from the equator. they are located in the south of the United States (which is not by chance) so it is meant to be closer to the equator.
- Also we note that all the Launch sites are always by proximity to coasts.

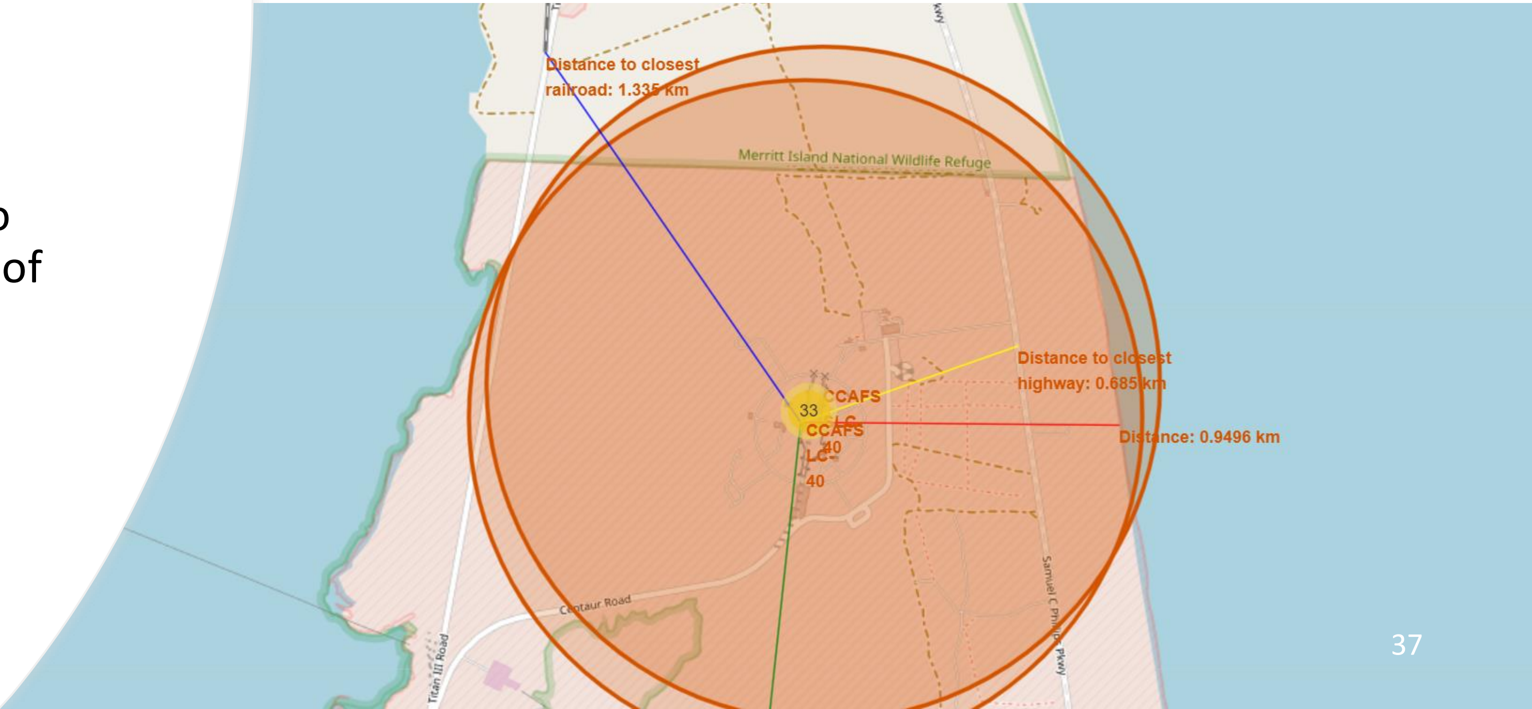




Using Folium to calculate distance and mark polylines between geo-coordinates

- In this step we built a distance calculation function that gets the calculated distance between two points on the map given their Latitudes and Longitudes. The distance is measured in Kilometers.
- Next marking both a polyline and the distance measures on the map is done to illustrate a better Geo-idea on the proximity of important places and facilities. Such as the Coast the Rail way, the Highway and the nearest city.



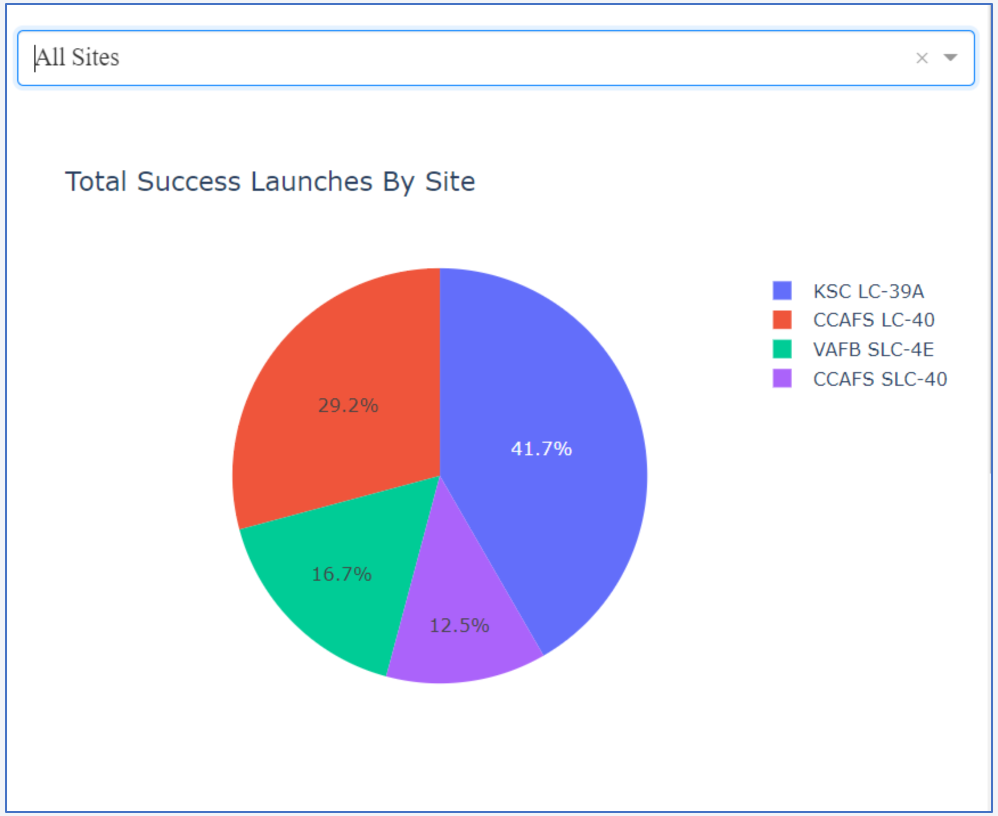


Section 4

Build a Dashboard with Plotly Dash

Total success launces by all sites

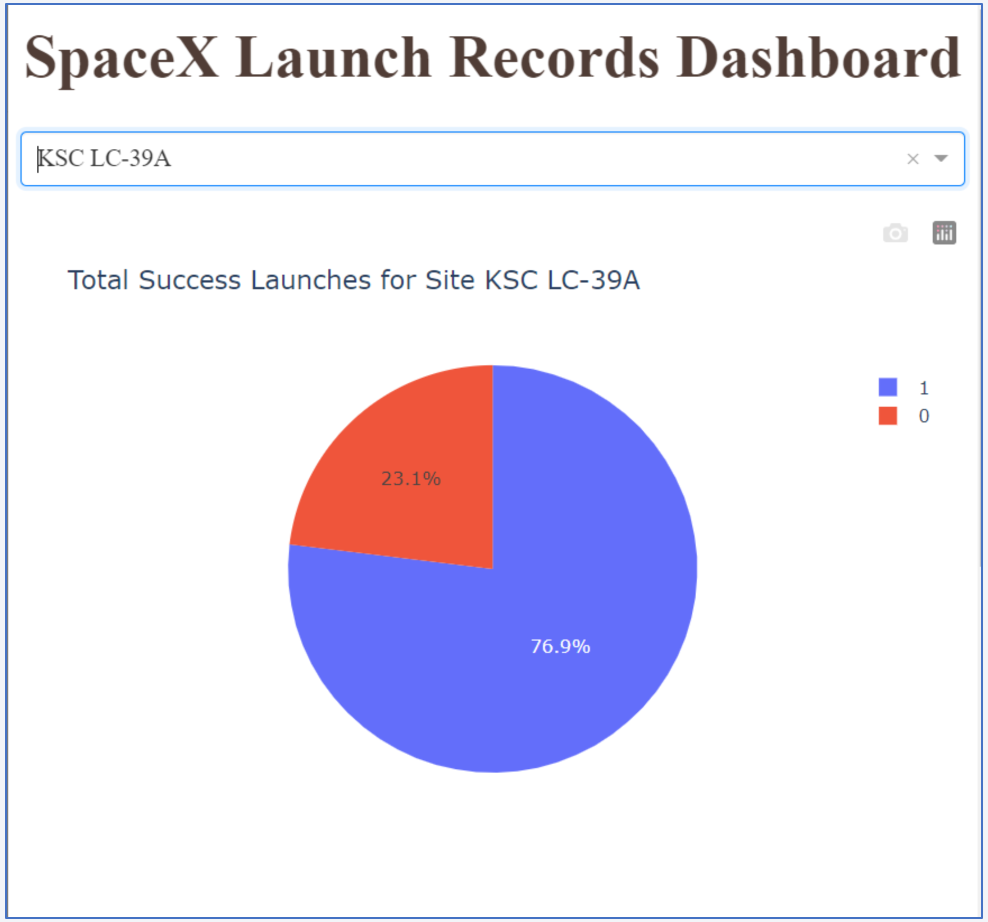
- We can see that KSC LC 39A had the most successful launches than the other sites.



Success rate by site

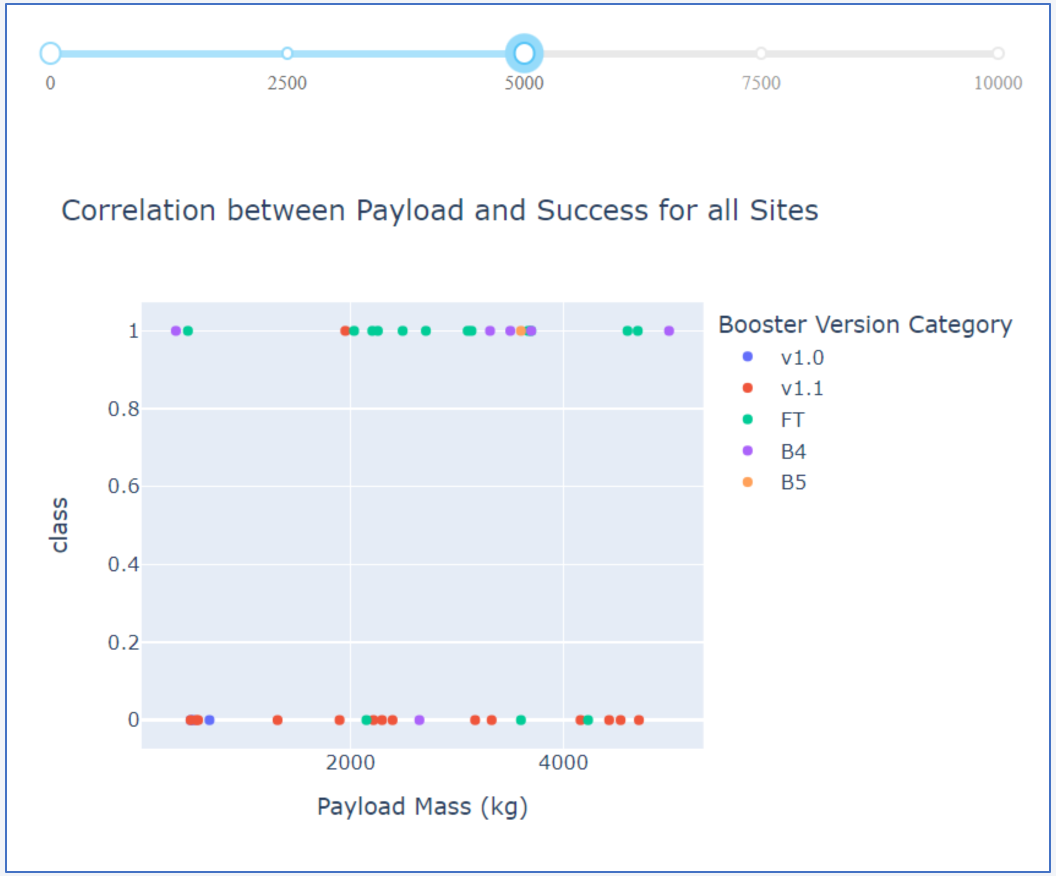
- Screenshot of the piechart for the launch site with highest launch success ratio

- We can see that KSC LC 39A achieved 76.9% success rate versus 23.1% failure rate.



Payload vs launch outcome

- We can see that the success rates for low weighted payloads is higher than the heavy weighed payloads as in the example below: (0-5000kg) and (4000-10000kg)

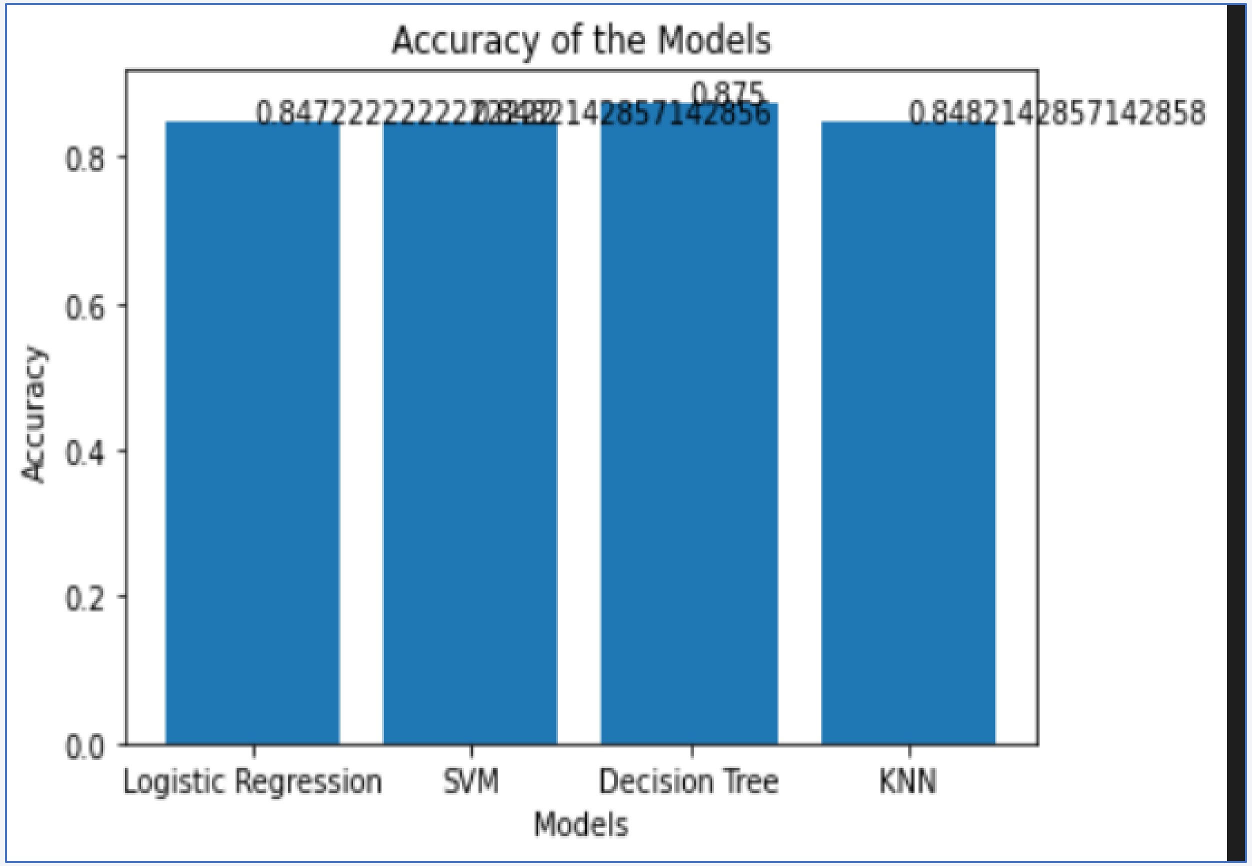


Section 5

Predictive Analysis (Classification)

Classification Accuracy

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

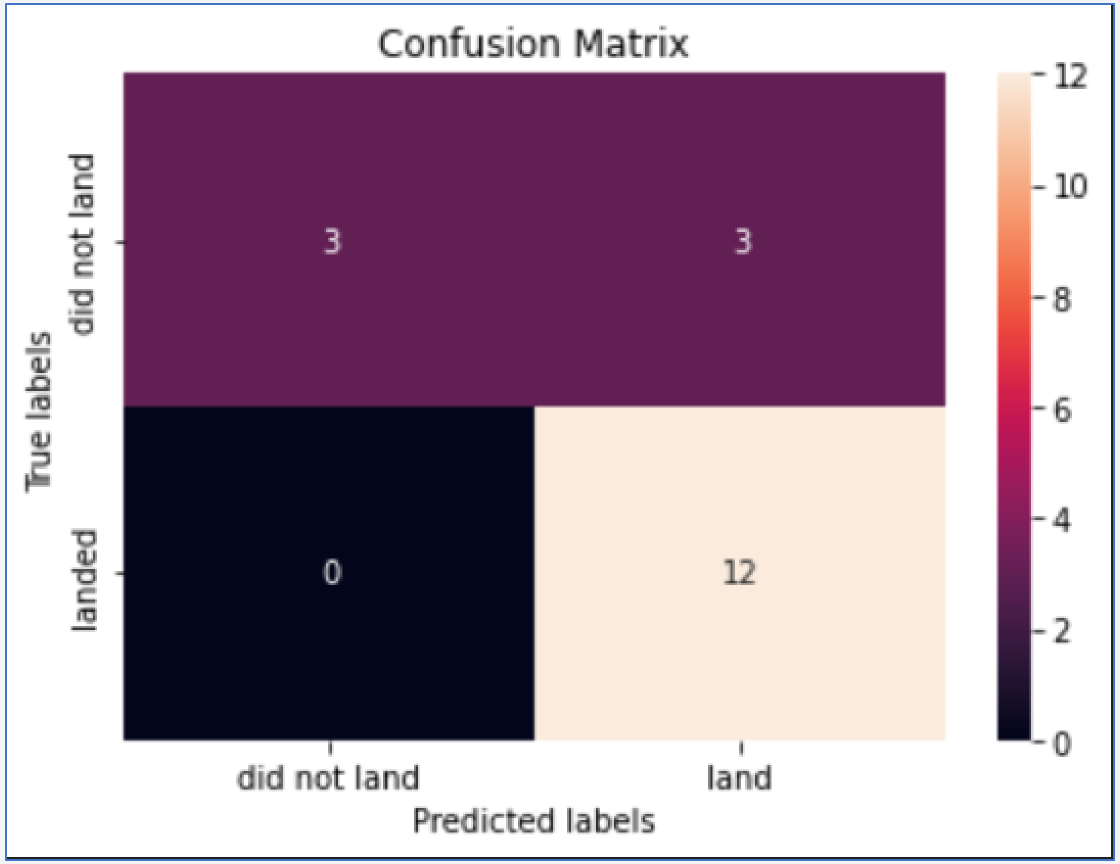


- Logistic Regression best score: 85 %
- SVM best score: 85 %
- Decision Tree best score: 88 %
- KNN best score: 85 %

Confusion Matrix

- Showing the confusion matrix of the best performing model

- The SVM, KNN and LR models are the best in terms of predicting an accurate result.



Conclusions

- Low weighed payloads perform better than the heavier ones
- The launch success rates of SpaceX are directly proportional to time as they get experience the failure probability decrease.
- The SVM, KNN and LR models are the best in terms of predicting an accurate result.
- KSC LC 39A can be said is the best location with the highest success launch rate.
- GEO, HEO, SSO and ES L1 orbits have a strong correlation with the missions success rates.

Appendix

- Refer to the GitHub repository of this project as is archives all the relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that were created during this project.
- Link:

GitHub - helpyassine/IBM-CapstoneProjectRepo

Thank you!