IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Shilpa Mehla
21st April 2023

Outline

Executive Summary

- **Summary of Methodologies**
  - ➢ -Data Collection through API
  - ➢ -Data Collection with Web Scraping
  - ➢ -Data Wrangling
  - ➢ -Exploratory Data Analysis with SQL
  - ➢ -Exploratory Data Analysis with Data Visualization
  - ➢ -Interactive Visual Analytics with Folium
  - ➢ -Interactive Visual Analytics with Plotly Dash
  - ➢ -Machine Learning Prediction
- **Summary of all Results**
  - ➢ -Exploratory Data Analysis Result
  - ➢ -Interactive analytics Results
  - ➢ -Predictive Analytics Results

Introduction

- **Project Background and Context**

In this project, 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. Therefore, if we can determine if the first stage will land, 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. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- **Problems we want to find answers**

➢ -What factors determine if the rocket will land successfully?

➢ -To determine if SpaceX will reuse the first stage, thus, determining the price of each launch.

Section 1

Methodology

Methodology

- Executive Summary
- Data collection methodology: Data was collected using SpaceX API and web scraping from Wiki pages.
- Perform data wrangling: One-hot encoding was applied to some of the features which were categorical.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

Data sets were collected using the below methods:

- -Data was collected using get request to the SpaceX API and also with the webscraping of the wiki pages.
- -For API one, the response content is decoded as a Json using .json() function call and converted to a pandas dataframe using .json_normalize().
- -For Webscraping, BeautifulSoup was used where launch record were extracted as HTML table, table was parsed and converted to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- Get request to the SpaceX API is used to collect data, cleaned the requested data and filled the missing values.

- Notebook link is:

https://github.com/Shilpa-Mehla/FinalRepo/blob/main/C10WK1-Data%20Collection%20API__1.1.ipynb

Now let's start requesting rocket launch data from SpaceX API with the following URL:

In [6]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [7]:
```
response = requests.get(spacex_url)
```

Check the content of the response

In [8]:
```
print(response.content)
```


s":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/94/f2/NN6Ph4
arge":"https://images2.imgbox.com/5b/02/QcxHUb5V_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"sm
ginal":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=0a_00nJ_Y88","youtube_id":"0a_00nJ_Y88","article":"https://www.space.c
ex-inaugural-falcon-1-rocket-lost-launch.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":"2006-03-17T00:0
static_fire_date_unix":1142553600,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","success":false,"failures":[{"time":33,"altitud
son":"merlin engine failure"}],"details":"Engine failure at 33 seconds and loss of vehicle","crew":[],"ships":[],"capsules":[],"payloads":
:3bb0006eeb1e1"],"launchpad":"5e9e4502f5090995de566f86","flight_number":1,"name":"FalconSat","date_utc":"2006-03-24T22:30:00.000Z","date_uni
),"date_local":"2006-03-25T10:30:00+12:00","date_precision":"hour","upcoming":false,"cores":[{"core":"5e9e289df35918033d3b2623","flight":
:false,"legs":false,"reused":false,"landing_attempt":false,"landing_success":null,"landing_type":null,"landpad":null}],"auto_update":true,"t
aunch_library_id":null,"id":"5eb87cd9ffd86e000604b32a"},{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":
{"patch":{"small":"https://images2.imgbox.com/f9/4a/ZboXReNb_o.png","large":"https://images2.imgbox.com/80/a2/bkWotCIS_o.png"},"reddit":{"ca
,"launch":null,"media":null,"recovery":null},"flickr":{"small":[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=
,"youtube_id":"Lk4zQ2wP-Nc","article":"https://www.space.com/3590-spacex-falcon-1-rocket-fails-reach-orbit.html","wikipedia":"https://en.wik
iki/DemoSat"},"static_fire_date_utc":null,"static_fire_date_unix":null,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","success":
es":[{"time":301,"altitude":289,"reason":"harmonic oscillation leading to premature engine shutdown"}],"details":"Successful first stage bu

Data Collection - Scraping

- BeautifulSoup is used for webscraping the Falcon 9 launch records
- Table parsed and converted into a pandas dataframe.
- Notebook link is:

https://github.com/Shilpa-Mehla/FinalRepo/blob/main/C10WK1-Data%20Collection%20with%20Webscraping__1.2.ipynb

To keep the lab tasks consistent, you will be asked to scrape the data from a snapshot of the `List of Falcon 9 and Falcon Heavy launches` Wikipage updated on `9th June 2021`

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
data  = requests.get(static_url).text
```

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

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data)
```

Print the page title to verify if the `BeautifulSoup` object was created properly

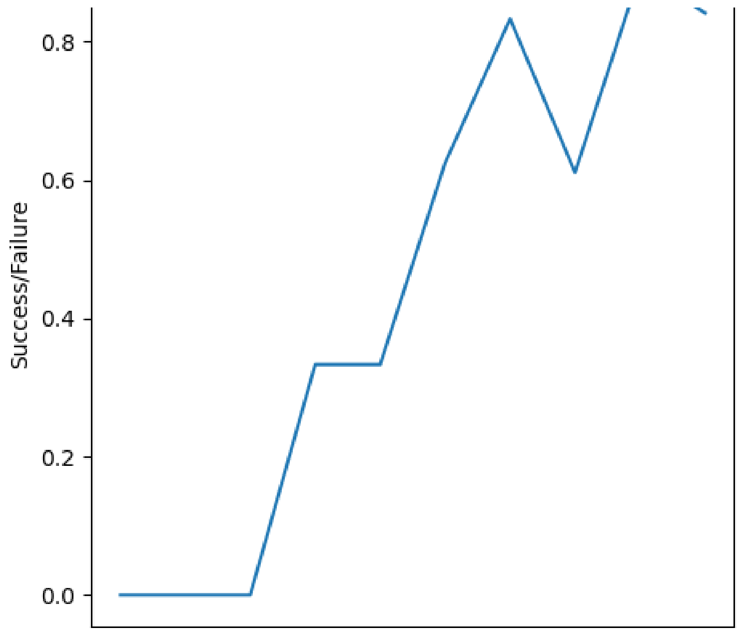
In [7]:
```
# Use soup.title attribute
soup.title
```

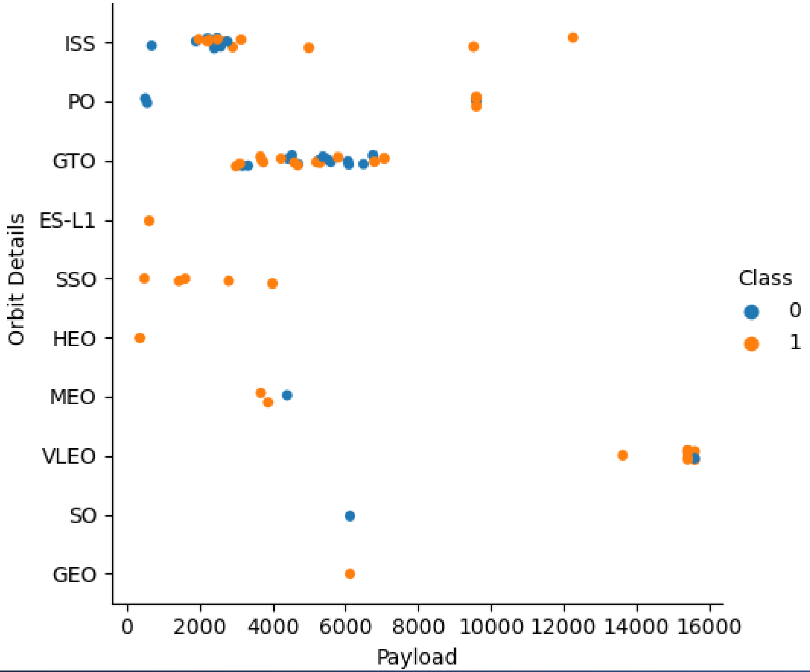
Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Data Wrangling

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- Notebook link is:

https://github.com/Shilpa-Mehla/FinalRepo/blob/main/C10WK1-spacex-Data%20Wrangling__1.3.jupyterlite.ipynb





EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly .

- Notebook link is:

https://github.com/Shilpa-Mehla/FinalRepo/blob/main/C10WK2-EDA%20With%20Visualization__2.2.ipynb

EDA with SQL

- SpaceX dataset is loaded into a SQL database first.
- Queries are used to get insights from the data like:

➢ **%sql** select Mission_Outcome, count(Mission_Outcome) from SPACEXTBL group by Mission_Outcome *// for total number of successful and failure mission outcomes*

➢ **%sql** select Booster_Version from SPACEXTBL where "PAYLOAD_MASS__KG_"=(select max("PAYLOAD_MASS__KG_") from SPACEXTBL) *//for names of the booster_versions which have carried the maximum payload mass*

➢ And many more..

- Notebook link is:

https://github.com/Shilpa-Mehla/FinalRepo/blob/main/C10WK2-EDA%20With%20SQL__2.1.ipynb

Build an Interactive Map with Folium

- All launch sites are marked, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- Feature launch outcomes (failure or success) are assigned to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Moreover, color-labeled marker clusters are used to identify which launch sites have relatively high success rate.
- Some of the questions are answered are as below:

➢ -Are launch sites near railways, highways and coastlines.

➢ -Do launch sites keep certain distance away from cities.

- Notebook link is:

https://github.com/Shilpa-Mehla/FinalRepo/blob/main/C10WK3-Data%20Visualization%20with%20Folium__3.1.ipynb

Build a Dashboard with Plotly Dash

Pie chart and Scatter plots are plotted using Plotly Dash.

Pie Chart showing the total launches by a certain site or all sites.

Scatter plot showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

Notebook link is:

https://github.com/Shilpa-Mehla/FinalRepo/blob/main/SpaceX-Plotly%20Dash.py

Predictive Analysis (Classification)

Data is loaded using numpy and pandas and other required libraries, transformed, and data is splitted into training and testing.

Different machine learning models are used and tuned different hyperparameters using GridSearchCV for each model.

Accuracy as the metric for our model is used to identify which model performs the best.

Notebook link is:

https://github.com/Shilpa-Mehla/FinalRepo/blob/main/C10WK4-Predictive%20Analysis%20Classification__4.1.ipynb

Results

Exploratory data analysis
Results

Interactive analytics
Results

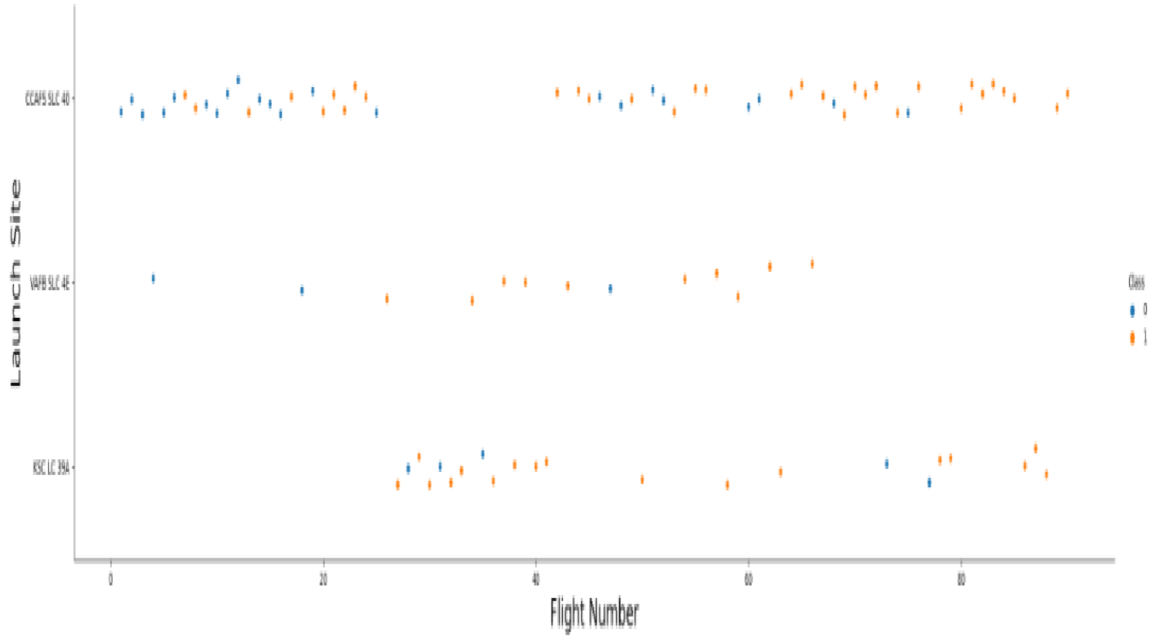
Predictive analysis
Results

Section 2

Insights drawn from EDA

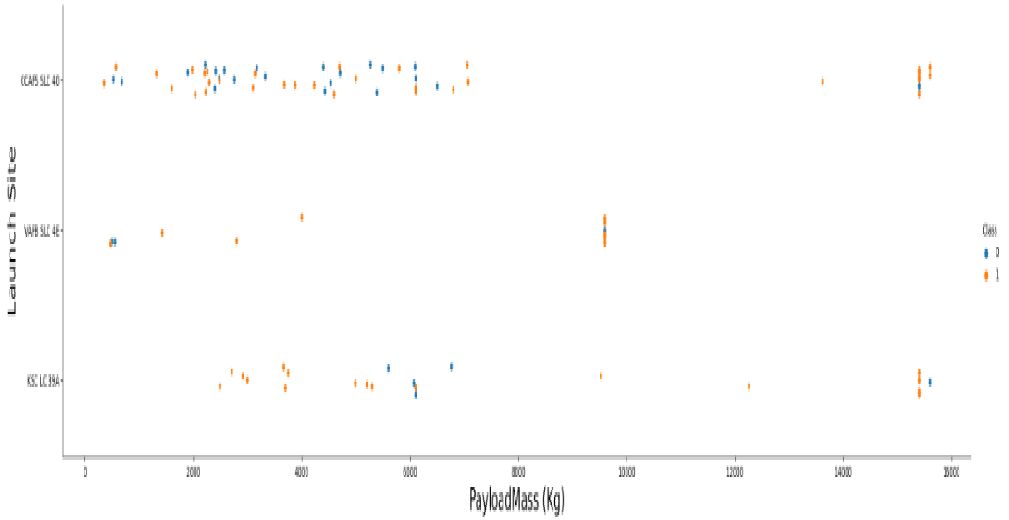
Flight Number vs. Launch Site

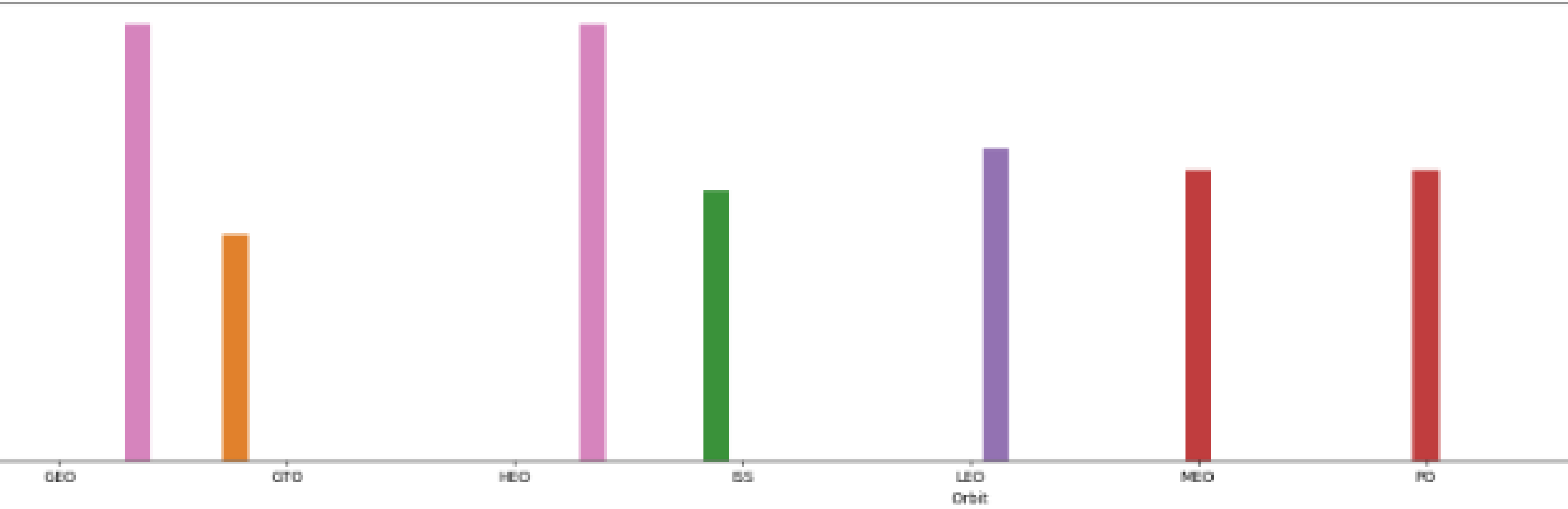
- It can be inferred as larger the flight amount at a launch site, the larger the success rate at a launch site.



Payload vs. Launch Site

- It can be inferred as greater the payload mass, the larger the success rate at a launch site.
- But for VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).



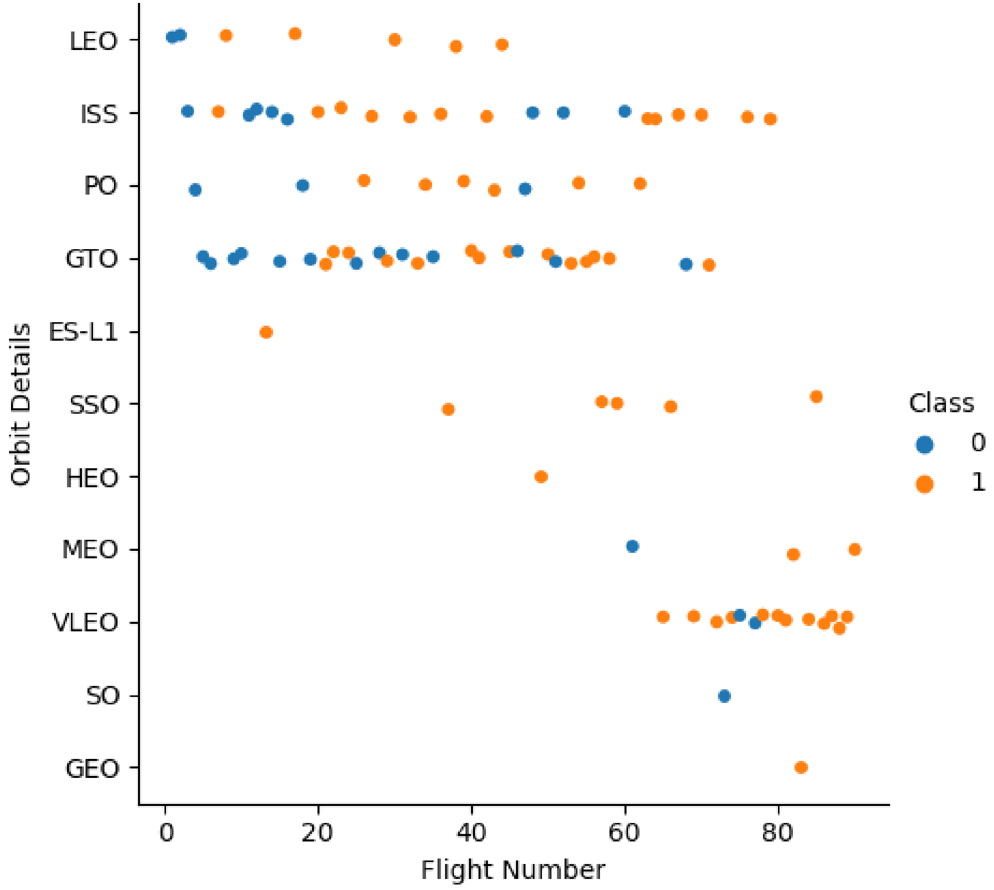


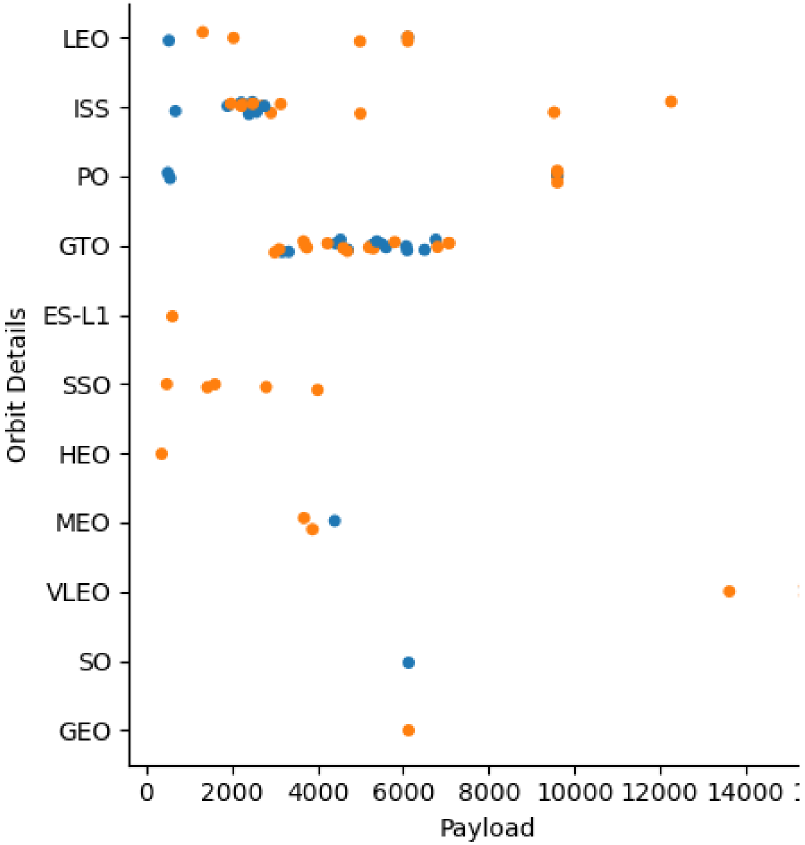
Success Rate vs. Orbit Type

- It can be inferred as ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

Flight Number vs. Orbit Type

- We can see that for the LEO orbit the Success appears related to the number of flights; whereas, for GTO orbit, there seems to be no relationship between flight number.



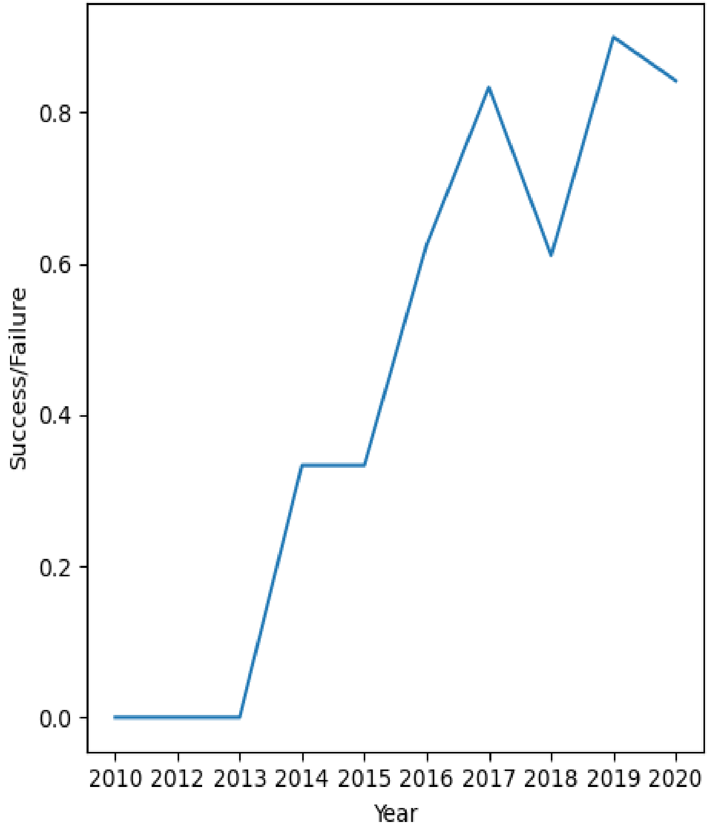


Payload vs. Orbit Type

- We can see that 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

- It can be observed that the success rate since 2013 kept increasing till 2020.



All Launch Site Names

Display the names of the unique launch sites in the space mission

In [7]:
```
%sql select distinct("Launch_Site") from SPACEXTBL
```

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

Out[7]:

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

DISTINCT key word is used to show only unique launch sites from the SpaceX data.

Launch Site Names Begin with 'CCA'

Below query is used.

Task 2

Display 5 records where launch sites begin with the string 'CCA'

In [8]:
```
%sql select * from SPACEXTBL where Launch_Site like "CCA%" limit 5
```

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

Out[8]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Below query is used:

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

In [9]:
```
%sql select sum("PAYLOAD_MASS__KG_") from SPACEXTBL where Customer="NASA (CRS)"
```

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

Out[9]:

sum("PAYLOAD_MASS__KG_")
45596

Task 4

Display average payload mass carried by booster version F9 v1.1

In [10]:
```
%sql select avg("PAYLOAD_MASS__KG_") from SPACEXTBL where Booster_Version="F9 v1.1"
```

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

Out[10]:

avg("PAYLOAD_MASS__KG_")
2928.4

Average Payload Mass by F9 v1.1

- Above query is used:

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

In [11]:
```
%sql select min(Date) from SPACEXTBL where "Landing _Outcome"="Success (ground pad)"
```

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

Out[11]:

min(Date)
01-05-2017

First Successful Ground Landing Date

- Above query is used:

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 600

In [12]:

```
%%sql select Booster_Version from SPACEXTBL where "Landing _Outcome"="Success (drone ship)" and
("PAYLOAD_MASS__KG_">4000 and "PAYLOAD_MASS__KG_"<6000)
```

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

Out[12]:

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

Successful Drone Ship Landing with Payload between 4000 and 6000

- Above query is used:

List the total number of successful and failure mission outcomes

In [13]:
```
%sql select Mission_Outcome, count(Mission_Outcome) from SPACEXTBL group by Mission_Outcome
```

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

Out[13]:

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

Total Number of Successful and Failure Mission Outcomes

- Above query is used:

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [14]:
```
%sql select Booster_Version from SPACEXTBL where "PAYLOAD_MASS__KG_"=(select max("PAYLOAD_MASS__KG_") from SPACEXTBL)
```

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

Out[14]:

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

Boosters Carried Maximum Payload

- Above query is used:

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, laun

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the month

```
In [15]:  %%sql SELECT substr(Date, 4, 2) as Month,"Landing _Outcome",Booster_Version,Launch_Site FROM SPACEXTB
          where substr(Date,7,4)='2015' and "Landing _Outcome"="Failure (drone ship)"

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

Out[15]:

Month	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

2015 Launch Records

- Above query is used:

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

In [32]:
```
%sql SELECT "Date", count("Landing _Outcome") as Count FROM SPACEXTBL \
WHERE "Date" BETWEEN "04-06-2010" AND "20-03-2017" and "Landing _Outcome" like "%Success%" \
group by "Date" \
order BY count("Landing _Outcome") desc
```

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

Out[32]:

Date	Count
19-02-2017	1
18-10-2020	1
18-08-2020	1
18-07-2016	1
18-04-2018	1
17-12-2019	1
16-11-2020	1

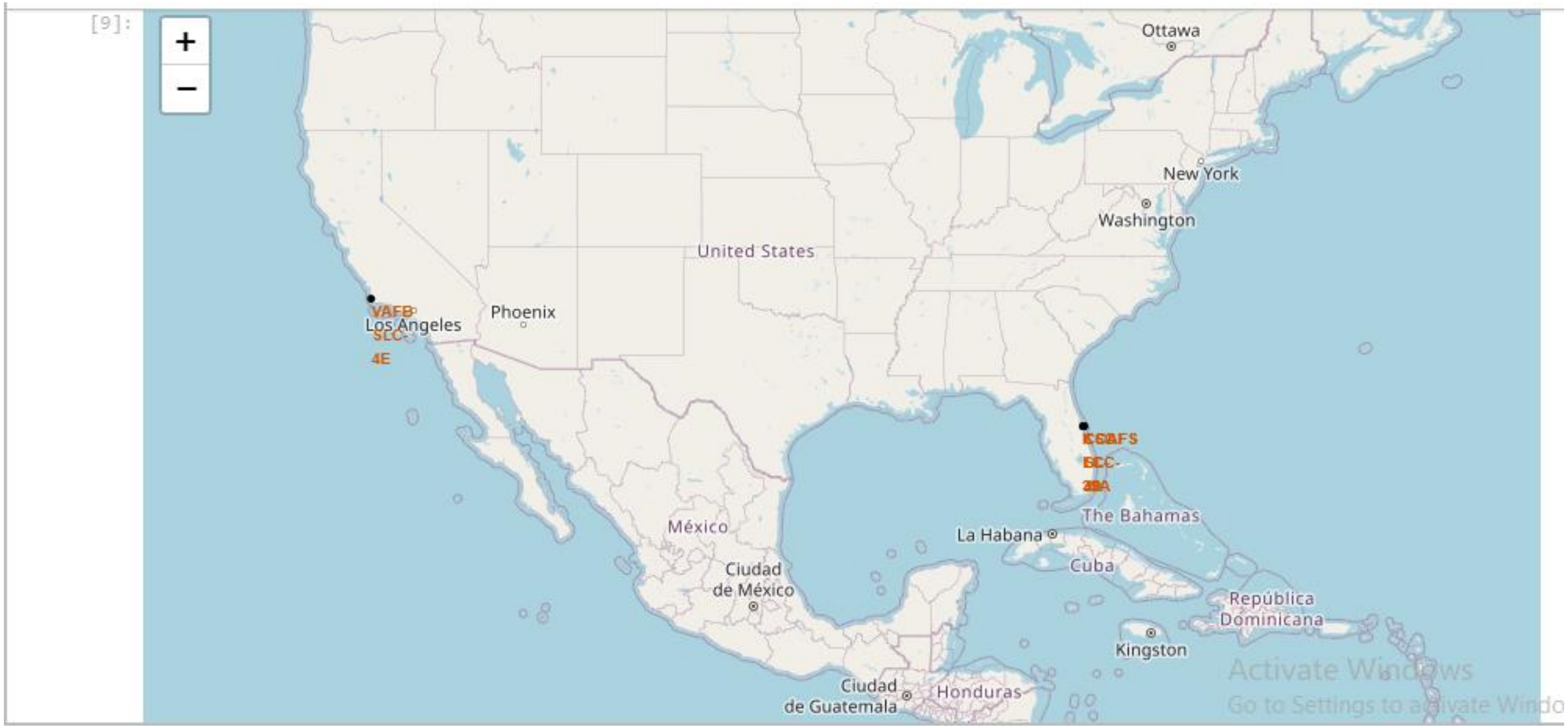
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- Above query is used:

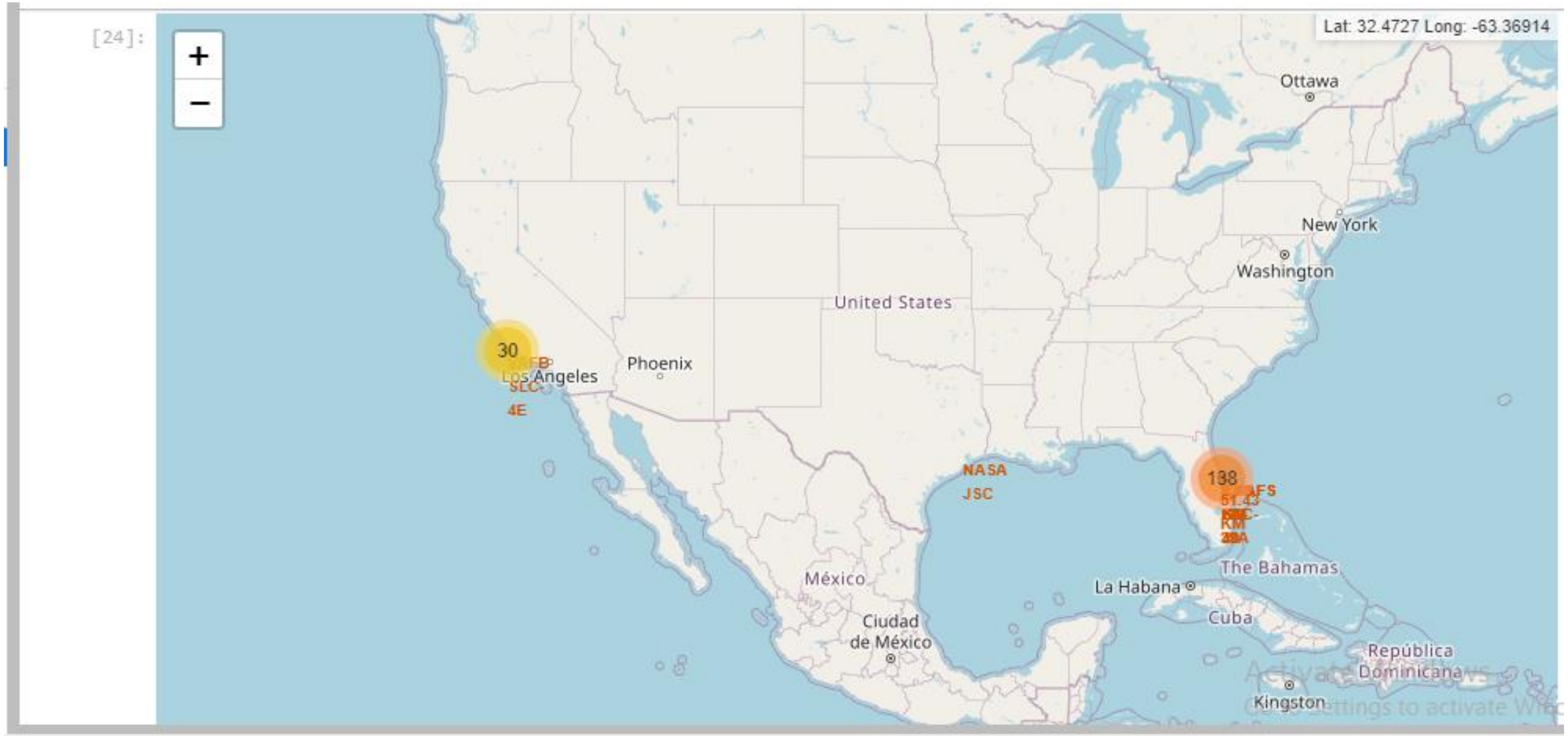
Section 3

Launch Sites Proximities Analysis

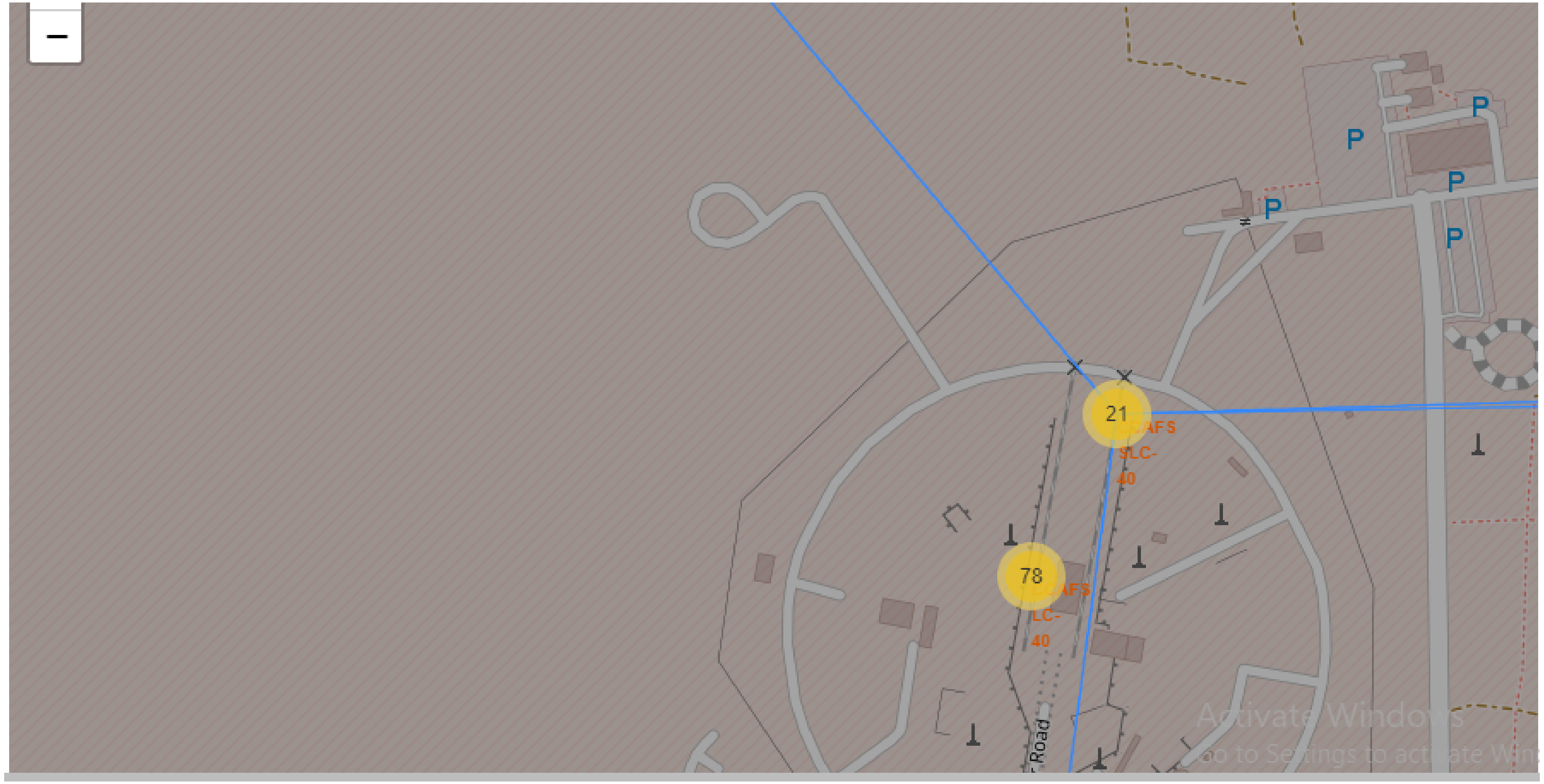
LAUNCH SITES ON A GLOBAL MAP



LAUNCH SITES WITH COLOR MARKERS



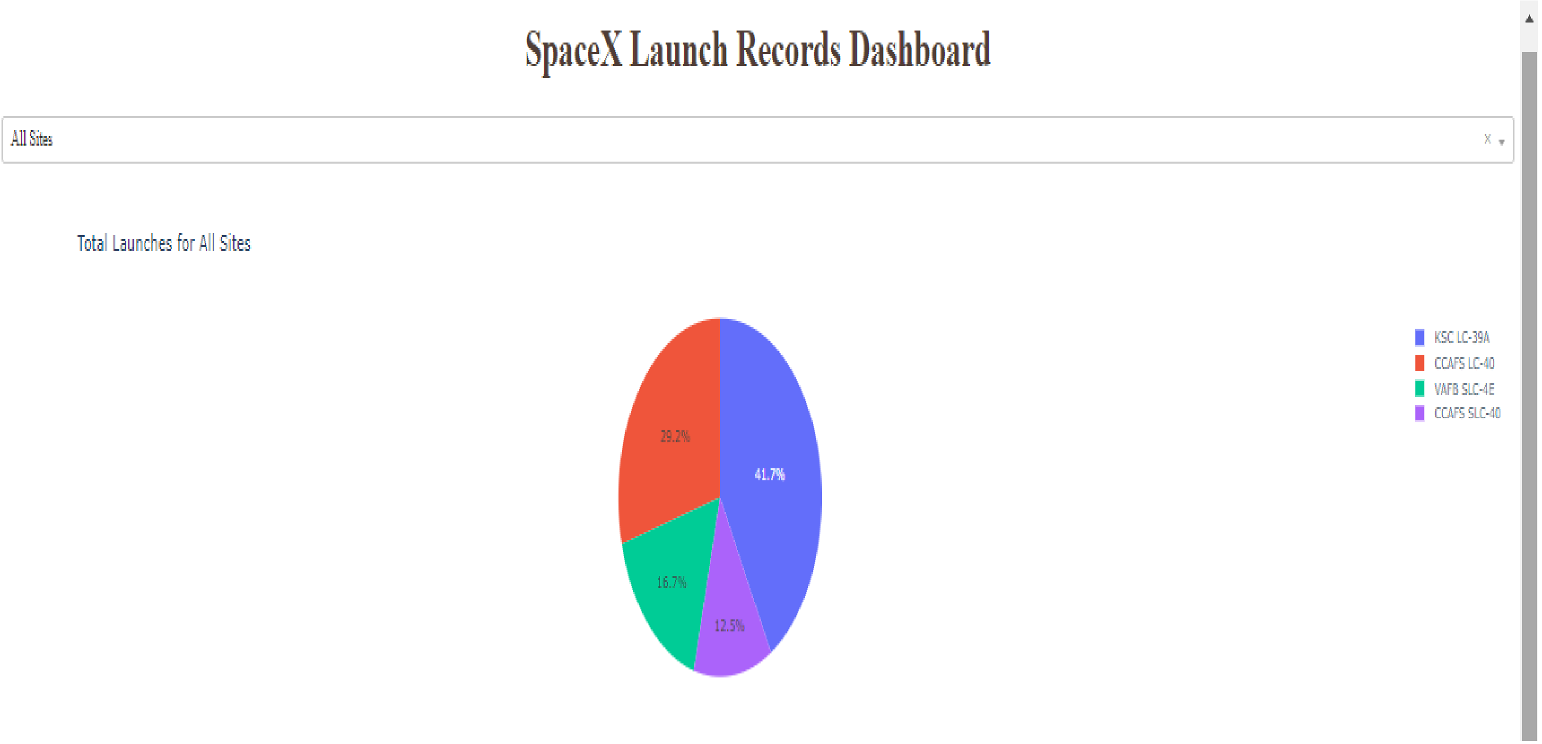
LAUNCH SITES DISTANCE TO LANDMARKS



Section 4

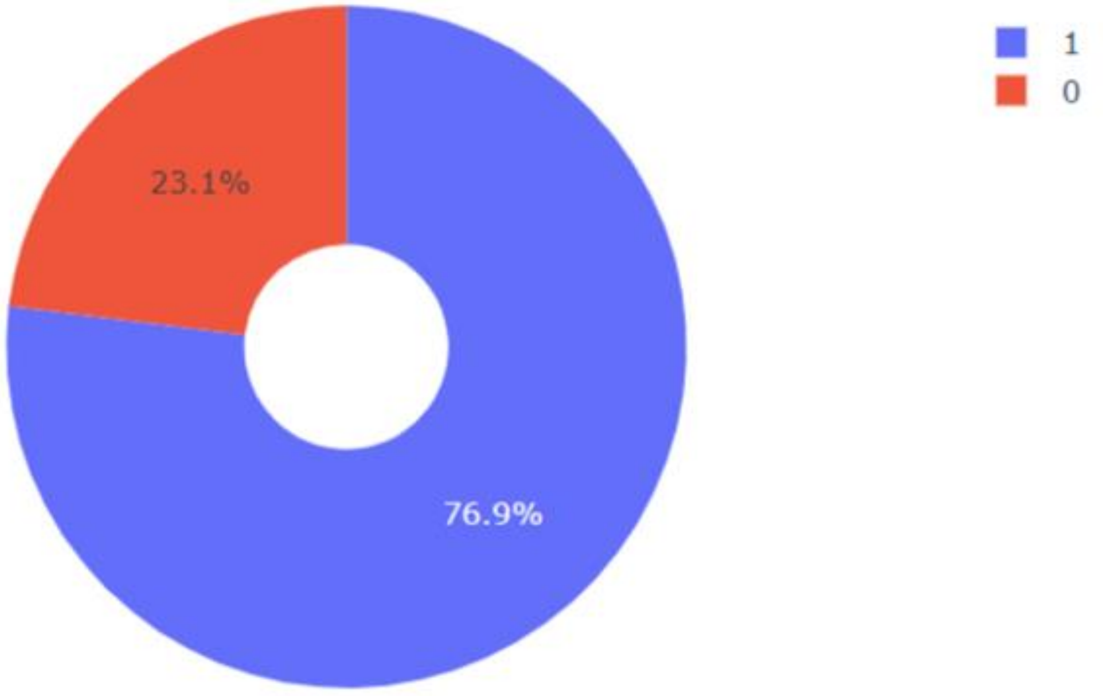
Build a Dashboard with Plotly Dash

PIE CHART SHOWING SUCCESS LAUNCHES FOR ALL SITES



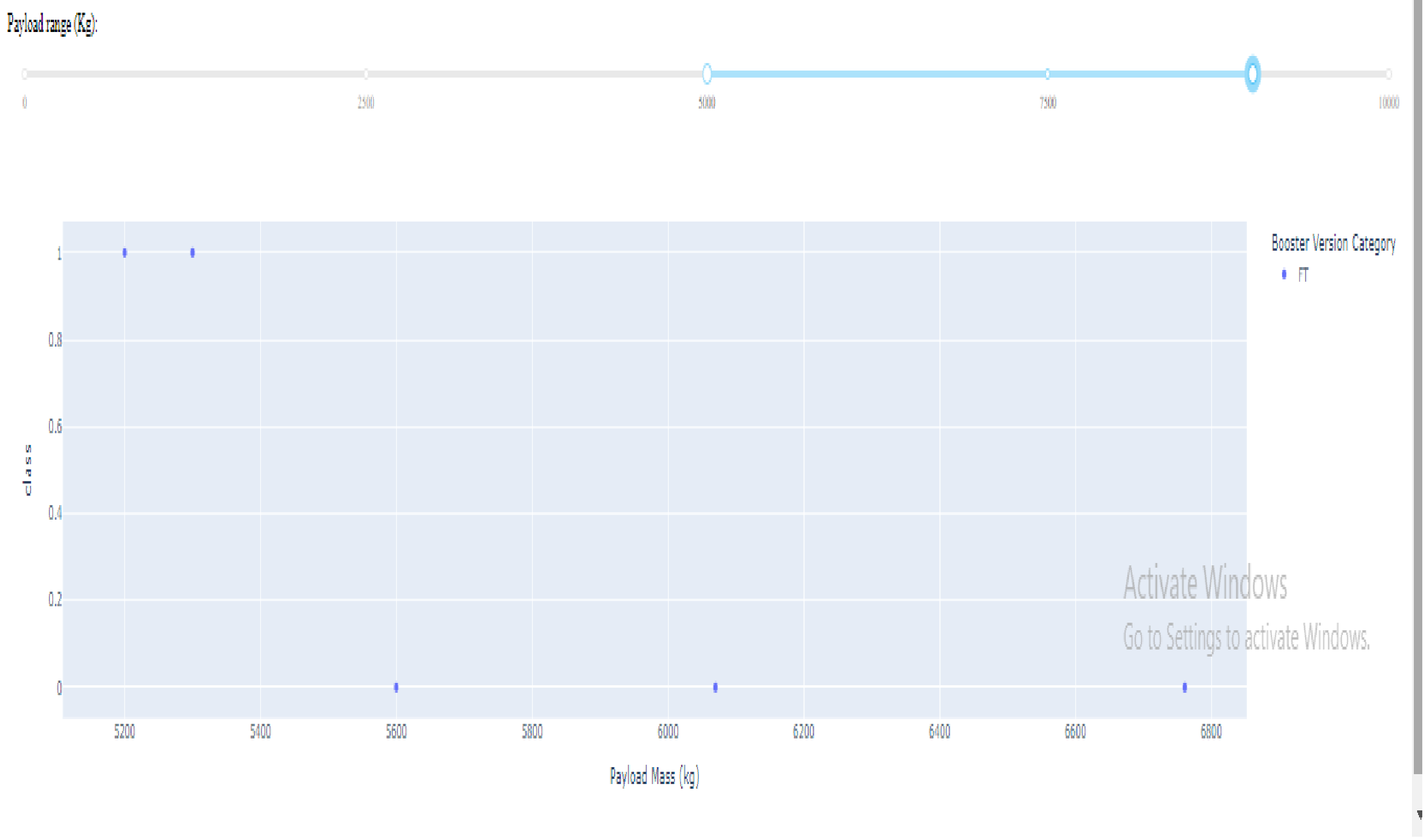
KSC LC-39A having the most successful launches from all sites.

PIE CHART SHOWING THE SITE HAVING THE HIGHEST SUCCESS LAUNCH RATE



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

SCATTER PLOT WITH DIFFERENT PAYLOAD SELECTED IN THE RANGE SLIDER



Section 5

Predictive Analysis (Classification)

Classification Accuracy

Find the method performs best:

In [38]:

```python
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearest neighbors method:', knn_cv.score(X_test, Y_test))
```

```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.8333333333333334
Accuracy for K nearest neighbors method: 0.8333333333333334
```

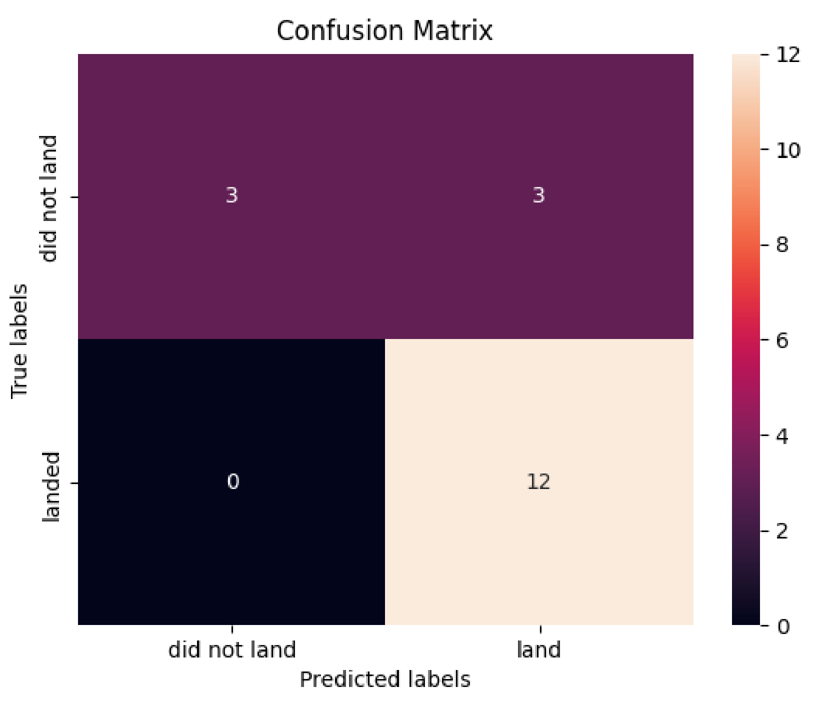
In [ ]:

```python
#All methods have equal accuracy
```

- It can be seen that all models have equal accuracy which is 83.3%

Confusion Matrix

- As all the models have equal accuracy , so confusion matrix is same for all, considering as presented.
- Classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

There are key points to be concluded so far as mentioned here:

The larger the flight amount at a launch site, the greater the success rate at a launch site.

Launch success rate started to increase in 2013 till 2020.

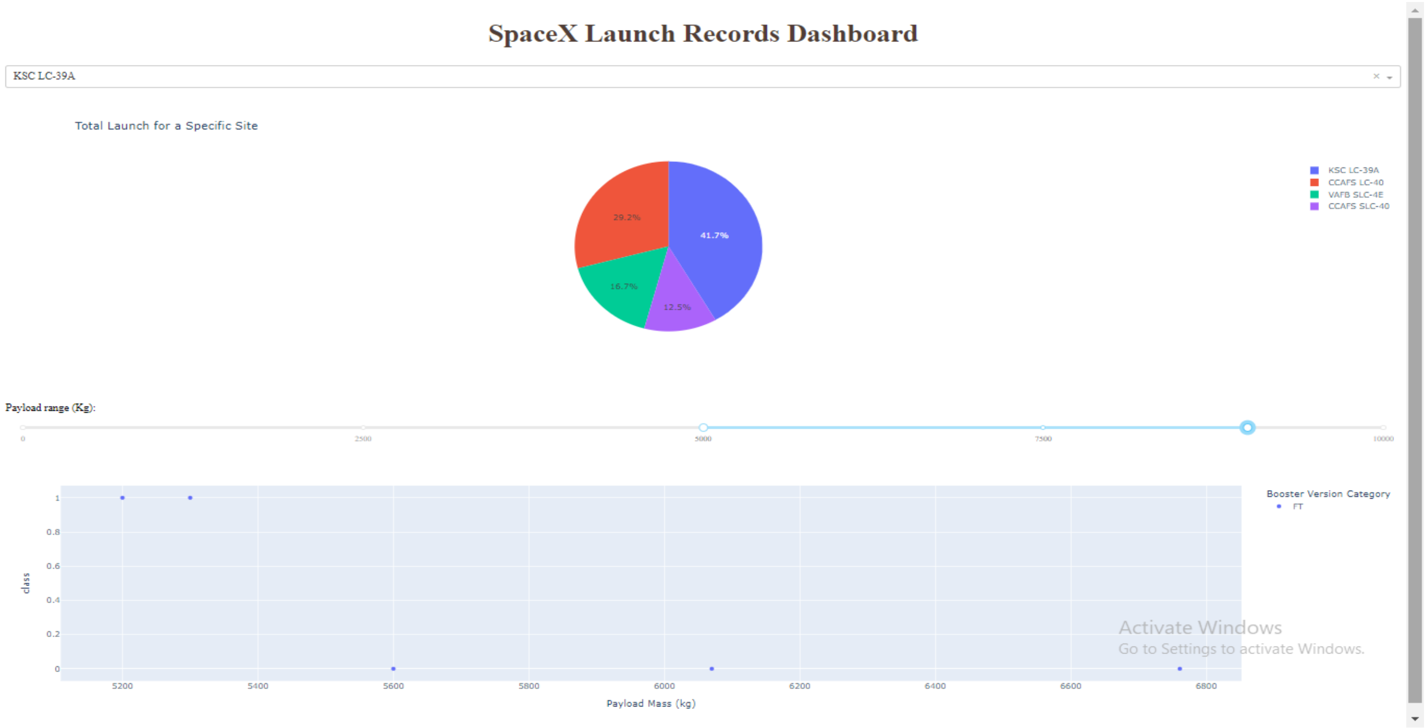
Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

KSC LC-39A had the most successful launches of any sites.

All the models we used have the equal accuracy, so can be used anyone. Be it Logistic Regression, Support Vector Machines(SVM), K-Nearest Neighbors(KNN) and Decision Tree.

Appendix

Showing the Pie chart and Scatter plot for KSC LC-39A which is having most successful launches.



One Innovative Insight:

Different Booster Versions have different Successful launches i.e., one booster version may appear as successful while other one may not at a time.

Thank you!