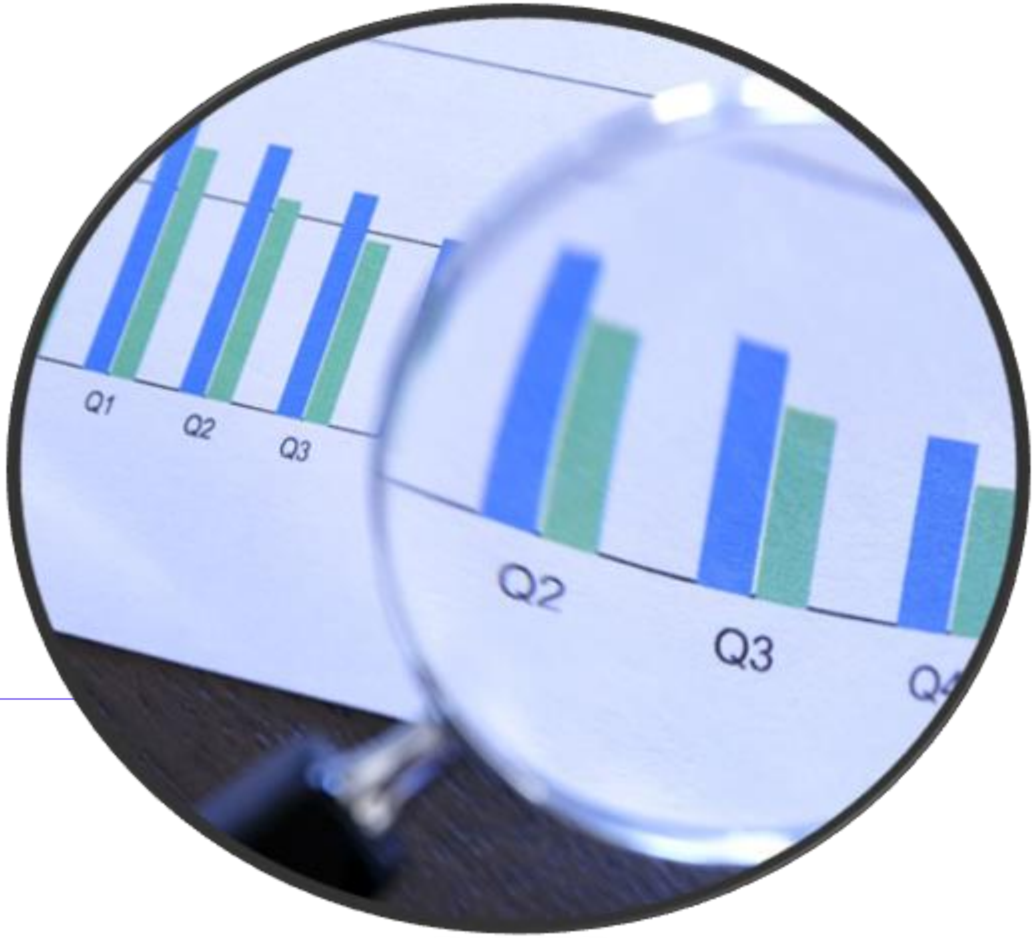
**Applied Data Science Capstone**

# Space Y Falcon 9 First Stage Landing Prediction

Alexander Yusyuk
01.13.2024




https://github.com/AlexYusyuk/IBM-Applied-Data-Science-Capstone/tree/main

# Presentation Contents

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

# EXECUTIVE SUMMARY

**Summary of Methodologies:**

- Data Collection via API, Web Scraping
- Exploratory Data Analysis(EDA) with Data Visualization
- Exploratory Data Analysis(EDA) with SQL
- Interactive Map with Folium
- Dashboards with Plotly Dash
- Predictive Analysis

**Summary of all results:**

- Exploratory Data Analysis results
- Interactive maps and dashboard
- Predictive results

# INTRODUCTION

## Project Background and context

The aim of this project is to predict if the Falcon 9 first stage will successfully land. Other providers cost upward of 165 million dollars each. The price difference is explained by the fact that SpaceX can reuse the first stage. By determining if the stage will land, we can determine the cost a launch. This information is interesting for another company if it wants to complete with SpaceX for a rocket launch.

## Problems you want to find answers

1. What are the main characteristics of a successful or failed landing?
2. What are the effect of each relationship of the rocket variables on The success or failure of a landing?
3. What are the conditions which will allow SpaceX to achieve the best Landing success rate?

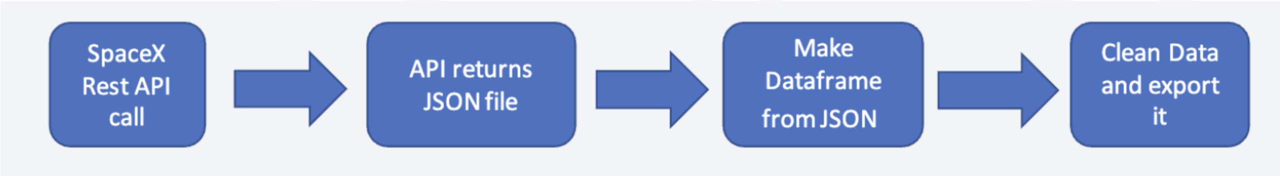
# Methodology

# METHODOLOGY

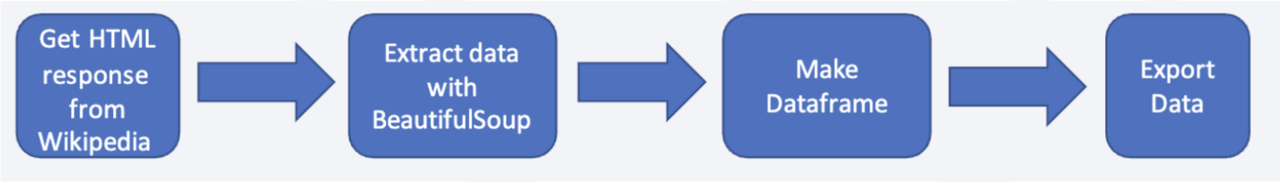
- Data collection methodology
  - Using SpaceX Rest API
  - Using Web Scrapping from Wikipedia
- Performed data wrangling
  - Filtering the data
  - Dealing with missing values
  - Using One Hot Encoding to prepare the data to a binary classification
- Performed exploratory data analysis (EDA) using visualization and SQL.
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models
  - Building, tuning and evaluation of classification models to ensure the best results

# Data Collection

- Datasets are collected from Rest SpaceX API and webscrapping Wikipedia
  - The information obtained by the API are rocket, launches, payload information
    - The Space X REST API URL is api.spacexdata.com/v4/



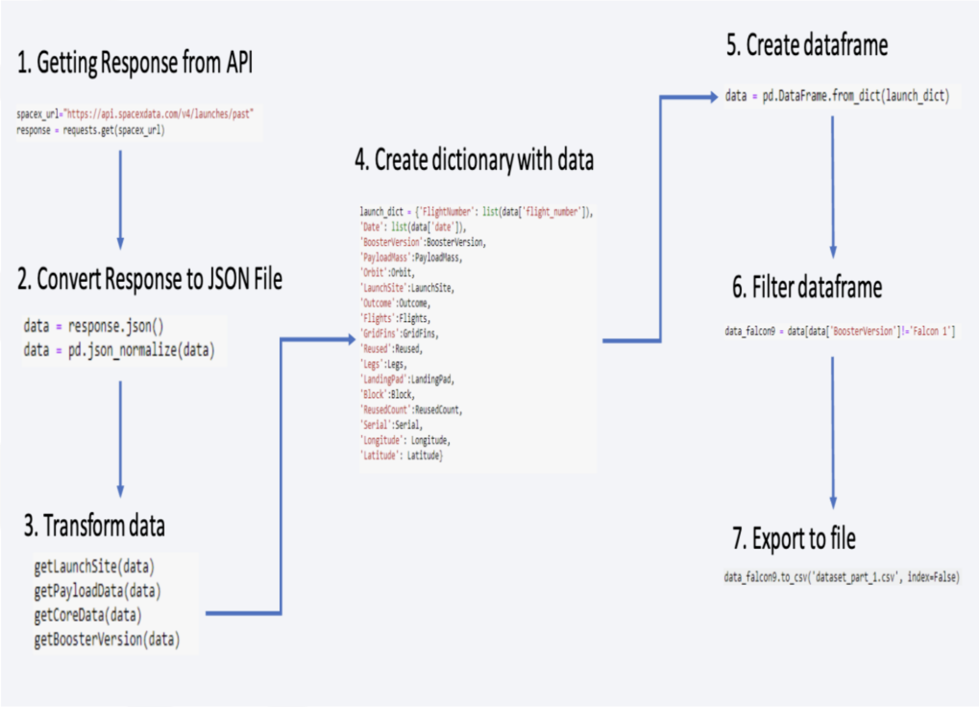

- The information obtained by the webscrapping of Wikipedia are launches, landing, payload information.
  - URL is https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922



# Data Collection – SpaceX API

**Steps**

- Request data from SpaceX API(rocket launch data)
- Decode responce using .json() and convert to a dataframe using .json_normalize()
- Request information about the launches from SpaceX API using costum functions
- Create dictionary from the data
- Create dataframe from the dictionary
- Filter dataframe to contain only Falcon 9 launches
- Replace missing values of Payload Mass with calculated .mean()
- Export data to cvs file



# Data Collection- Scraping

### 1. Getting Response from HTML

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

### 2. Create BeautifulSoup Object

```
soup = BeautifulSoup(response.text, "html5lib")
```

### 3. Find all tables

```
html_tables = soup.findAll('table')
```

### 4. Get column names

```
for th in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0 :
        column_names.append(name)
```

### 5. Create 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. Add data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is a
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.stri
                flag=flight_number.isdigit()
```

**See notebook for the rest of code**

### 7. Create dataframe from dictionary

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

### 8. Export to file

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

# Data Wrangling

- Perform EDA  and determine data labels
- Calculate:
  - #of launches for each site
  - #occurrence of orbit
  - #occurence of mission outcome per orbit type
- Create binary landing outcome column (dependent variable)
- Export data to csv file.

**Landing Outcome:**

- In the dataset, there are several cases where the booster did not land successfully.
  - True Ocean, True RTLS, True ASDS means the mission has been successful.
- We need to transfor string variables intoccategoricla variables where 1 means the mission has been successful and 0 means the mission was a failure

# EDA with Data Visualization

- Scatter Graphs
  - 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
- Bar Graph
  - Success rate vs. Orbit
- Line Graph
  - Success rate vs Year

# EDA with SQL

- We performed SQL queries to gather and understand data from dataset:
  - Displaying the names of the unique launch site in the space mission.
  - Display 5 records where launch sites begin with the string 'CCA'.
  - .Display the total payload mass carrier by boosters launched by NASA.
  - Display average payload mass carried by booster version F9 v1.1.
  - List the date when the first successful landing outcome in group pad was achived.
  - 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
  - List the records which will display the month names, failure in drone ship, booster version, launch site for the months in a year 2015

# All Launch Site Names

- SQL Query:

SELECT DISTINCT "LAUNCH SITE" FROM SPACEXTBL

**Results**

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

# Launch Site Names Begin with 'CCA'

- SQL Query

SELECT * FROM SPACEXTBL WHERE 'LAUNCH_SITE' LIKE '%CCA%' LIMIT 5

**Results**

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer |
|---|---|---|---|---|---|---|---|
| 04-06-2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX |
| 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 |
| 22-05-2012 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) |
| 08-10-2012 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) |
| 01-03-2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) |

# Total Payload Mass

%sql select sum(PAYLOAD_MASS_KG) from SPACEXTBL where COSUMER = 'NASA (CRS)'

45596

# Average Payload Mass by F9 v1.1

- %sql select avg(PAYLOAD_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 BOOSTER_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 SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = ' Failure (in flight)'

100

# Booster Carrier 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 Record

- %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] |

# Data Collection

## The data sets are collected using 2 methods :

- **Request to the SpaceX API**
  - Gathered SpaceX's past launch data via their open-source API.
  - Retrieved and processed this data with GET request.
  - Ensured the data included only Falcon 9 launches
  - Filled in missing payload weights from secret missions with average values
- **Web Scraping**
  - Request past Falcon 9 and Falcon Heavy launch data from Wikipedia's relevant page.
  - Accessed the Falcon 9 Launch page via its direct Wikipadia link.
  - Extracted all the column names from the HTML table.
  - Parsed and transformed the table into a Pandas data frrame suitable for analysis.

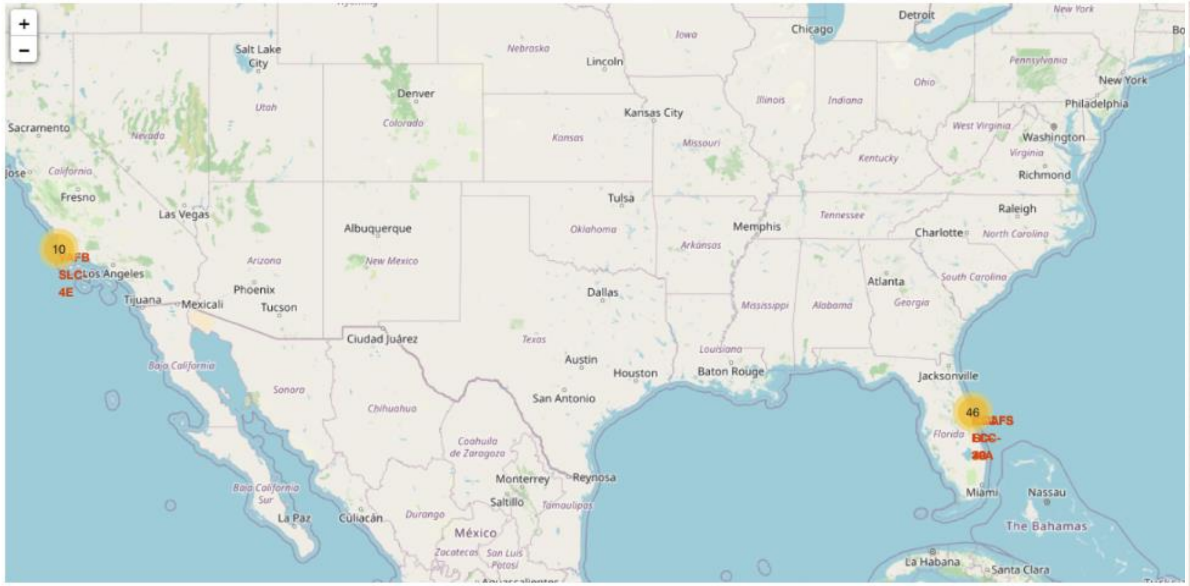
# Build an Interactive Map with Folium

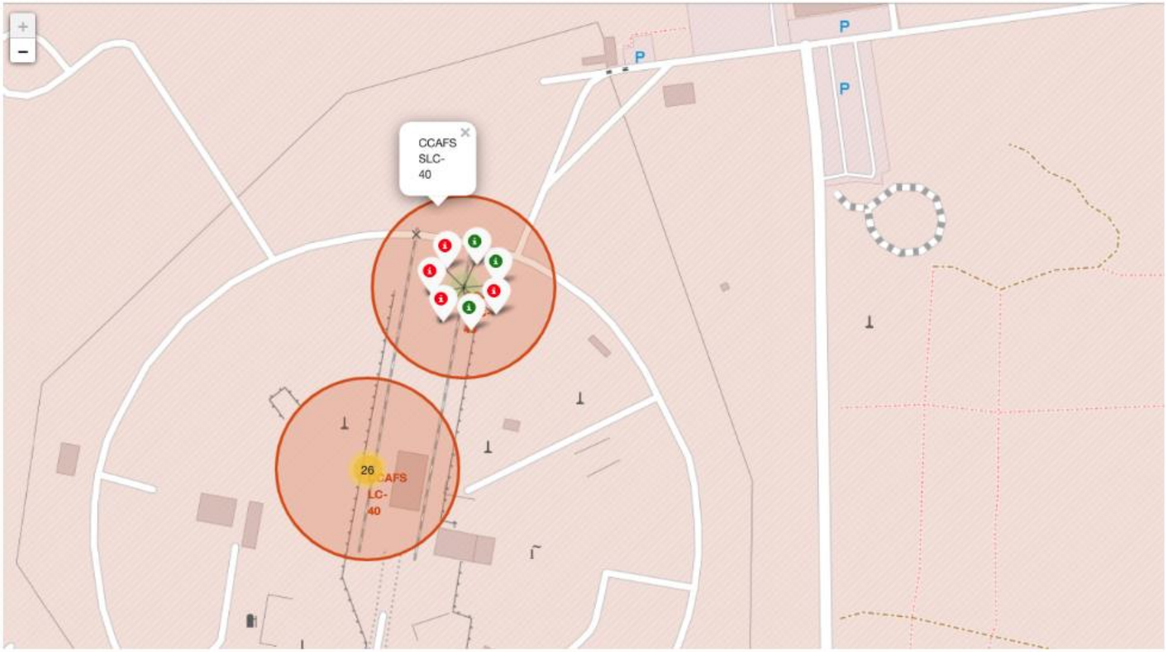
**Folium map object is a map centered on NASA Johnson Space Center at Huston, Texas**

- Red circle at NASA Johnson Space Center's coordinate with label showing it's name
- Red circles at each launch site coordinates with label showing launch site name
- The grouping of points in a cluster to display multile and different information for the same coordinates
- Markers to show successful and unsuccessful landing. Green for successful landing and Red for unsuccessful landing
- Markers to show distance between launch site to key locations and plot a line between them.

**These objects are created in order to understand better the problem and the data. We can show easily all launch sites, their surroundings and the number of successful and unsuccessful landing**

# Folium Map Slides



# Prediction modes used and best performing model.

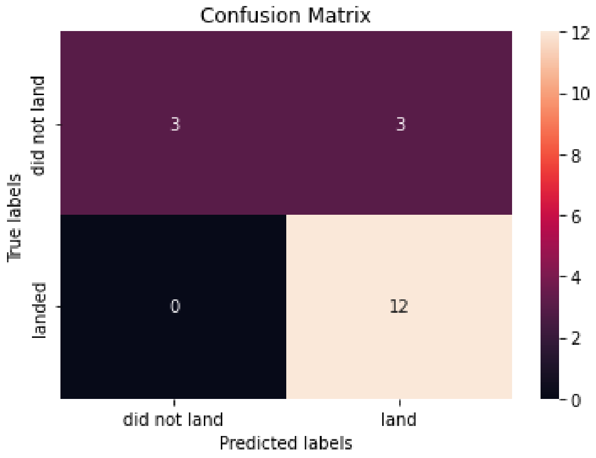
Calculate the accuracy on the test data using the method `score` :

In [54]:
```
test_accuracy = logreg_cv.score(X_test, Y_test)
print("Test data Accuracy: ", test_accuracy)
```

```
Test data Accuracy:  0.8333333333333334
```

Lets look at the confusion matrix:

In [56]:
```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```




Performed method score to analyse and build test and train models using logreg to the dataset

**Model accuracy is 83%**

# SVM model performance

**TASK 7**

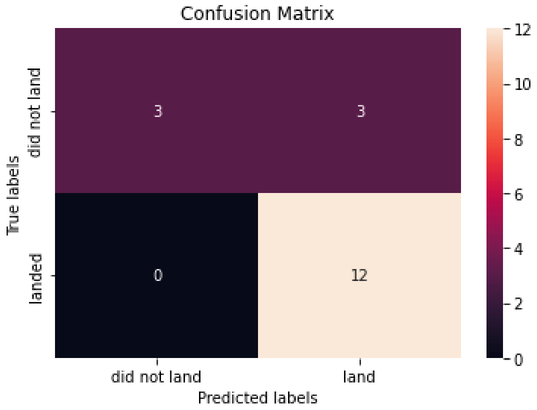
Calculate the accuracy on the test data using the method `score` :

In [78]:
```
test_accuracy = svm_cv.score(X_test, Y_test)
print("Test data Accuracy: ", test_accuracy)
```

```
Test data Accuracy:  0.8333333333333334
```

We can plot the confusion matrix

In [72]:
```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```




## Accuracy of the SVM model is 83%

# KNN model prediction

### TASK 11

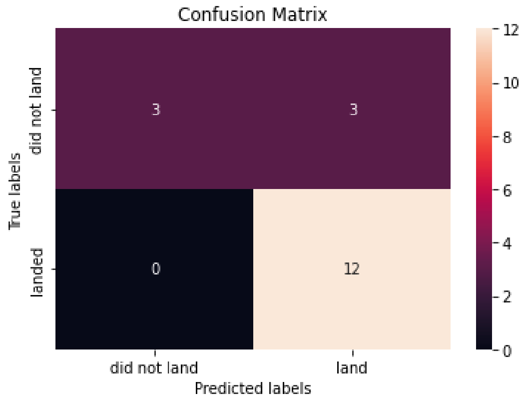
Calculate the accuracy of knn_cv on the test data using the method `score`:

In [96]:
```
knn_cv_accuracy = knn_cv.score(X_test, Y_test)
print(" Model accuracy :", knn_cv_accuracy)
```

```
 Model accuracy : 0.8333333333333334
```

We can plot the confusion matrix

In [98]:
```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```




**The accuracy of the KNN model is 83%**

# Methods performed the best

Find the method performs best:

```
print("Logistic Regression accuracy :",logreg_cv.best_score_)
print("Support Vector Machine accuracy :",svm_cv.best_score_)
print("Decigion Tree accuracy :",tree_cv.best_score_)
print("K-nearest neighboors accuracy :",knn_cv.best_score_)
```

```
Logistic Regression accuracy : 0.8464285714285713
Support Vector Machine accuracy : 0.8482142857142856
Decigion Tree accuracy : 0.9035714285714287
K-nearest neighboors accuracy : 0.8482142857142858
```

The best model which can perfor successful landing of the rocket booster is Decigion Tree with the accuracy 90% of prediction.

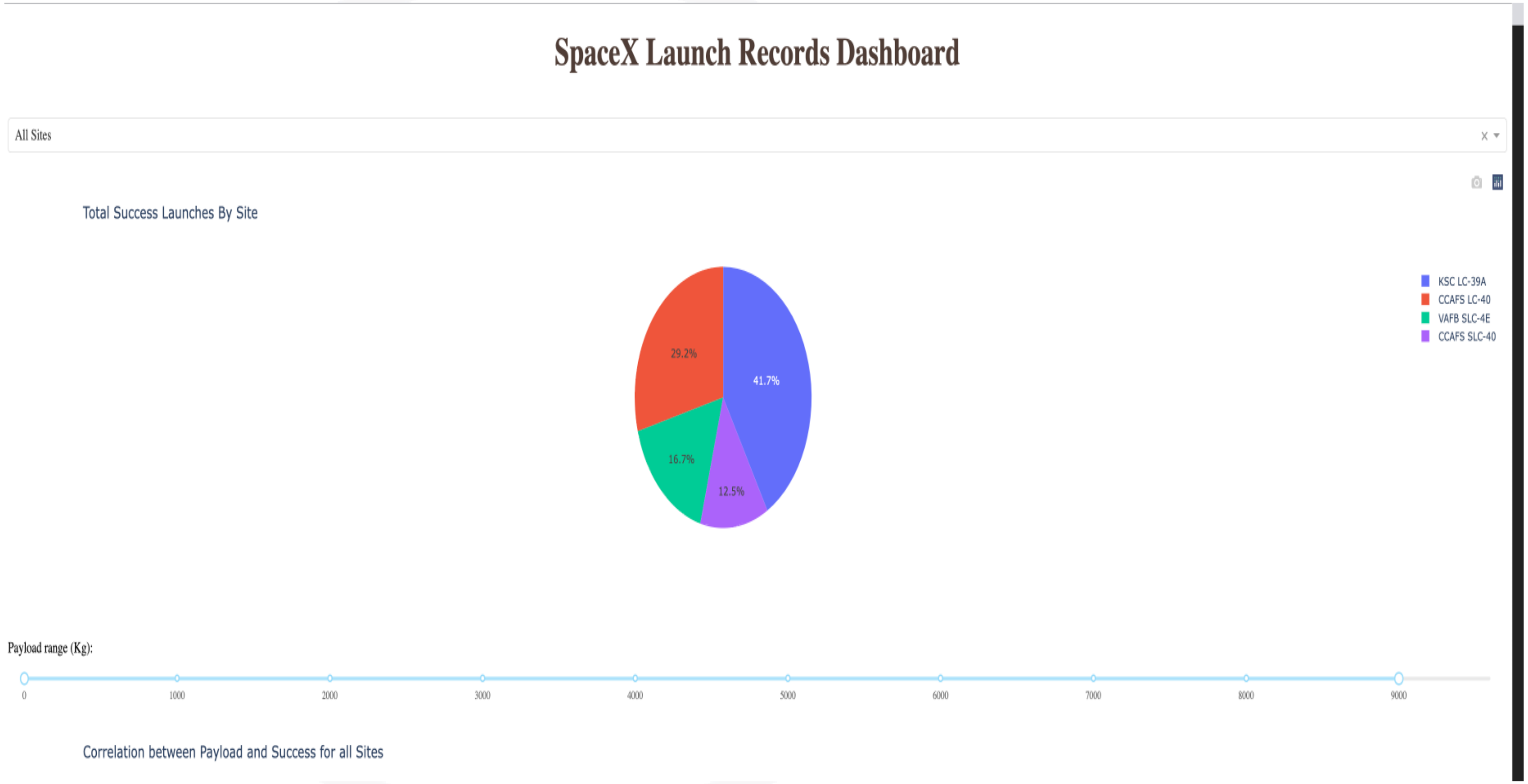
# DASHBOARD app with Plotly Dash

- Dashboard has dropdown, pie chart, rangeslider and scatter plot components
  - Dropdown allows a user to choose the launch site or all launch sites
  - Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component
  - Rangeslider allows a user to select a payload mass in a fixed range
  - Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass

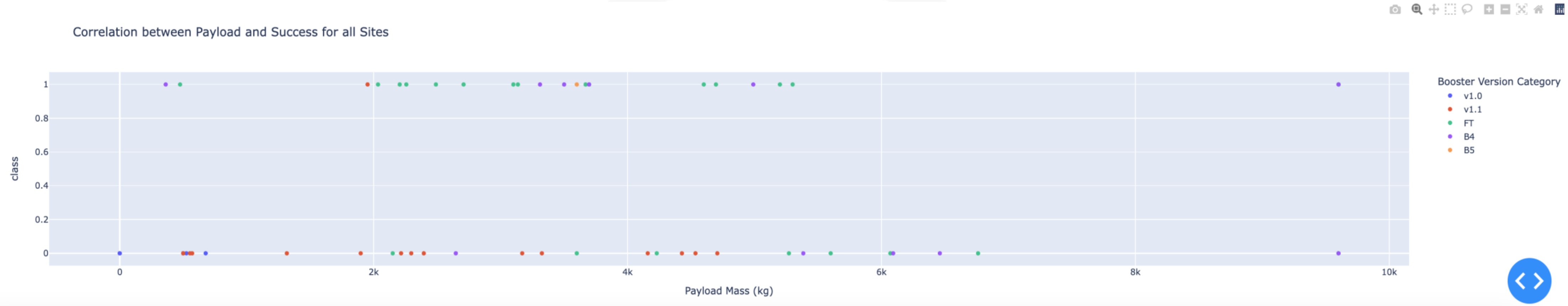
https://github.com/AlexYusyuk/IBM-Applied-Data-Science-Capstone/blob/main/Spacex%20Dash%20App.ipynb

# DASHBOARD SpaceX Launch Record

The Slide represents the most successfull launch. Which is KSC-LC.

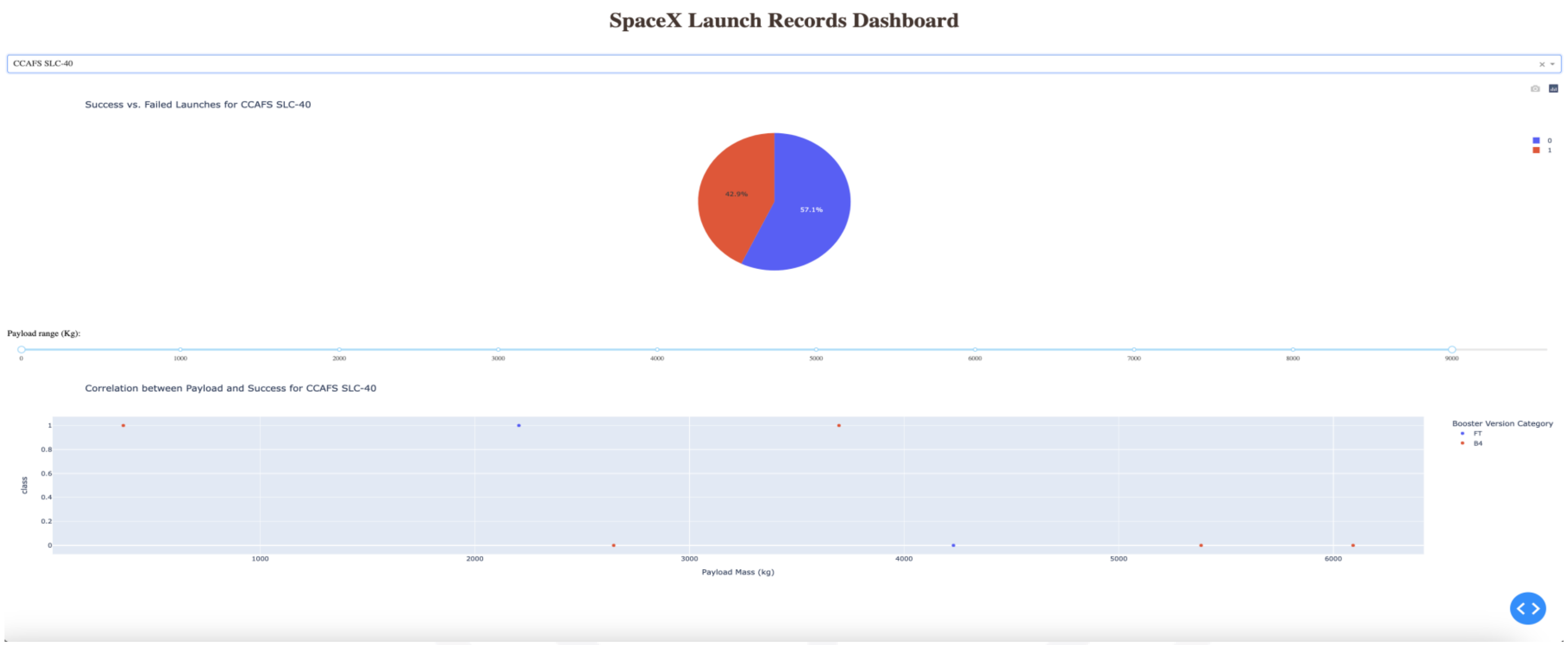


# Success of Laumches for all types



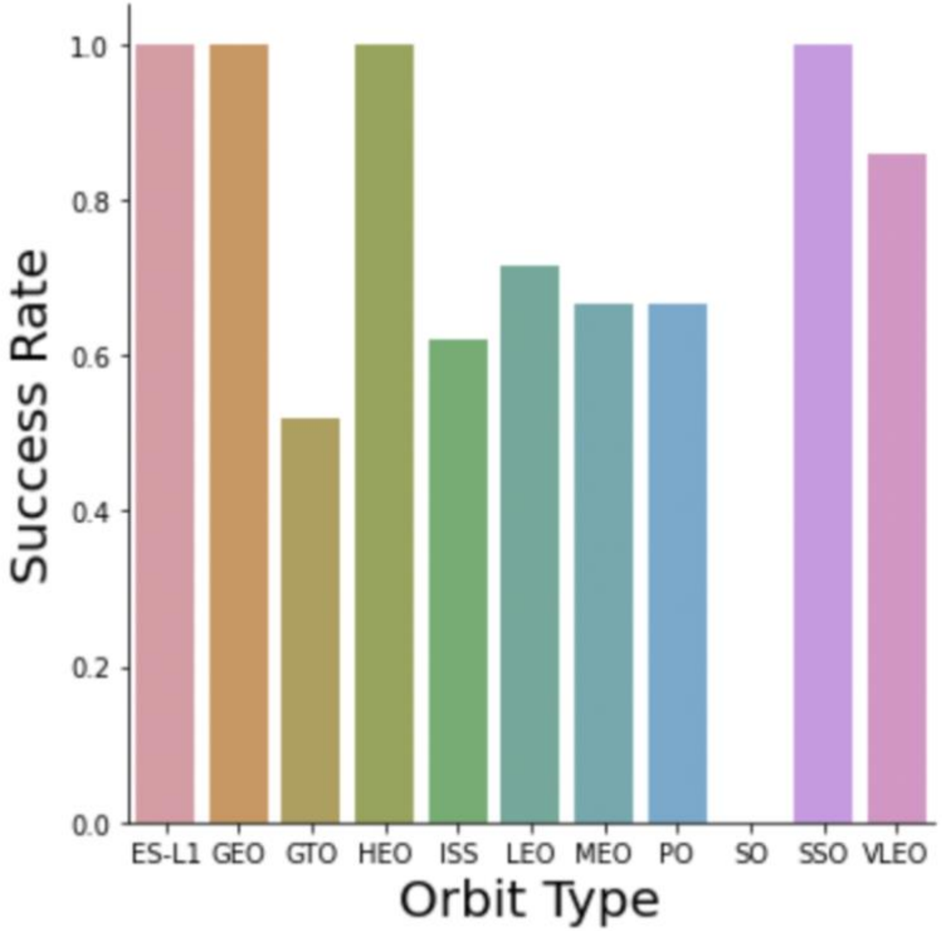

**The most Successfull payload for the launch is between 2K and 4K for all types of rockets**

# Worse successful Rocket type and Payload correliation

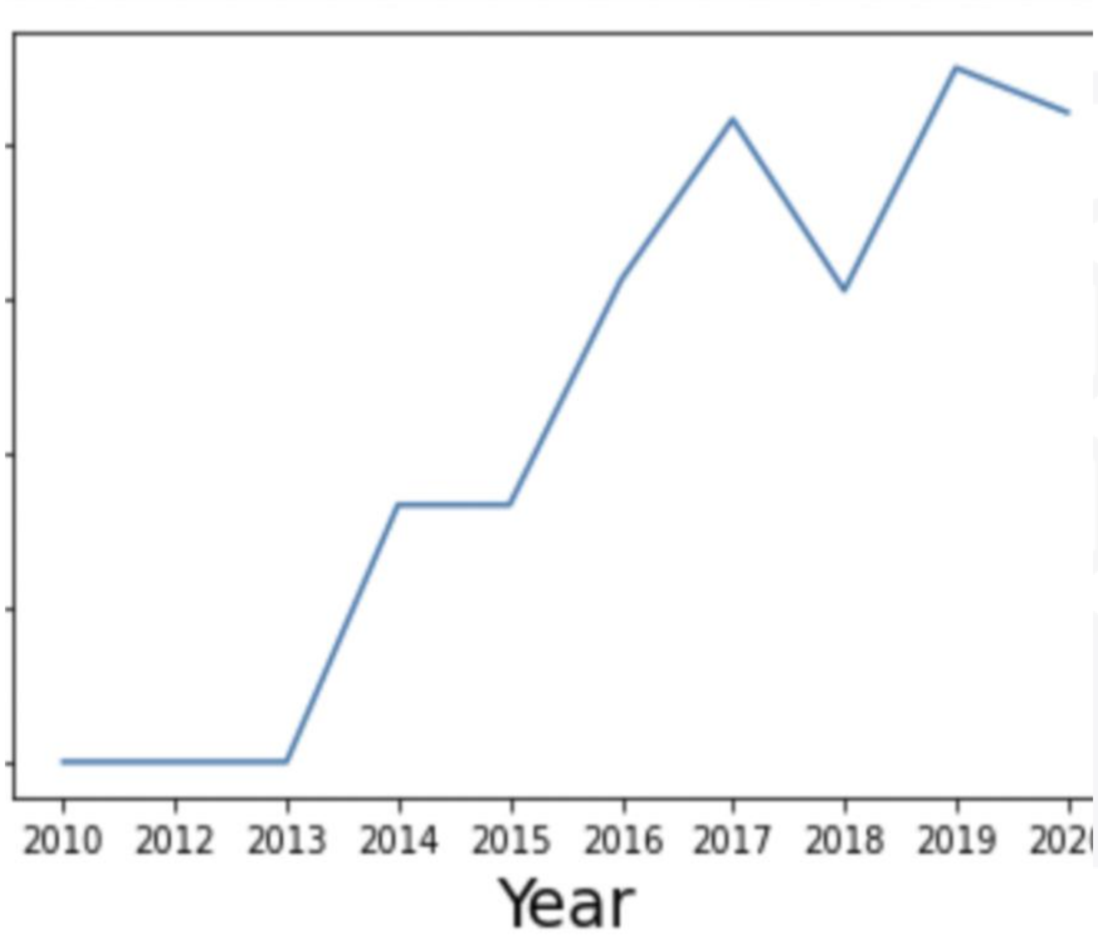


# Success rate vs. Orbit type

- Explanation:
  - Orbits with 100% success rate: ES-L1, GEO, HEO, SSO
  - Orbits with 0% success rate: SO
  - Orbits with success rate between 50% and 85%:
  - GTO, ISS, LEO, MEO, PO



# Launch success yearly trend



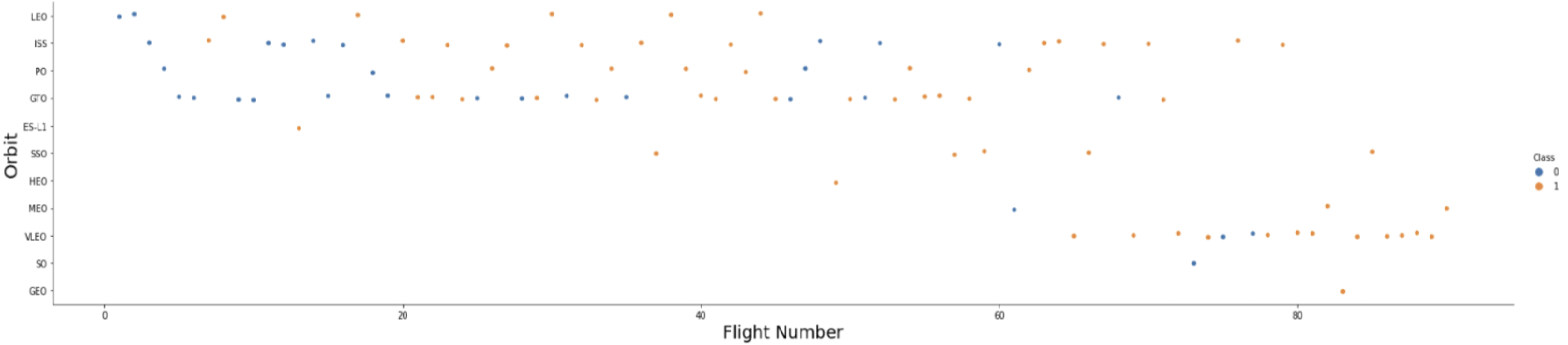

Explanation:

The success rate since 2013 kept till 2020

# OVERALL FINDINGS & IMPLICATIONS

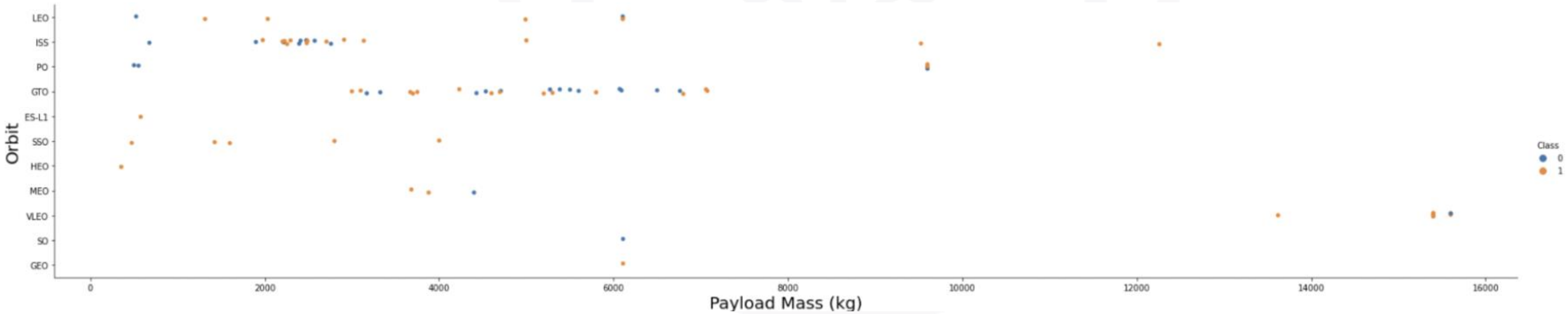
**Explanation:**

In the LEO orbit the Success appear relateds to the number of flights. On the other hand, there seems to be no relationsip between flight number when in GTO orbit



# Payload Mass vs. Orbit type

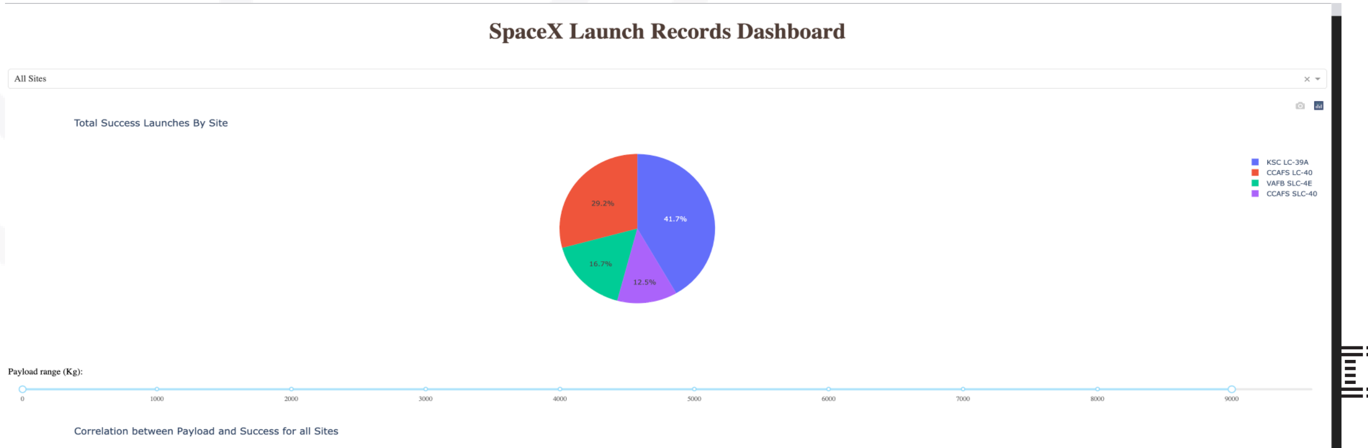
- Explanation:
  - Heavy payload have a negative influence on GTO orbits and positive on GTO and Polar LEO(ISS) orbits.



# CONCLUSION

- LR, SVM, KNN are top-performing models for forecasting outcomes in this data.
- Lighter payloads have a higher performance compared heavier ones.
- The likelihood of a SpaceX launch succeeding increases with the number of years of experience, suggesting a trend towards flawless launches over time.
- Launch Complex 39A at Kennedy Space Center has the highest number of successful launches compared to other launch sites
- GEO,HEO,SSO.ES L1 orbit types exhibit the highest rates of successful launches.

KSC LC-39A, has the most successful launches overall



# Thank you!