IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

PANACEA TEVERAH
02 JULY 2022

Outline

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

Section 1

Executive Summary

EXECUTIVE SUMMARY

Summary of methodologies

- Data Collection
- Data Wrangling
- EDA with SQL and data visualization
- Building interactive dashboard with Plotly
- Building interactive map with Folium
- Predictive Analysis

Summary of all results

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

Introduction

Project background and context

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

Problems that need answers

- What influences if the rocket will land successfully?
- What conditions does SpaceX need to have to ensure the best rocket success landing rate

Methodology

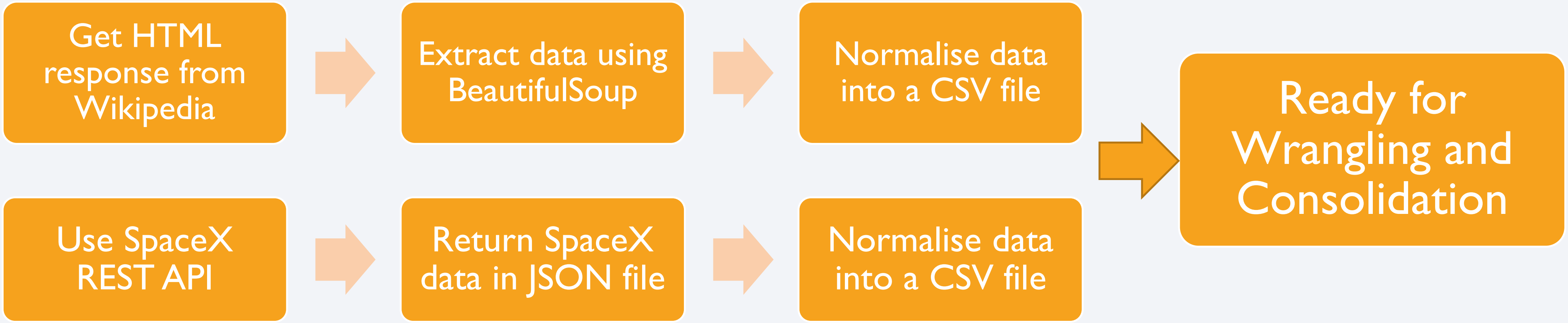
Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping from Wikipedia

- Perform data wrangling (that is transforming data for machine learning)
  - One hot encoding data fields for machine learning and dropping irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL;
  - Plotting : Scatter graphs, Bar graphs to show relationship between variables to show patterns of data
- 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

- The following data sets were collected:
  - SpaceX launch data was gathered from 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
- The SpaceX REST API endpoints , or URL , starts with api.spacexdata.com/v4/.
- Falcon 9 launch data was also found by web/scrapping using BeautifulSoup.



Data Collection – SpaceX API

1.Getting a response from API

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

2.converting a response to a .JSON file

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

3.Apply custom files to clean data

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

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

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

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

4.Assign list to a dictionary then data frame

```
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
data2 = pd.DataFrame.from_dict(launch_dict)
```

5.Filter dataframe and export to a flat file (.csv file)

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = data2[data2['BoosterVersion']!='Falcon 1']
```

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

- https://github.com/Panacea020/Data-Science-Capstone-project-SpaceX/blob/main/REST%20API%20data%20collection.ipynb

Data Collection – Web Scraping from Wikipedia

1.Getting Response from HTML

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

2.Creating BeautifulSoup object

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html')
```

3.Finding tables

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

4.Getting column names

```
column_names = []

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

5.Creation of 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']=[]
```

6.Appending data to keys

```
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.`
            #print(flight_number)
            launch_dict['Flight No.'].append(flight_number)

            datatimelist=date_time(row[0])

            # Date value
            # TODO: Append the date into launch_dict with key `Date`
            date = datatimelist[0].strip(',')
            #print(date)
            launch_dict['Date'].append(date)
```

7.Converting dictionary to Dataframe

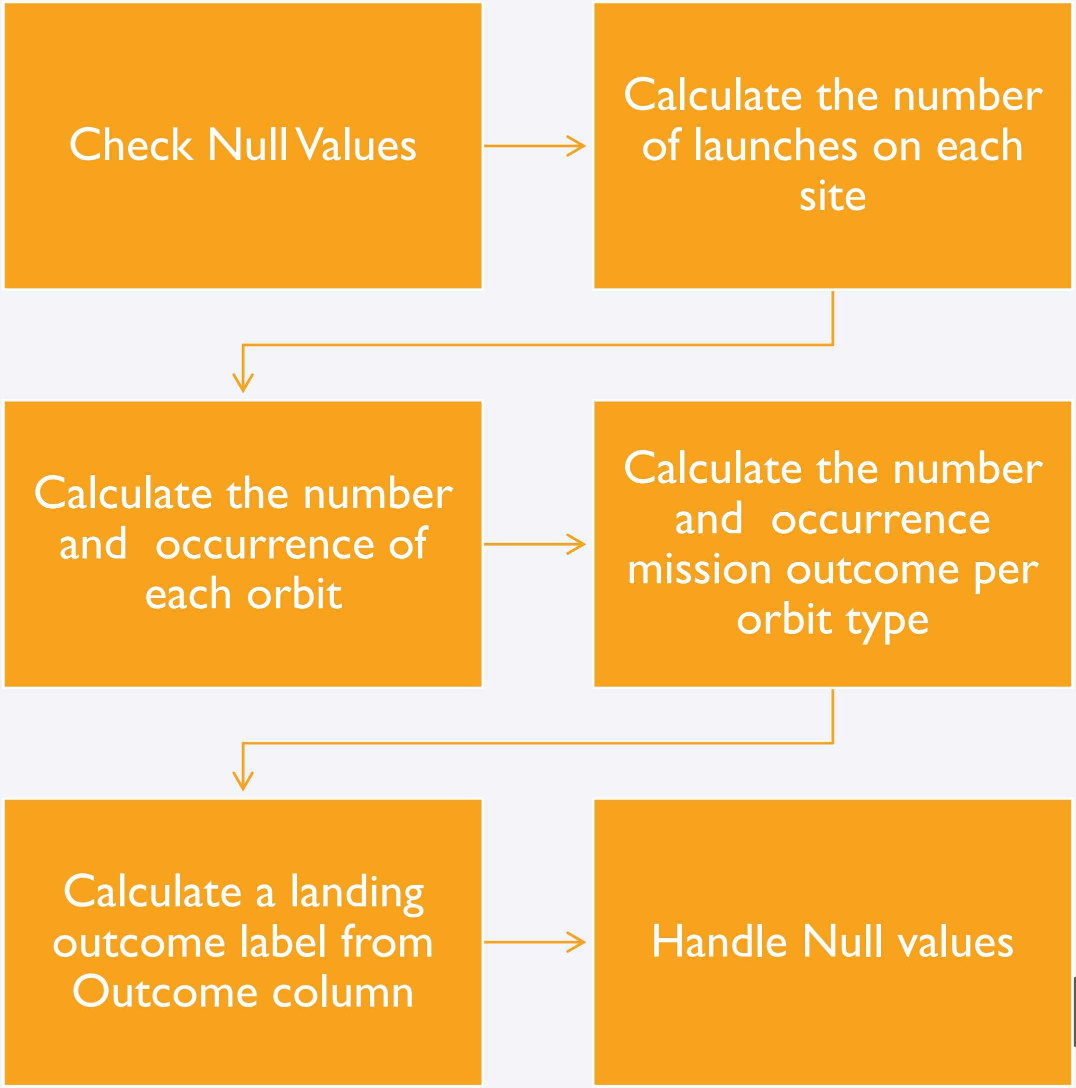
```
launch_dict
df=pd.DataFrame(launch_dict)
df
```

8.Data frame to .csv

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

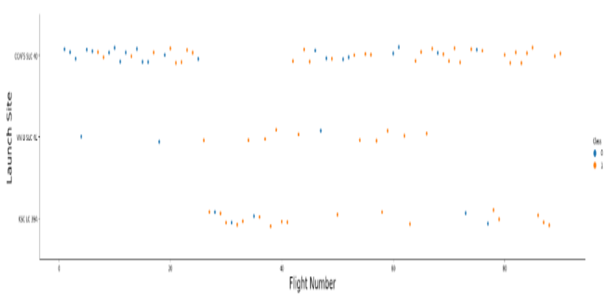
- https://github.com/Panacea020/Data-Science-Capstone-project-SpaceX/blob/main/Web%20scrap%20data%20-%20Wikipedia.ipynb

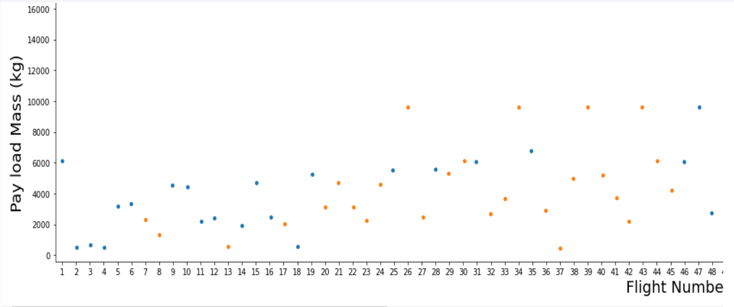
Data Wrangling - EDA

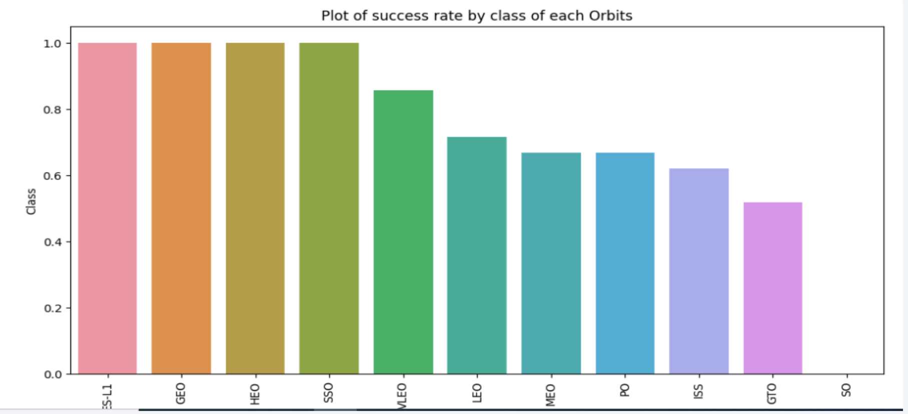


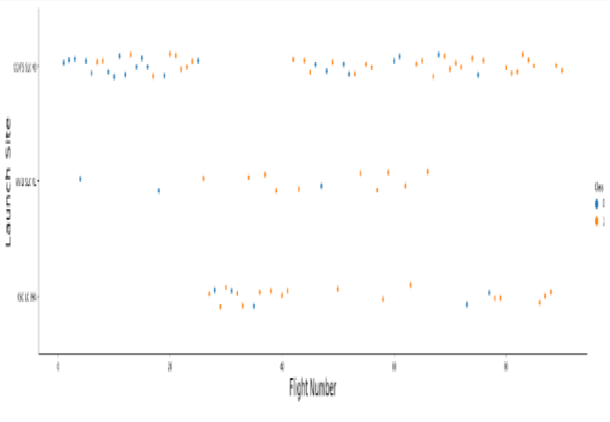
- https://github.com/Panacea020/Data-Science-Capstone-project-SpaceX/blob/main/REST%20API%20data%20collection.ipynb

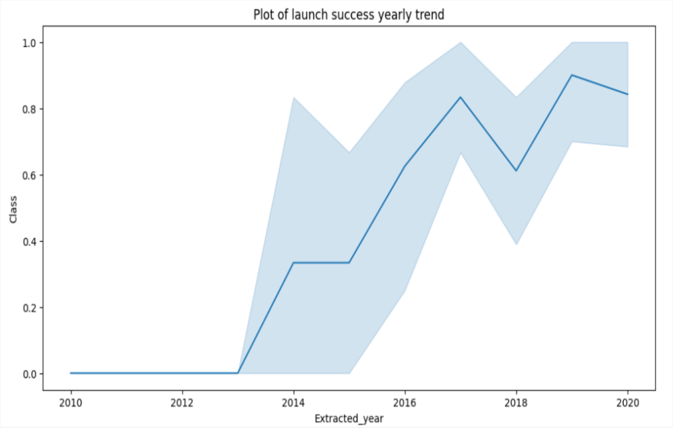
EDA with Data Visualization

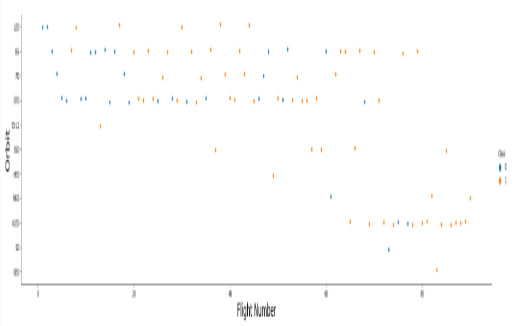


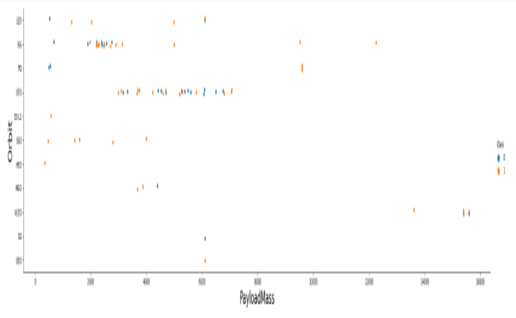












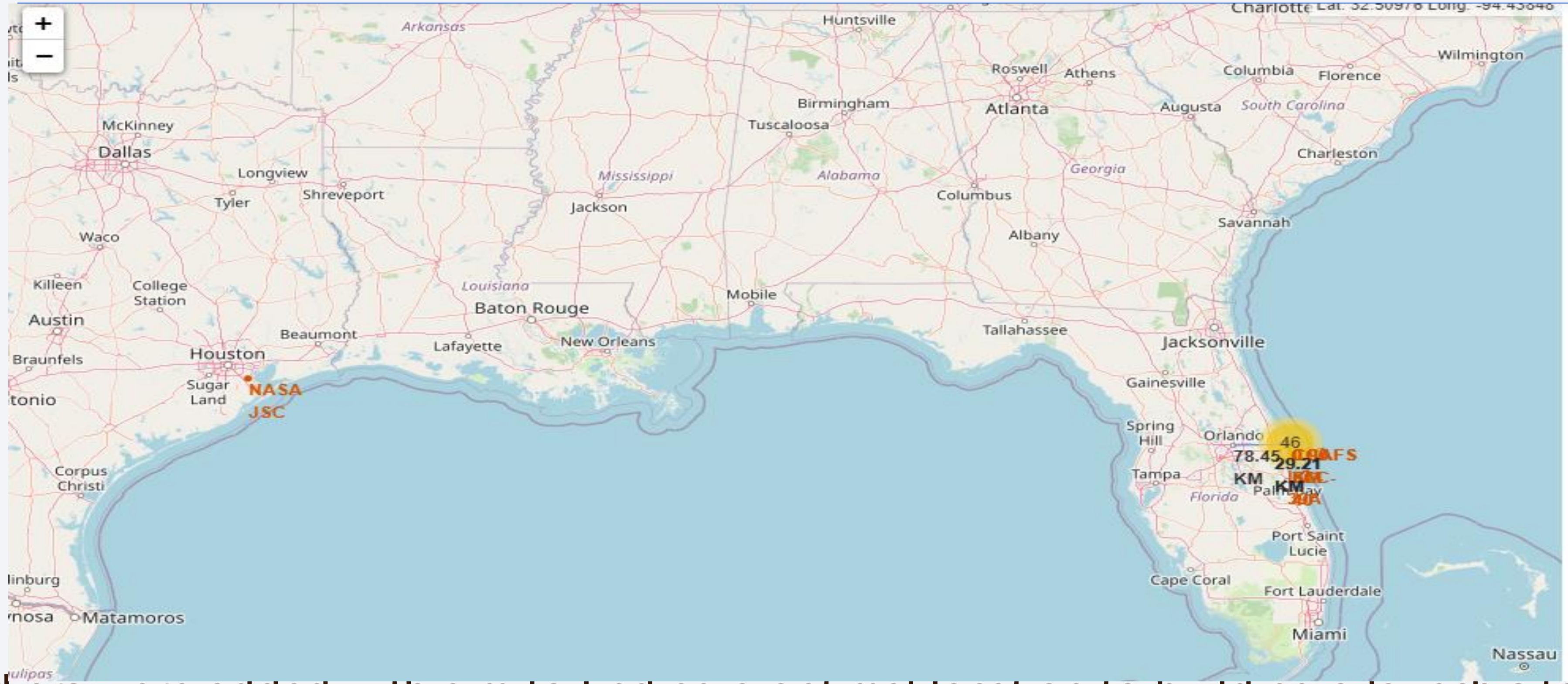
- https://github.com/Panacea020/Data-Science-Capstone-project-SpaceX/blob/main/EDA%20with%20Data%20Visualization%20spaceX.ipynb

EDA with SQL

- SQL queries performed include
  - Displaying the names of the unique launch site in the space mission
  - Displaying top 5 records where launch sites begin with the string 'KSC'
  - Displaying the total payload mass carried by booster version F9 V1.1
  - Listing the date where the successful landing outcome in drone ship was achieved.
  - Listing the name of Boosters which had success in ground and have payload mass greater than 4000 but less than 6000
  - Listing a total number of successful and failure mission outcomes
  - Listing of booster versions which have carried the maximum payload mass
  - Listing the record which will display the month names, successful landings outcomes in ground pad, booster versions, launch site for the months in 2018
  - Ranking the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

- https://github.com/Panacea020/Data-Science-Capstone-project-SpaceX/blob/main/IBM-EDA%20with%20SQL.ipynb%20at%20main%20%C2%B7%20chuksoo_IBM-Data-Science-Capstone-SpaceX.html

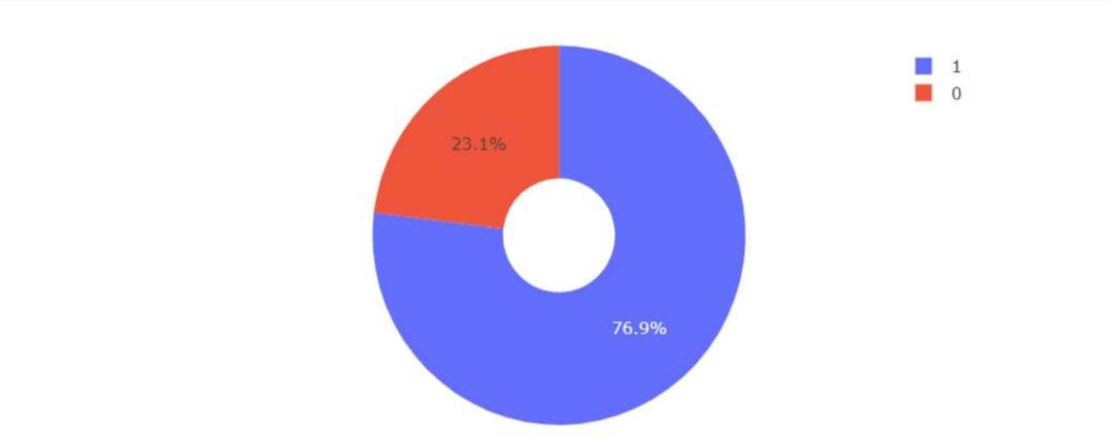
Interactive Map with Folium



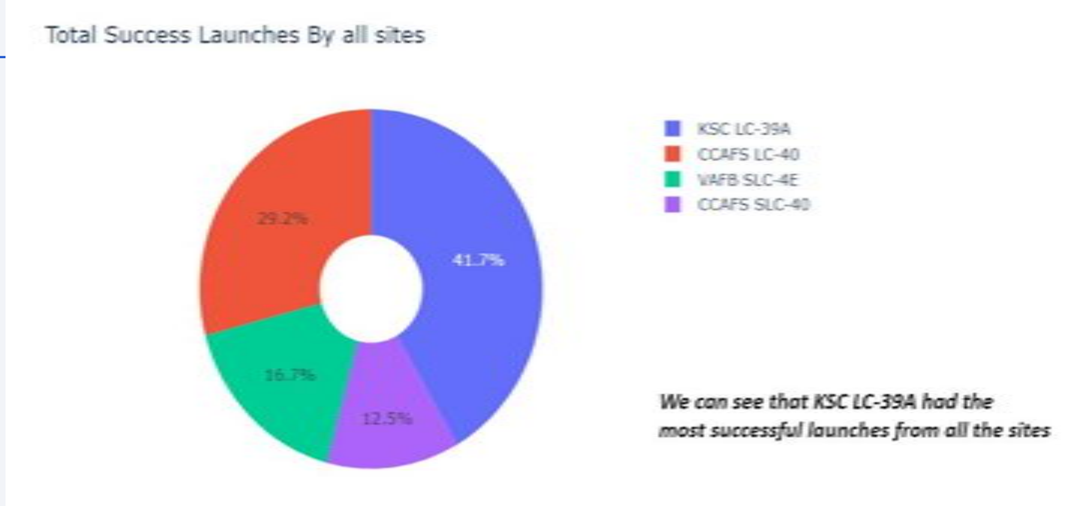
- Markers were added with aim to finding a optimal location to building a launch site.
- https://github.com/Panacea020/Data-Science-Capstone-project-SpaceX/blob/main/INTERACTIVE%20FOLIUM%20MAP.ipynb

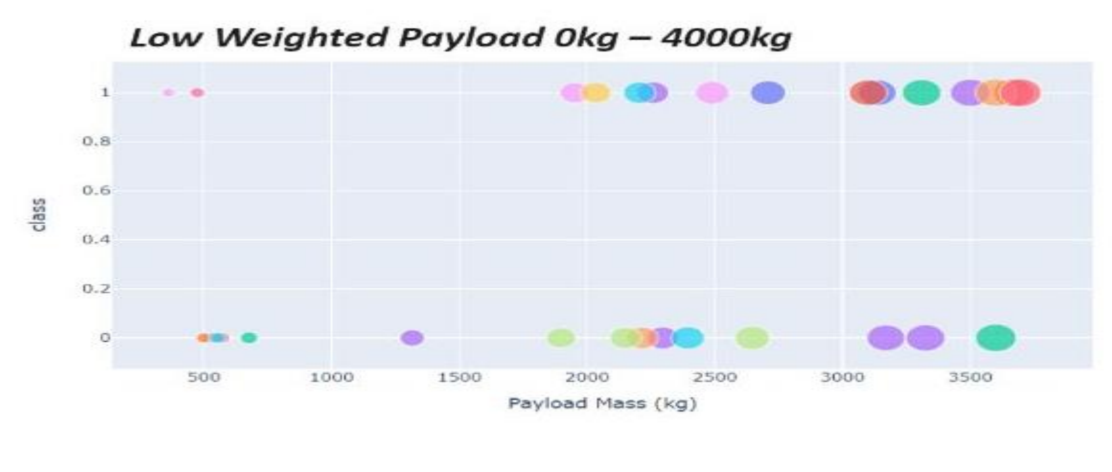
Build a Dashboard with Plotly Dash

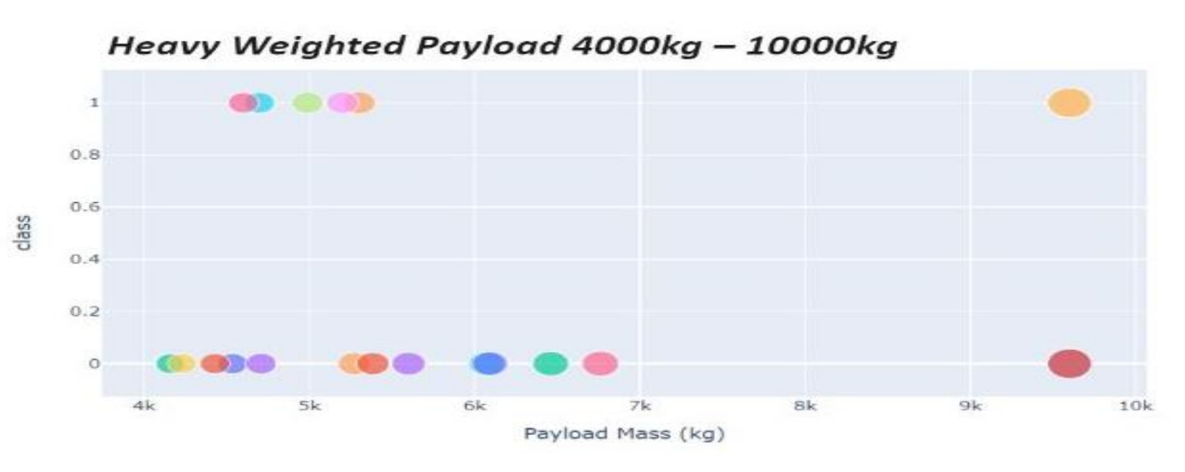
https://github.com/Panacea020/Data-Science-Capstone-project-SpaceX/blob/main/SpaceX%20plotly.html



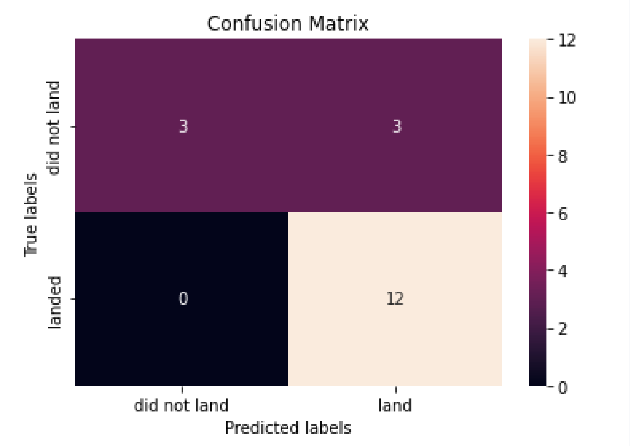
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

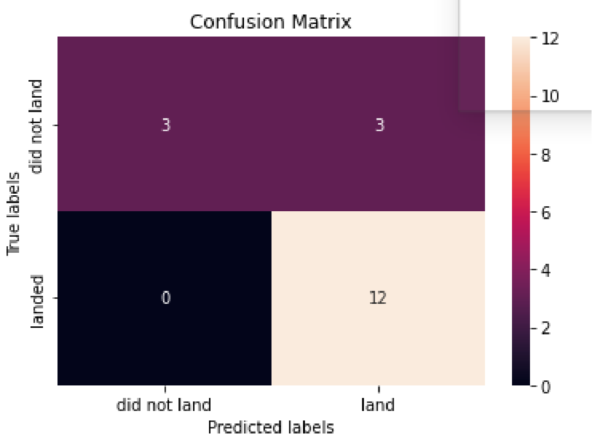


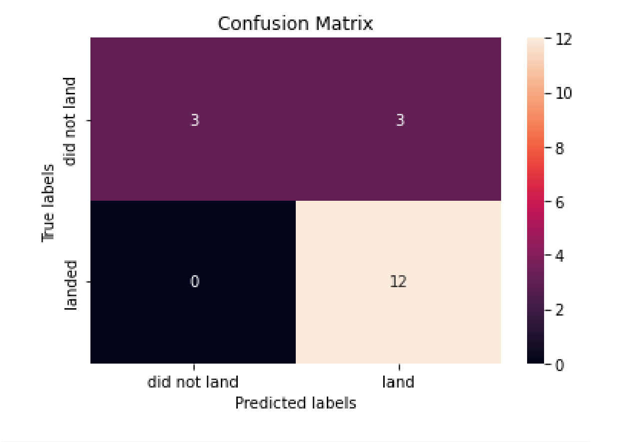


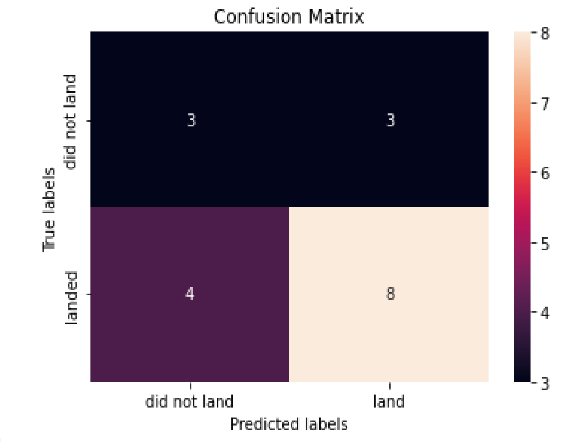


Predictive Analysis (Classification)









Find the method performs best:

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.8892857142857145
Best Params is : {'criterion': 'gini', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10,
 'splitter': 'best'}
```

- The SVM ,KNN and logistics regression produced the highest accuracy at 88.9%. SVM performs best in terms of area under curve at 0.95
- https://github.com/Panacea020/Data-Science-Capstone-project-SpaceX/blob/main/Predictive%20Analysis%20ML.ipynb

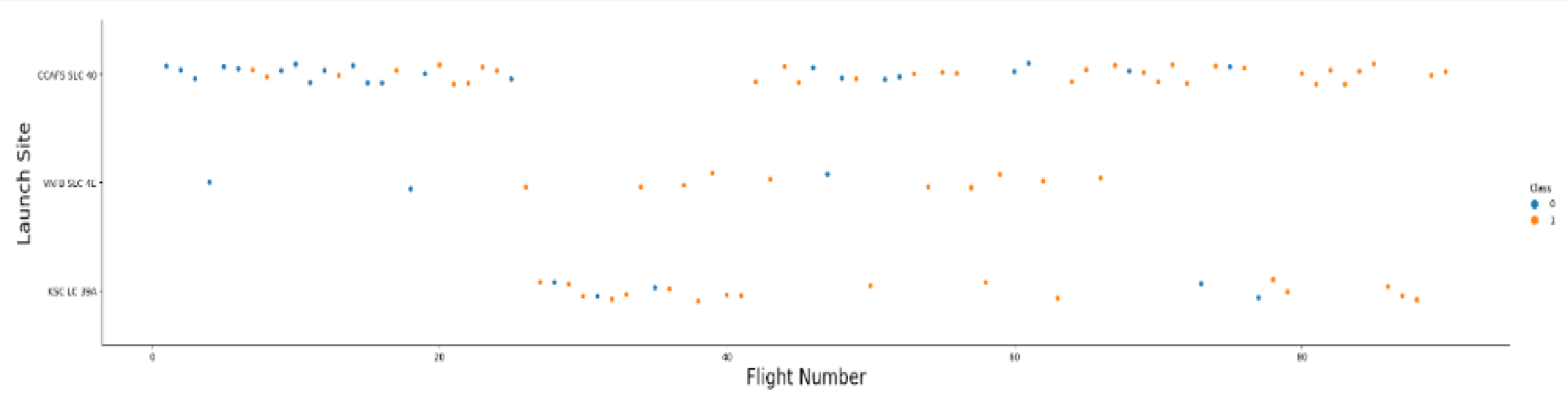
Results

- The SVM , KNN, and logistic Regression are the best in terms of prediction accuracy for this SpaceX dataset.
- Low weighted payloads perform better than the heavier payloads.
- The success rate for Space X launches is directly proportional time in years they will eventually perfect the launches.
- KSC LC 39A had the successful Launches from the sites.
- Orbit GEO, HEO, SSO ,ES L1 have the best Success rate

Section 2

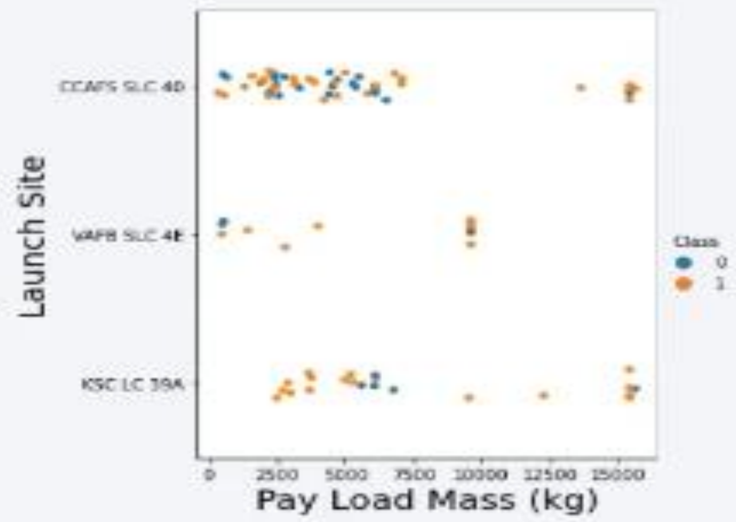
Insights drawn from EDA

Flight Number vs. Launch Site



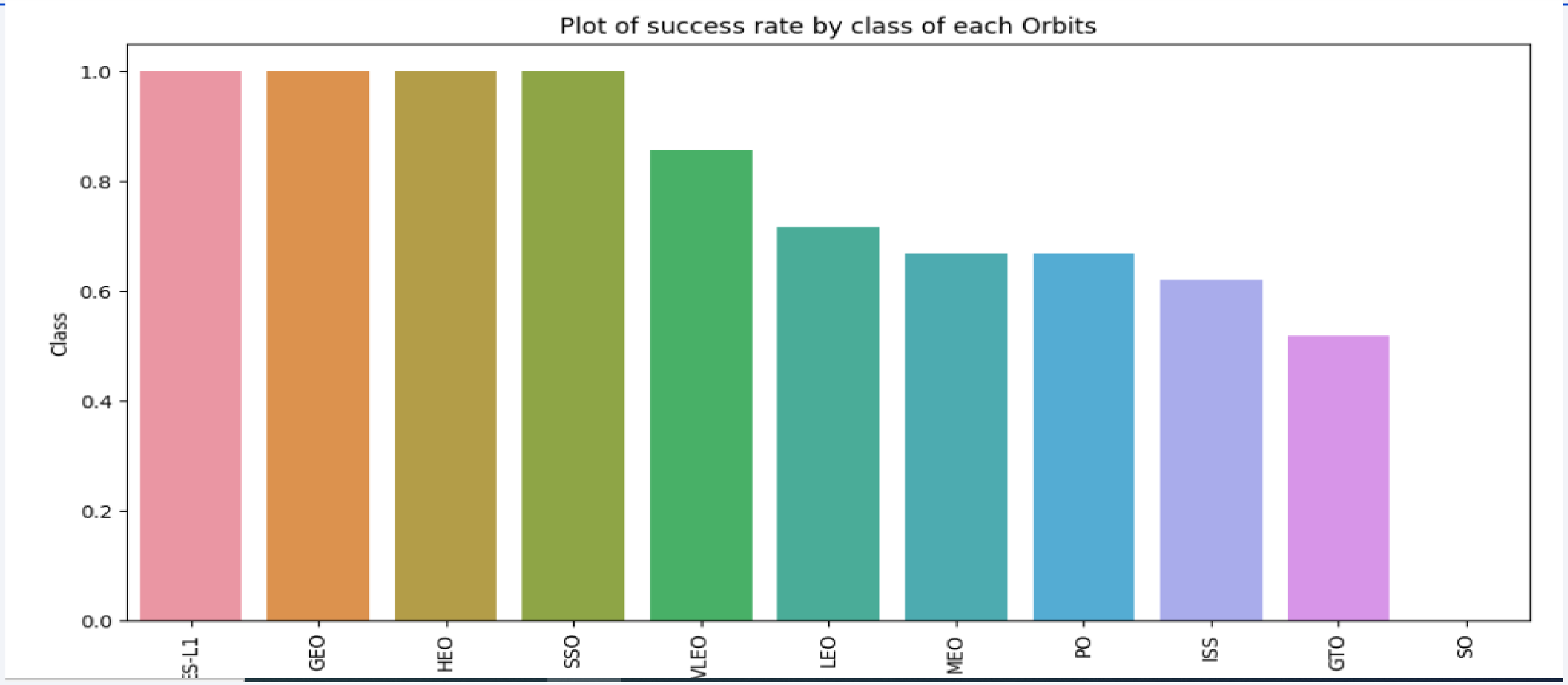
- Flight Number vs. Launch Site
- Show the screenshot of the scatter plot with explanations
- CCAFS 40 has a significantly higher launches than other sited

Payload vs. Launch Site



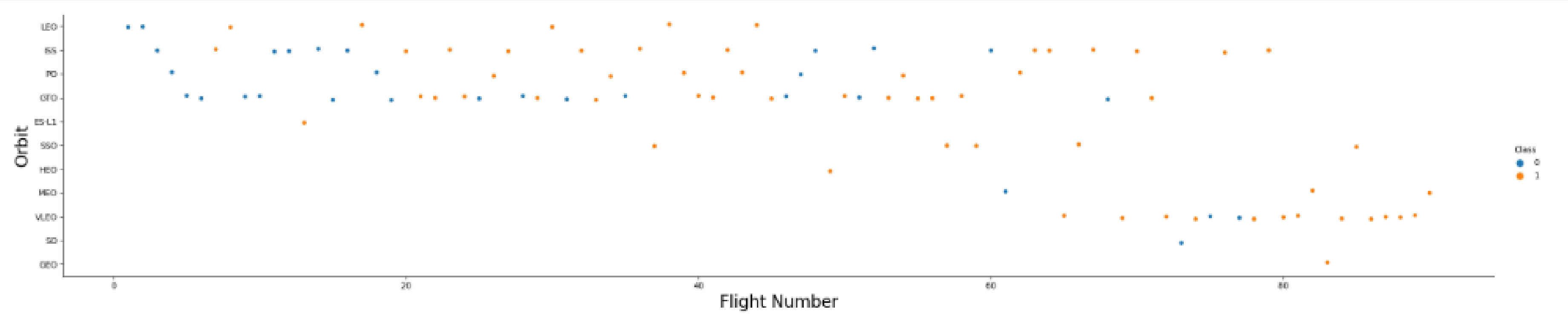
- Payload vs. Launch Site
- The majority of Pay loads with lower mass have been launched from CCAFS SLC40

Success Rate vs. Orbit Type



- Show the screenshot of the scatter plot with explanations
- The orbit Types of ES-L1, GEO, HEO, SSO are among the highest success rate

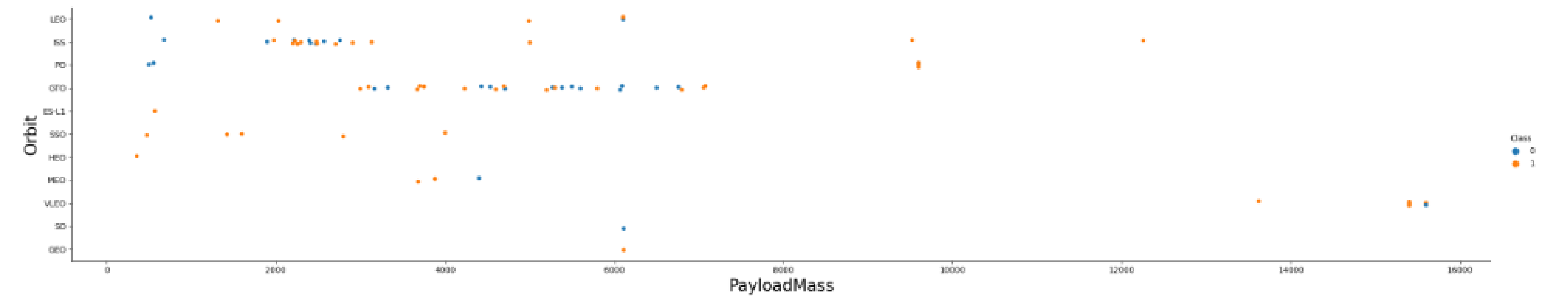
Flight Number vs. Orbit Type



Flight number vs. Orbit type

- A trend can be observed of shifting to VLEO launches in recent years

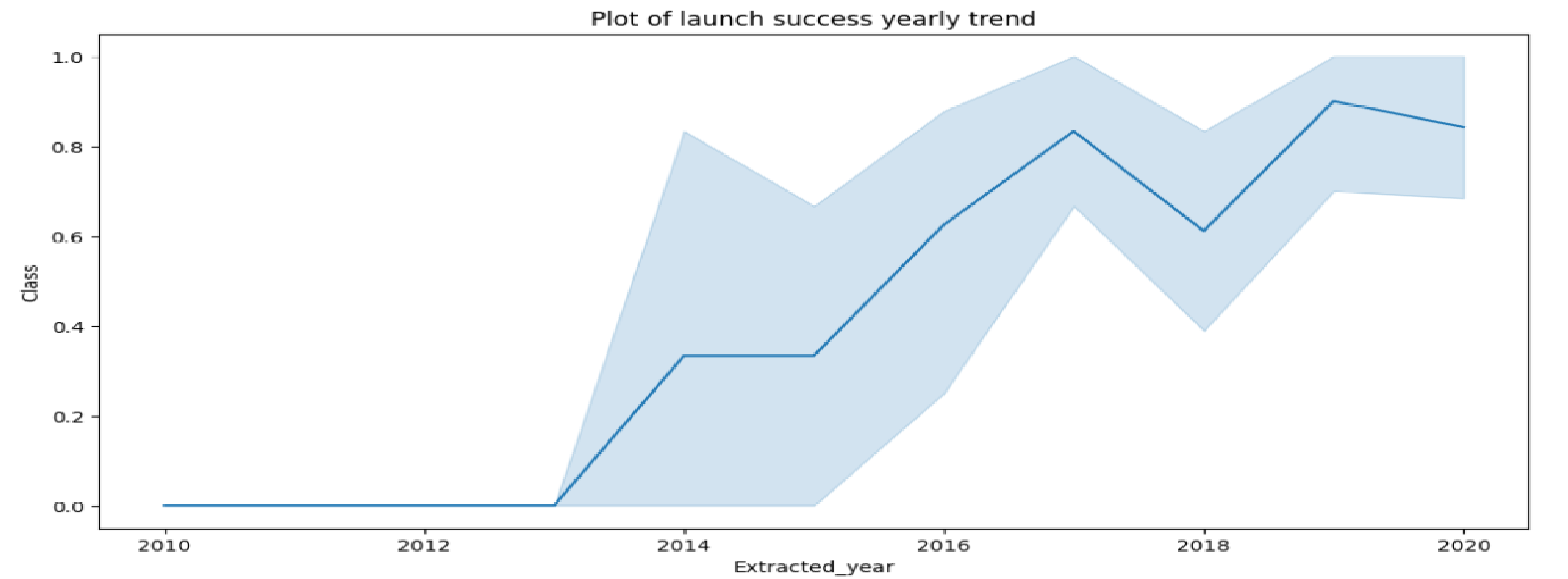
Payload vs. Orbit Type



- payload vs. orbit type

- There is a strong correlation between ISS and Payload at the range around 2000, as well as between GTO and the range of 4000 – 8000

Launch Success Yearly Trend



- The launch success rate has increased significantly since 2013 and has established since 2019, potentially due to advance in technology and the lessons learnt over the past years.

All Launch Site Names

- %sql select distinct(LAUNCH_SITE) from SPACEXTBL

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

Launch Site Names Begin with 'CCA'

- %sql select* from SPACETBL where LAUNCH_SITE like CCA% LIMIT 5

DATE	time_utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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	Failure (parachute)
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
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- %sql select sum(PAYLOAD_MASS_KG_) from SPACETBL where CUSTOMER = 'NASA(CRS)'

45596

Average Payload Mass by F9 v1.1

- %sql select avg(AVG_MASS_KG_) from SPACEXTBL where BOOSTER_VERSION = ‘F9 v1.1’

2928.400000

First Successful Ground Landing Date

- %sql select min(DATE) from SPACEXTBL where Landing Outcome = ‘Success(ground pad)’

2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- %sql select OOSTER_VERSION from SPACEXTBL where Landing Outcome = 'Success (drone ship)' and PAYLOAD_MASS_KG_>4000 and Payload_MASS_KG_<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

- %sql select count(MISSION_OUTCOME) from SPACEXTL where MISSION_OUTCOME = ‘Failure(in flight)’

100

Boosters Carried Maximum Payload

- %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

- %sql select ‘ from SPACEXTBL where Landing _Outcome like ‘Success%’ and (DATE between ‘2015-01-01’ and ‘2015-12-31’) order by date desc

time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
[illegible]	[illegible]	CCAFS LC-	[illegible]	[illegible]	[illegible]	SKY Perfect JSAT	[illegible]	[illegible]

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

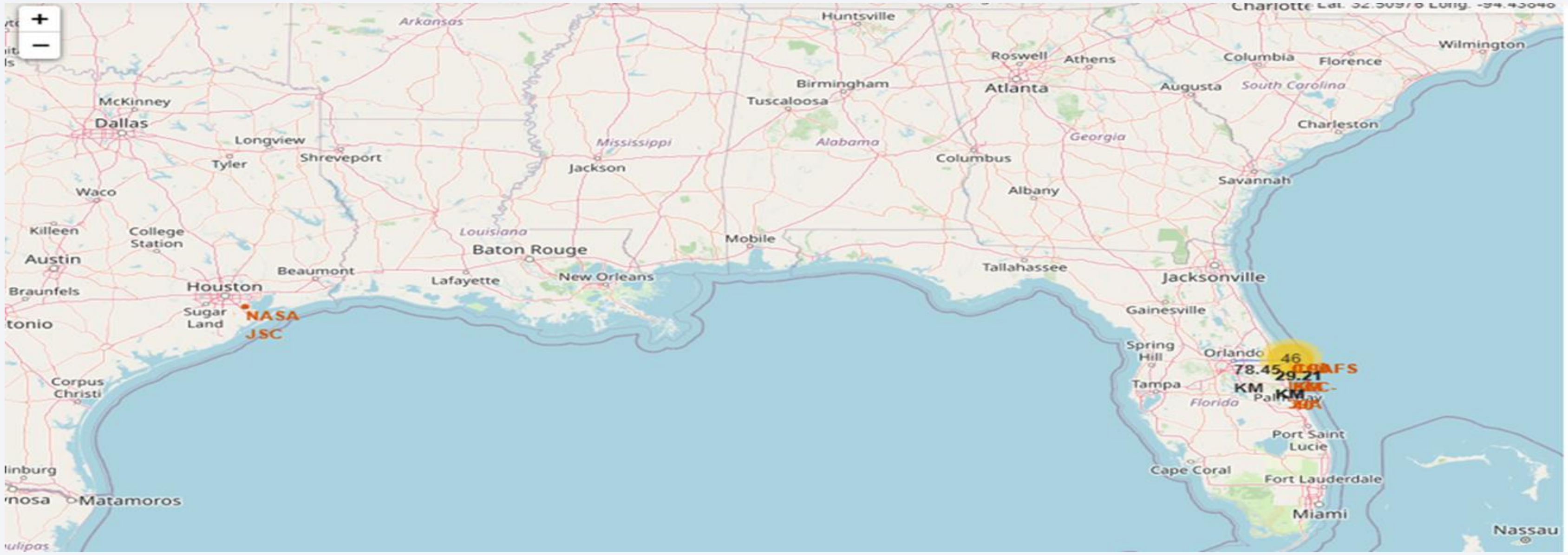
- %sql select ‘ from SPACEXTBL where Landing _Outcome like ‘Success%’ and (DATE between ‘2010-06-04’ and ‘2017-03-20’) order by date desc

2016-03-27	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

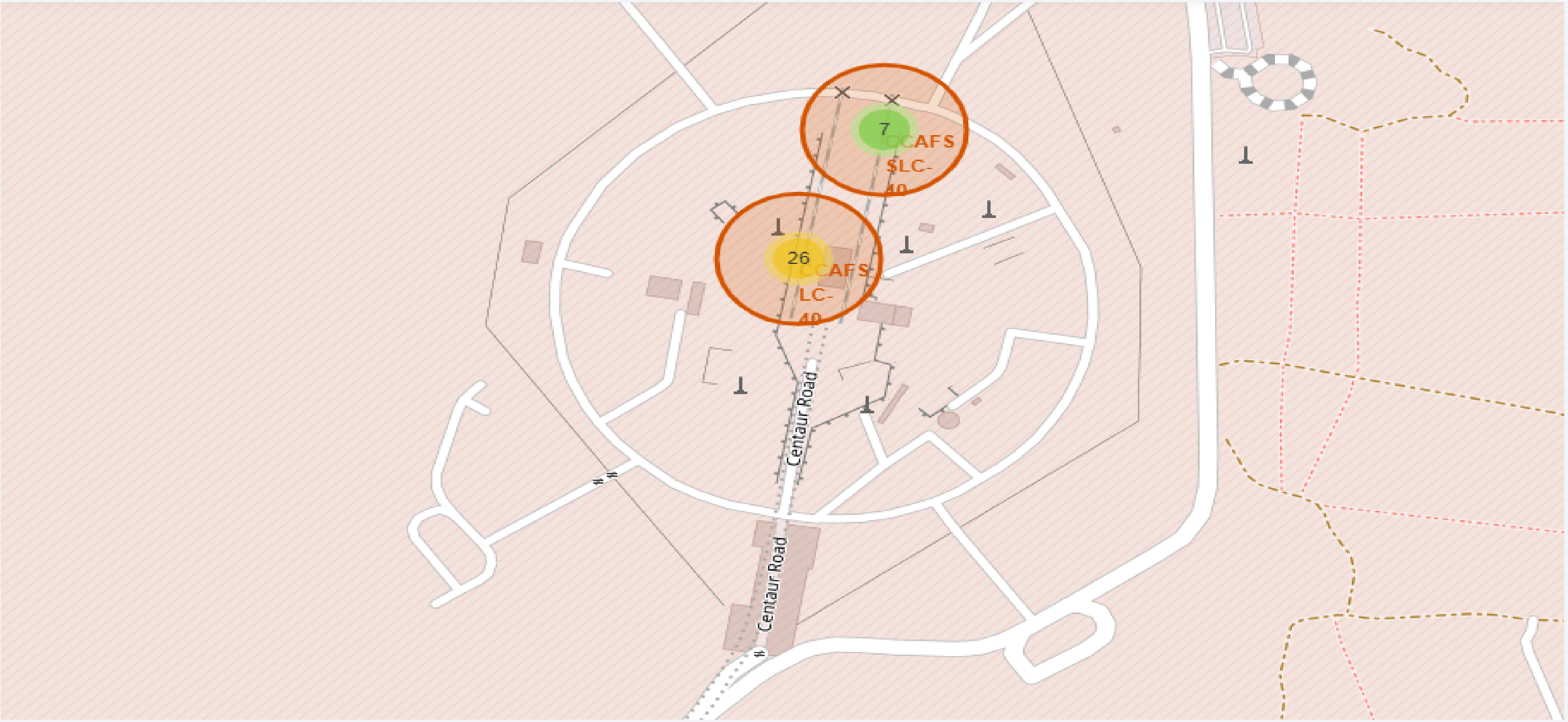
Launch Sites Proximities Analysis

ALL LAUNCH SITES MARKED ON THE MAP

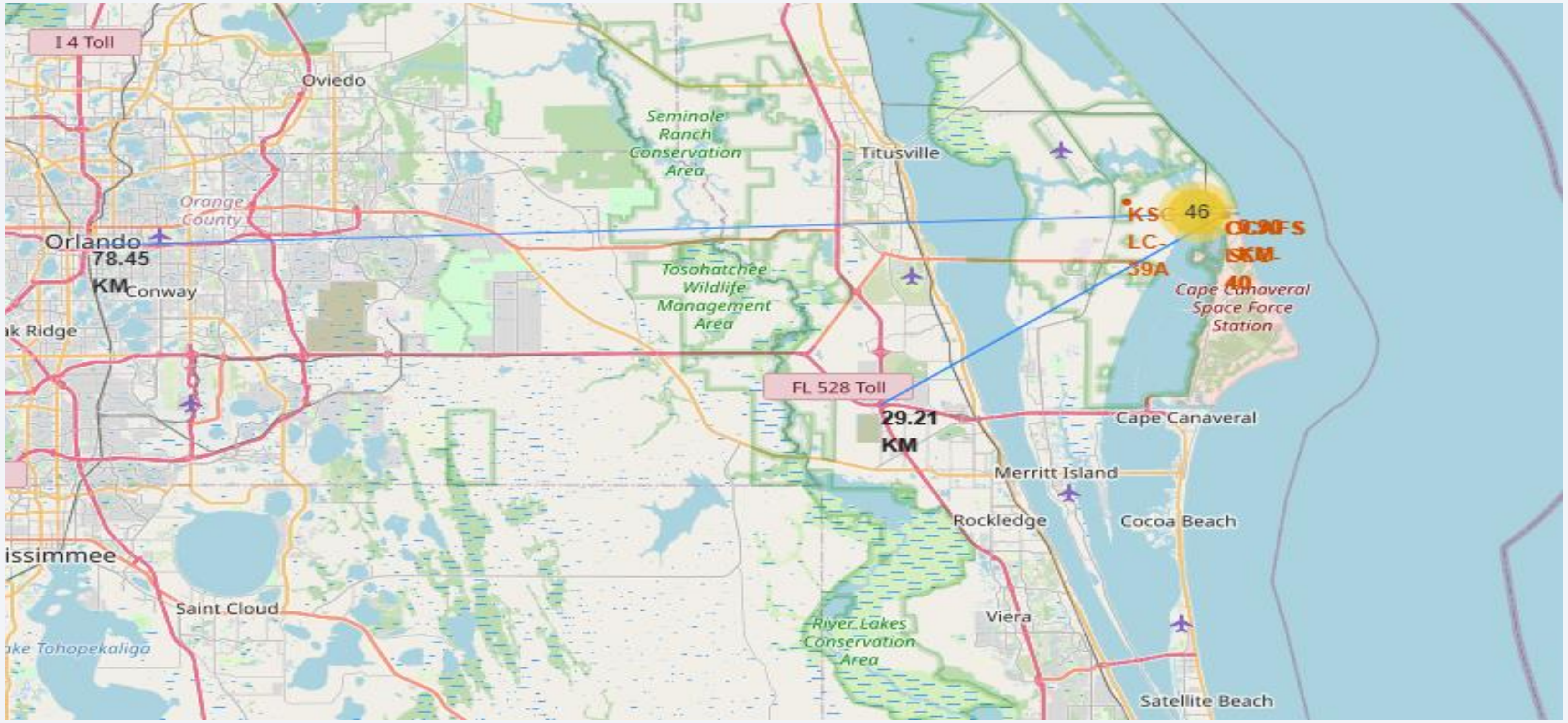


- Explain the important elements and findings on the screenshot

Success/failed launches for each site on the map



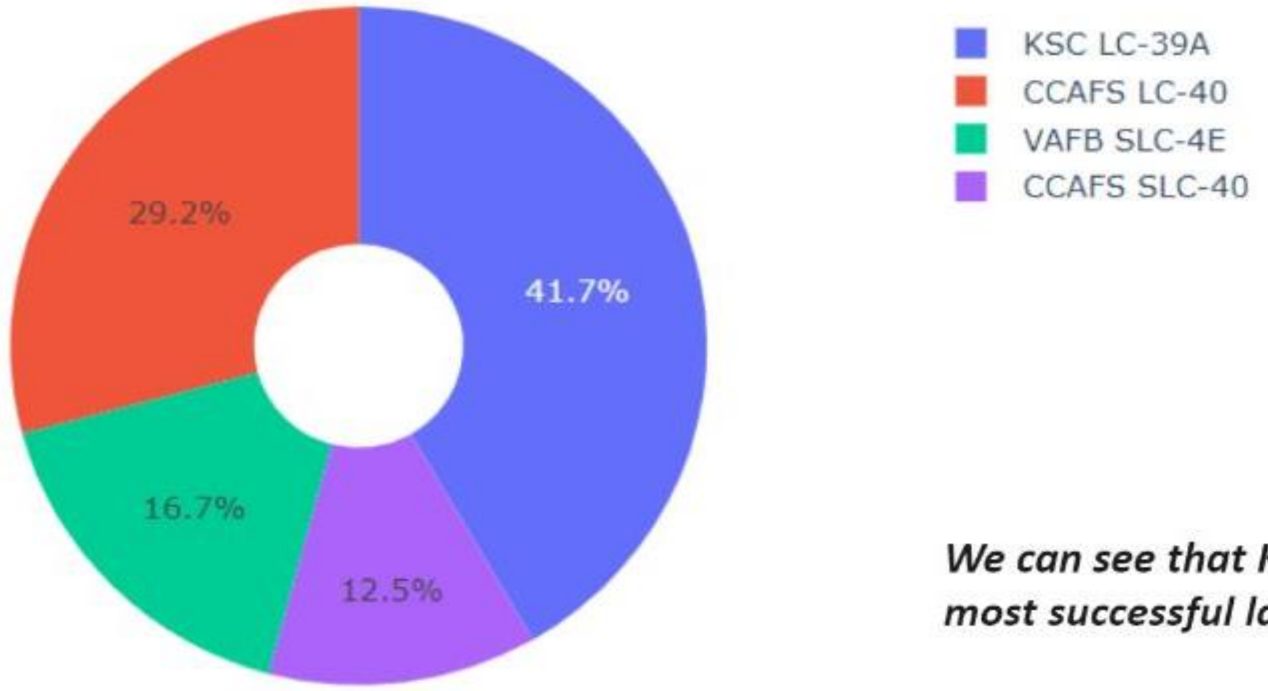
Distance between launch site and its proximities



Section 4

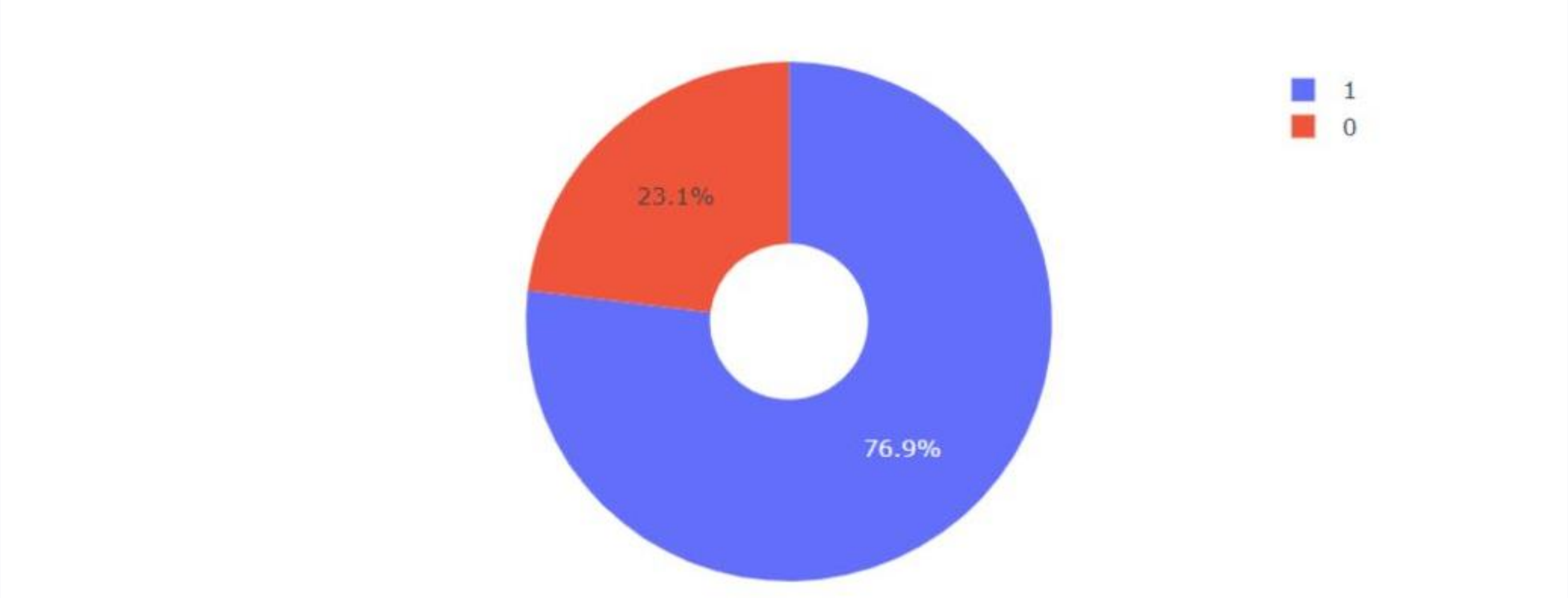
Build a Dashboard with Plotly Dash

Total Success Launches by all sites



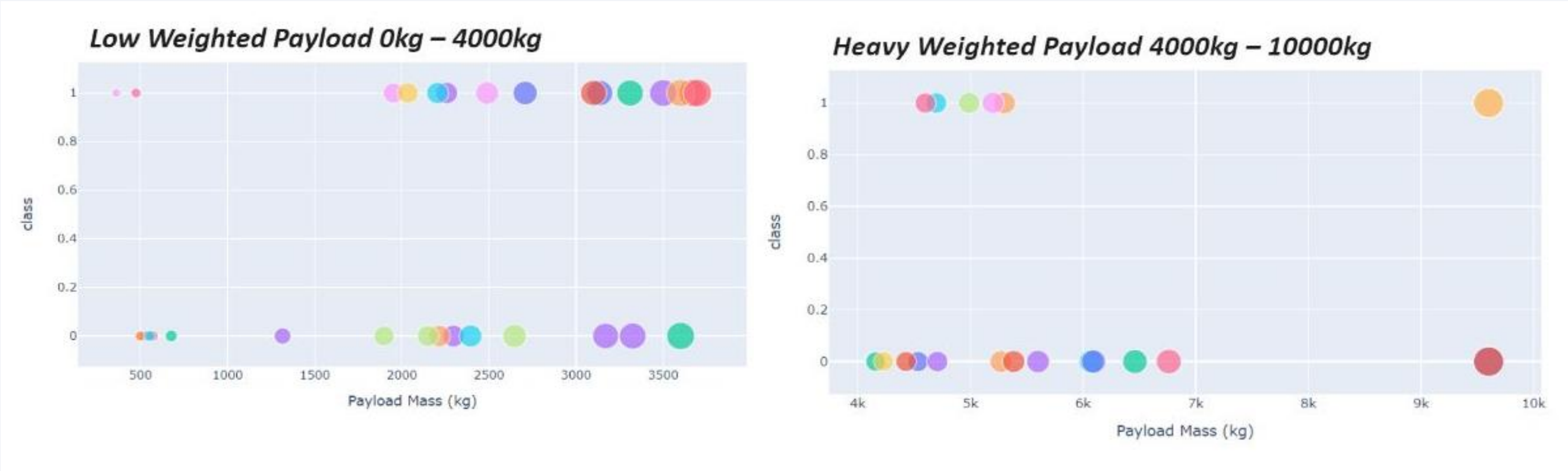
We can see that KSC LC-39A had the most successful launches from all the sites

KSC LC-39 success rate



- KSC LC-39 Achieved a 76.9% success rate while getting a 23.1% failure rate.

Payload vs Launch Outcome

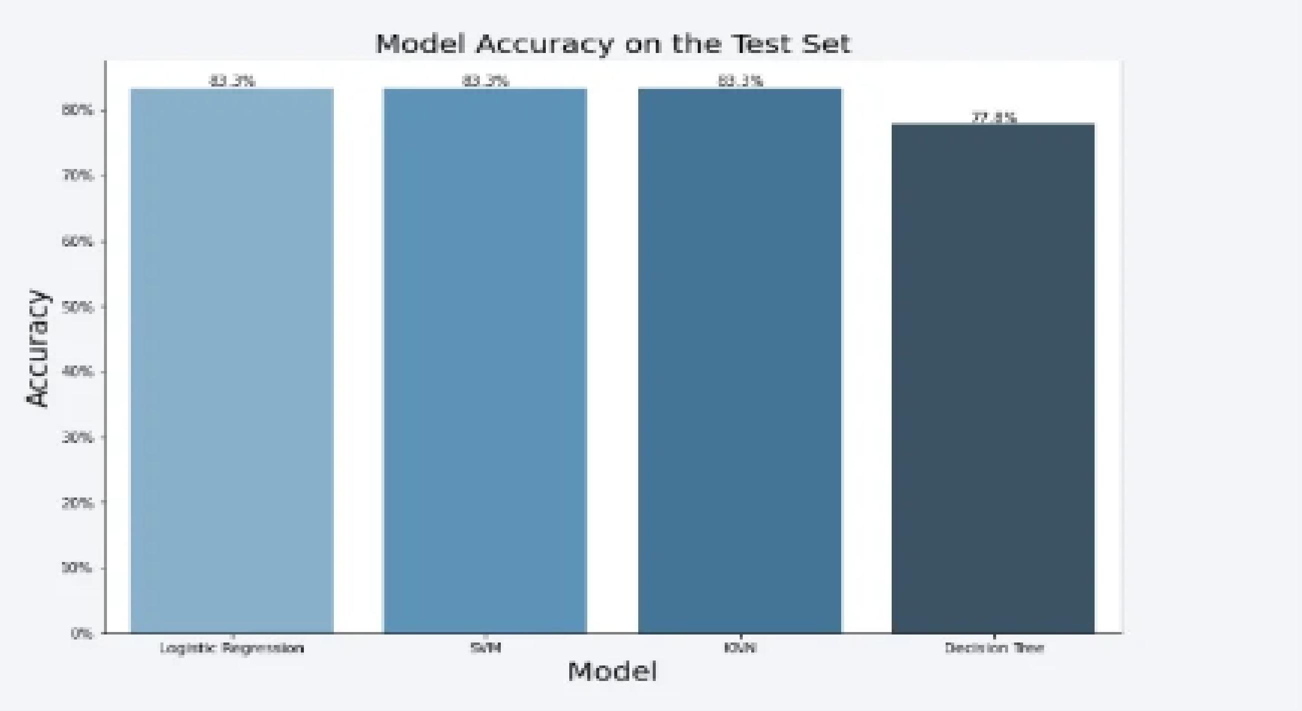


- We can see the success rate for low weighted payload is higher than the heavy weighted payloads

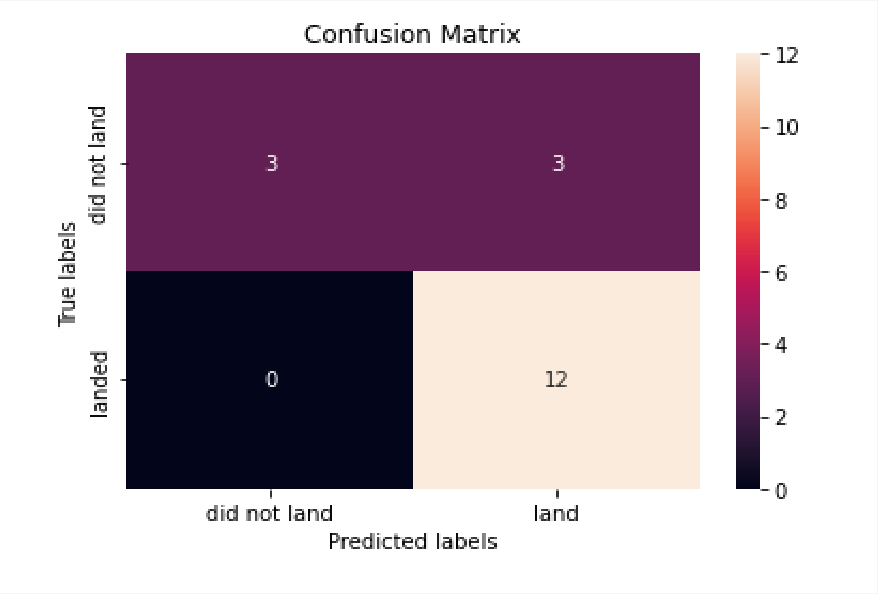
Section 5

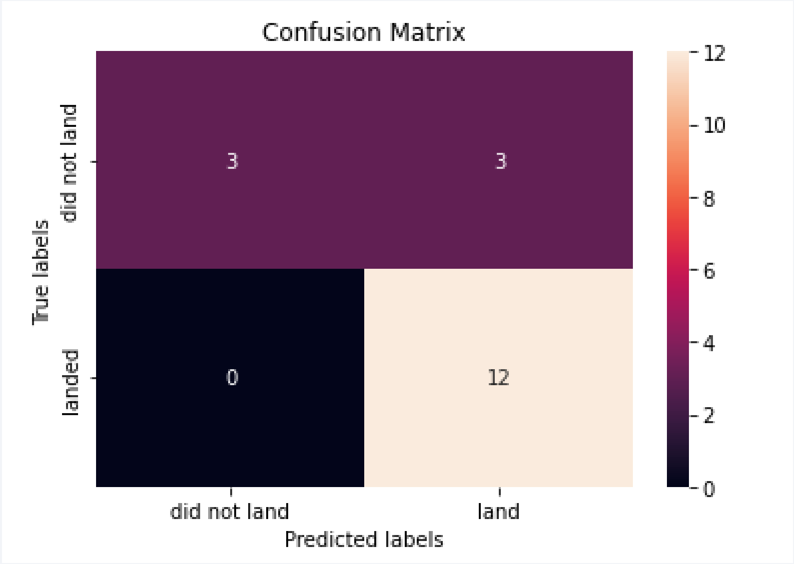
Predictive Analysis (Classification)

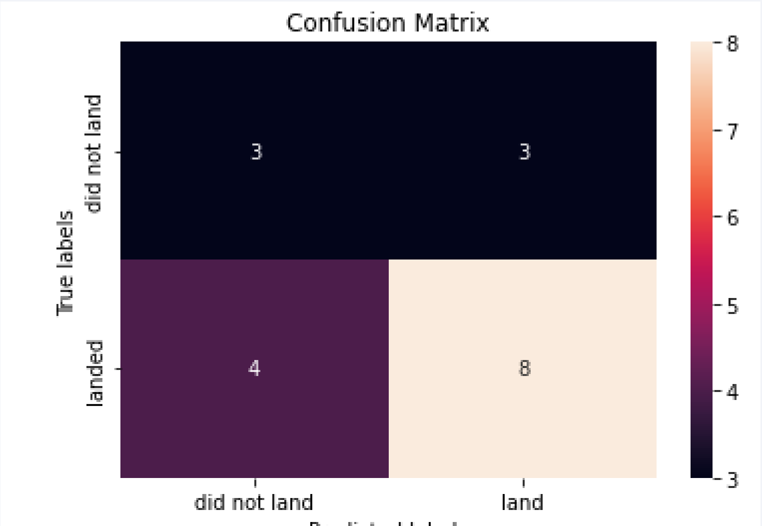
Classification Accuracy

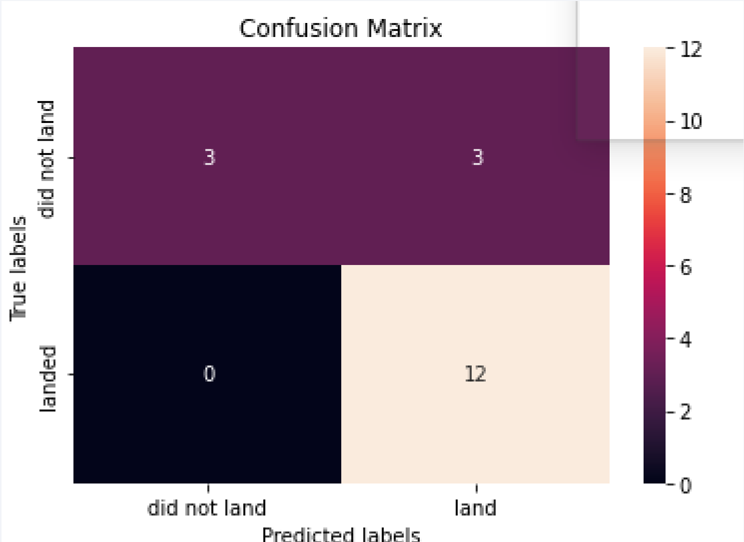


Confusion Matrix









Conclusions

- The SVM , KNN AND logistics regression models are the best in terms of prediction accuracy for this data set
- Low weighted payloads perfom better than heavier payloads
- KSC LC 39A had the most successful launches from all the sites
- Orbit GEO ,HEO, SSO, ES L1 has the best success Rate.
- The success rates for spaceX launches is directly proportional to time in years they will eventually perfect the launch.
- …

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!