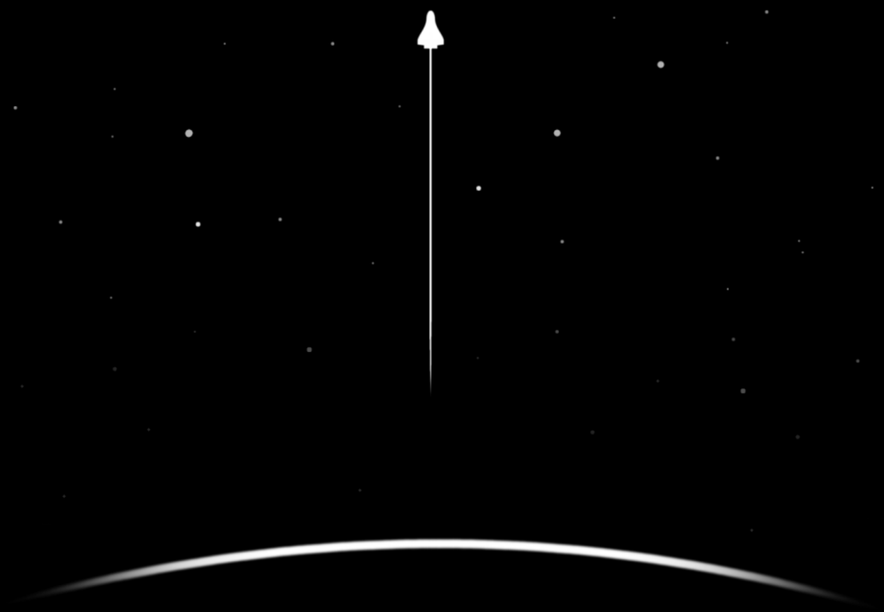
Winning Space Race with Data Science

SpaceX Falcon 9 Landing Analysis

Saurabh Kumar Singh
25-05-2022

OUTLINE

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



Executive Summary

- The process followed the following steps – Data Collection, Data Wrangling, Exploratory Data Analysis, Interactive Visual Analytics, Predictive Analysis.
- The data was collected form the public SpaceX API and SpaceX Wikipedia page. The labels were created and the successful landings were classified. After exploring data with various methods. Best parameters were generated using GridSearchCV. Accuracy scores were also measured.
- Decision Tree Classfier, Support Vector, Linear Regression and K Nearest neighbors were used as machine learning models. The accuracy rate was found to be 83.33 %

Introduction

- SpaceX launched Flacon 9 for 62 million $ which was much cheaper than any other providers. This led to savings by utilization of the successfully landed SpaceX parts for reuse.
- Space Y is a competitor and has tasked to train a machine learning model to predict a successful Stage 1 recovery.

Methodology

Data Collection Overview

Data collection combined API requests from SpaceX public API and data from the Wikipedia page. The flowcharts are presented in the next slide.

The Data columns for SpaceX API are as follows:

FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

The Data columns for SpaceX Wikipedia Page Webscraping are as follows:

Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

Data Collection – SpaceX API

GET response to the SpaceX REST API

Response converted to .json file

Converted to Pandas Dataframe

Lists defined for data

Call functions

Dataset constructed

Pandas DataFrame constructed from dictionary dataset

Filter out Falcon 9 launches
Flight Number Column reset
Missing values of PayloadMass replaced with mean value

Data Collection –Web Scraping- SpaceX Wiki

HTML page for Wikipedia was requested from Static HTML

Response function was assigned with an object

Beautiful Soup object from HTML response object was created to extract tables from the HTML page

All columns were collected

Using the column names as dictionaries , custom functions were used to parse all launch table.

The dictionary values were filled

The Pandas Dataframe was created using the dictionary values

Data Wrangling

Context

The training label was created Successful = 1 and failure = 2
'Mission Outcome' and 'Landing Location' are the outcome column

True Ocean – Mission outcome was successfully in specific region of ocean
False Ocean – Mission outcome was unsuccessful in specific region of ocean
True RTLS – Mission outcome was successfully landed to ground pad
False RTLS – Mission outcome unsuccessful
True ASDS – landed successful to drone ship
False ASDS – landed unsuccessful to done ship
None ASDS and NN – Failed to land

Data Wangling

Defining bad_outcome, landing_class =0 if Outcome in bad_outcomes otherwishe its 1
Creating class clumn to list values of landing_Class
Exported as .csv file

URL

https://github.com/srslysaurabh0710/DataScience-Final-Capstone-Project/blob/main/Week%201/Data%20wrangling%20.ipynb

EDA with SQL

Plots used

Used local Jupyter installation.

Queried using SQL Python

Information retrieved about launch site names, mission outcomes, various payload sizes of customers and booster versions, and landing outcomes

URL

https://github.com/srslysaurabh0710/DataScience-Final-Capstone-Project/blob/main/Week%202/EDA%20with%20SQL.ipynb

Interactive Map with Folium

Folium Map

Folium maps mark Launch sites, successful and unsuccessful landing and a proximity example to key locations : Railway, Highway, Coast and City

The relation between Successful landing and relative to locations.

URL

https://github.com/srslysaurabh0710/DataScience-Final-Capstone-Project/blob/main/Week%203/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Dashboard with Plotly Dash

Plotly Dash

The distribution of successful landing across all launch sites was mapped on a pie chart and successful launch site were selected

Pie chart is used to visualize launch site success rate.

Scatter plot shows the relation between launch sites, payload mass and booster version category.

URL

https://github.com/srslysaurabh0710/DataScience-Final-Capstone-Project/blob/main/Week%203/spacex_dash_app.py

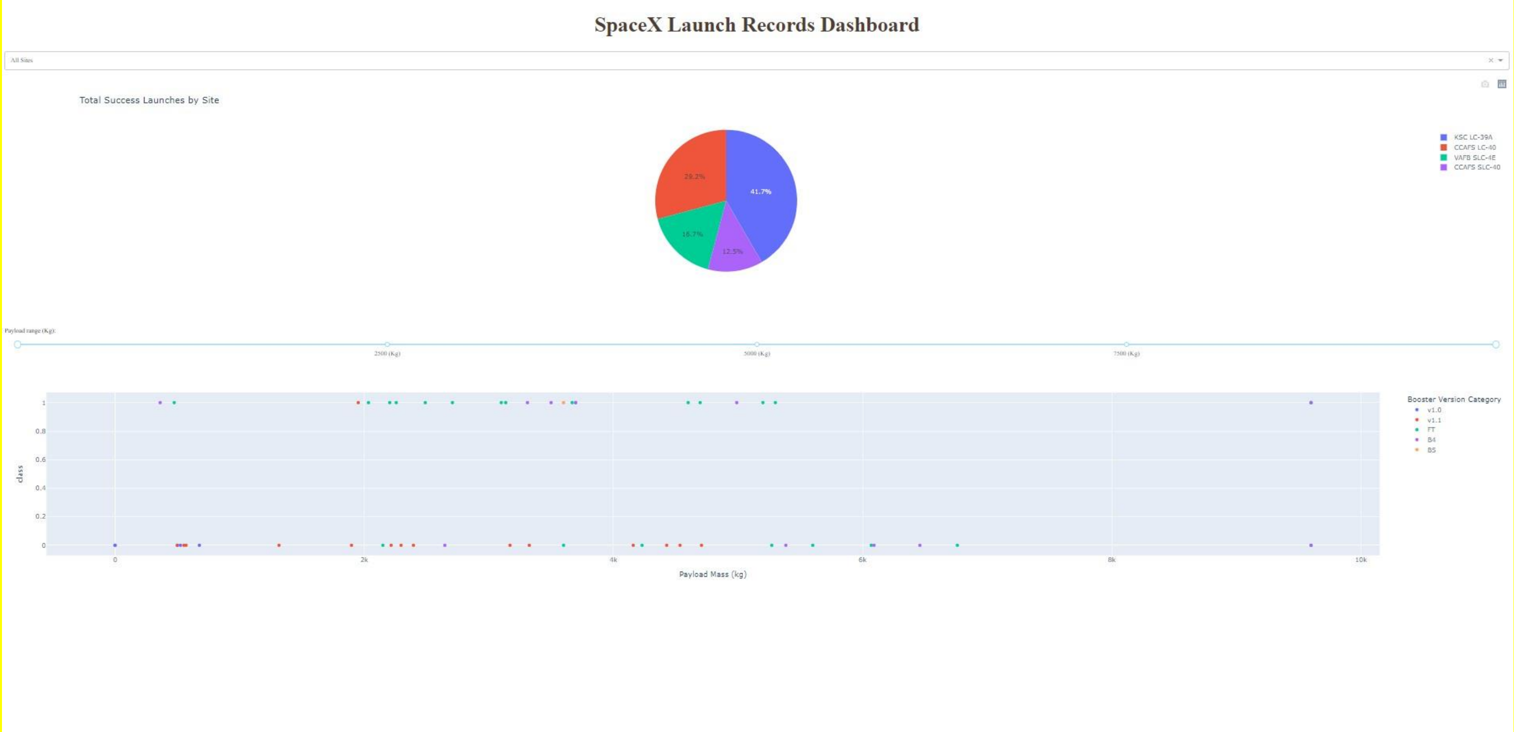
Predictive Analysis (classification)

- The label column 'Class' from datatset was split.
- They were fit and transform using a StandardScaler.
- The Train_test_split data was generated.
- Then GridSearchCV (cv=10) were used to find optimal paramters.
- Then Grid SearchCV on LogReg, SVM, Decision Tree and KNN Models were used to score models on split test set.
- Confusion Matrix for all models were created and Barplot was made to compare scores of models

URL

https://github.com/srslysaurabh0710/DataScience-Final-Capstone-Project/blob/main/Week%204/Machine%20Learning%20Prediction.ipynb

Results



Plotly Dash Visualization

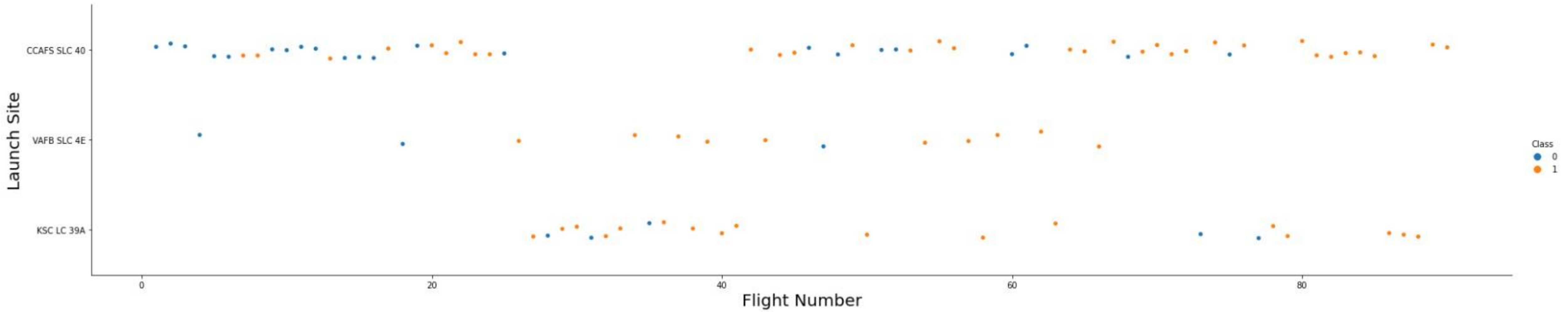
Flight Number vs Launch Site

TASK 1: Visualize the relationship between Flight Number and Launch Site

Use the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter `x` parameter to `FlightNumber`,set the `y` to `Launch Site` and set the parameter `hue` to `'class'`

```python
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

Python



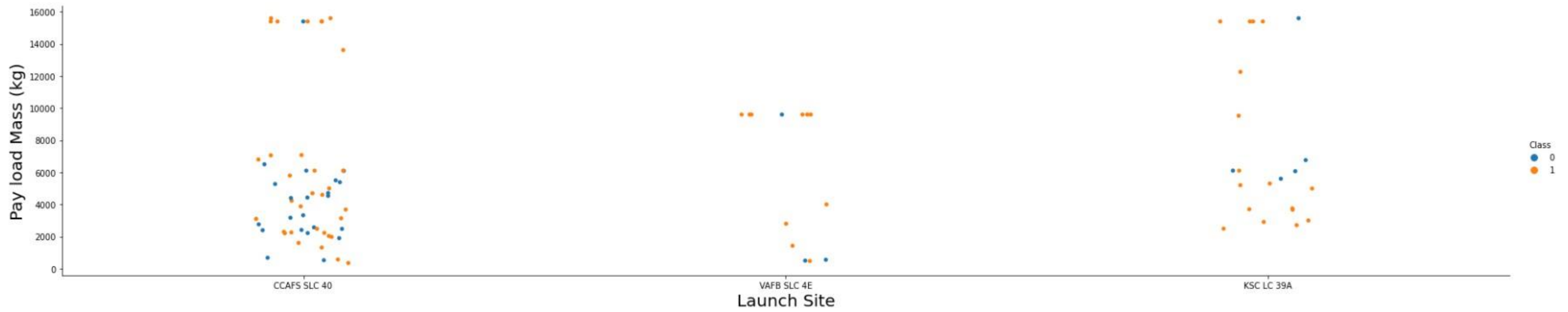
Payload vs Launch Site

TASK 2: Visualize the relationship between Payload and Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

```python
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="PayloadMass", x="LaunchSite", hue="Class", data=df, aspect = 5)
plt.xlabel("Launch Site",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```

Python



Success rate vs Orbit Type

TASK 3: Visualize the relationship between success rate of each orbit type

Next, we want to visually check if there are any relationship between success rate and orbit type.

Let's create a `bar chart` for the sucess rate of each orbit

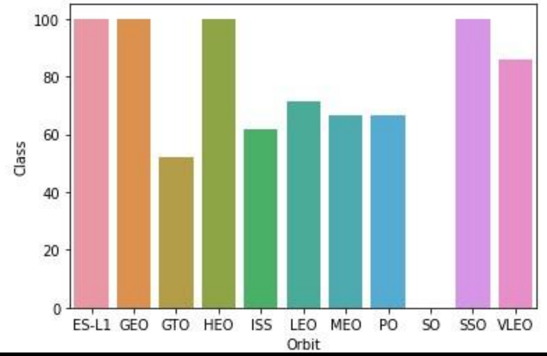
```python
# HINT use groupby method on Orbit column and get the mean of Class column
temp = df.groupby(["Orbit"]).mean().reset_index()
temp2 = temp[["Orbit", "Class"]]
temp2["Class"] = temp2["Class"]*100
sns.barplot(x = "Orbit", y = "Class", data = temp2)
```

Python

```
<ipython-input-48-7e88f588b437>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  temp2["Class"] = temp2["Class"]*100

<AxesSubplot:xlabel='Orbit', ylabel='Class'>
```



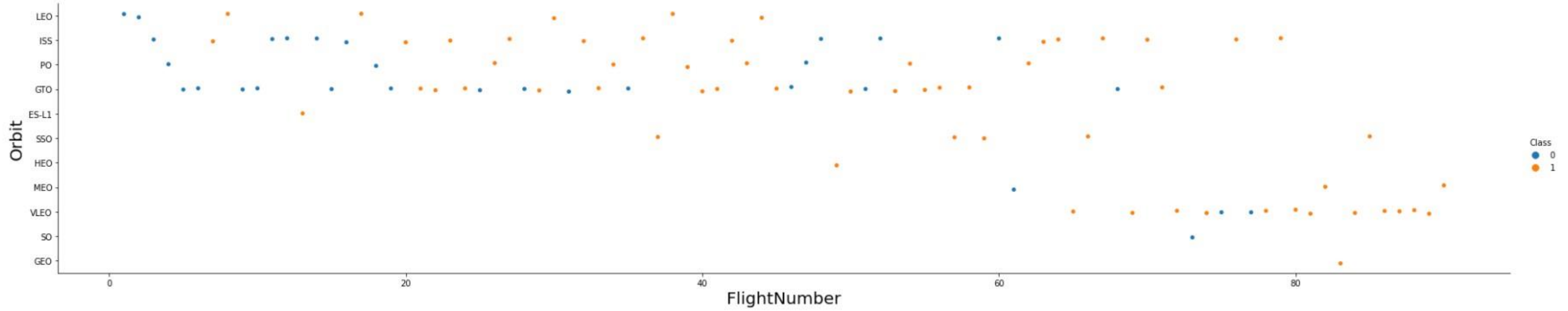
Flight Number vs Orbit Type

TASK 4: Visualize the relationship between FlightNumber and Orbit type

For each orbit, we want to see if there is any relationship between FlightNumber and Orbit type.

```python
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

Python



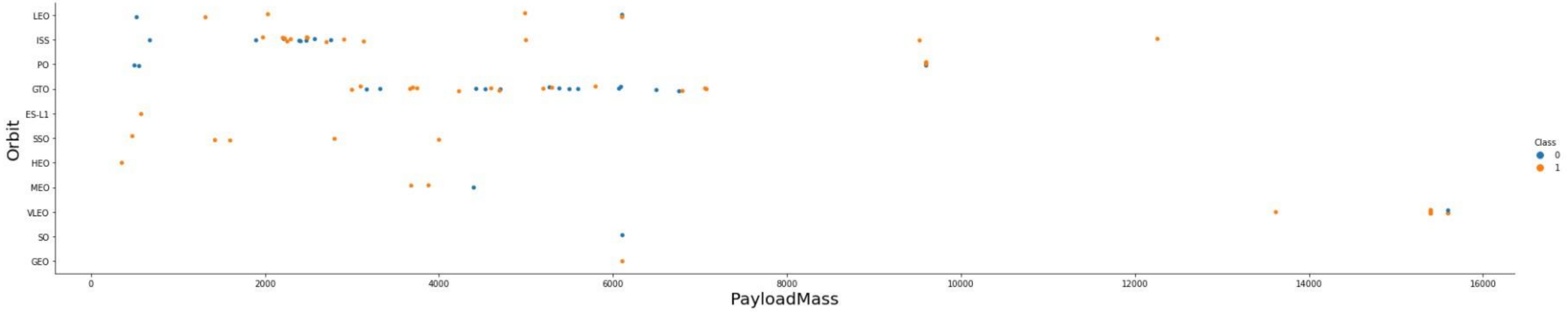
Payload vs Orbit Type

TASK 5: Visualize the relationship between Payload and Orbit type

Similarly, we can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and Orbit type

```python
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

Python

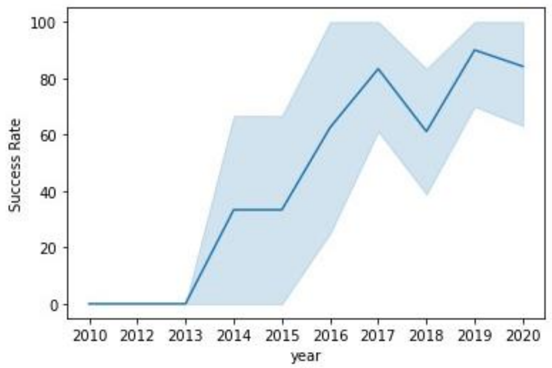


Launch Success Yearly Trend

```python
# A function to Extract years from the date
def Extract_year(year):
    for i in df["Date"]:
        year.append(i.split("-")[0])
    return year
```

```python
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
year = []
df["year"] = Extract_year(year)
df["Success Rate"] = df["Class"] * 100
sns.lineplot(data = df, x = "year", y = "Success Rate")
```

```
<AxesSubplot:xlabel='year', ylabel='Success Rate'>
```



3

All Launch Sites

```
%sql select DISTINCT LAUNCH_SITE from SPACEXDATASET
```

[redacted]

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

Launch Site Names beginning with CCA

Task 2

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

```
%sql select * from SPACEXDATASET where launch_site like 'CCA%' limit 5
```

[redacted]

TE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__ou
0-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	(para
0-08	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	(para
2-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No a
2-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No a
3-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No a

3

Total Payload Mass from NASA

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

```
%sql select sum(payload_mass__kg_) as sum from SPACEXDATASET where customer like 'NASA (CRS)'
```

[redacted]

SUM
45596

Average Payload Mass by Falcon 9

Task 4

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

```
%sql select avg(payload_mass__kg_) as Average from SPACEXDATASET where booster_version like 'F9 v1.1%'
```

[redacted]

average
2534

First Successful Ground Pad Landing Date

```
%sql select min(date) as Date from SPACEXDATASET where mission_outcome like 'Success'
```

DATE
2010-06-04

Successful Drone Ship Landing (Payload >4000 <6000)

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql select booster_version from SPACEXDATASET where (mission_outcome like 'Success')
AND (payload_mass__kg_ BETWEEN 4000 AND 6000) AND (landing__outcome like 'Success (drone ship)')
```

Python

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

Total Number of Each Mission Outcome

Task 7

List the total number of successful and failure mission outcomes

+ Code + Markdown

```
%sql SELECT mission_outcome, count(*) as Count FROM SPACEXDATASET GROUP by mission_outcome ORDER BY missio
```

Python

mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters that carried maximum payload

Task 8

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

```
maxm = %sql select max(payload_mass__kg_) from SPACEXDATASET
maxv = maxm[0][0]
%sql select booster_version from SPACEXDATASET where
payload_mass__kg_=(select max(payload_mass__kg_) from SPACEXDATASET)
```

Python

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

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015

```
%sql select MONTHNAME(DATE) as Month, landing__outcome, booster_version, launch_site
from SPACEXDATASET where DATE like '2015%' AND landing__outcome like 'Failure (drone ship)'
```

Python

MONTH	landing__outcome	booster_version	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

Task 10

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

```
%sql select landing__outcome, count(*) as count from SPACEXDATASET
where Date >= '2010-06-04' AND Date <= '2017-03-20'
GROUP by landing__outcome ORDER BY count Desc
```

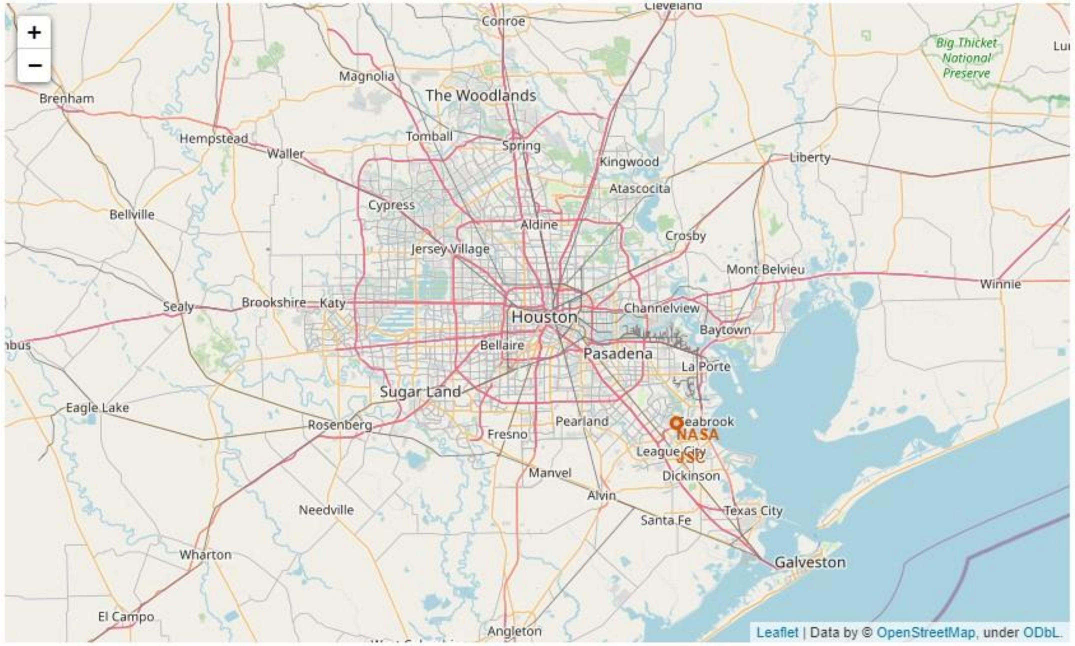
Python

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

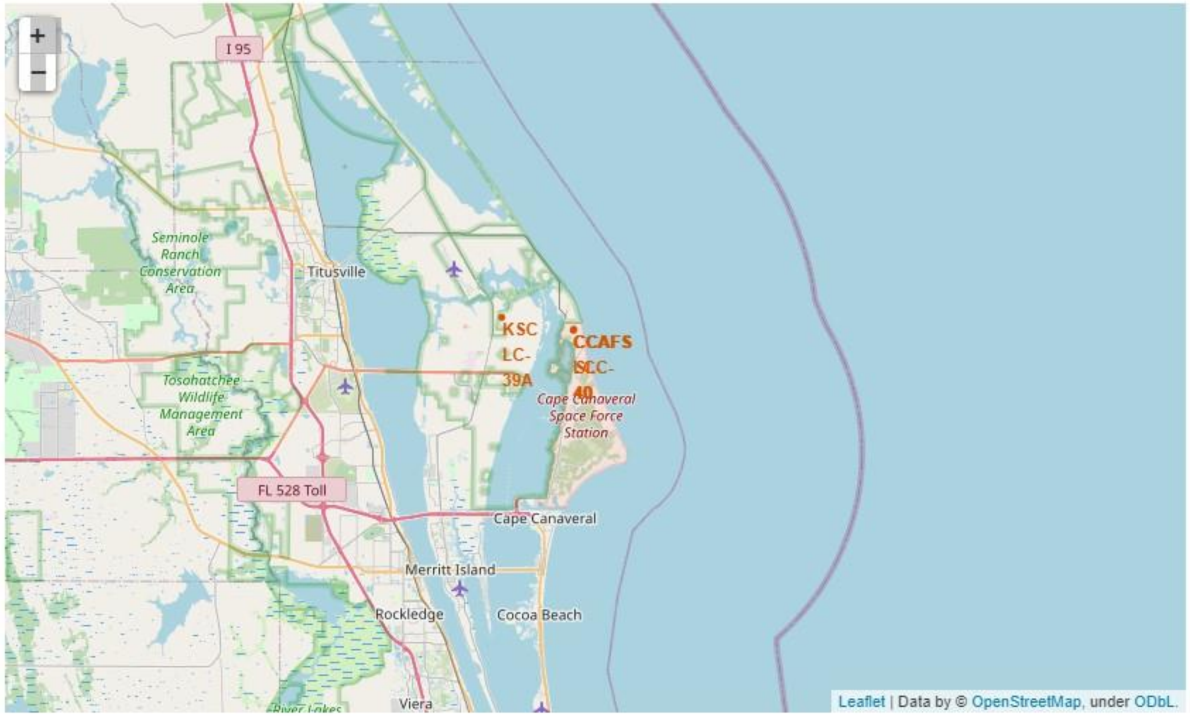
Folium Maps – Marking all Launch Sites

```python
# Create a blue circle at NASA Johnson Space Center's coordinate with a popup label showing its name
circle = folium.Circle(nasa_coordinate, radius=1000, color='#d35400', fill=True).add_child(folium.Popup('N
# Create a blue circle at NASA Johnson Space Center's coordinate with a icon showing its name
marker = folium.map.Marker(
    nasa_coordinate,
    # Create an icon as a text label
    icon=DivIcon(
        icon_size=(20,20),
        icon_anchor=(0,0),
        html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % 'NASA JSC',
        )
    )
site_map.add_child(circle)
site_map.add_child(marker)
```

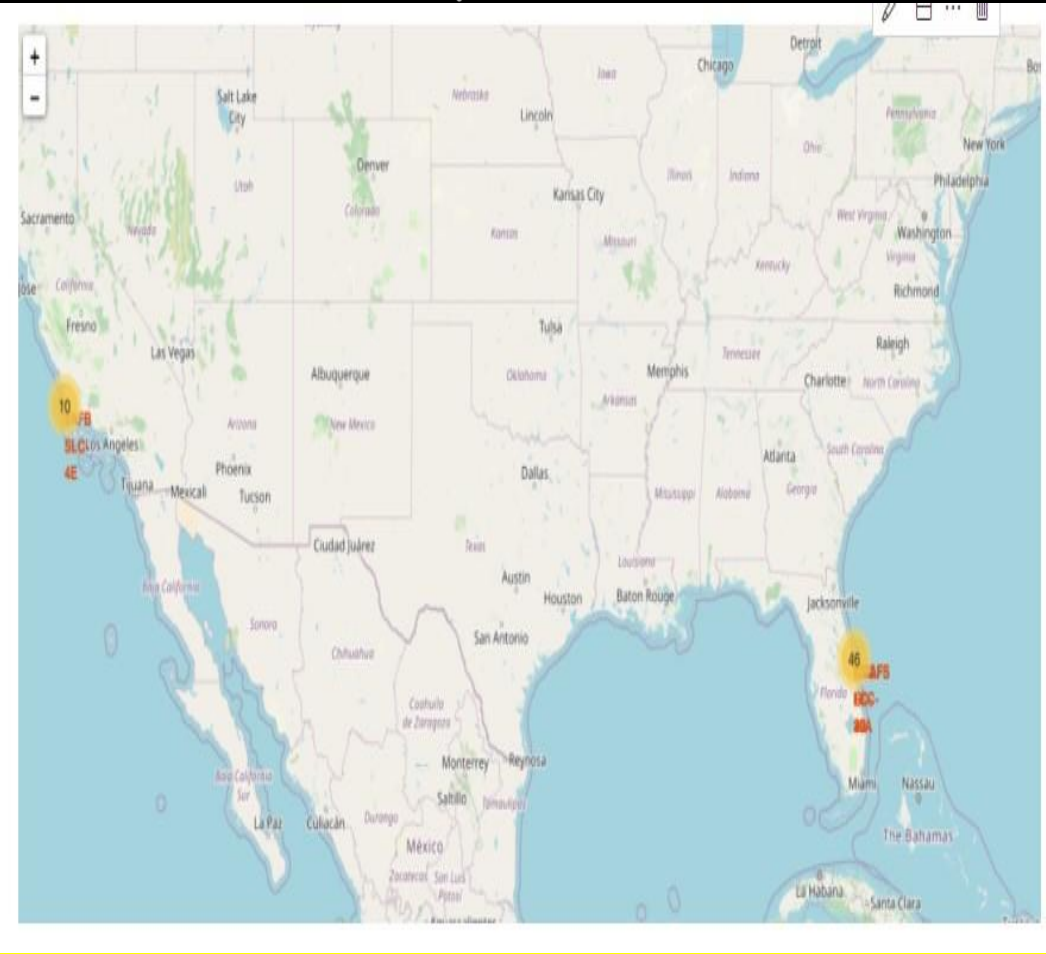
Python



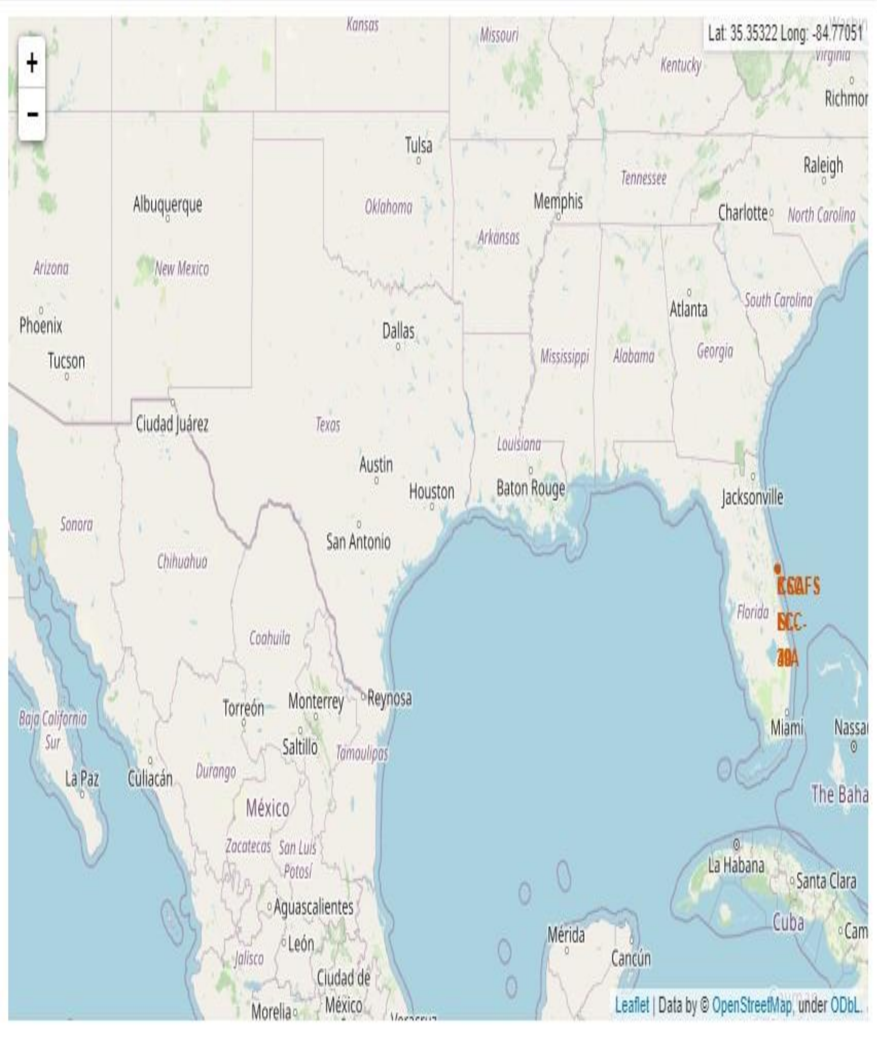
Python



Folium Maps – Marking Success/failed

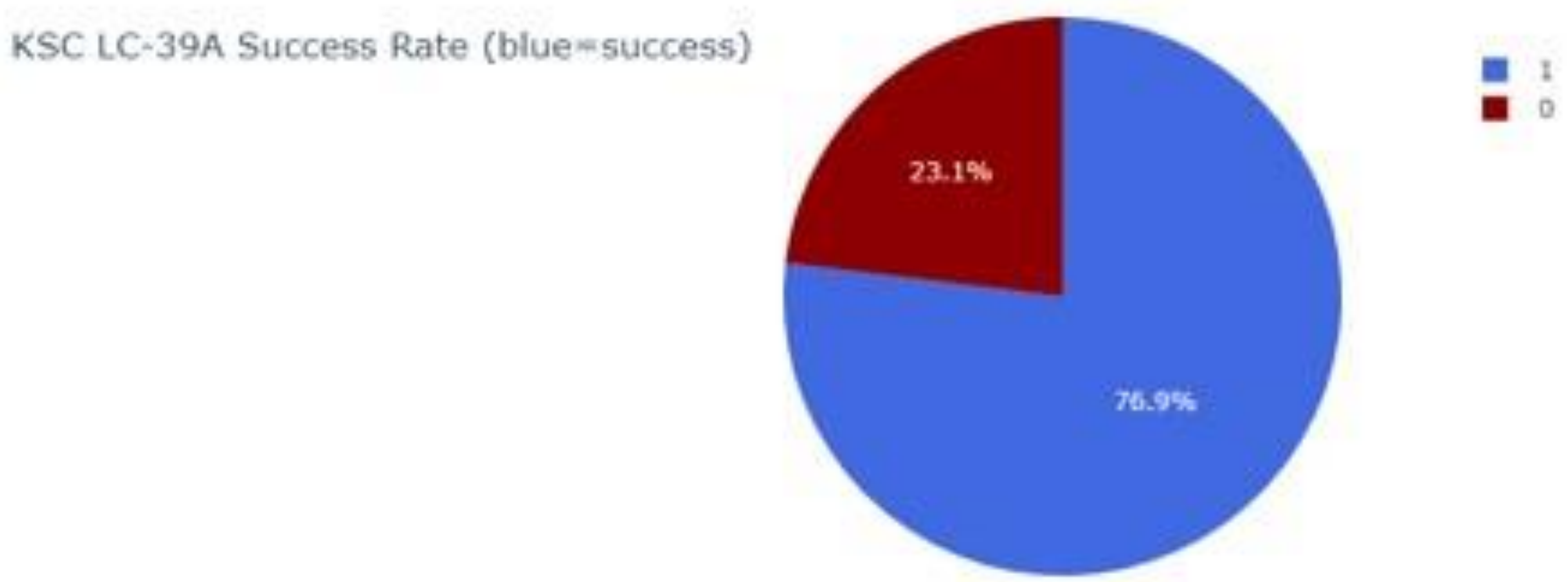


Folium Maps – Distances between lunch site



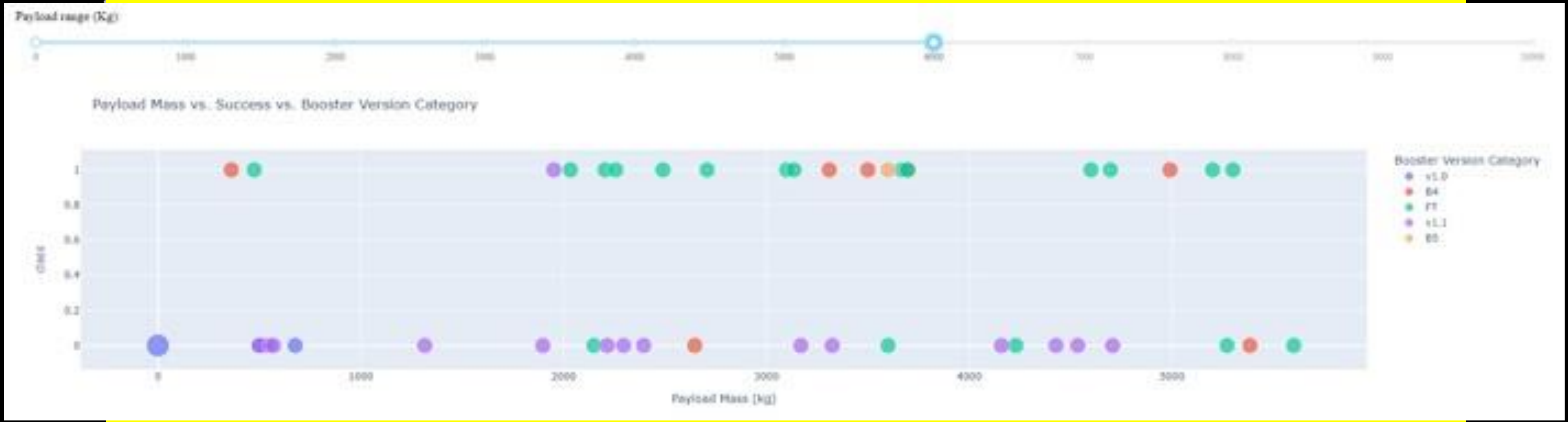
SpaceX Launch Dashboard

Highest Success Rate Launch Site

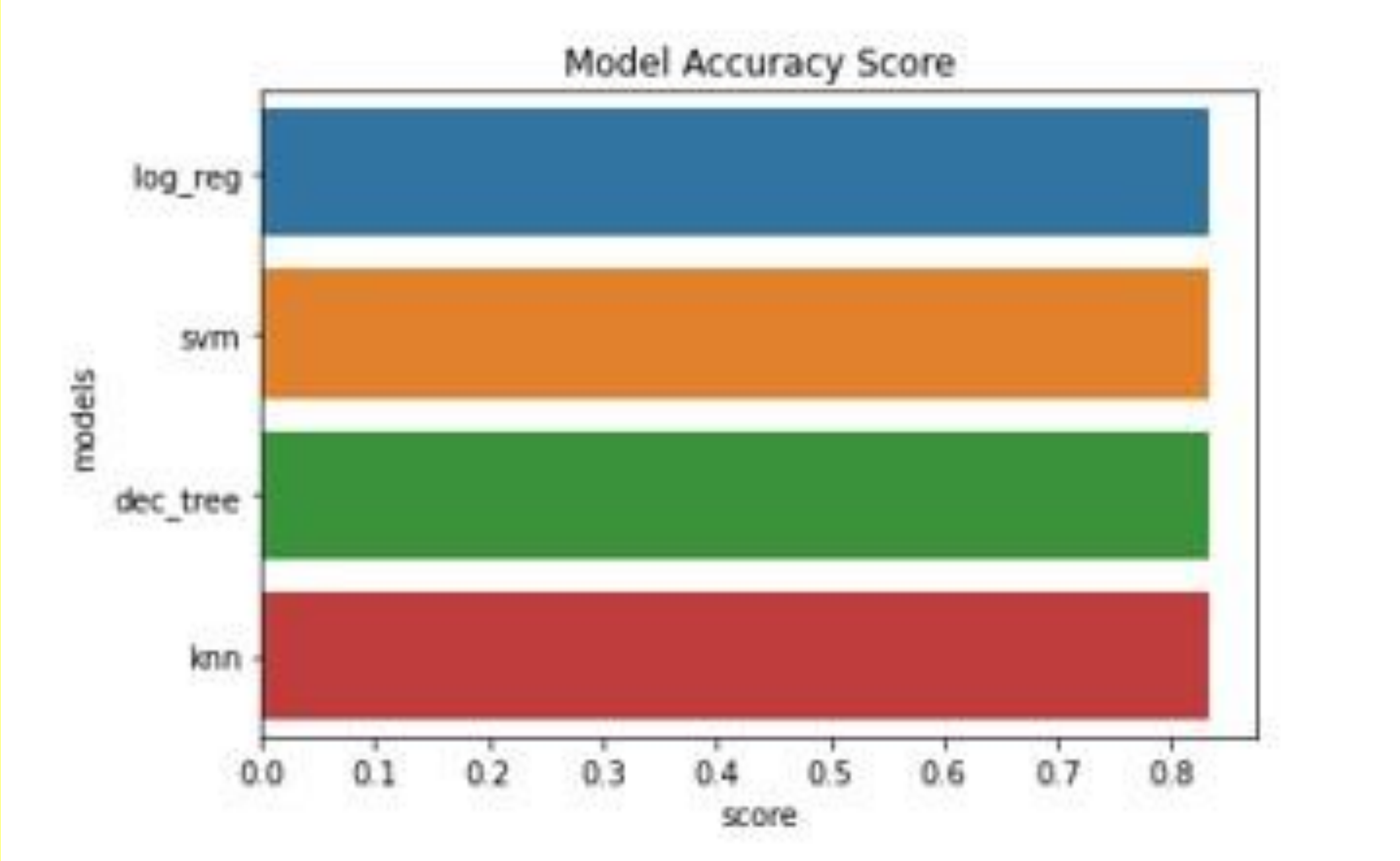


KSC LC-39A has the highest success rate with 10 successful landings and 3 failed landings.

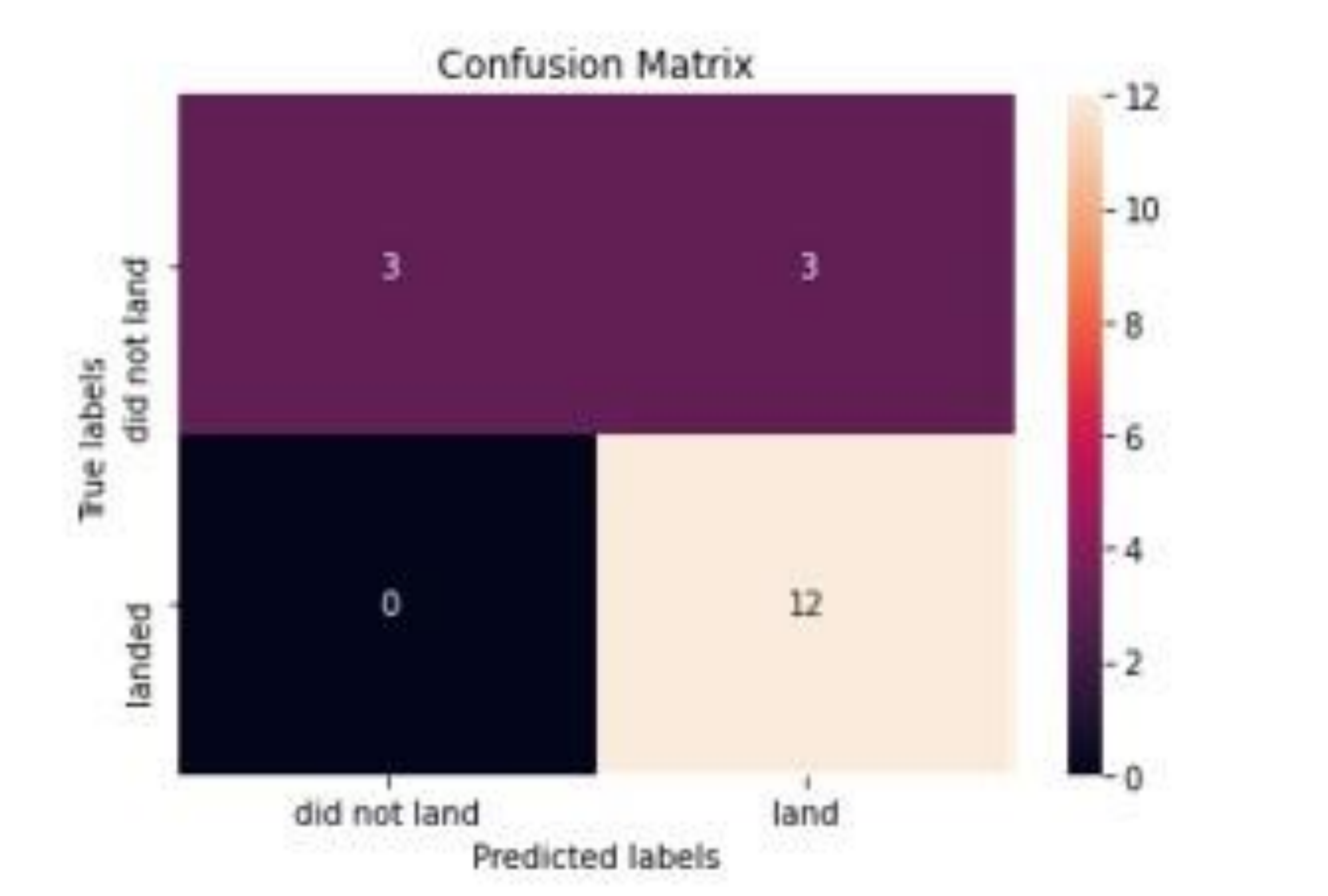
Payload vs. Launch Outcome scatter plot



Predictive Analysis – Classification Accuracy



Predictive Analysis – Classification Accuracy



Conclusion

SpaceX API and Wiki pages were used to generate data and machine learning model was created with 83.33% accuracy.

Allon Musk can utilize the model with a relatively medium-high accuracy of 80% < x < 85%.

Stage 1 success can be determined utilizing the model and more data can be collected for better modelling and improve accuracy above 90%

Thank You