IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ajisa Muthayil Ali
18/08/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection using requests to the Request to the SpaceX API and cleaning data.
  - Web scarping using BeautifulSoup.
  - Exploratory data analysis (EDA) visualisation analysis using Pandas and Matplotlib
  - EDA using SQL
  - Interactive visual analytics using Dashboard with Plotly Dash and Folium.
  - Predictive analysis using Machine learning algorithms.
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytical results
  - Predictive analysis results

Introduction

- Project background and context

  Space-X-Falcon-9-First-Stage-Landing-Prediction Space X 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 Space X can reuse the first stage. The aim of this project is to predict if the first stage will land so that we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch.

- Problems you want to find answers
  - To assess main characteristics of the successful or failed landing.
  - To find if there is any correlation between rockets variables with successful and or failed landing.
  - To better understand the conditions favor SpaceX to achieve successful landing.

Section 1

Methodology

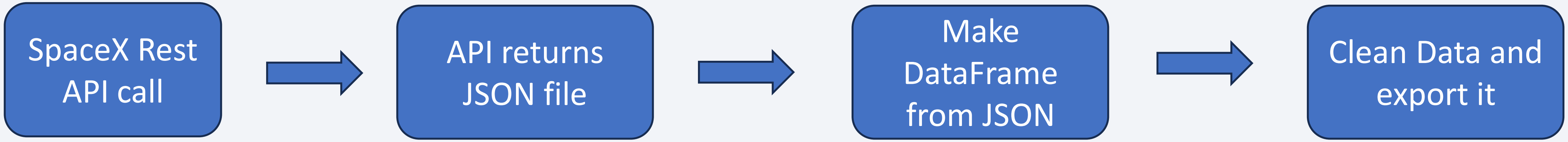
Methodology

Executive Summary

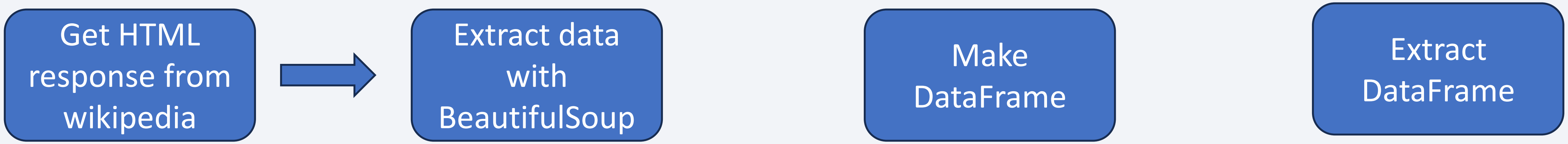
- Data collection methodology:
  - SpaceX REST API.
  - Webscraping from Wikipedia.
- Perform data wrangling
  - Dropping not applicable columns.
  - One hot encoding to classify models.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data sets were collected from SpaceX Rest API and webscraping Wikipedia.
  - SpaceX API and Wikipedia URL were:
  - api.spacexdata.com/v4/
  - https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches



API generated information about booster versions, launch sites/dates, payload mass



Webscraping Wikipedia generated information about launches, landings and payload

Data Collection – SpaceX API

1. Got response from API

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

2. Converted response into JSON File

```
response1 = requests.get(static_json_url)
resp_dict = response1.json()
data= pd.json_normalize (resp_dict)
```

3. Transformed Data

```
getLaunchSite(data)
getPayload(data)
getCoreData(data)
getBoosterVersion(data)
```

4. Created Dictionary with Data

```
launch_dict = {'FlightNumber':
(data['flight_number']), 'Date':
(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. Created DataFrame

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

6. Filtered DataFrame

```
data_falcon9 = data_falcon9 [data_falcon9 ['BoosterVersion'] != 'Falcon 1']
```

7. Exported to file

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

https://github.com/AjisaMuthayilAli/Space-X-Falcon-9-First-Stage-Landing-Prediction/blob/main/Lab1_DataCollection.ipynb

Data Collection - Scraping

1. Got response from HTML

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

2. Created BeautifulSoup object

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

3. Found all tables

```
html_tables = soup.find_all("table")
```

4. Got column names

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

5. Created 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. Added data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in
(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
```

SEE NOTEBOOK FOR THE WHOLE CODE

7. Created DataFrame from Dictionary

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

8. Exported to file

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

https://github.com/AjisaMuthayilAli/Space-X-Falcon-9-First-Stage-Landing-Prediction/blob/main/Lab2_WebScraping.ipynb

Data Wrangling

- In the data set, there are several different cases where the booster did not land successfully. They were represented as categorical variables (False Ocean: unsuccessfully landed into ocean, False RTLS: unsuccessfully landed into ground pad, True RTLS: successfully landed into ground pad, True ASDS: successfully landed into drone ship etc.)
- Converted these categorical variables into binary variables using dummies; where 1 means successful mission and 0 means unsuccessful mission.

1. Calculated number of launches from each launch site

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

CCAFS SLC 40     55
KSC LC 39A       22
VAFB SLC 4E      13
Name: LaunchSite, dtype: int64
```

2. Calculated number and occurrence of each orbit

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

GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

3. Calculated number and occurrence of mission outcome for each orbit type

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

True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

4. Created landing outcome label from Outcome column

```
landing_class= []
for i in df['Outcome']:
    if i in set(bad_outcomes):
        landing_class.append(0)
    else:
        landing_class.append(1)
```

5. Exported to file

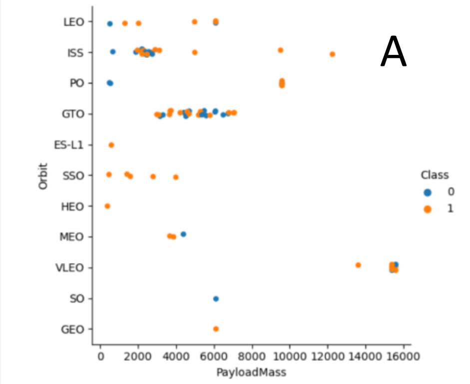
df.to_csv('dataset_part_2.csv', index=False)

https://github.com/AjisaMuthayilAli/Space-X-Falcon-9-First-Stage-Landing-Prediction/blob/main/Lab3_DataWrangling.ipynb

EDA with Data Visualization

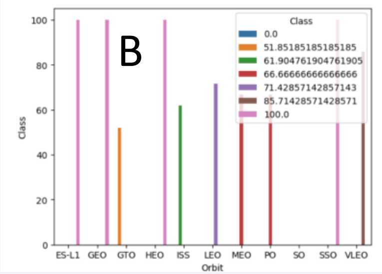
A. Scatter graphs to show the correlation between variables

- Flight Number vs PayLoad Mass
- Flight Number vs Launch Site
- Payload vs Launch Site
- Orbit vs Flight Number
- Payload vs Orbit Type
- Orbit vs Payload Mass

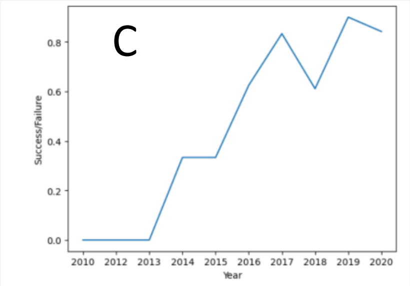


https://github.com/AjisaMuthayilAli/Space-X-Falcon-9-First-Stage-Landing-Prediction/blob/main/Lab5_EDA_DataVisualisation.ipynb

B. Bar plot to show the relation between categorical and numerical variable



C. Line chart to show the trend



EDA with SQL

- Displayed the names of the unique launch sites in the space mission.
- Displayed 5 records where launch sites begin with the string 'CCA'.
- Displayed the total payload mass carried by boosters launched by NASA (CRS).
- Displayed average payload mass carried by booster version F9 v1.1.
- Listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- Listed the total number of successful and failure mission outcomes
- Listed the names of the booster_versions which have carried the maximum payload mass.
- Listed the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- Ranked 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.

https://github.com/AjisaMuthayilAli/Space-X-Falcon-9-First-Stage-Landing-Prediction/blob/main/Lab4_ExploratoryDataAnalysisUsingSQL.ipynb

Build an Interactive Map with Folium

- Created a folium map object centered around NASA Johnson Center at Houston, Texas.
  - Blue circle showing NASA Johnson Space Center's name.
  - Red circle shows the launch site names.
  - Green marker shows successful mission and red marker shows unsuccessful mission.
  - Markers shows the distance between launch sites to key locations such as railway, highway, city etc.
- This map aided to better understand if the geographical locations and or surroundings has any effect on the mission outcome.

https://github.com/AjisaMuthayilAli/Space-X-Falcon-9-First-Stage-Landing-Prediction/blob/main/Lab6_Launch%20SitesLocationsAnalysiswithFolium.ipynb

Build a Dashboard with Plotly Dash

- Dashboard included drop down, pie and scatter plots.
  - Drop down: to choose launch sites.
  - Pie chart: shows the successful and unsuccessful missions.
  - Scatter plot: correlation between variables such as payload mass vs successful landing.

https://github.com/AjisaMuthayilAli/Space-X-Falcon-9-First-Stage-Landing-Prediction/blob/main/Lab7_DashboardinteractiveappwithPlotlyDash.ipynb

Predictive Analysis (Classification)

- Data preparation: Loaded, Normalised and split train/test datasets.
- Model preparation: selected ML algorithms, set parameters for each type of model and trained with training sets.
- Model Evaluation: chose best paraments for each model, computed accuracy for each model using test dataset and plotted confusion matrix.
- Model comparison: Compared models based on their accuracy and the model with the best accuracy was chosen.

https://github.com/AjisaMuthayilAli/Space-X-Falcon-9-First-Stage-Landing-Prediction/blob/main/Lab8_Machine%20Learning%20Prediction.ipynb

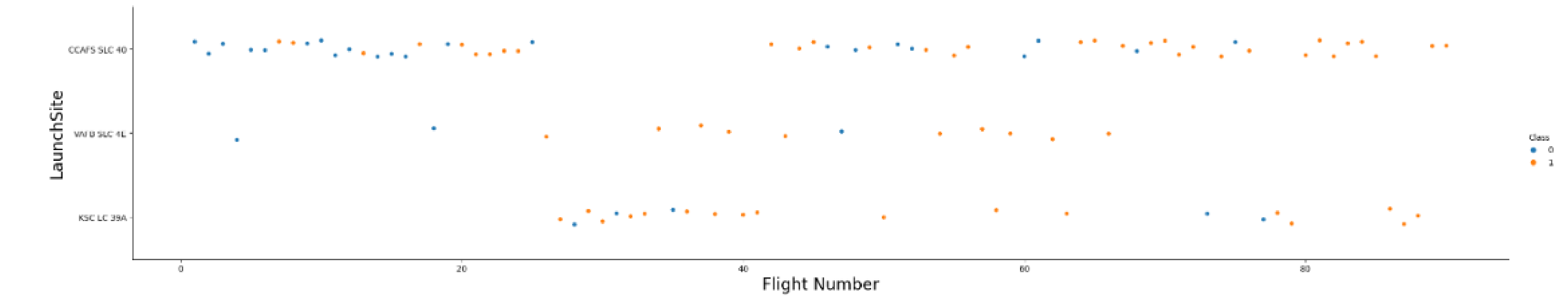
Results

- SVM, KNN and Logistic regression models west best in terms of prediction accuracy.
- The success rates of spaceX launches is directly proportional to time in years.
- KSC LC 39A had the most successful launches from all sites.
- Orbit GEO, HEO, SSO, ES L1 has the best success rate

Section 2

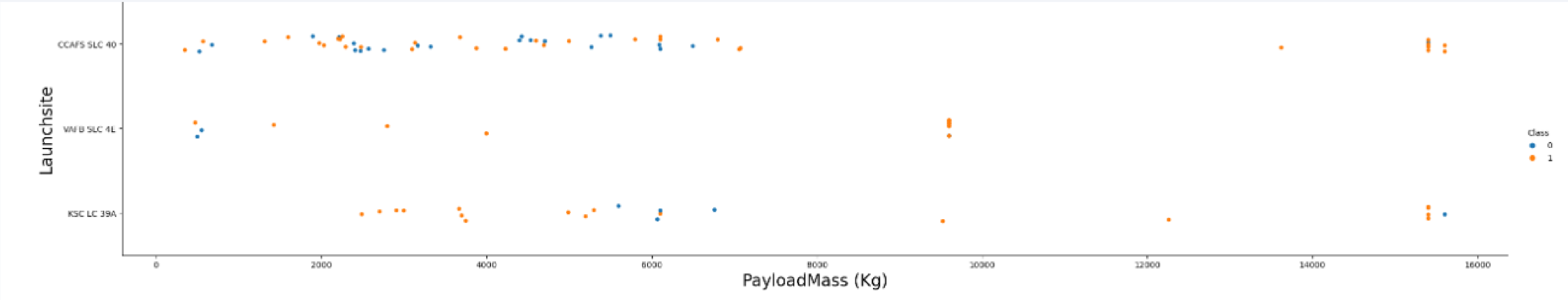
Insights drawn from EDA

Flight Number vs. Launch Site



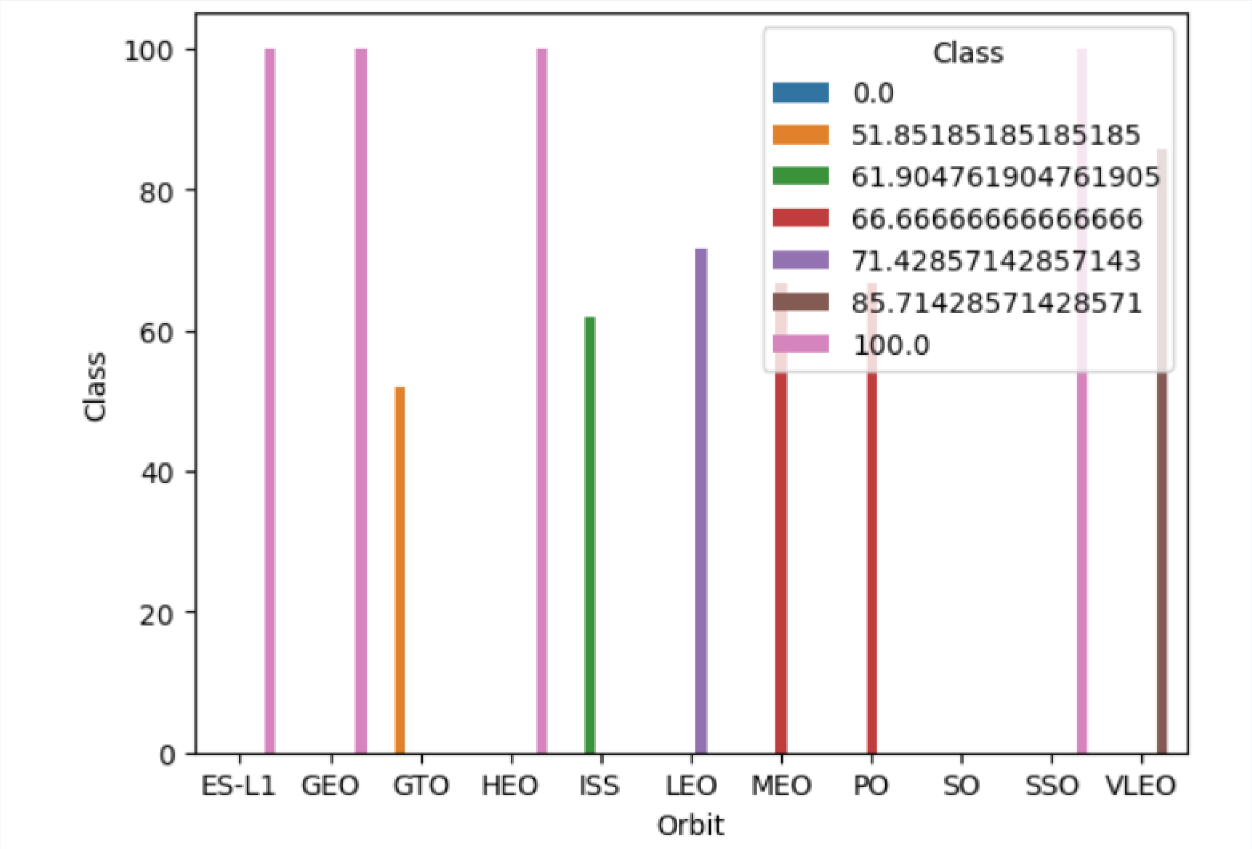
- Success rate (orange data points) is increasing at all launch sites.

Payload vs. Launch Site



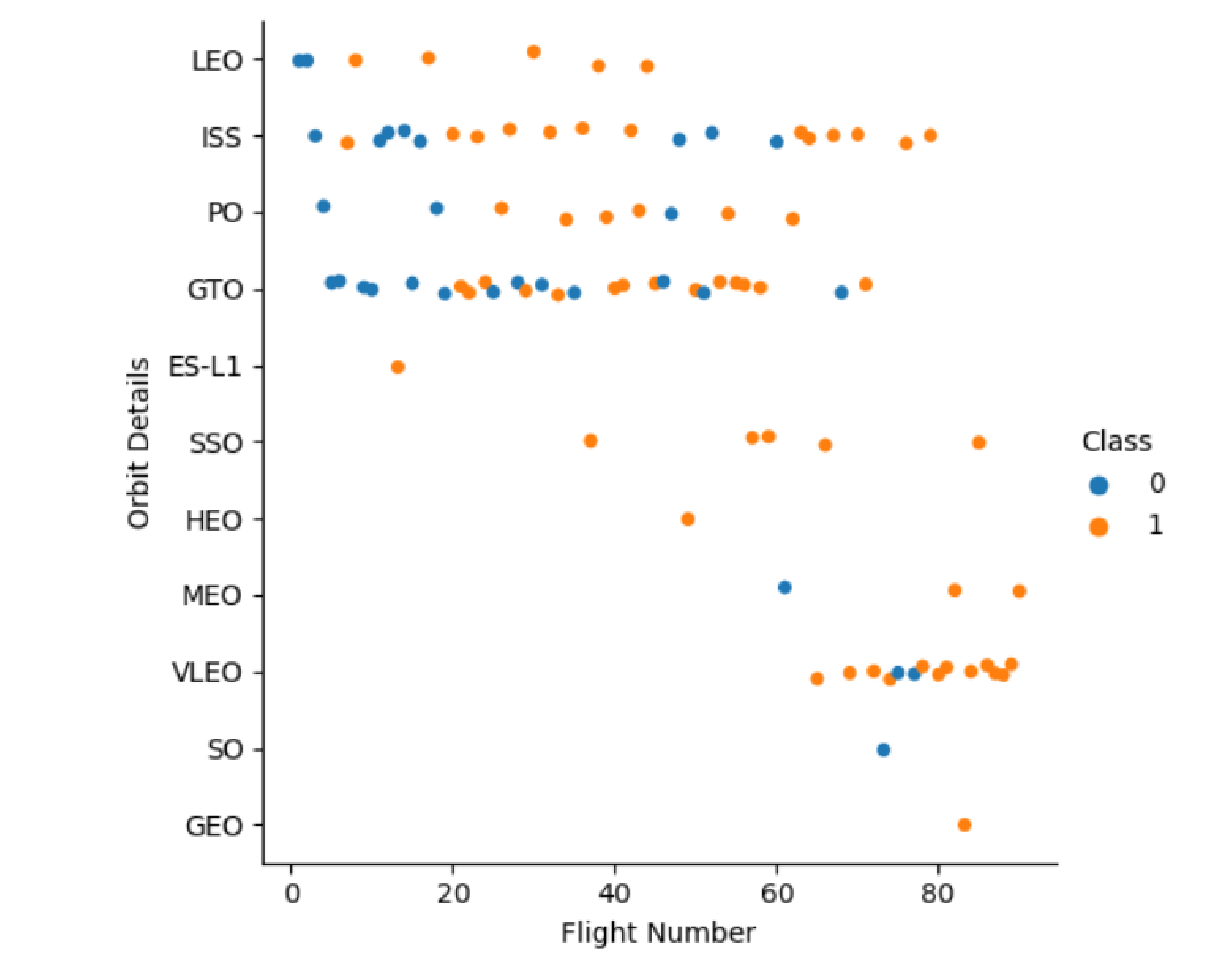
- Success rate is higher with higher payload mass.

Success Rate vs. Orbit Type



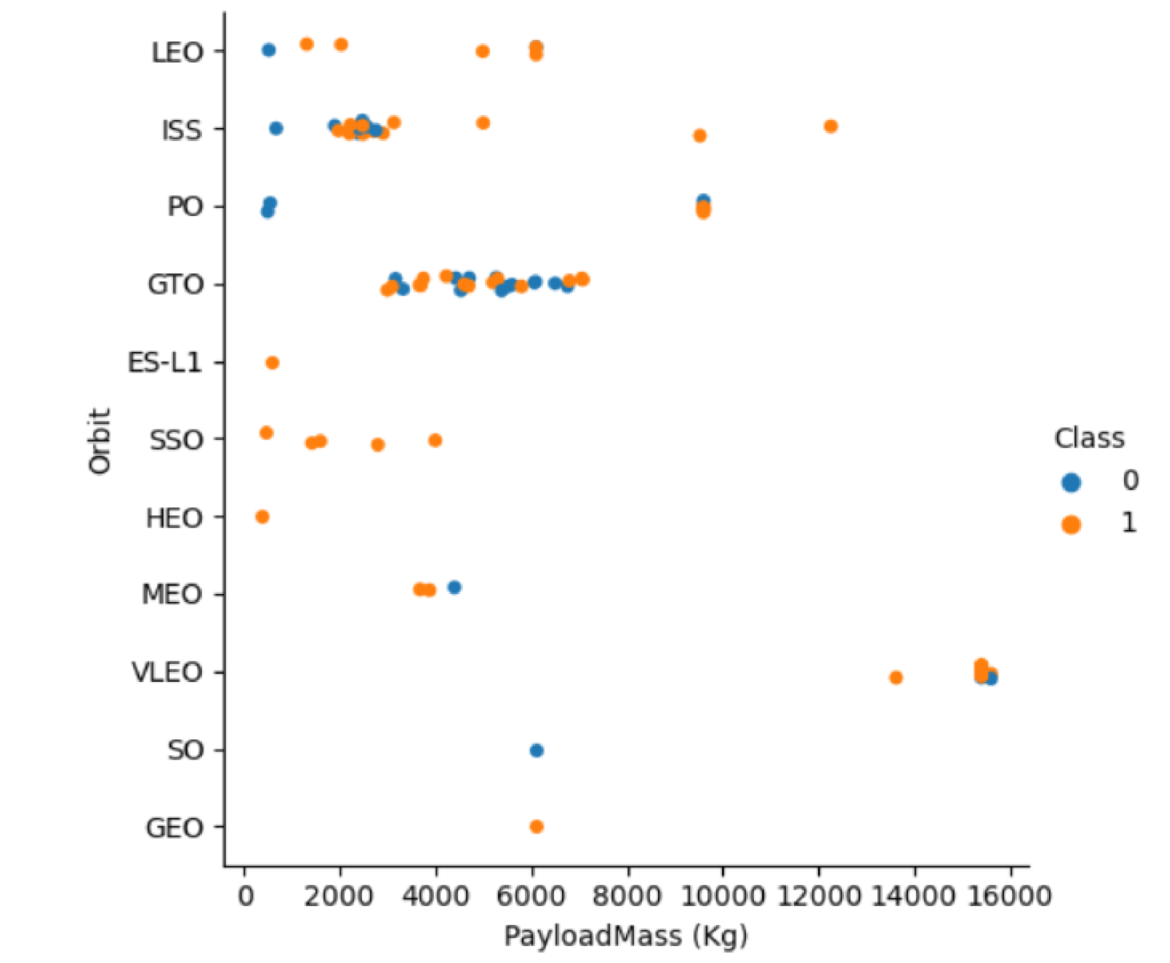
- Success rate us higher for SD-L1, GEO, HEO, SSO

Flight Number vs. Orbit Type



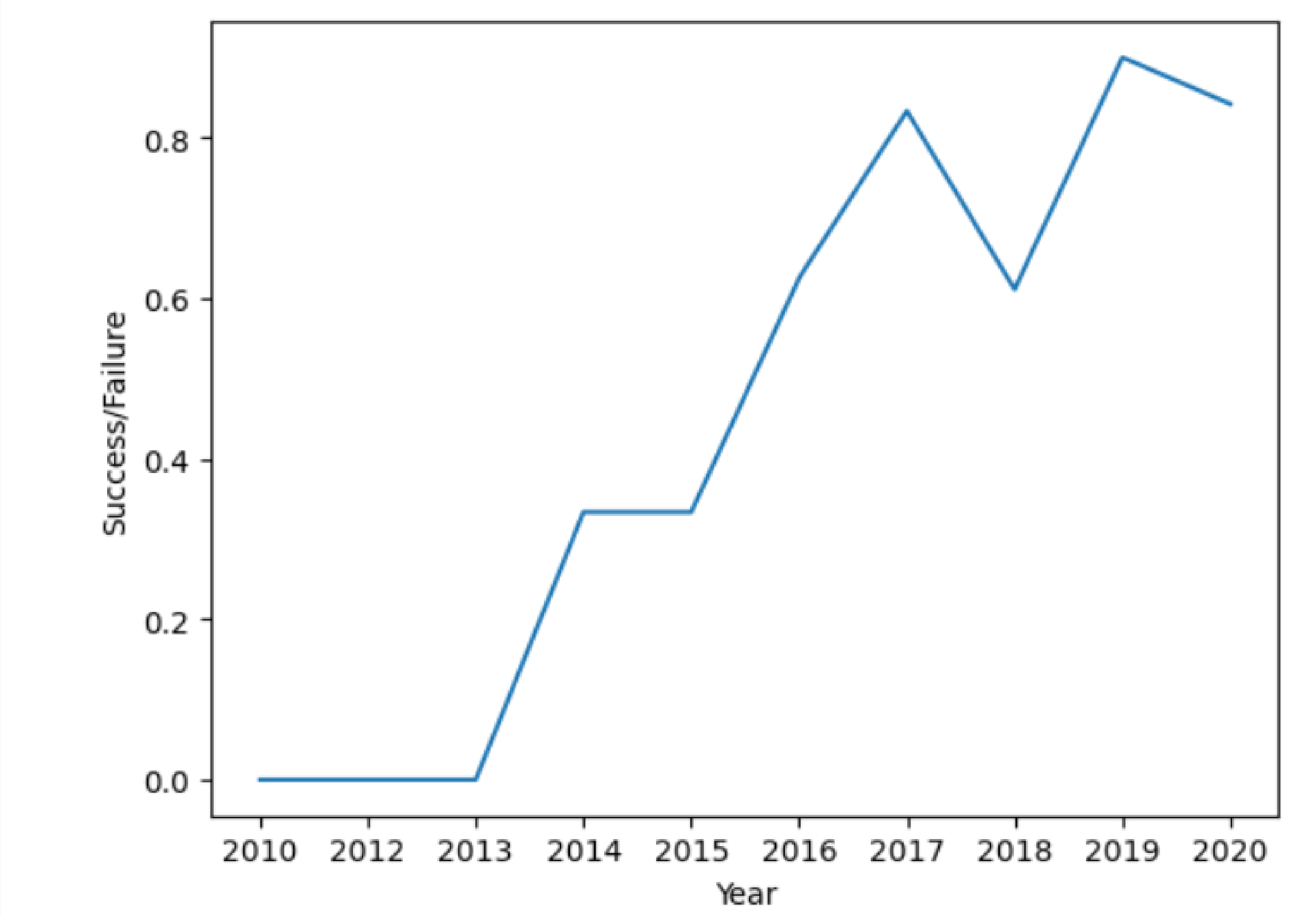
- Success rate increases with the number of flights for LEO, ISS, PO and GTO.
- Doesn't correlate success rate with VLEO, SSO orbits as the flight number increases.

Payload vs. Orbit Type



- Success rate increases with payload mass for certain orbits (ISS, LEO)
- Conversely, the success rate decreased with payload mass for GTO.

Launch Success Yearly Trend



- Success rate increased since 2013 until 2020.

All Launch Site Names

```
SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

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

- Distinct query removed duplicate attributes (LAUNCH_SITE)

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTABLE 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
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Like function aids to sub-filter with part of the word.
- Limit (5) function limits the rows to 5.

Total Payload Mass

```
SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
```

SUM("PAYLOAD_MASS__KG_")
45596

- The total payload mass for NASA (CRS) is 45596 kg.

Average Payload Mass by F9 v1.1

```
%sql SELECT Avg (PAYLOAD_MASS__KG_) from SPACEXTABLE where "Booster_Version" = 'F9 v1.1'
```

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

Avg (PAYLOAD_MASS__KG_)
2928.4

- Average payload mass carried by booster version F9 v1.1 is 2928.4 kg

First Successful Ground Landing Date

```
%sql SELECT Date from SPACEXTABLE where "Landing_Outcome" = 'Success (ground pad)'
```

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

Date
2015-12-22

- The first successful landing on ground pad was made on 22nd December 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select Booster_Version from SPACEXTABLE where (PAYLOAD_MASS__KG_  between 4000 and 6000) and "Landing_Outcome" = 'Success (drone ship)'
```

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

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

- We can filter payload mass and limit specific weight with between and function.

Total Number of Successful and Failure Mission Outcomes

```
%sql Select "Mission_Outcome", count (*) as total_number from SPACEXTABLE group by "Mission_Outcome"
```

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

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

- Group by can be used to separate groups within a column and count function can be used to get total sum of each group.

Boosters Carried Maximum Payload

```
%sql select "Booster_Version", PAYLOAD_MASS__KG_ from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select Max(PAYLOAD_MASS__KG_) from SPACEXTABLE);

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

- Subquery can be used to filter maximum value of an observation (Payload mass) for a booster version.

2015 Launch Records

```
%sql select substr("Date", 6, 2) as month, "Date", "Booster_Version", "Launch_Site", \
"Landing_Outcome" from SPACEXTABLE where "Landing_Outcome" = 'Failure (drone ship)' and substr("Date",1,4)='2015';
```

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

month	Date	Booster_Version	Launch_Site	Landing_Outcome
10	2015-10-01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

- Can sub-filter Date and specify the character position and length and saved using ‘as’ in a new column.
- Year sub-filtered using (1,4) where 1 is the position of year starting character and 4 is the total number of characters need to filter (i.e. 2015).

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

```
%sql select "Landing_Outcome", count (*) as outcome from SPACEXTABLE where "Date" between "2010-06-04" and "2017-03-20"\
group by "Landing_Outcome" order by outcome desc;
```

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

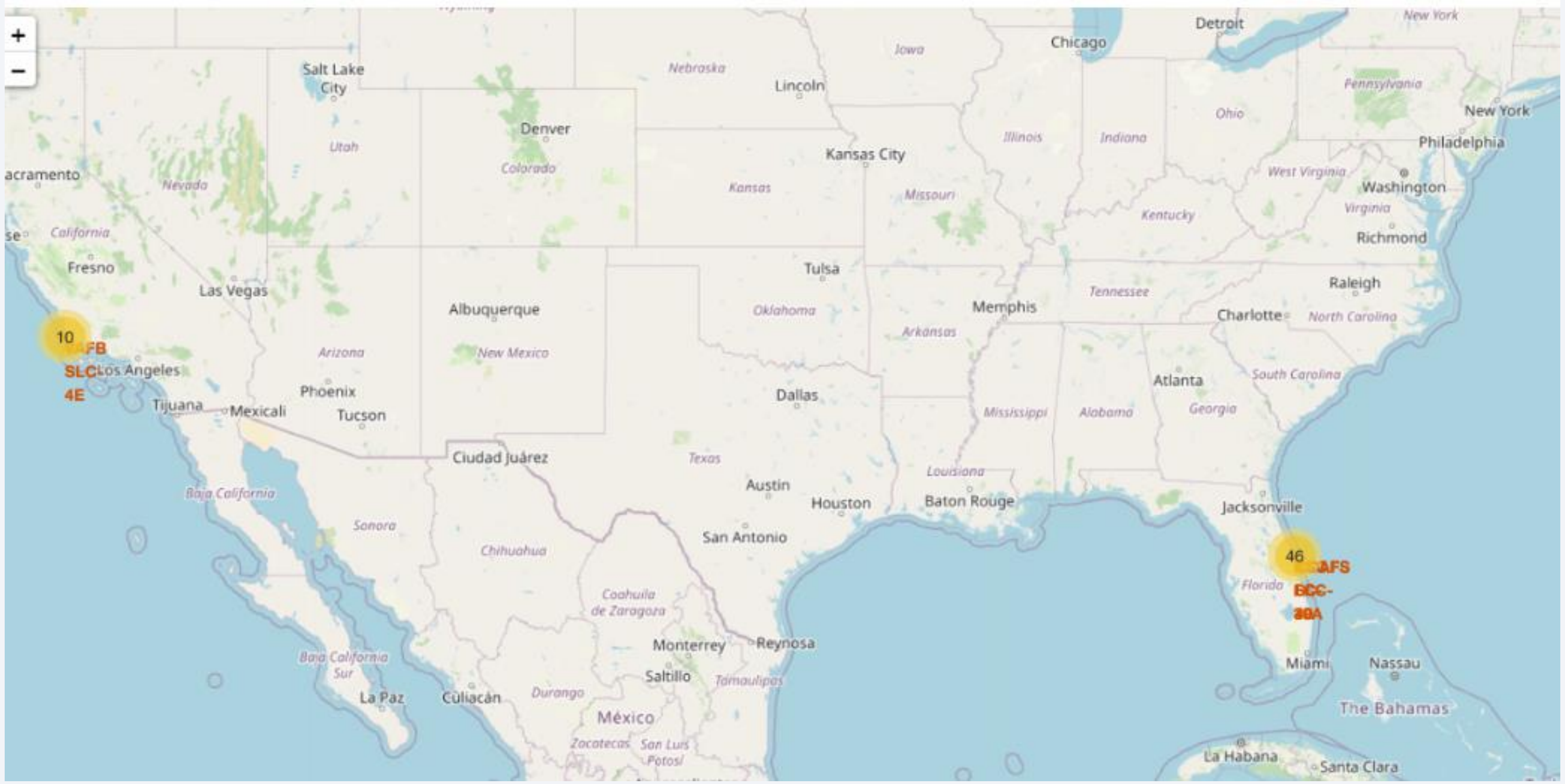
Landing_Outcome	outcome
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

- Specific dates were sub-filtered from a column (Date), counted total number using ‘count’ and ordered using ‘order’.
- ‘desc’ function helps to order in descending order.

Section 3

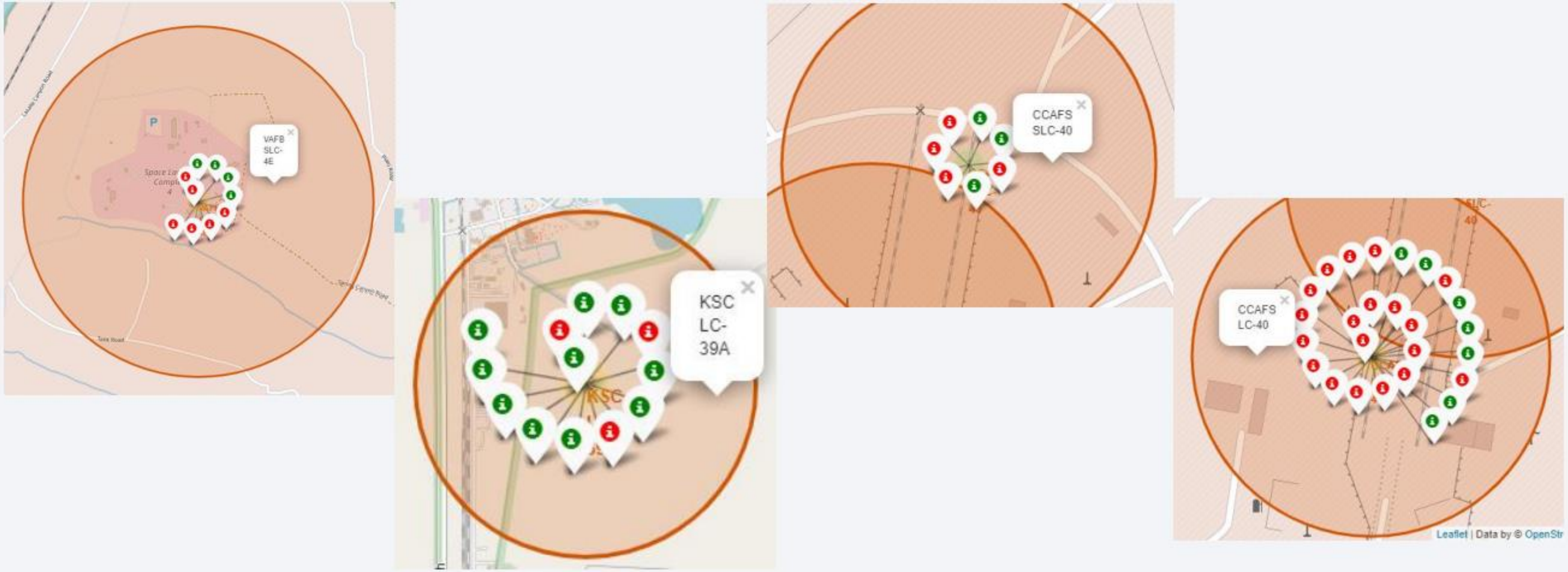
Launch Sites Proximities Analysis

Folium Map



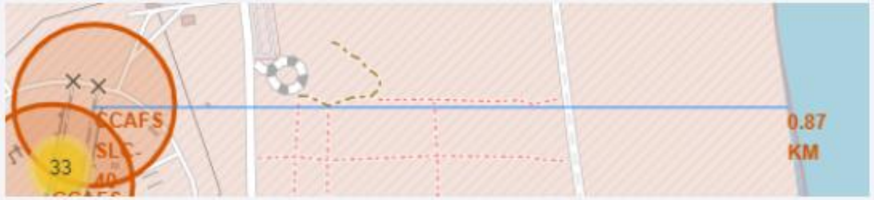
- Launch sites are clearly marked

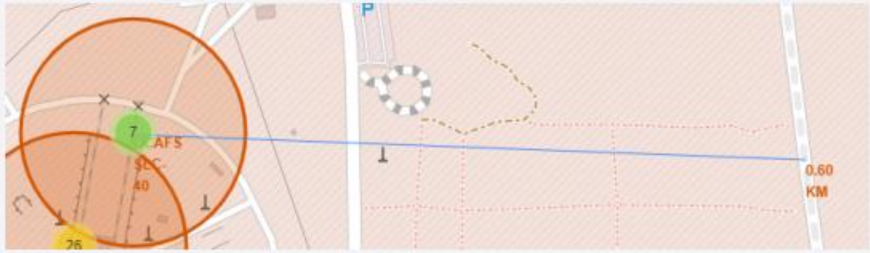
Folium Map: Markers



- Green represents successful and red represents unsuccessful missions.
- KSC LC-39A has the highest success rate.

Folium Map: distance between CCAFS SLC-40 and its proximities



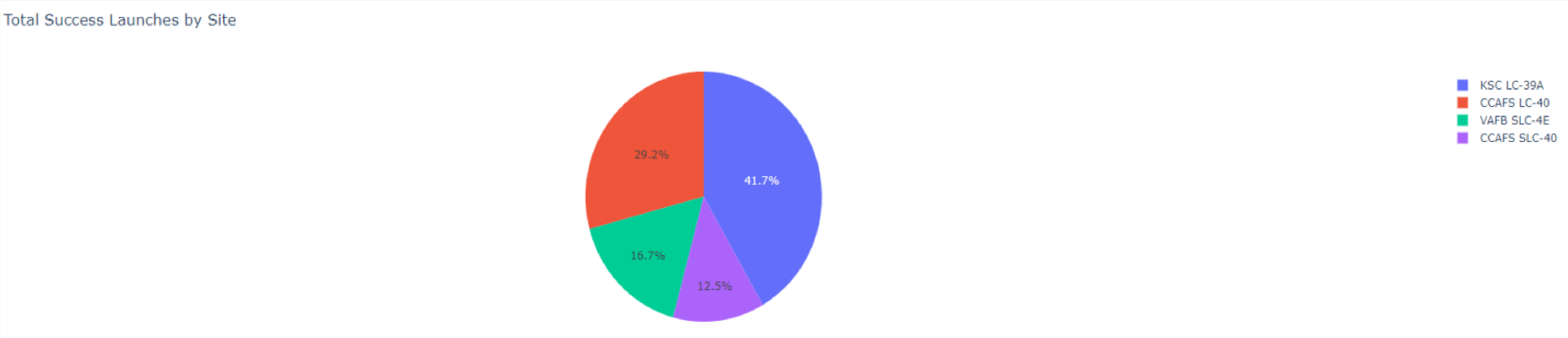


- CCAFS SLC-40 is close to railways, highways, coastline and city: Yes

Section 4

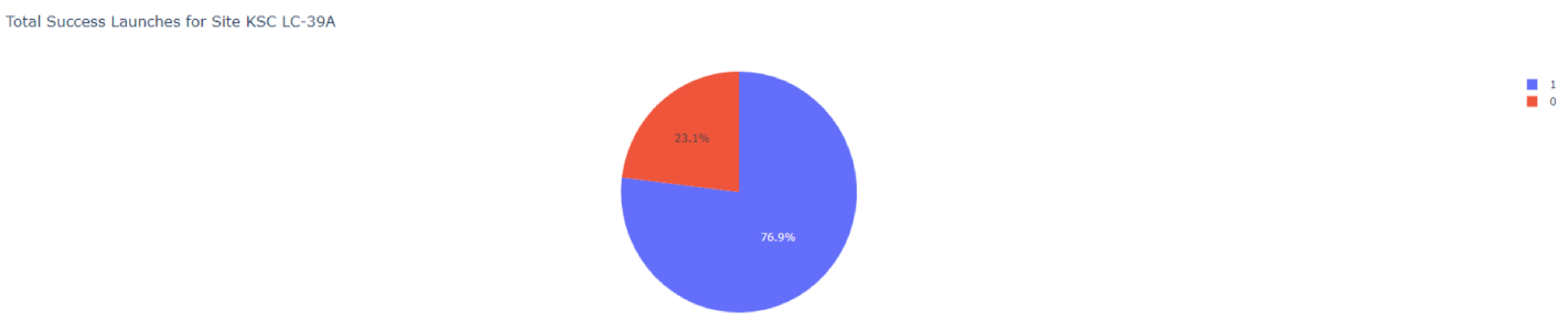
Build a Dashboard with Plotly Dash

Dashboard: Total success grouped by launch sites



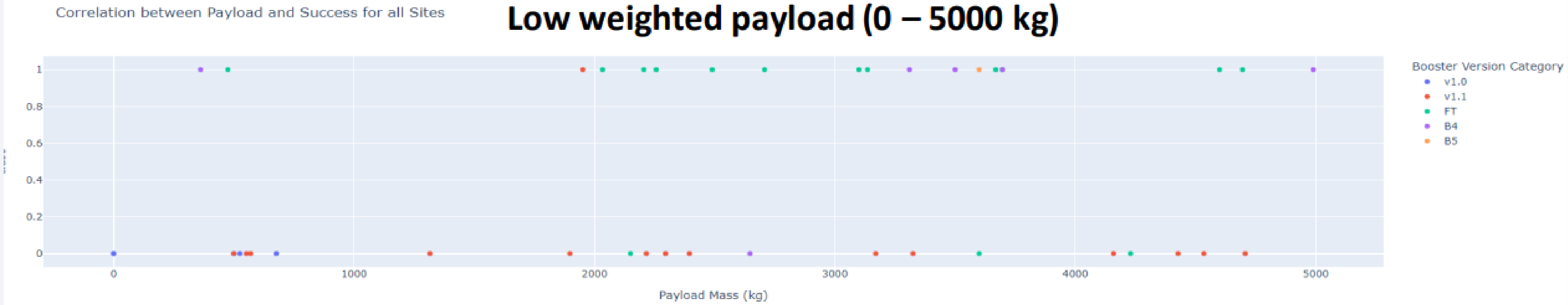
- KSC LC 39A has the highest success rate

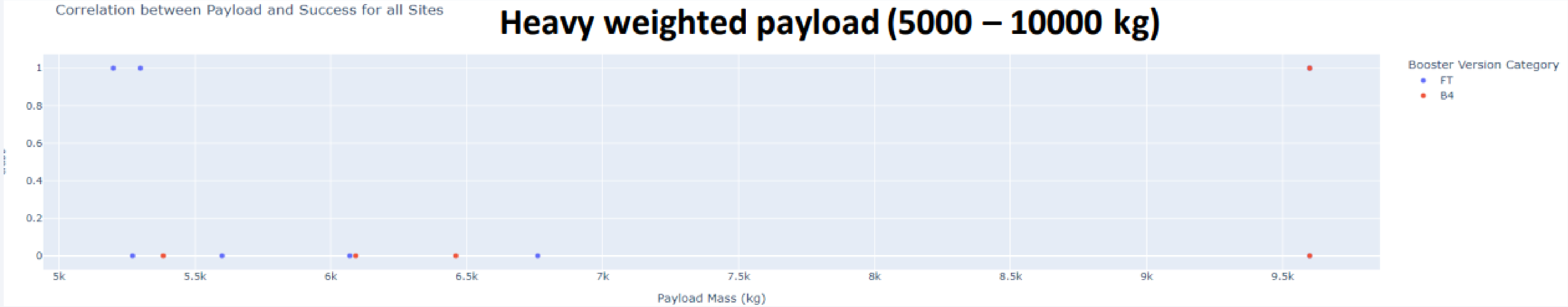
Dashboard: success rate breakdown of KSC LC-39A site



- KSC LC-39A has a breakdown of 76.9% successful missions and 23.1% failure rate.

Dashboard: Payload Mass vs outcome.



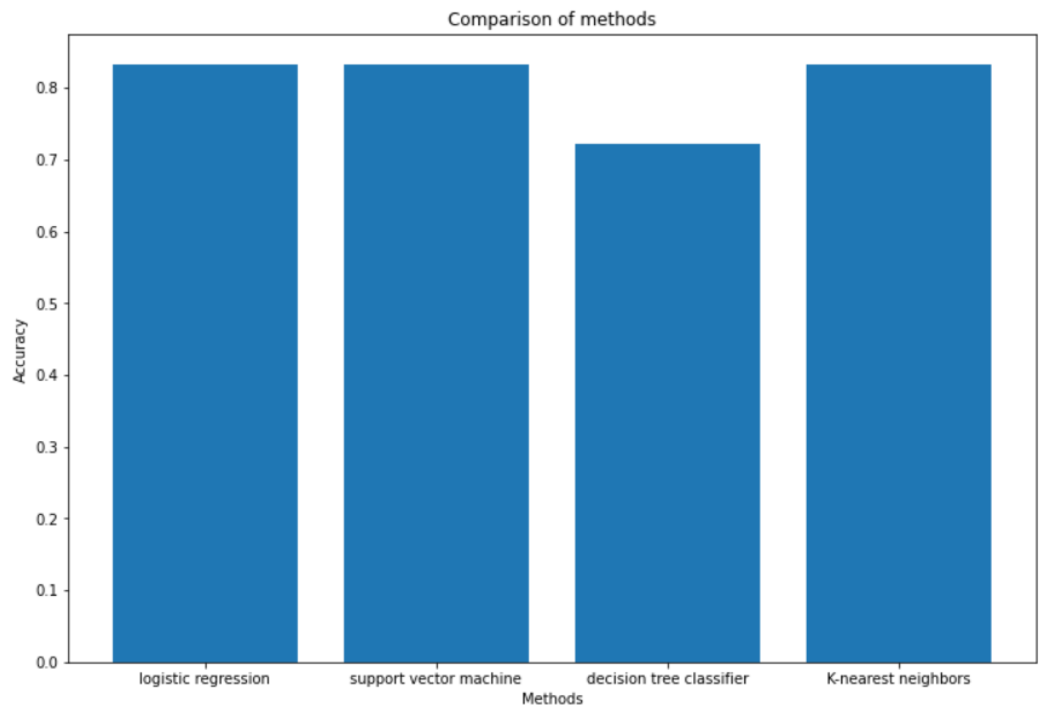


- Low payload mass has better success rate.

Section 5

Predictive Analysis (Classification)

Classification accuracy

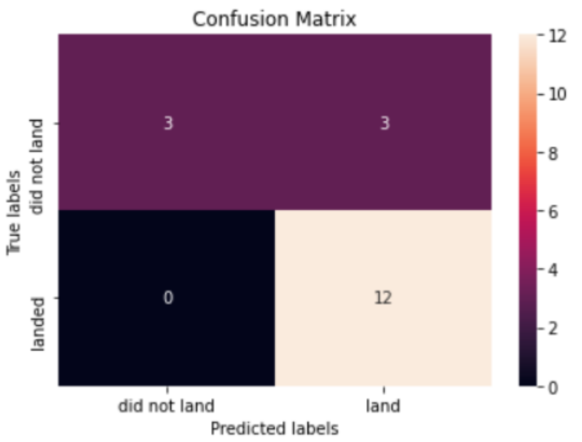


Logistic regression, **Support vector machine** and **K-nearest neighbors**, all of these three methods give the best performance, with accuracy of 0.8333333333333334.

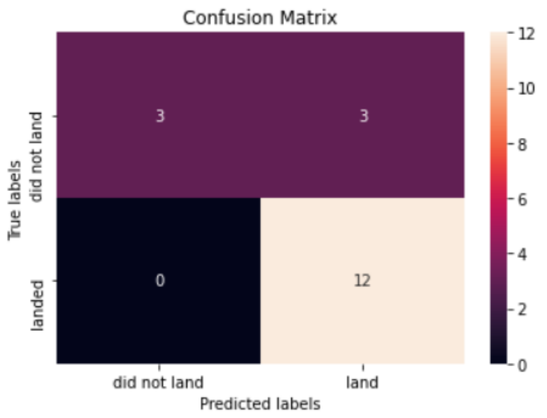
- Decision tree model has the lowest accuracy.
- Logistic regression, SVM and KNN show similar accuracy.

Confusion Matrix

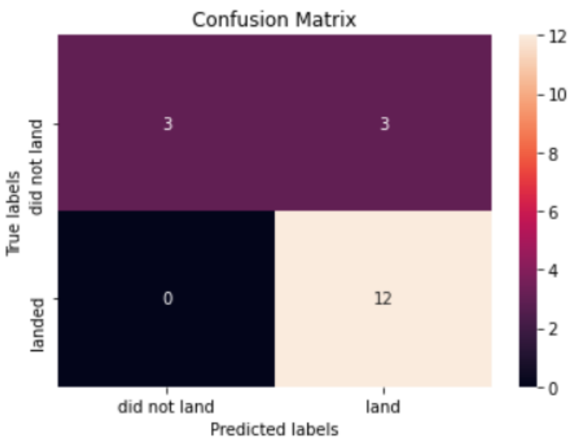
```
yhat = logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test, yhat)
```



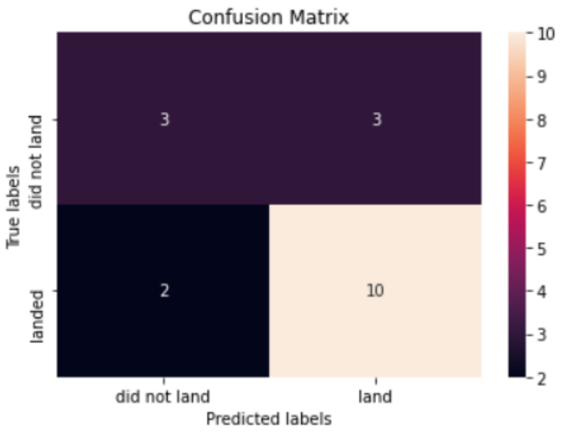

```
yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test, yhat)
```




```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test, yhat)
```




```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test, yhat)
```



- Confusion matrix is same for Logistic regression, SVM and KNN.
- Decision tree model has lower number of True Negatives.

Conclusions

- Success rate of a mission does vary with variables such as Launch site, payload mass, orbit and number of launches.
- Orbits with highest success rate are GEO, HEO, SSO, ES-L1.
- Some orbits showed better success rate with heavier payload and vice-versa.
- KNN, Logistic regression and SVM showed highest accuracy and therefore any of these models can be used.

Appendix

- https://github.com/AjisaMuthayilAli/Space-X-Falcon-9-First-Stage-Landing-Prediction

Thank you!