IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Kholoud Al Bittar>
<May/22/2024>

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

. Predictive Analysis (Classification)

❑Summary of all results

. EDA Results

. Interactive Analytics

. Predictive Analysis

Introduction

❑ Project background and context

❑ In this capstone assignment, I will be working with SpaceX launch data that is gathered from an API, specifically the SpaceX REST API.

This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome

❑Problems you want to find answers

❑ Predict whether SpaceX will attempt to land a rocket or not.

Section 1

Methodology

Methodology

Executive Summary

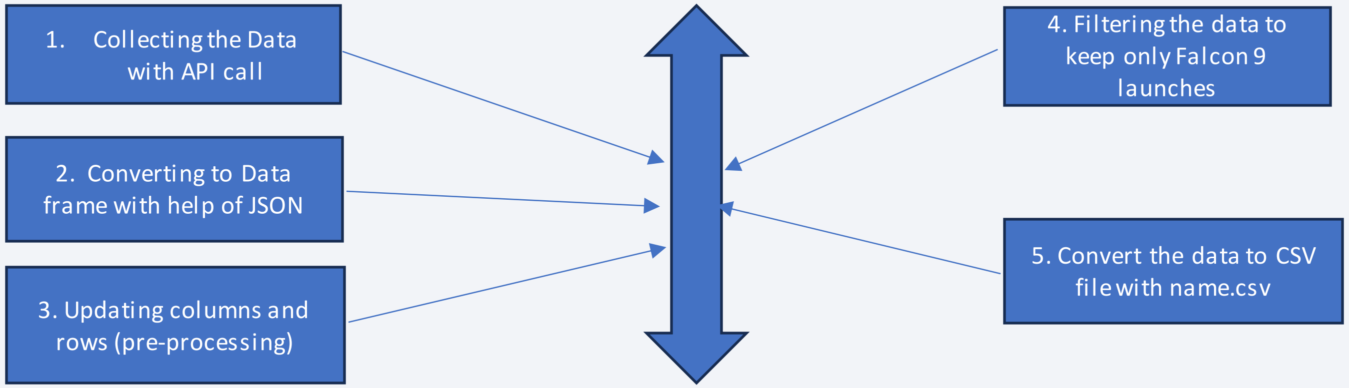
- ❑ Data collection methodology:
  - With Rest API and Web Scrapping
- ❑ Perform data wrangling
  - Data were transformed and one hot encoded to be apply later on the Machine Learning models
- ❑ Perform exploratory data analysis (EDA) using visualization and SQL
  - Discovering new patterns in the data with Visualization like scatter plots
- ❑ Perform interactive visual analytics using Folium and Plotly Dash
  - Dash and Folium were used to achieve the goal
- ❑ Perform predictive analysis using classification models
  - Classification Machine Learning models were built to achieve the goal

Data Collection

❑Describe how data sets were collected.

- The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/.
- We will be working with the endpoint api.spacexdata.com/v4/launches/past.
- We will perform a get request using the requests library to obtain the launch data, which we will use to get the data from the API.
- Another popular data source for obtaining Falcon 9 Launch data is web scraping related Wiki pages.

❑You need to present your data collection process use key phrases and flowcharts



Data Collection – SpaceX API

- Data collection with SpaceX REST calls using key phrases and flowcharts

- https://github.com/kholoudbittar/Build-an-Interactive-Dashboard-with-Ploty-Dash/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

1- Collecting the Data with API call

```
. spacex_url="https://api.spacexdata.com/v4/launches/past"
. response = requests.get(spacex_url)
. print(response.content)
```

2- Converting to Data frame with help of JSON

```
. response.status_code
. data=pd.json_normalize(response.json())
```

3- Updating columns and rows (pre-processing)

```
[18]: # Call getLaunchSite
getLaunchSite(data)

[19]: # Call getPayloadData
getPayloadData(data)

[20]: # Call getCoreData
getCoreData(data)
```

Finally lets construct our dataset using the data we have obta

```
[21]: 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}
```

4- Filtering the data to keep only Falcon 9 launches

```
[28]: # Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

Now that we have removed some values we should reset the FlgihtNumber column

```
[29]: data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

5- Convert the data to CSV file with name.csv

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

Data Collection - Scraping

- Web scraping process using key phrases and flowcharts

- https://github.com/kholoudbittar/Build-an-Interactive-Dashboard-with-Ploty-Dash/blob/main/jupyter-labs-webscraping.ipynb

1. Html response

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

2. Creating BeautifulSoup

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

3. Getting column names

```
. temp = soup.find_all('th')
for x in range(len(temp)):
   try:
    name =
extract_column_from_header(temp[x])
    if (name is not None and len(name) > 0):
      column_names.append(name)
   except:
    pass
```

4. Creating Launch 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. Converting to final Data

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

6. Convert to CSV file name.csv

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

Data Wrangling

1. Load Dataset

```
. df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv")

. df.head(10)
```

2. Calculate the number of launches

```
. df['LaunchSite'].value_counts()
```

3. Calculate the number and occurrence of each orbit

```
. landing_outcomes = df['Outcome'].value_counts()

. landing_outcomes
```

4. Calculate landing outcome

```
. landing_class = df['Outcome'].replace({'False Ocean': 0, 'False ASDS': 0, 'None None': 0, 'None ASDS': 0, 'False RTLS': 0, 'True ASDS': 1, 'True RTLS':
1, 'True Ocean': 1}, inplace = True)
. df['Outcome'] = df['Outcome'].astype(int)
. df.info()
```

5. Determine the success

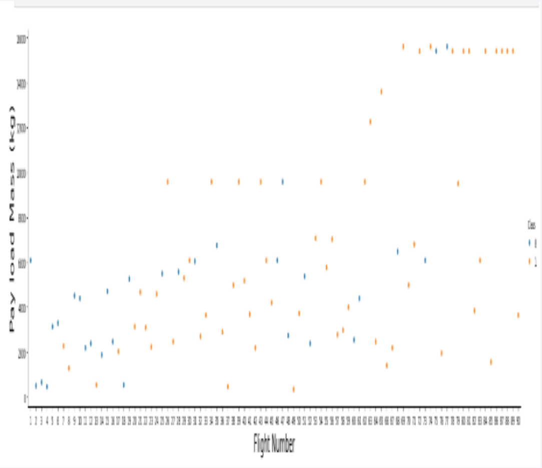
```
. df["Class"].mean()
```

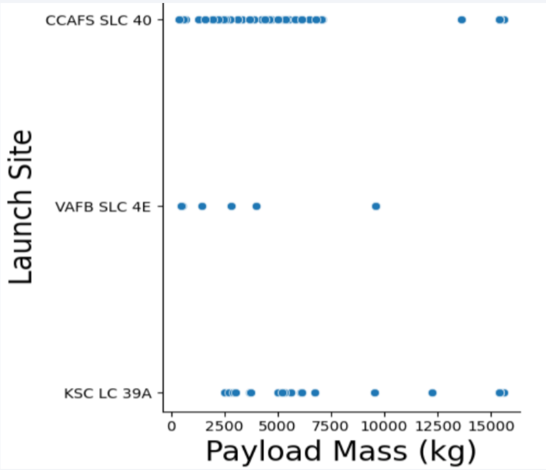
6. Convert to CSV file name.csv

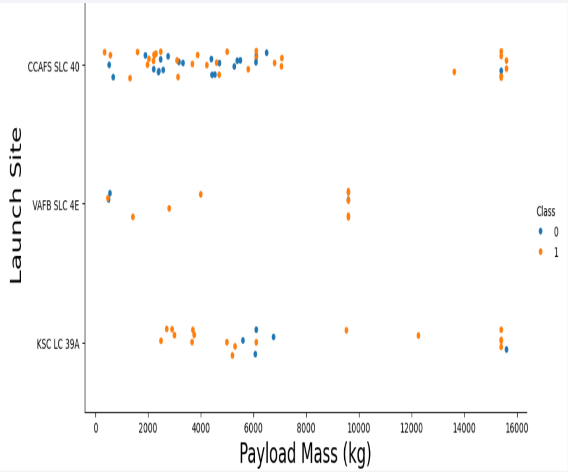
```
. df.to_csv("dataset_part_2.csv", index=False)
```

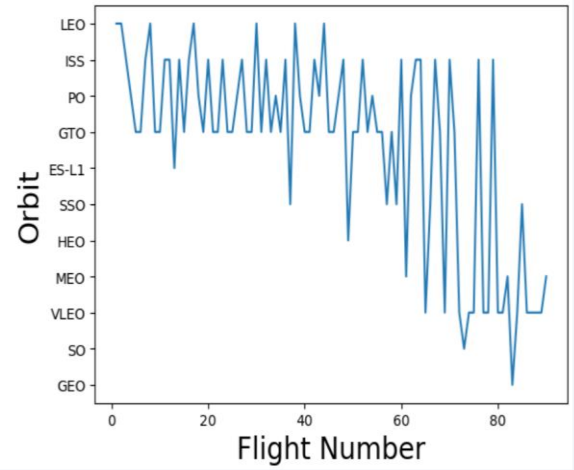
. https://github.com/kholoudbittar/Build-an-Interactive-Dashboard-with-Ploty-Dash/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

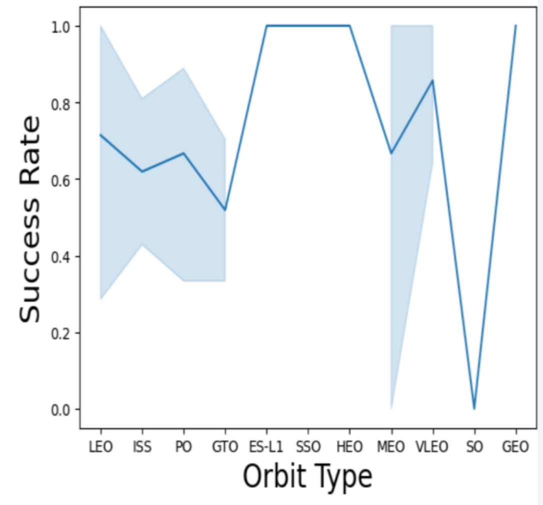
EDA with Data Visualization

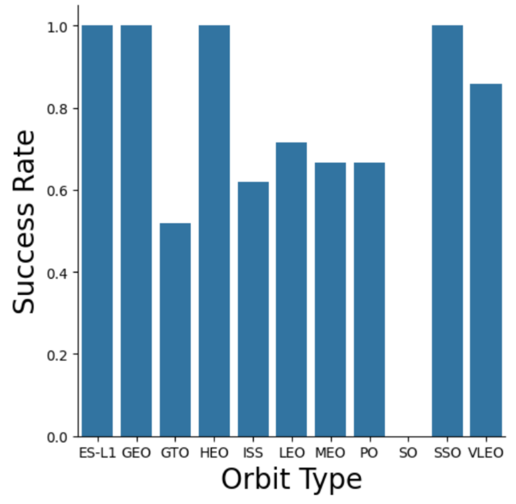


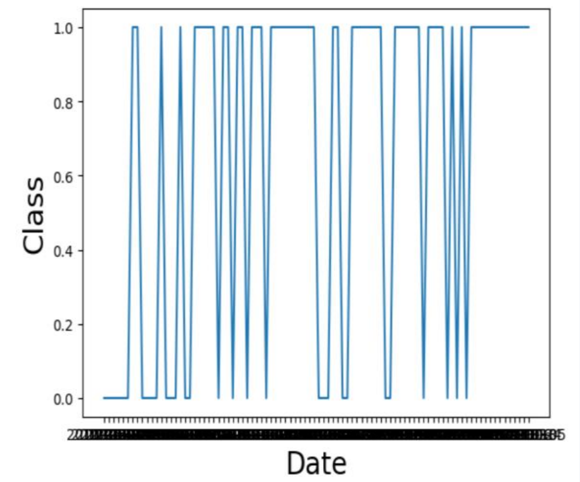


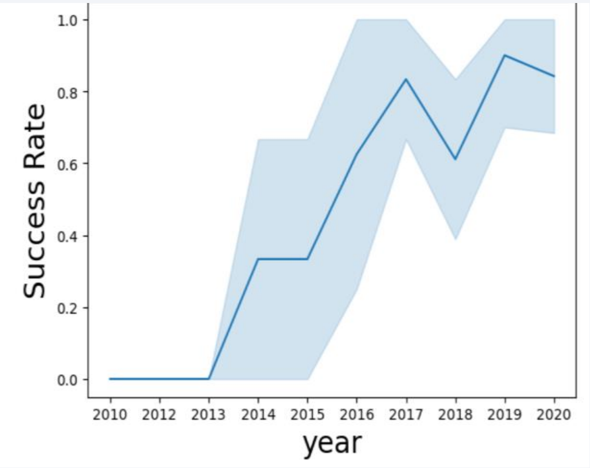












- https://github.com/kholoudbittar/Build-an-Interactive-Dashboard-with-Ploty-Dash/blob/main/edadataviz.ipynb

EDA with SQL

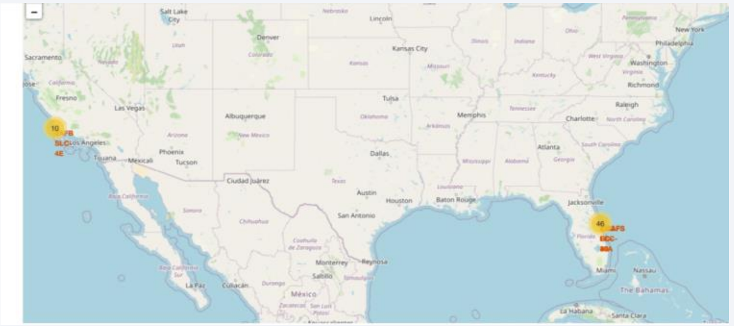
SQL Queries Include:

- Connect to the database
- 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
- display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- 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

Url= https://github.com/kholoudbittar/Build-an-Interactive-Dashboard-with-Ploty-Dash/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Folium.Circle() used to create circle above markers on the map.
- Folium.Marker() used to create marks on the map.
- `Folium.Icon()` used to create Icon on the map.
- `Folium.plugins()` used to create animated lines between the point.



- Marker.Cluster() used to simplify maps contain several markers with identical coordination.
- Folium.PolyLine() was used to create polynominal line between the point.

- https://github.com/kholoudbittar/Build-an-Interactive-Dashboard-with-Ploty-Dash/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- I install Dash and HTML to get access into graphs, tables, etc...
- I import pandas to create a dataframe.
- I import plotly to create plots to the graphs.
- I import Pie Chart and Scatter for the other shapes of the graphs presentation.
- I import dependencies for input and output.
- Dropdown was used for launch sites.

- https://github.com/kholoudbittar/Build-an-Interactive-Dashboard-with-Ploty-Dash/blob/main/README.md

Predictive Analysis (Classification)

- ❑ 1st we build the model by creating column, split data into train and test sets and build Grid Search CV model and fit the data.
- ❑ 2nd Evaluating the model by calculating the accuracies and the confusion matrixes than plot the results.
- ❑ 3rd find the best hyperparameters for the models, the best model with highest accuracy and confirm the optimal model.

❑ https://github.com/kholoudbittar/Build-an-Interactive-Dashboard-with-PlotyDash/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- ❑ The SVM, KNN and Logistic Regression models are better in prediction accuracy than Trees.
- ❑ Low weighted payloads is better than the heavier ones.
- ❑ The success rates for SpaceX launches will be better in years.
- ❑ KSC LC 39A had the best successful launches from all.
- ❑ Orbit GEO,HEO,SSO,ES L1 has the best success rate.

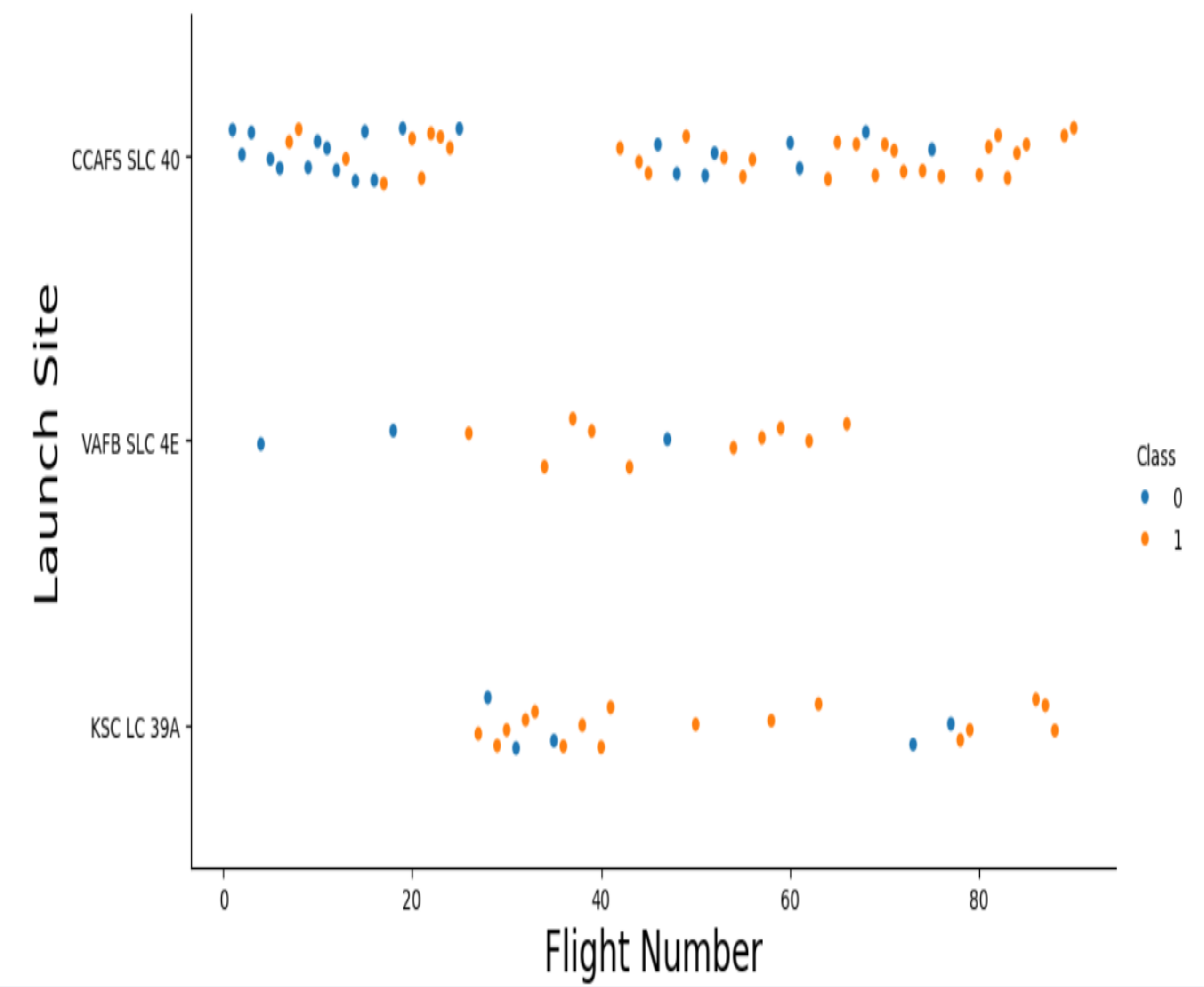
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- ❑The Scatter plot of Flight Number vs. Launch Site

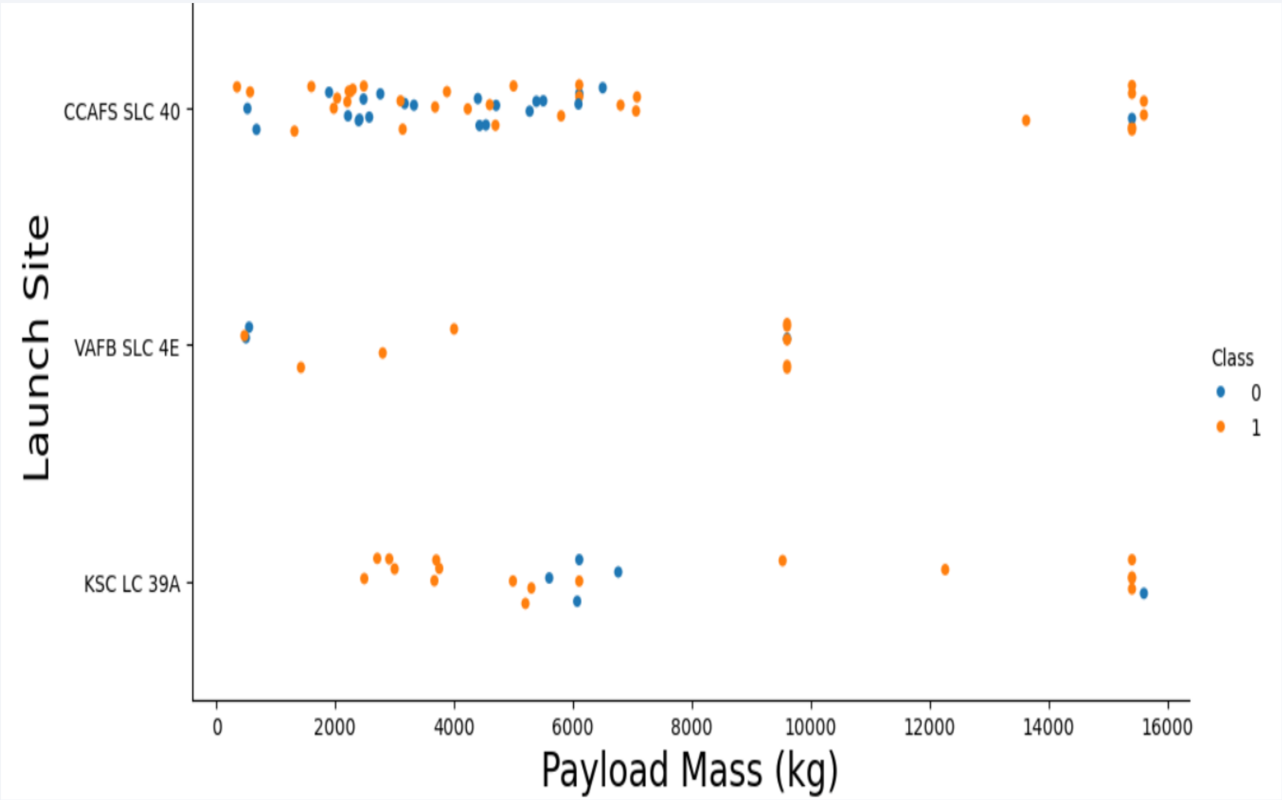
- ❑We can see that CCAFS SLC 40 are higher than the launches from other sites.



Payload vs. Launch Site

- The Scatter plot of Payload vs. Launch Site

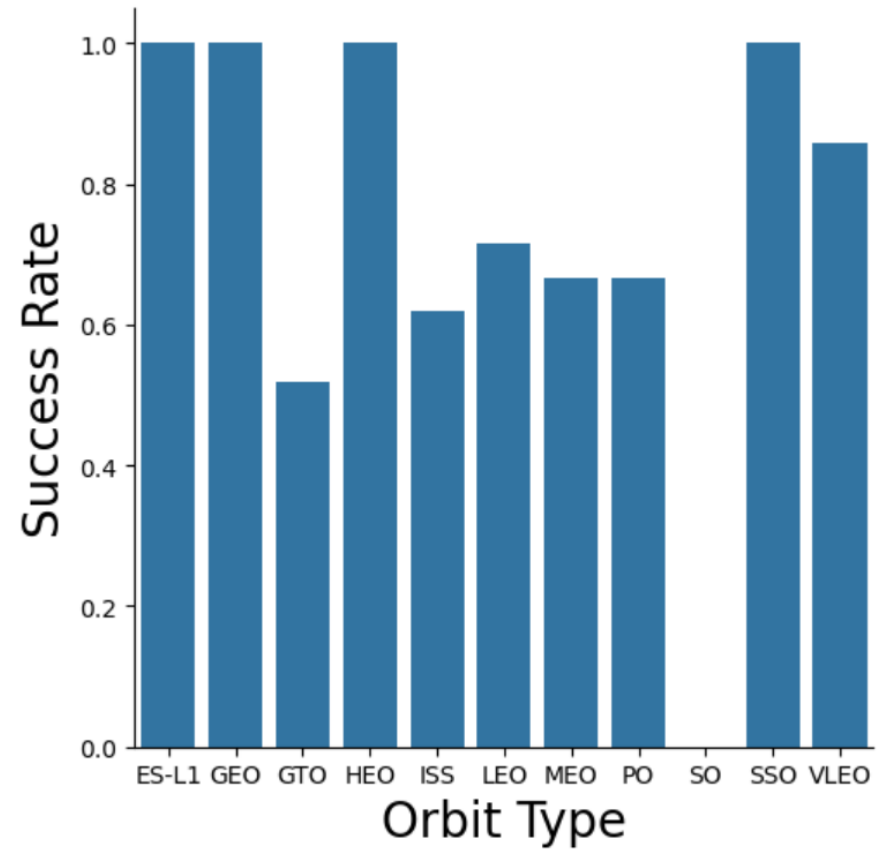
- In VAFB-SLC launch site there are no rockets launched for heavy payload Mass(greater than 10000).
- With the increase of the payload Mass the launch sites decrease.



Success Rate vs. Orbit Type

- The bar chart for the success rate of each orbit type

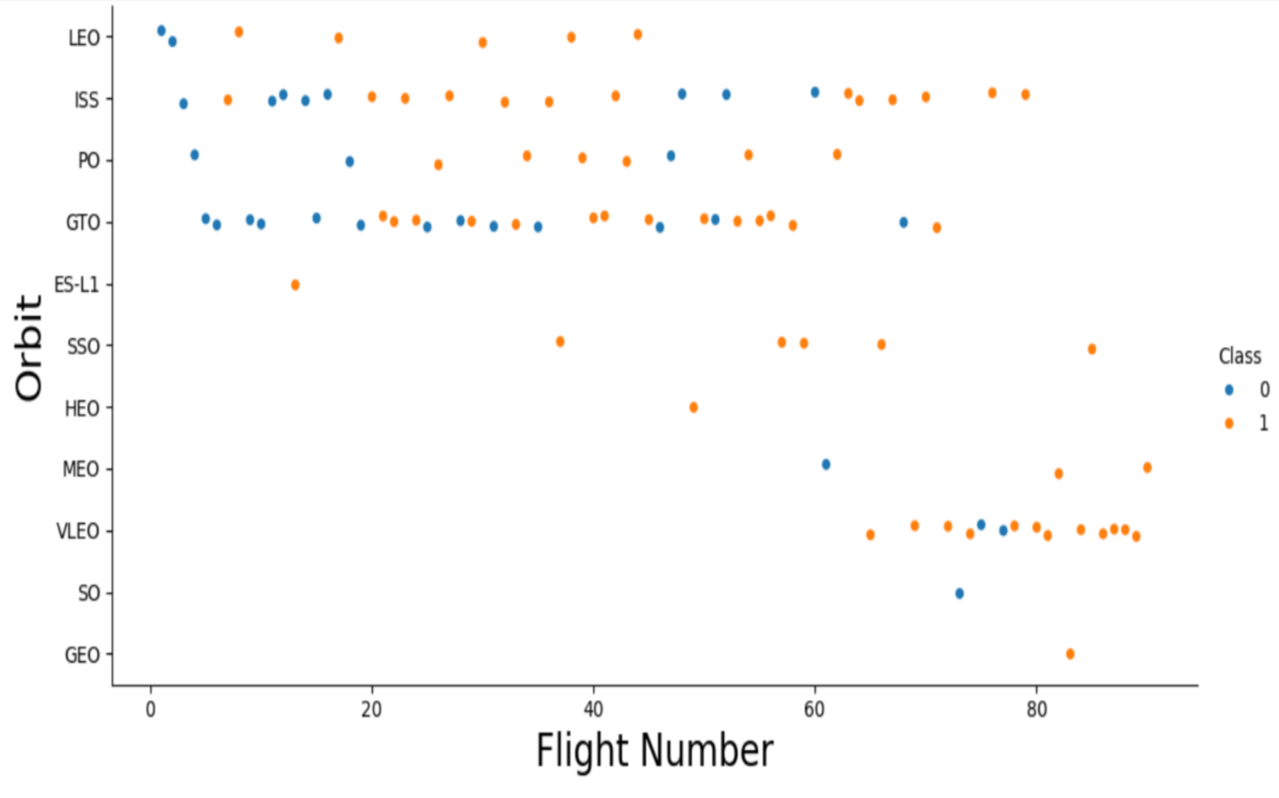
- We can see that the Orbit types ES_L1, GEO, HEO and SSO have the most successful rate up to 100%, on the other side the Orbit type SO have a null success rate.



Flight Number vs. Orbit Type

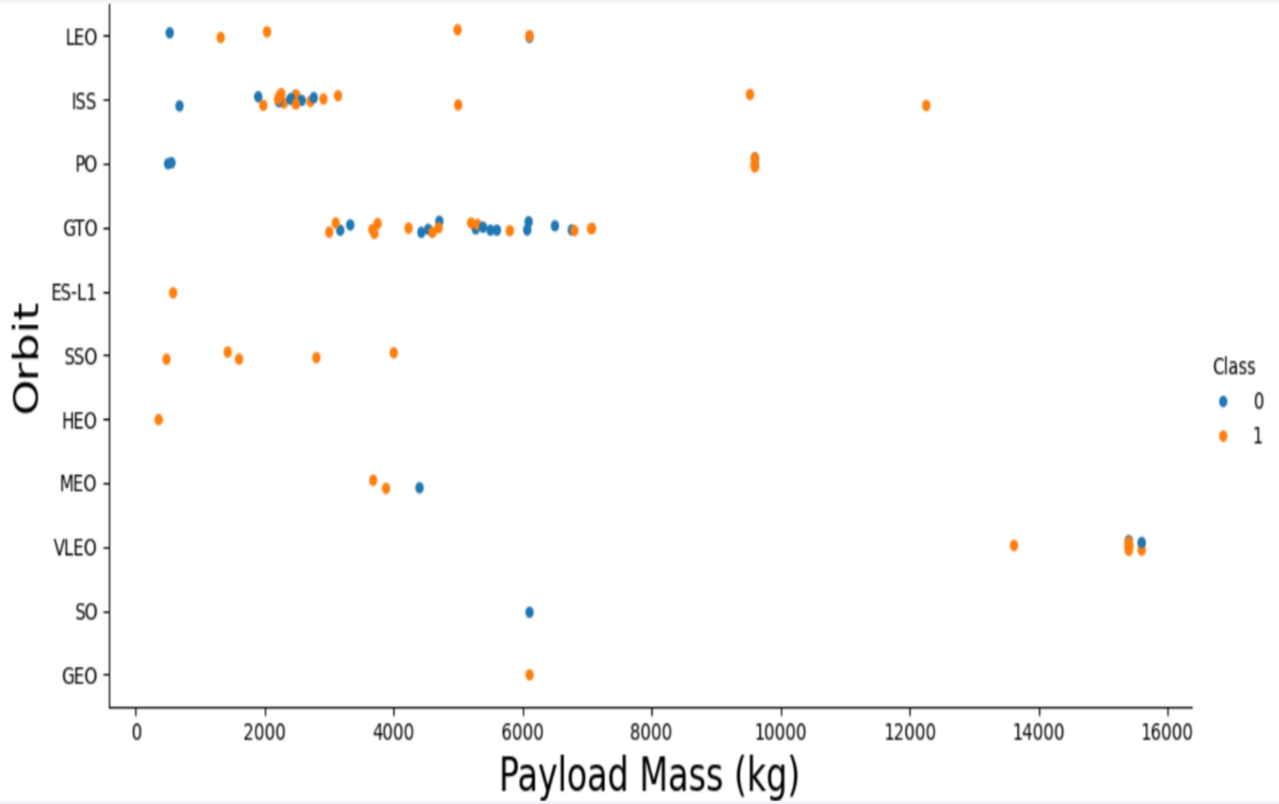
- ❑ The scatter point of Flight number vs. Orbit type

- ❑ You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit



Payload vs. Orbit Type

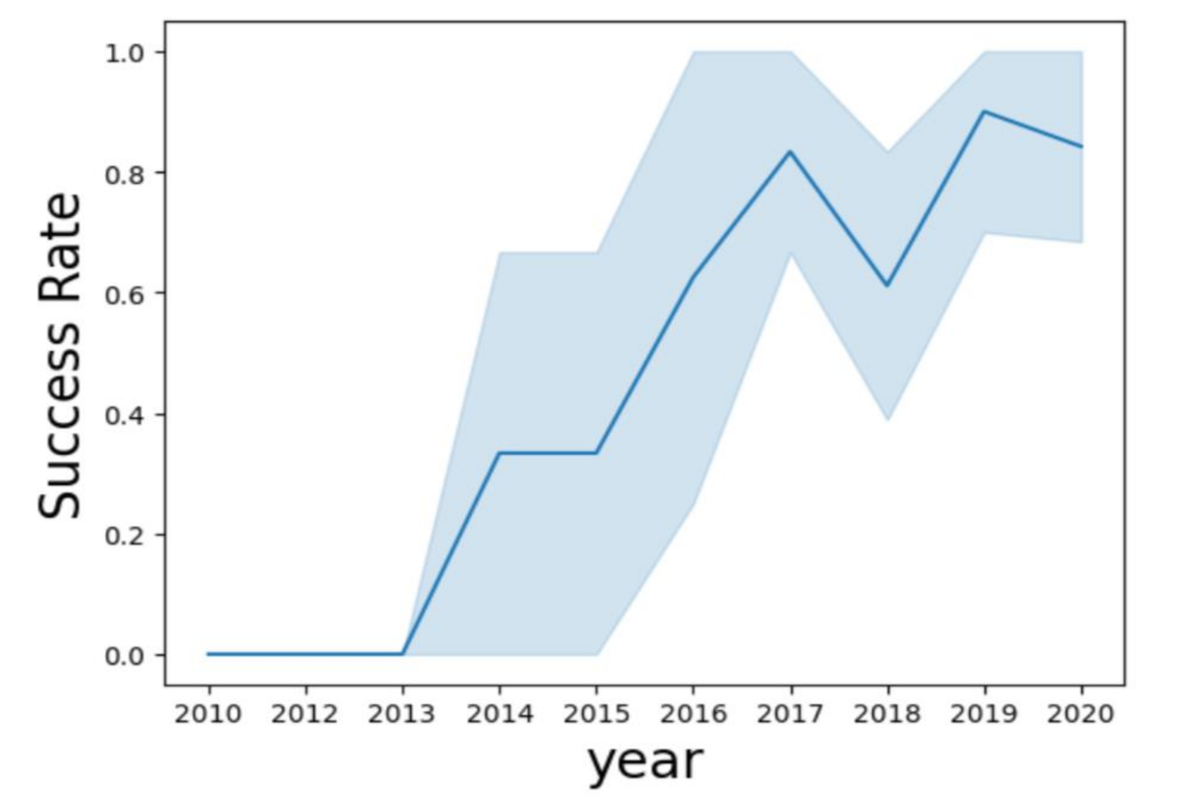
- The scatter point of payload vs. orbit type
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



Launch Success Yearly Trend

- The line chart of yearly average success rate

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



All Launch Site Names

- %sql SELECT DISTINCT LAUNCH_SITE as 'launch_sites' from SpaceX.

LAUNCH_SITES
CCAFS LC_40
CCAFS SLC_40
KSC LC_39A
VAFB SLC_4E

We can get the unique value by using 'DISTINCT' command

Launch Site Names Begin with 'CCA'

- ❑ %sql SELECT * from SpaceX where LAUNCH_SITE like 'CAA'% LIMIT 5;

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad
0	1	2010	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN
1	2	2012	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN
2	3	2013	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN
3	4	2013	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN
4	5	2013	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN

- ❑ LIMIT 5

Total Payload Mass

❑Calculate the total payload carried by boosters from NASA

❑ **%sql** SELECT SUM(PAYLOAD_MASS__KG_) \

❑ FROM SPACEXTBL \

❑ WHERE CUSTOMER = 'NASA (CRS)';

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- **%sql** SELECT AVG(PAYLOAD_MASS__KG_) \
- FROM SPACEXTBL \
- WHERE BOOSTER_VERSION = 'F9 v1.1';

AVG(PAYLOAD_MASS__KG_)
2928.4

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 = 'Success (ground pad)'

First Succesful Landing Outcome in Ground Pad
2015-12-22

- **We select min(date) because it's the same of the first Date.**

Successful Drone Ship Landing with Payload between 4000 and 6000

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

- ❑ **%sql** SELECT PAYLOAD \
- ❑ FROM SPACEXTBL \
- ❑ WHERE LANDING__OUTCOME = 'Success (drone ship)' \
- ❑ AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;

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

Total Number of Successful and Failure Mission Outcomes

- ❑Calculate the total number of successful and failure mission outcomes
- ❑ **%sql** SELECT MISSION_OUTCOME, COUNT(*) as total_number \
- ❑ FROM SPACEXTBL \
- ❑ GROUP BY MISSION_OUTCOME;

Mission_Outcome	total_number
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- ❑ We have 98+1+1=100 Success
- ❑ We have a 1 Failure.

Boosters Carried Maximum Payload

- ❑List the names of the booster which have carried the maximum payload mass
- ❑ **%sql** SELECT BOOSTER_VERSION \
- ❑ FROM SPACEXTBL \
- ❑ WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);

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

2015 Launch Records

- ❑List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- ❑ **%sql** SELECT substr(Date,4,2) as month, DATE,BOOSTER_VERSION, LAUNCH_SITE, [Landing _Outcome] \
- ❑ FROM SPACEXTBL \
- ❑ where [Landing _Outcome] = 'Failure (drone ship)' **and** substr(Date,7,4)='2015';

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

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(*) as count_outcomes \
- FROM SPACEXTBL \
- WHERE DATE between '04-06-2010' **and** '20-03-2017' group by [Landing _Outcome] order by count_outcomes DESC;

2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Section 3

Launch Sites Proximities Analysis

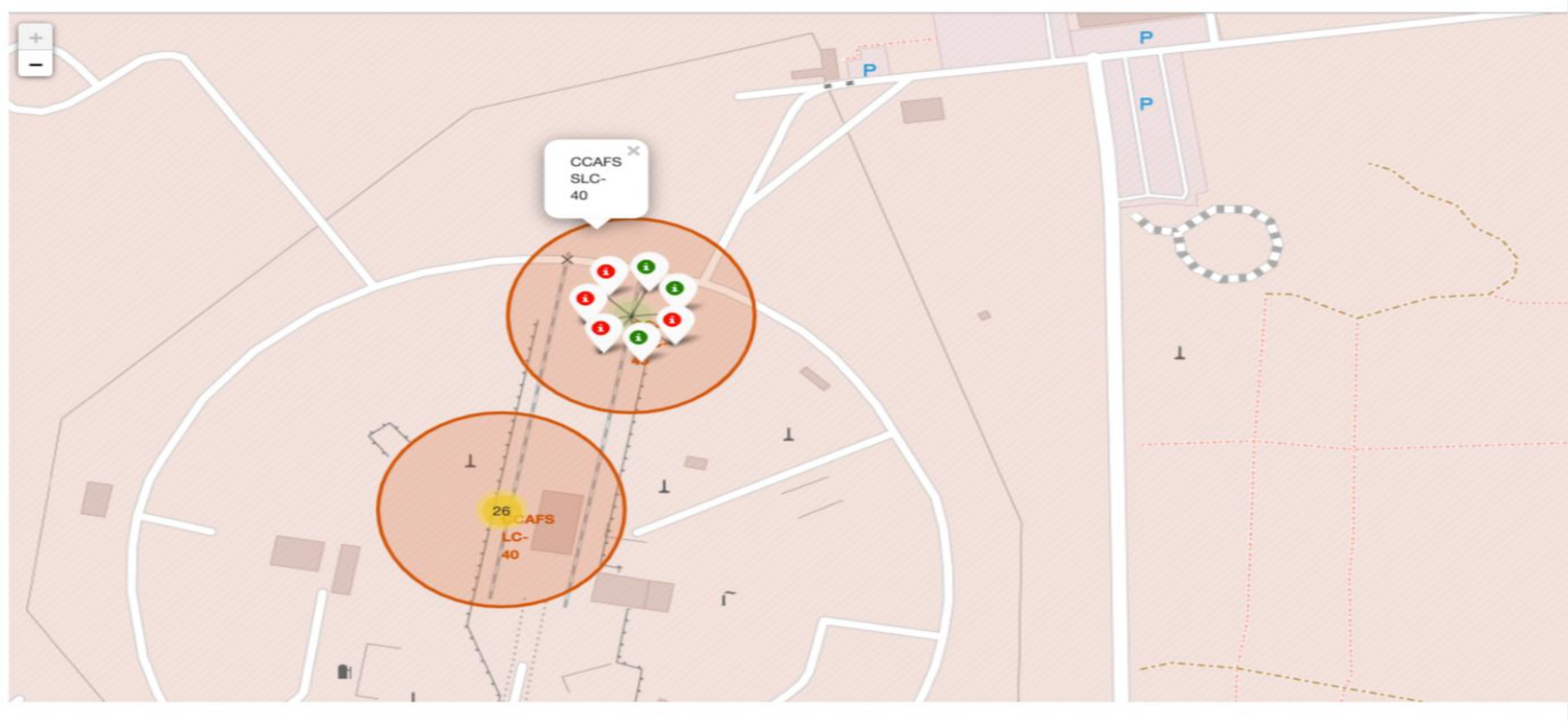
<Folium Map Screenshot 1>

The generated map with marked launch sites should look similar to the following:

- ❑All the Launches are in USA, Florida and California.

<Folium Map Screenshot 2>

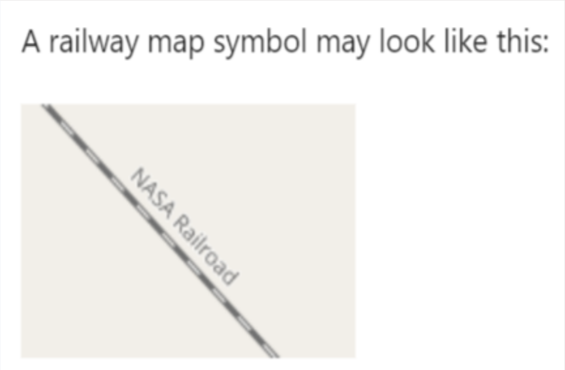
- ❑Green mean Success

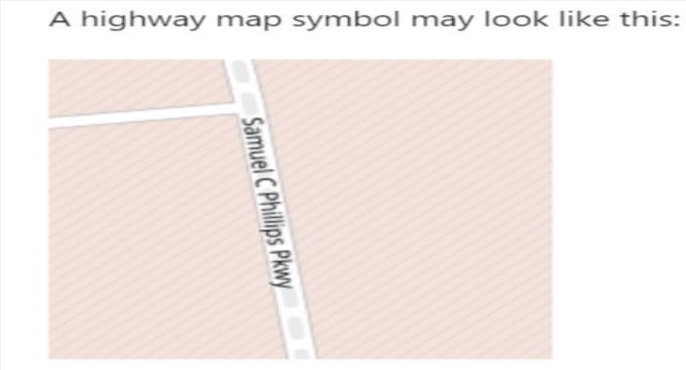


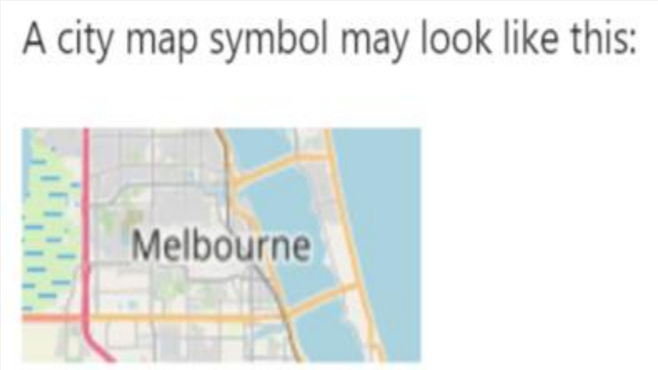
- ❑Red mean Failure

<Folium Map Screenshot 3>

❑Railway, Highway and City map and Distance.



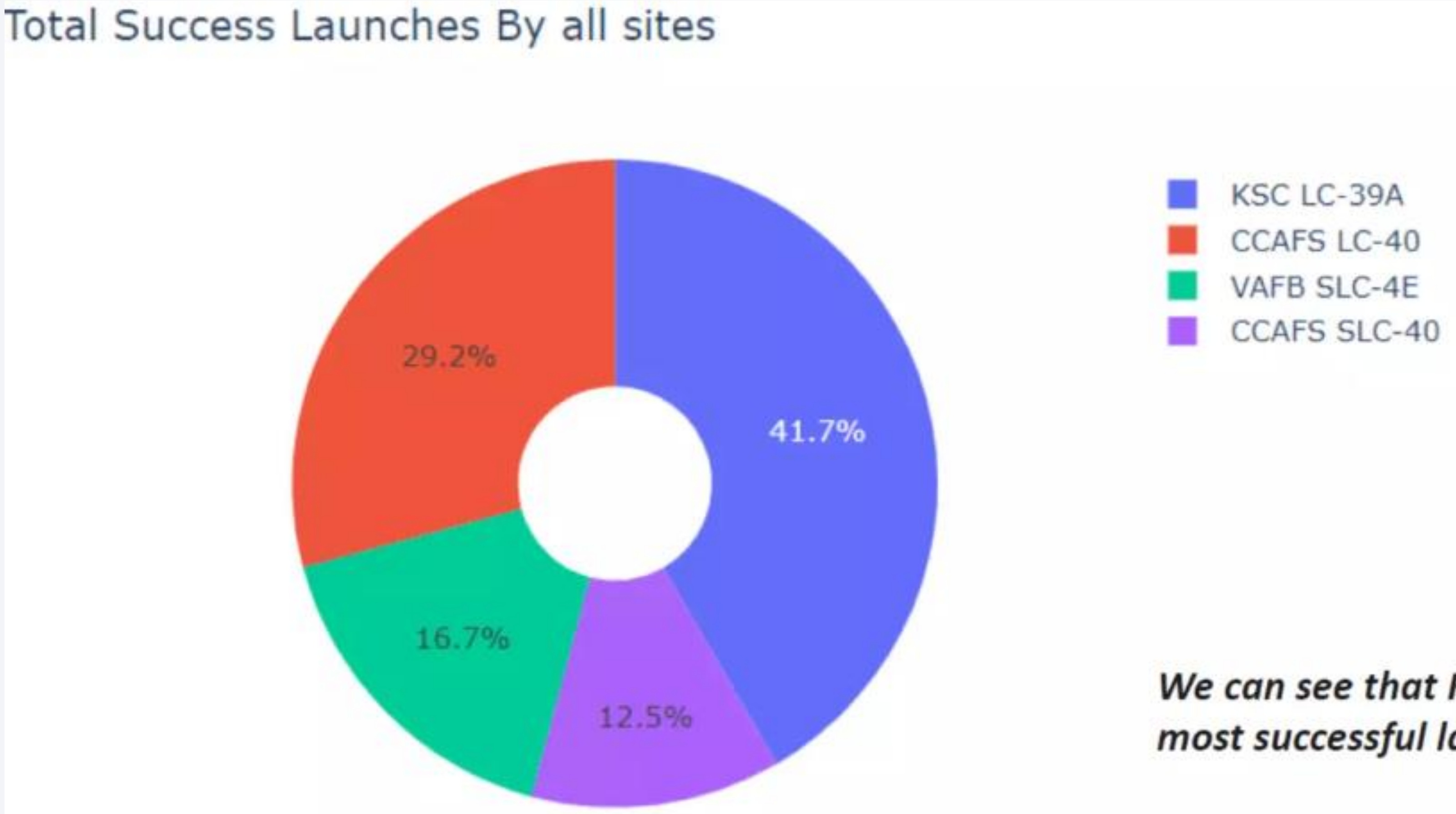




Section 4

Build a Dashboard with Plotly Dash

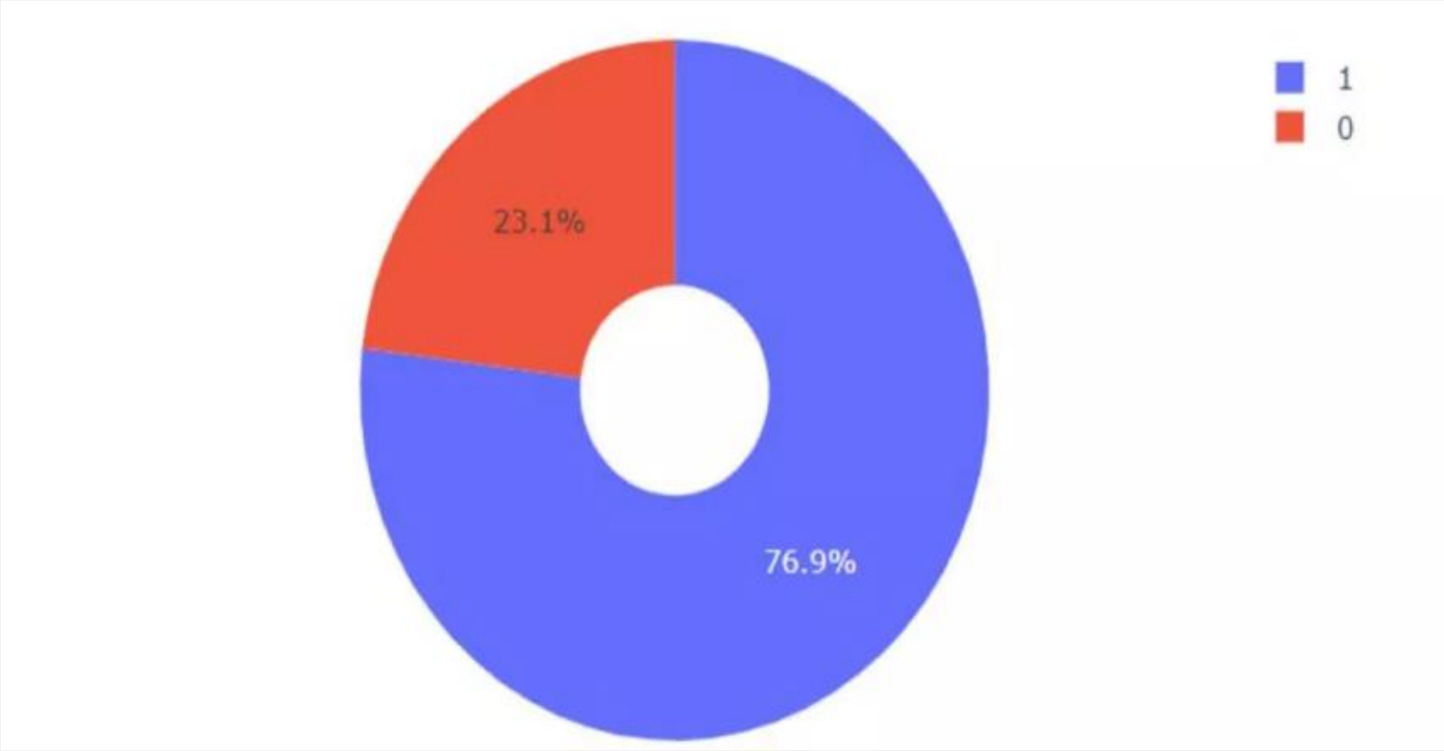
<Dashboard Screenshot 1>



We can see that KSC LC-39A had the most successful launches from all the sites

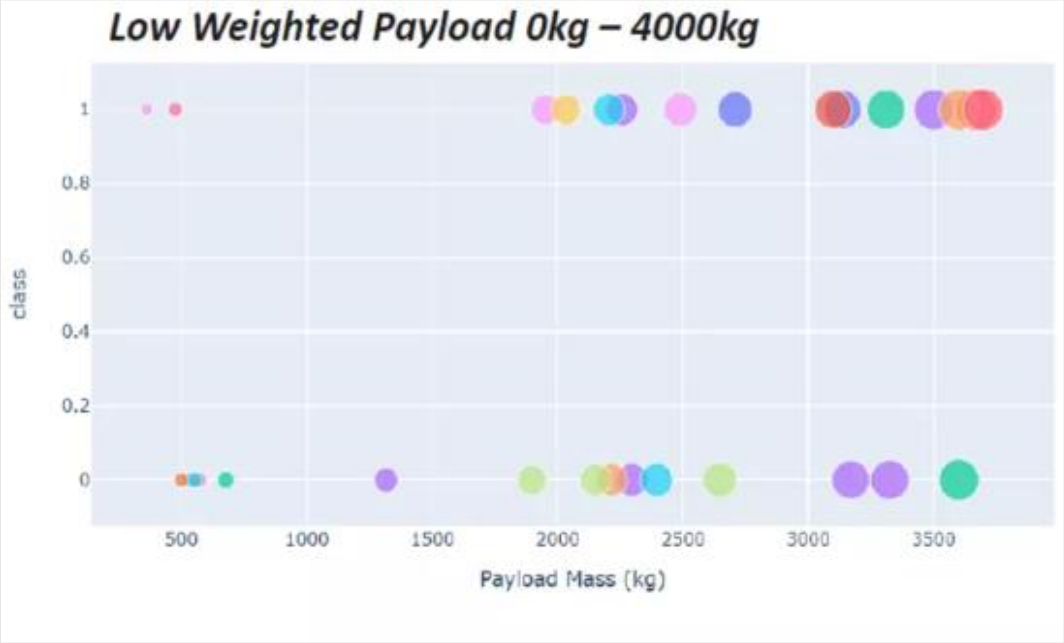
<Dashboard Screenshot 2>

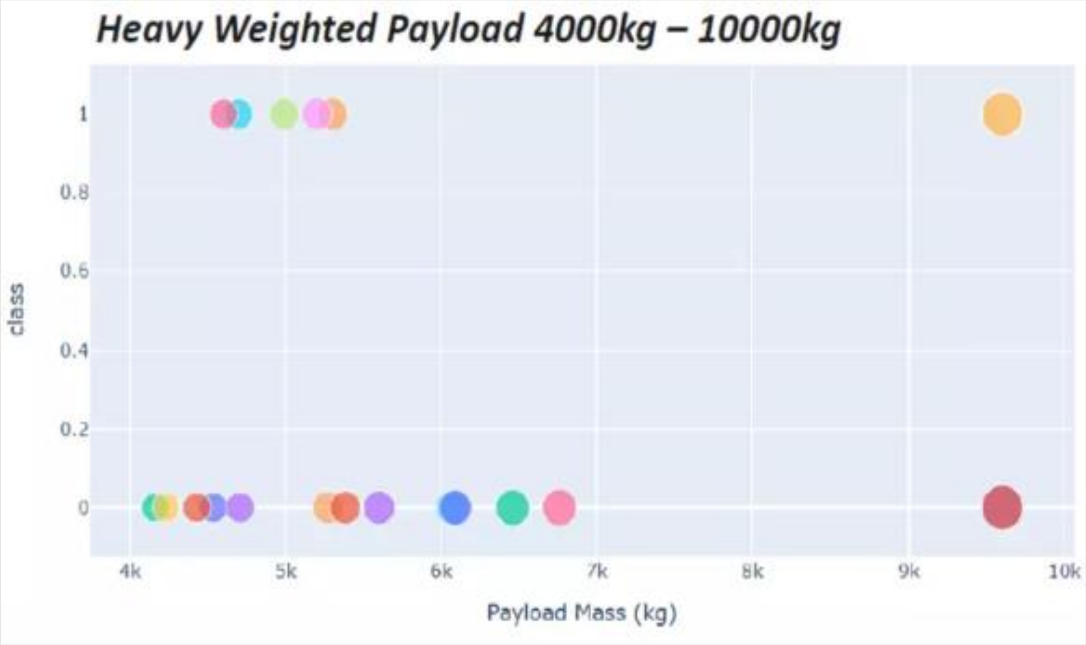
❑KSC LC-39A achieve a 76.9% Success rate while getting a 23.1% Failure.



<Dashboard Screenshot 3>

- ❑ We can see that the Success for the low weighted playload is better than the heavy weighted playload.



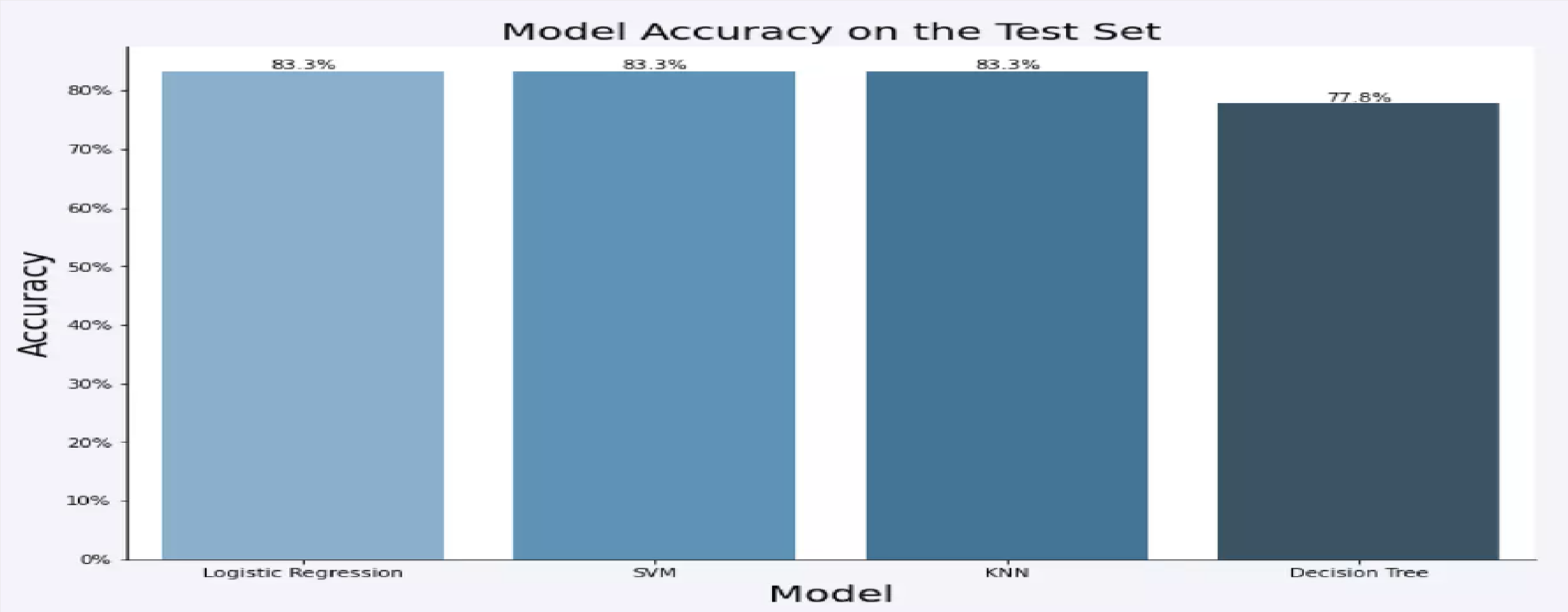


Section 5

Predictive Analysis (Classification)

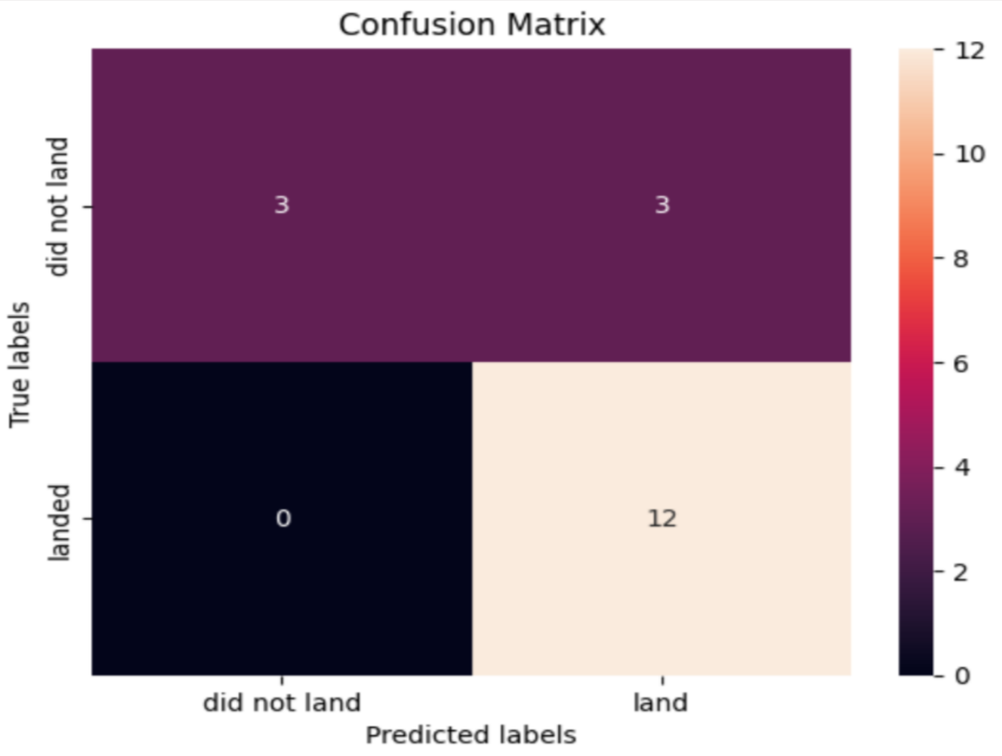
Classification Accuracy

❑Decision Tree is the lowest between all of them.



Confusion Matrix

- Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives.



Conclusions

- ❑Point 1: KNN, SVM and Logistic Regression models are better than the tree model accuracy.
- ❑Point 2: Low weighted playload are more success than the Heavy weighted playload.
- ❑Point 3: The Success Rate for SpaceX launches in years are better in time.
- ❑Point 4: KSC LC_39A had the most successful launches from all sites.
- ❑Orbit GEO,HEO,SSO,ES L1 HAS THE BEST SUCCESS RATE.

Appendix

- https://github.com/kholoudbittar/Build-an-Interactive-Dashboard-with-Ploty-Dash/blob/main/README.md

Thank you!