
IBM Developer
SKILLS NETWORK
Winning Space Race
with Data Science
IBM Data Science Capstone project SpaceX
By
Vaipoj Mesombat

# Outline

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

# EXECUTIVE SUMMARY

## Using Methodology

- Data Collection
- Data Warping
- EDA Data Visualization
- SQL Enquiry
- Create Visualize Map using Folium
- Predictive Analysis (Clustering, KNN and etc)

## Result

- Exploration data
- Example of each analysis step
- Enquiry to Explore EDA  data set
- Interactive Folium Map
- Real time Dash Board
- Predictive result

# Project Background

In case we are predicting 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. It is very challenging for data scientist that use the historical data and analyse so it will make more benefit to lunch SpaceX

## Mission & Key Challenge

- Know How & Knowledge to adjust from lesson learn in case of improve the future success
- Related key to success
- Find out the variable that depend upon of outcome

Section 1

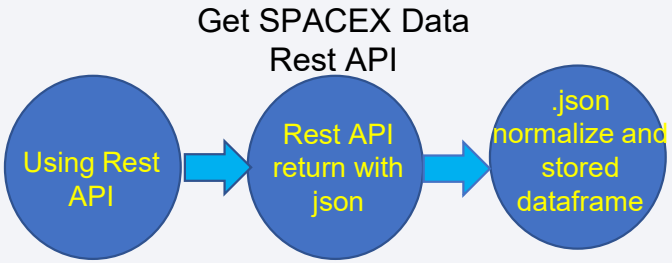
# Methodology

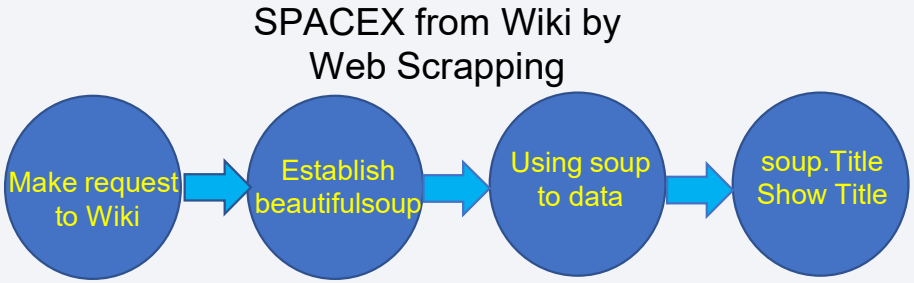
# Key Methodology

- Data Collection & Warping via Rest API from defined URL
- Transform the data that ready to use for the model
- Find the relationship between interesting variable using data visualization method
- Focus on the pairs of variable that higher depend by Folium and Plotly Dashboard
- Try to predict outcome with unsupervised model such as K-Nearest , SVM and Decision Tree

# Data Collection Methodology

- For this project we will use Rest API to get data from defined URL
- This API get the data of rocket such as lunch time. Load and etc
- This data is used for predict the rocket attempt to land or not
- End point is https://api.spacexdata.com/v4/
- (Core, Rocket, Lunchpad and PayLoadmass)
- We also request information from Wikipedia via web wrapping by Beautifulsoup



# Get data with Rest API

1. Get response for API

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

2. Convert response to json file

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

3. Preparing Data

```
# Call getBoosterVersion
getBoosterVersion(data)
 # Call getLaunchSite
 getLaunchSite(data)
  # Call getCoreData
  getCoreData(data)
# Call getCoreData
getCoreData(data)
```

GitHub

4. Prepare structure for 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}
```

5. Apply to Data Frame

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

# Get Data with Web Scrapping

```
# use requests.get() method with the provided static_url
# assign the response to a object
page = requests.get(static_url)
page.status_code
```

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

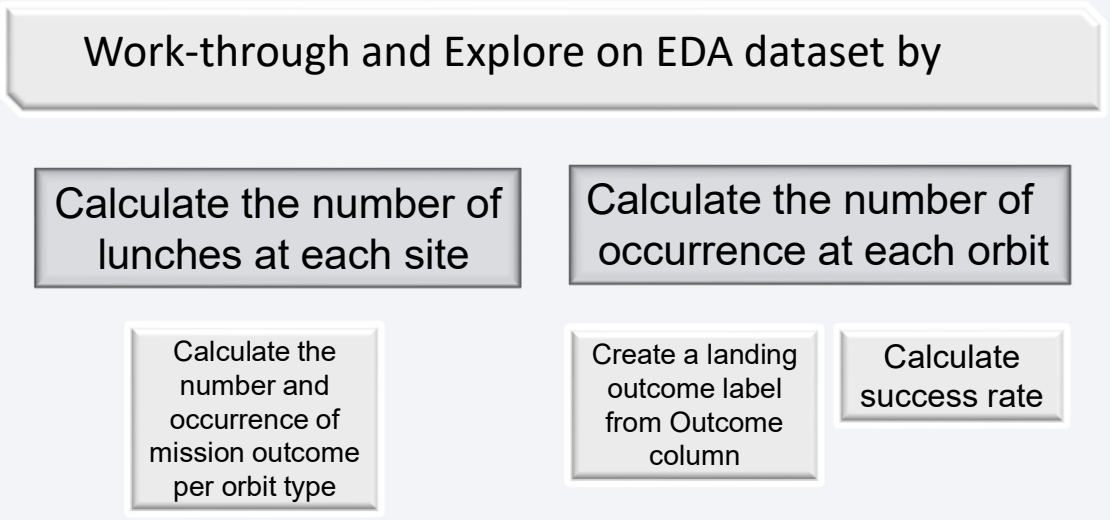
```
# Use soup.title attribute
soup.title
```

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

GitHub

# Data wrangling (cont.)

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad.True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.




GitHub

# Perform exploratory data analysis (EDA) using Visualization and SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.
- List the date when the first successful landing outcome in ground pad was achieved.
- 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. Use a subquery
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Rank 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

# Perform interactive visual analytics using Folium and Plotly Dash

- **Folium**

The launch success rate may depend on many factors such as payload mass, orbit type, and so on. It may also depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analysing the existing launch site locations. We will performing more interactive visual analytics using "Folium"


GitHub

- **Plotly Dash**

you will be building a Plotly Dash application for users to perform interactive visual analytics on SpaceX launch data in real time. This dashboard application contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart.


URL link to Web site

# Perform predictive analysis using Classification models

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. In this lab, you will create a **machine learning** pipeline to predict.

**Machine Learning Prediction Model**

- Logistic regression
- Support Vector Machine
- Decision Tree

**Key step for Machine Learning Model**

- Build Model
- Evaluate model of each model
- Tunning Model
- Finding the best performance

GitHub

# Results (Data Analytics by SQL)

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

“select DISTINCT Launch_Site from spacextbl “

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

GitHub

# Results (Data Analytics by SQL)

***List the total number of successful and failure mission outcomes***

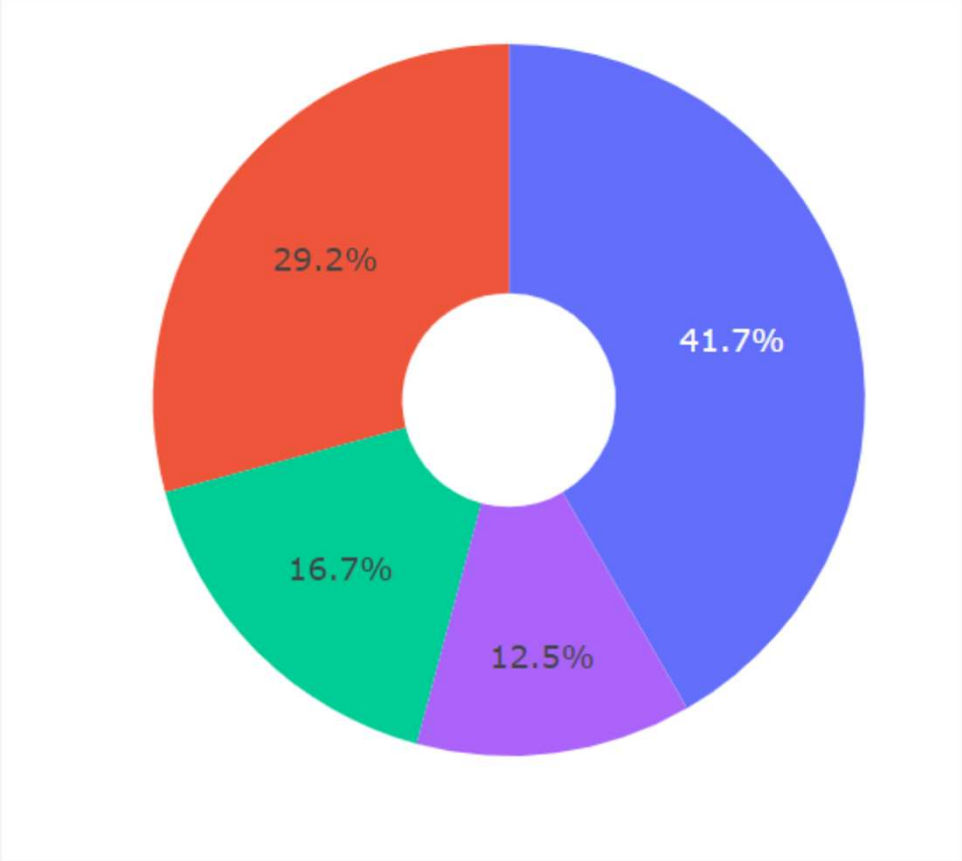
```
select(select count(Mission_Outcome) from spacextbl where
Mission_Outcome like '%Success%') as Successful_Mission_Outcomes,
 (select count(Mission_Outcome) from spacextbl where
Mission_Outcome like '%Failure%') as Failure_Mission_Outcomes
```

| Successful_Mission_Outcomes | Failure_Mission_Outcomes |
|---|---|
| 100 | 1 |

GitHub

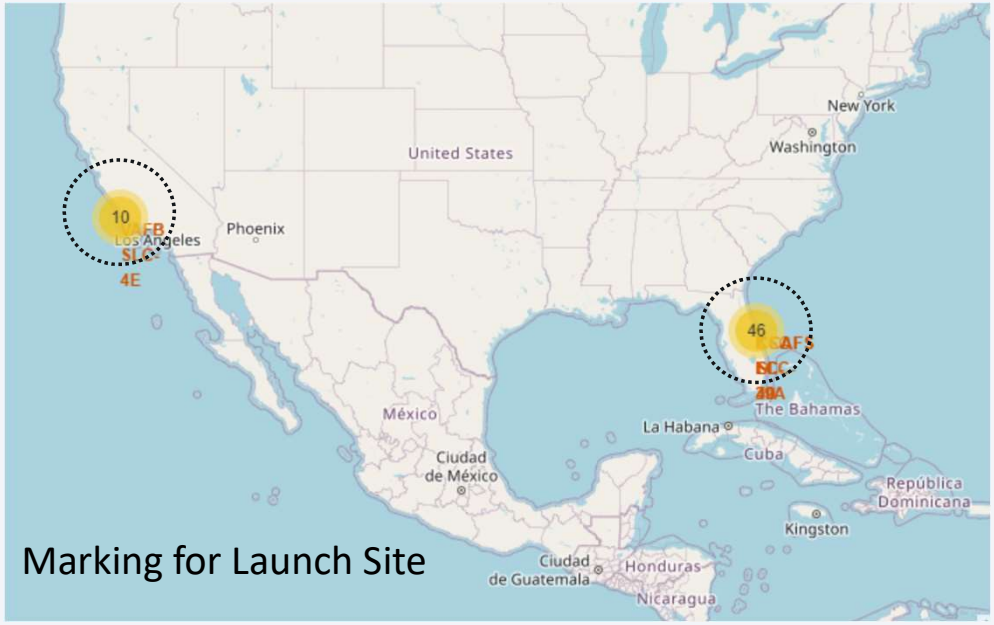
# Results (Interactive analytic demo in screenshots)

**Total Success Launch By all sites**

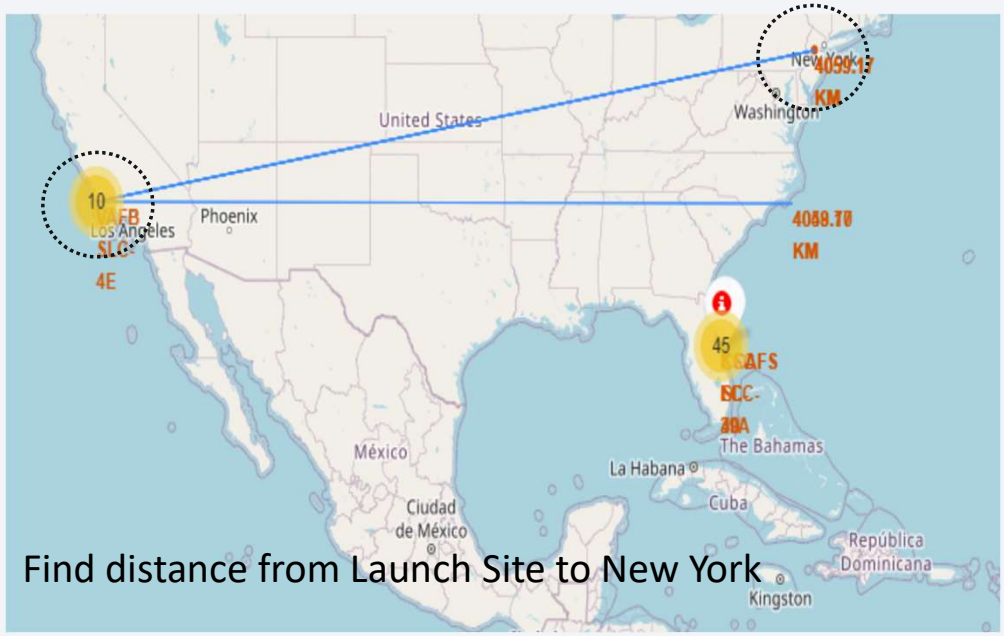



GitHub

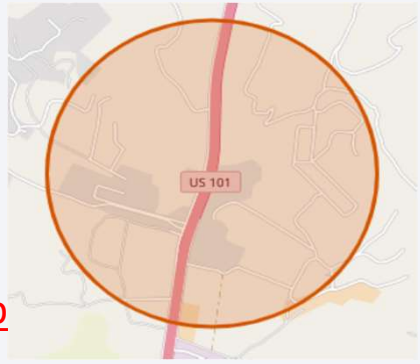
# Results (Interactive analytic demo in screenshots)



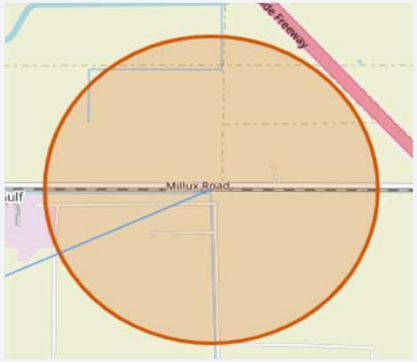

Marking for Launch Site




Find distance from Launch Site to New York




Highway from Launch Site

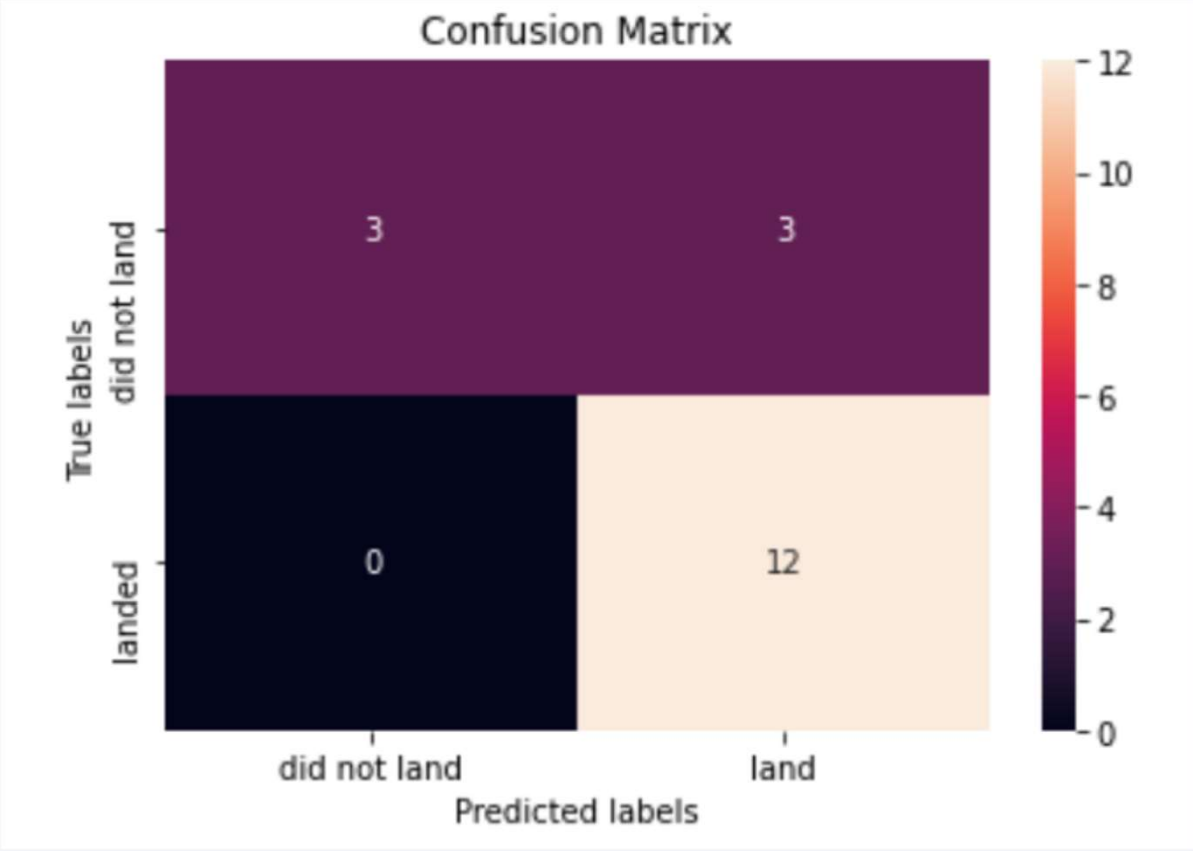



Railway from Launch Site

GitHub

# Results (Predictive analysis results)

Decision Tree Model

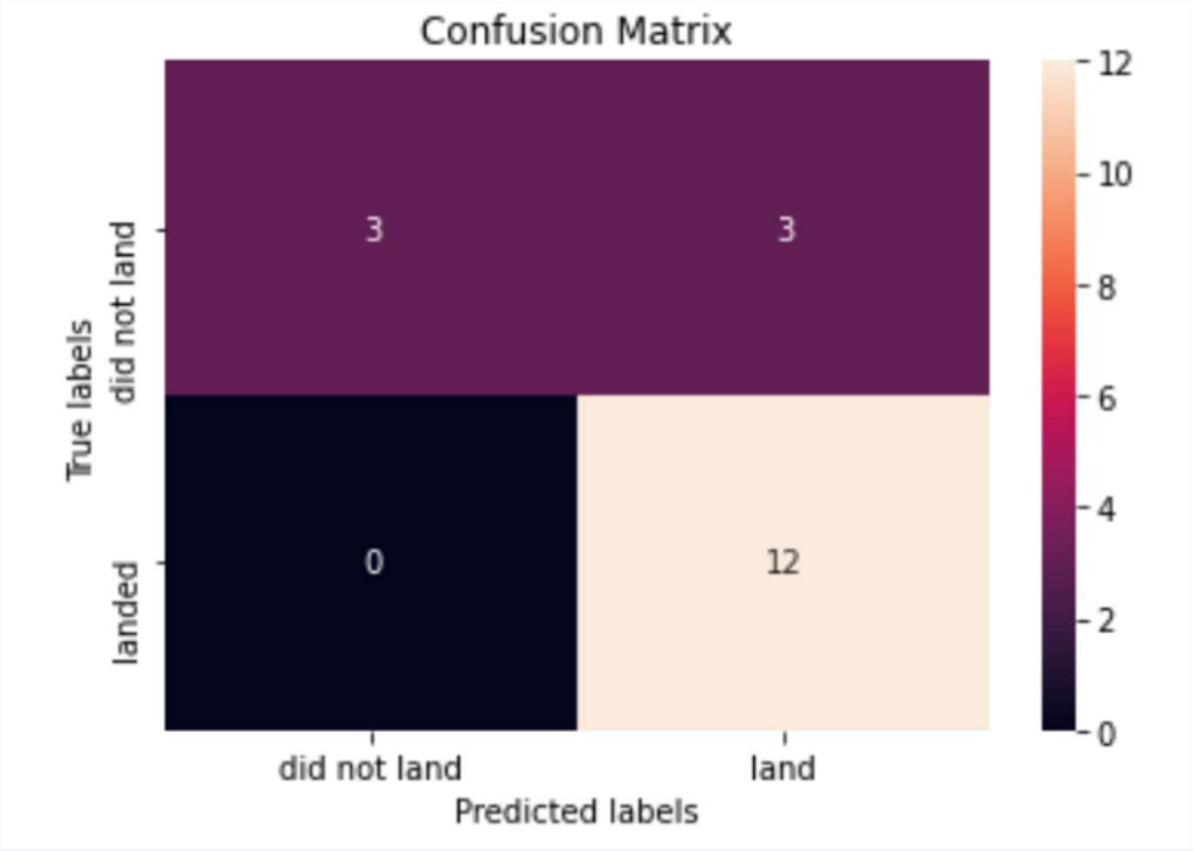



**Accuracy:**
0.8875

GitHub

# Results (Predictive analysis results)

Logistic Regression Model

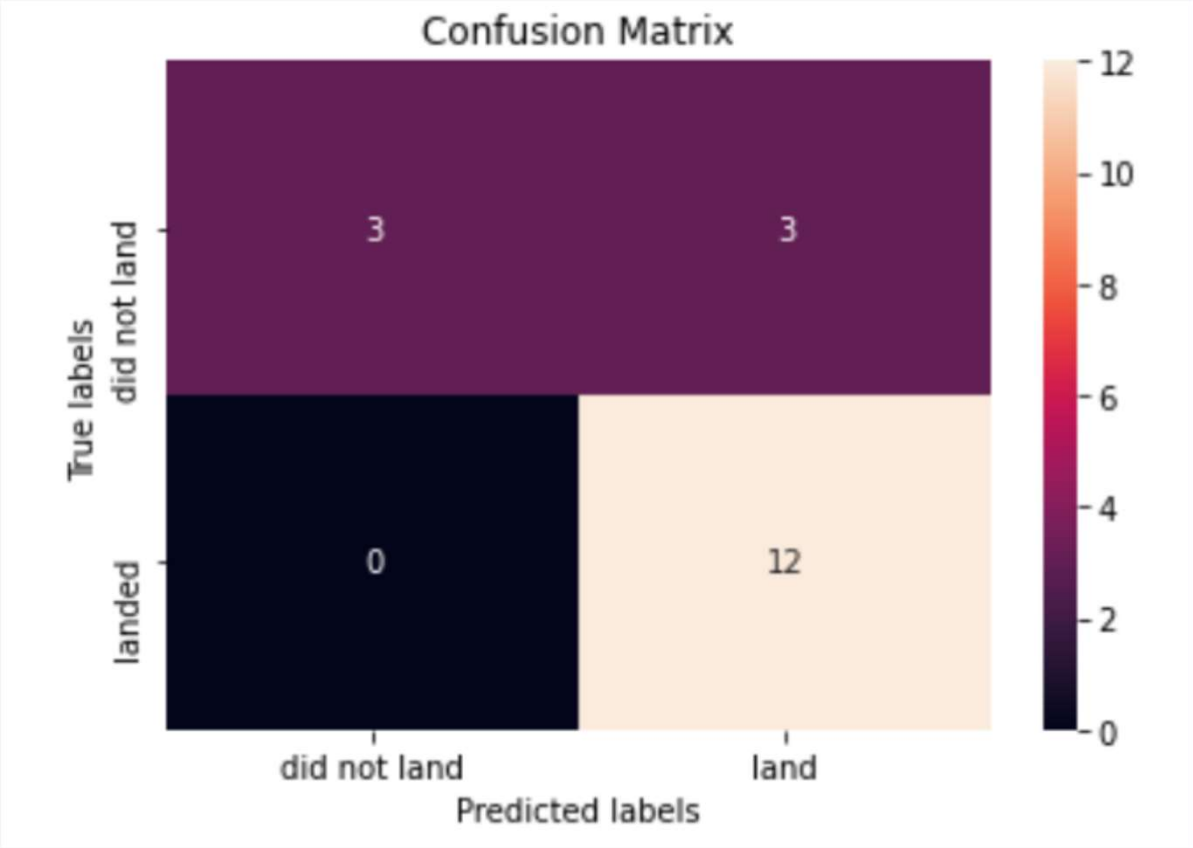



**Accuracy:**
0.8333333333333334

GitHub

# Results (Predictive analysis results)

## Support Vector Machine (SVM)

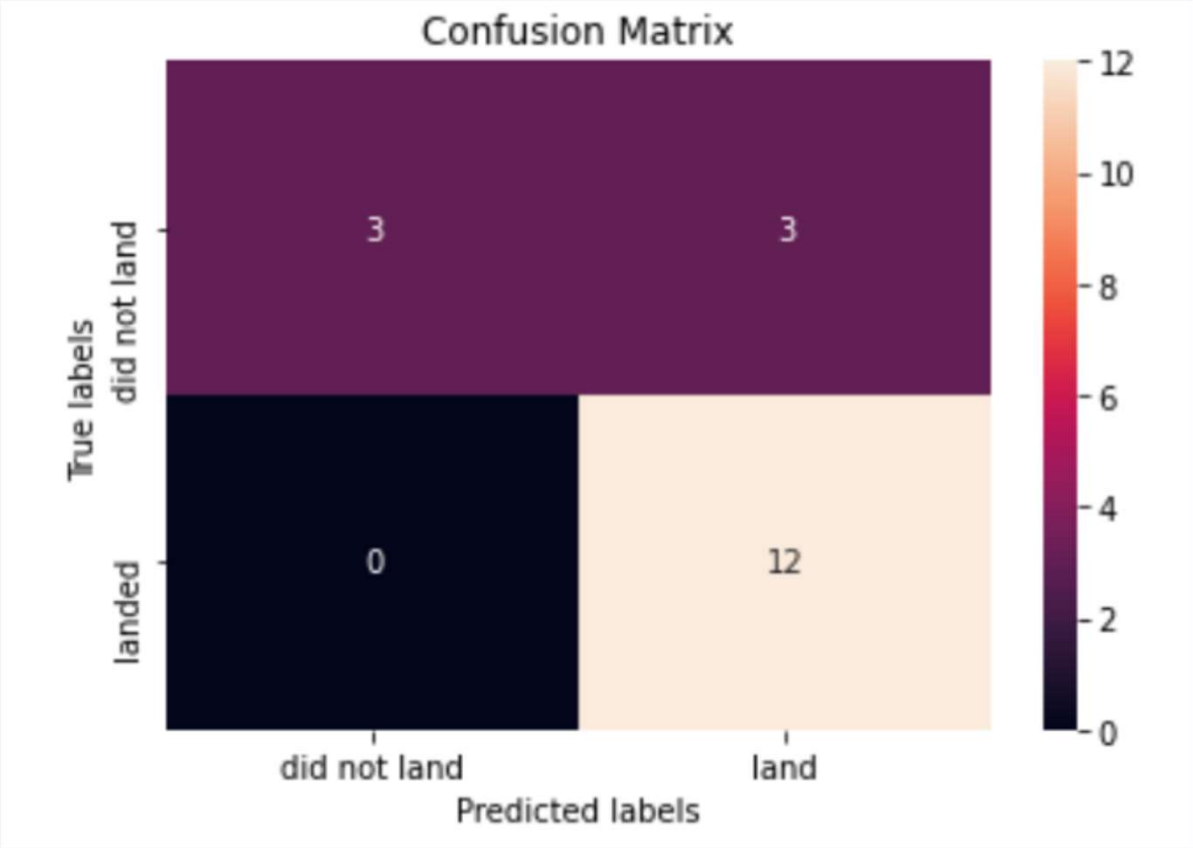



**Accuracy:**
0.8482142857142856

GitHub

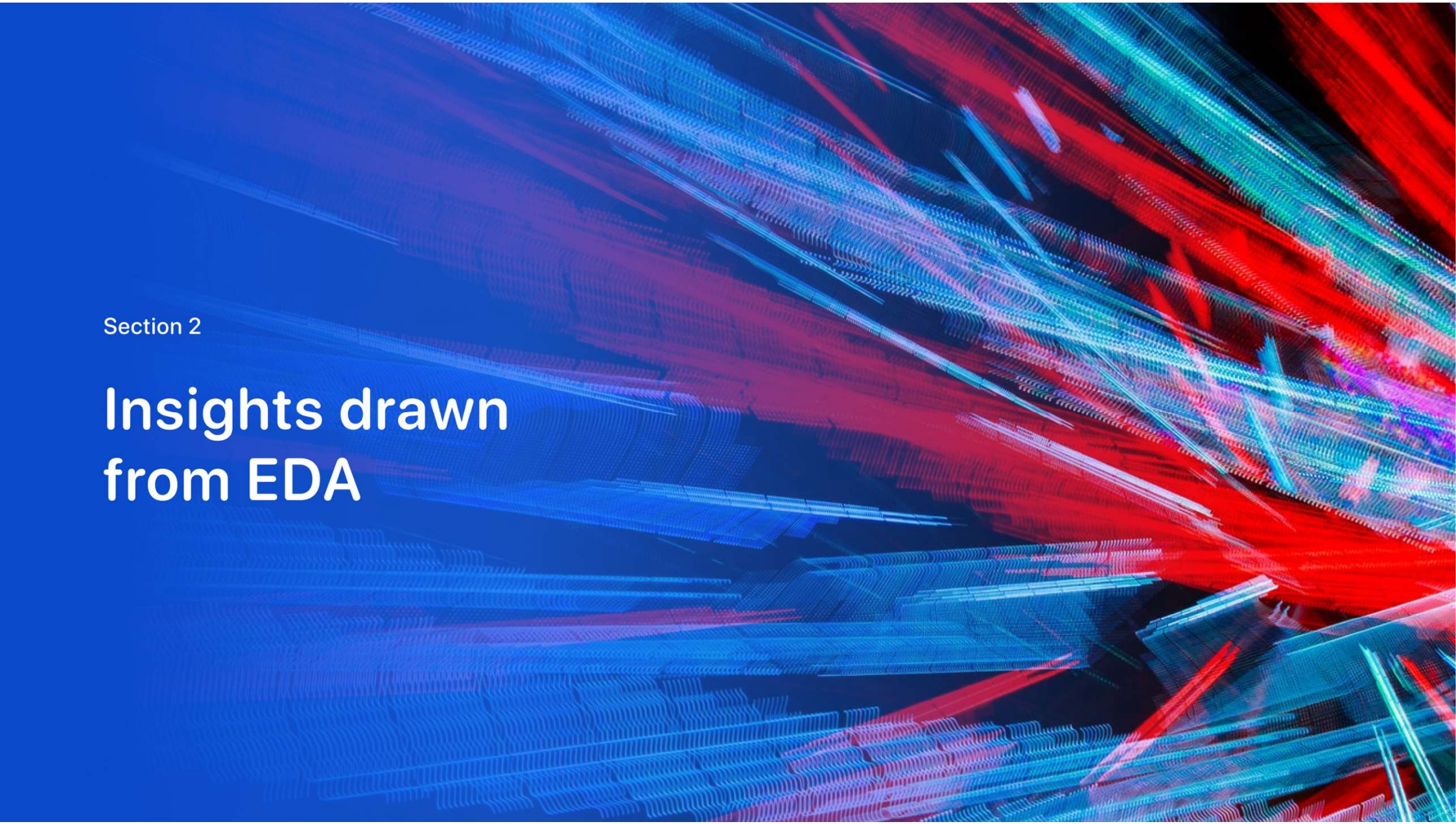
# Results (Predictive analysis results)

## K-Nearest

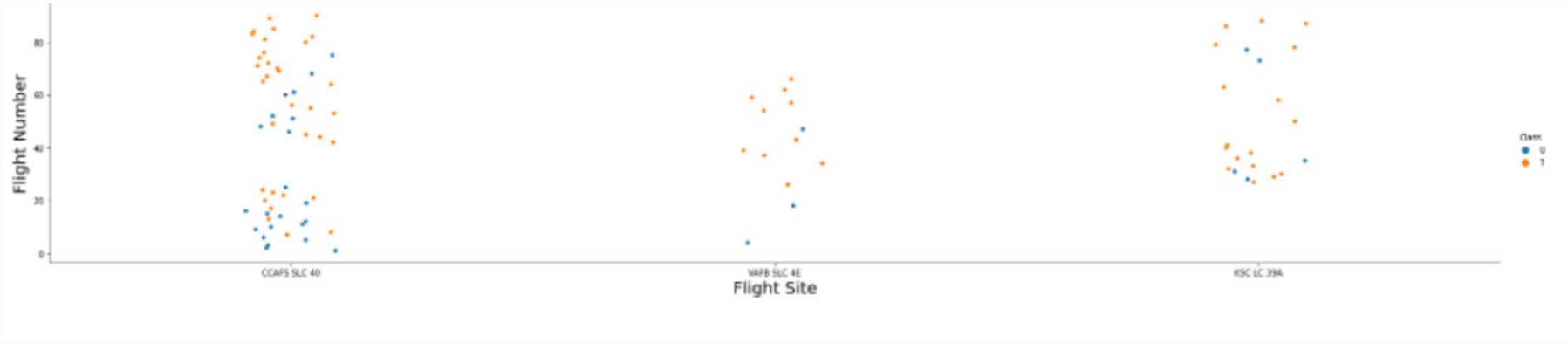



**Accuracy:**
0.8482142857142858

GitHub


Section 2
Insights drawn
from EDA

# Flight Number vs. Launch Site




In the scatter chart is showing launch site at launch Site CCAFS-SLC 40 has fight more then other launch list. The evidence  may come from PayLoadMass of rocket, because success rate of heavy load at this site is 100%

GitHub

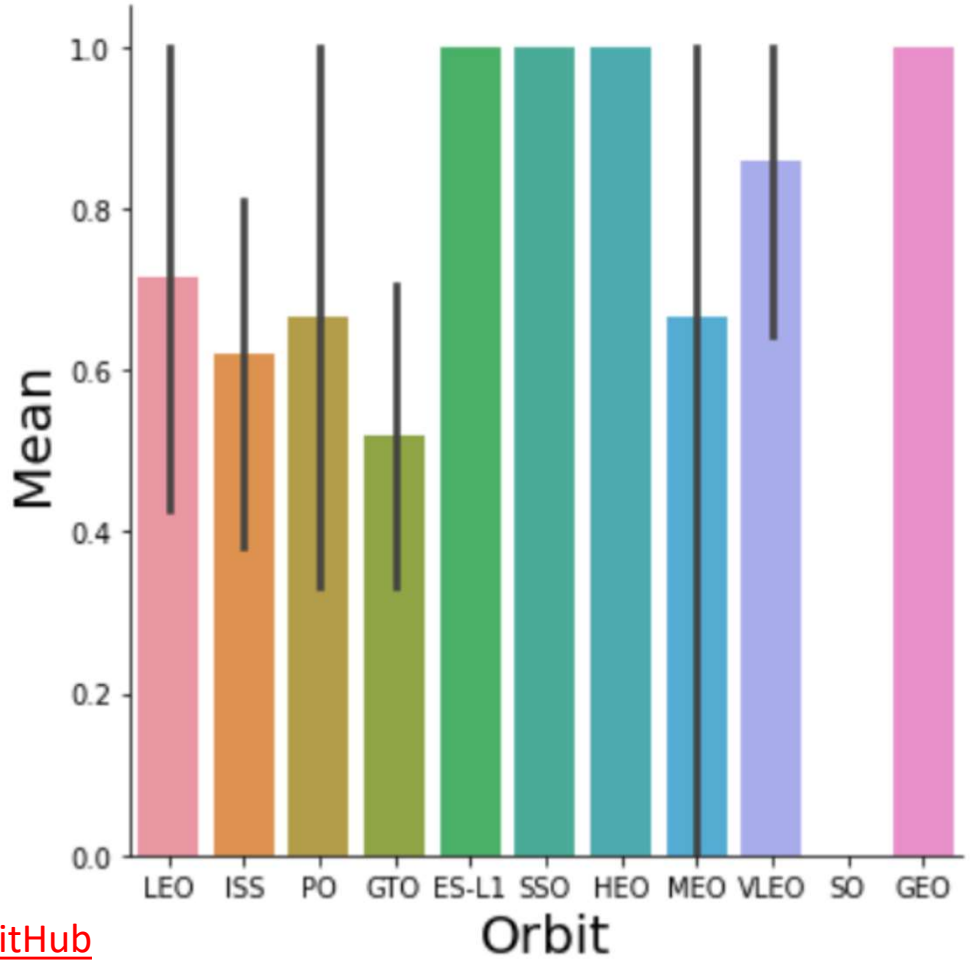
# Payload vs. Launch Site


The scatter chart explain the PayLoadMass with Launch site, because CCAFS-LC40.

It appropriates for all size of rocket, because the scatter chart between PayLaodMass and Fight Number is support chart for  success rate of big rocket is very high success rate when we launch at this site.

GitHub

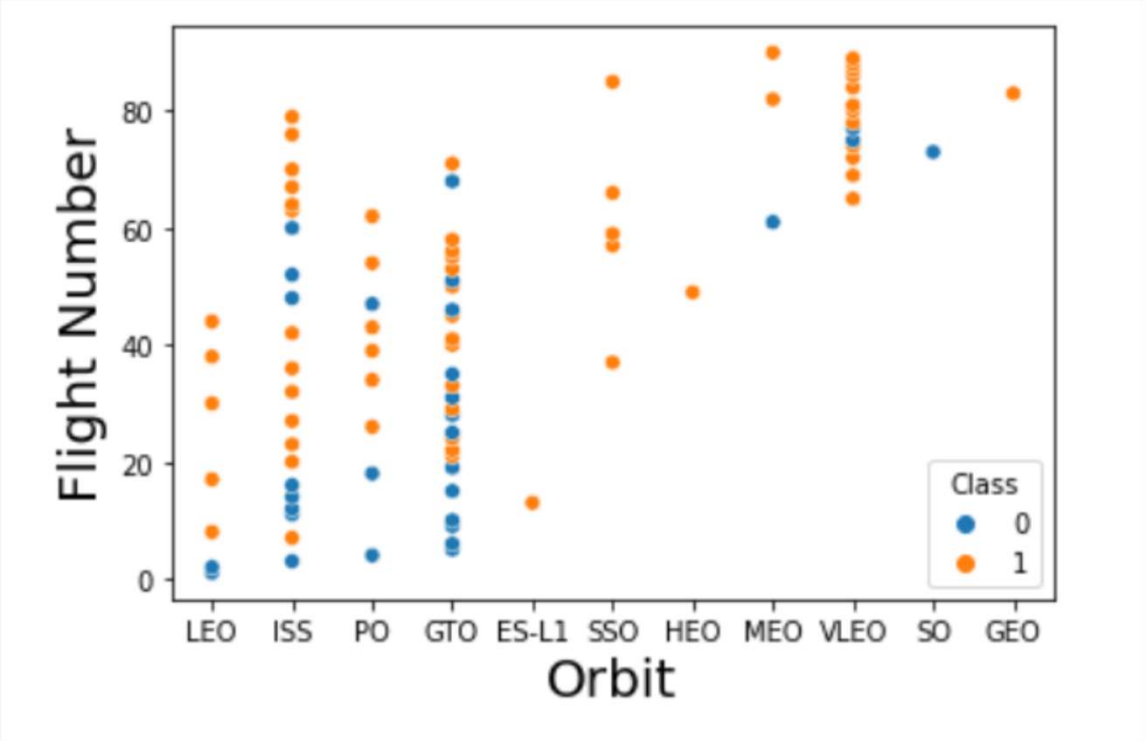
# Success Rate vs. Orbit Type




GitHub

Orbit that it has good success rate are GEO,HEO,SSO and ES-L1

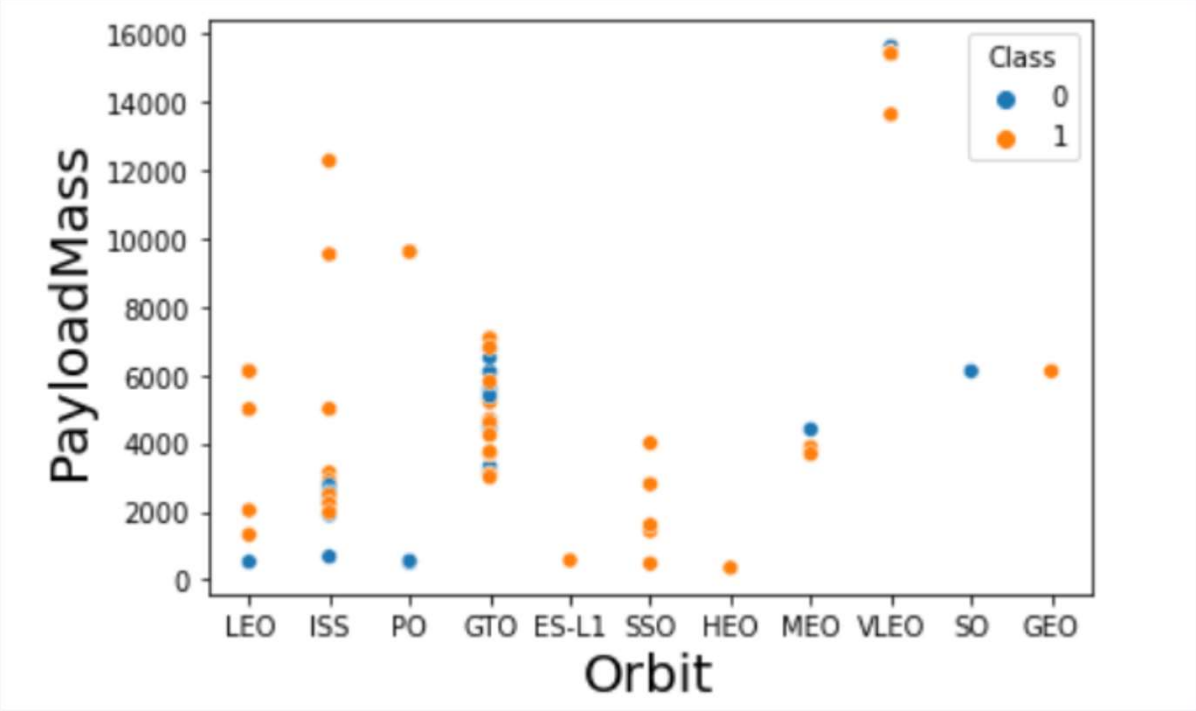
# Flight Number vs. Orbit Type




- LEO which it just only one failure
- VELO is more than fight and one failure too
- GEO can not compare to other, because it just only on fight to go

GitHub

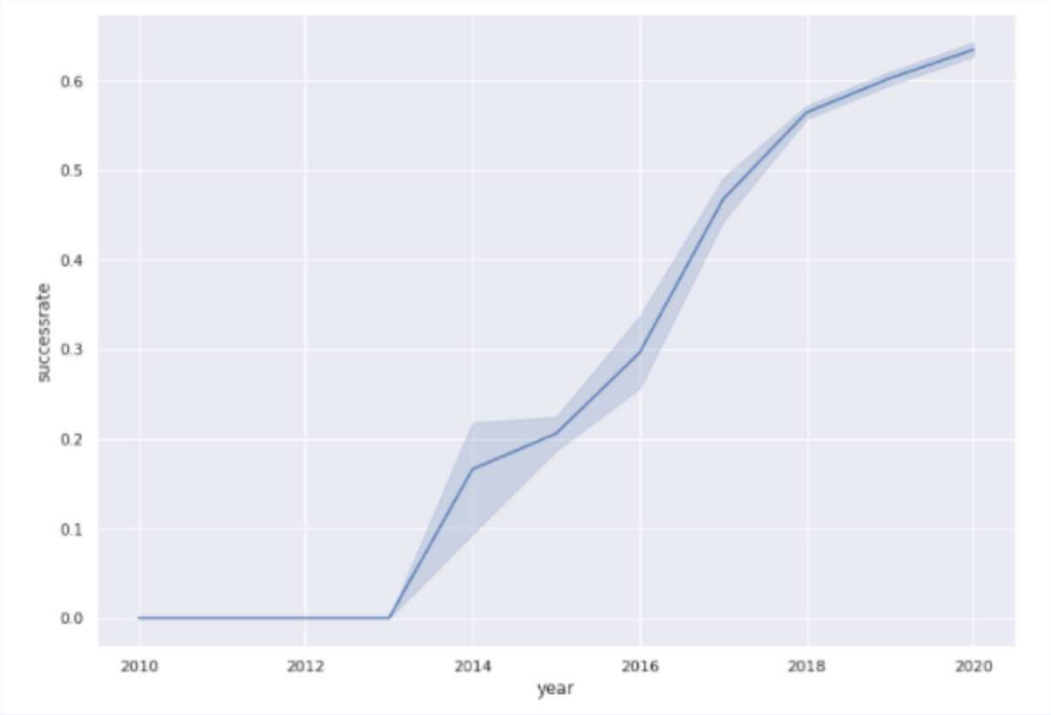
# Payload vs. Orbit Type




GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here

GitHub

# Launch Success Yearly Trend




**So far so good**: from 2013 the success rate increase with high slope

GitHub

# All Launch Site Names

"select DISTINCT Launch_Site from spacextbl "

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

By this enquiry use "DISTINCT" build in function to find the Site Names

GitHub

# Launch Site Names Begin with 'CCA'

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

```
%sql select launch_site from spacextbl where launch_site like "CCA%" limit 5
```

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

| Launch_Site |
|---|
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |

In this statement use “like” to find site name is starting with “CCA” by sign “%”

GitHub

# Total Payload Mass

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

```
%sql select sum (PAYLOAD_MASS__KG_)as "Sum Loading (KG)" from spacextbl where Customer = "NASA (CRS)"
```

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

| Sum Loading (KG) |
| --- |
| 45596 |

“SUM” build in function to calculate the total PayLoadMass

GitHub

# Average Payload Mass by F9 v1.1

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

```
%sql select avg (PAYLOAD_MASS__KG_) as "Average Loading (KG)" from spacextbl where Booster_Version = "F9 v1.1"
```

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

| Average Loading (KG) |
| --- |
| 2928.4 |

Average the PayLoadMass can be use “AVG” in SQL statement

GitHub

# First Successful Ground Landing Date

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

*Hint:Use min function*

```
%sql select min (Date) as " First succesful landing","Mission_Outcome","Landing _Outcome" from SPACEXTBL where "Landing _Outcom
e" = "Success (ground pad)"
```

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

| First succesful landing | Mission_Outcome | Landing _Outcome |
|---|---|---|
| 01-05-2017 | Success | Success (ground pad) |

“MIN” is build in Function to apply at “Date” filed

GitHub

# Successful Drone Ship Landing with Payload between 4000 and 6000

***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,PAYLOAD_MASS__KG_ from spacextbl where ("Landing _Outcome" = "Success (drone ship)") and ("PAYLOAD_MASS__KG_" >= 4000 and "PAYLOAD_MASS__KG_" <= 6000)
```

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

| Booster_Version | PAYLOAD_MASS__KG_ |
|---|---|
| F9 FT B1022 | 4696 |
| F9 FT B1026 | 4600 |
| F9 FT B1021.2 | 5300 |
| F9 FT B1031.2 | 5200 |

Multi condition in SQL will be use “AND” logic checking

GitHub

# Total Number of Successful and Failure Mission Outcomes

*List the total number of successful and failure mission outcomes*

```
%sql select(select count(Mission_Outcome) from spacextbl where Mission_Outcome like '%Success%') as Successful_Mission_Outcomes,
(select count(Mission_Outcome) from spacextbl where Mission_Outcome like '%Failure%') as Failure_Mission_Outcomes
```

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

| Successful_Mission_Outcomes | Failure_Mission_Outcomes |
|---|---|
| 100 | 1 |

**Task 8**

Sub query applies into the query and use “%” for string finding

GitHub

# Boosters Carried Maximum Payload

```
%sql select distinct Booster_Version, max("PAYLOAD_MASS__KG_") as "Maximum Payload Mass" from spacextbl group by Booster_Version
order by [Maximum Payload Mass] DESC
```

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

| Booster_Version | Maximum Payload Mass |
|---|---|
| F9 B5 B1060.3 | 15600 |
| F9 B5 B1060.2 | 15600 |
| F9 B5 B1058.3 | 15600 |
| F9 B5 B1056.4 | 15600 |
| F9 B5 B1051.6 | 15600 |
| F9 B5 B1051.4 | 15600 |
| F9 B5 B1051.3 | 15600 |
| F9 B5 B1049.7 | 15600 |
| F9 B5 B1049.5 | 15600 |
| F9 B5 B1049.4 | 15600 |
| F9 B5 B1048.5 | 15600 |
| F9 B5 B1048.4 | 15600 |

Sorting the maximum Payload with “DESC” couple command “ORDER BY”

GitHub

# 2015 Launch Records

*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.* ¶

**Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.**

```
%sql select "Mission_Outcome","Landing _Outcome",substr(Date, 4, 2) as Month ,substr(Date,7,4) as Year from spacextbl where "Landing _Outcome" = "Failure (drone ship)" and  substr(Date,7,4)='2015'
```

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

| Mission_Outcome | Landing _Outcome | Month | Year |
|---|---|---|---|
| Success | Failure (drone ship) | 01 | 2015 |
| Success | Failure (drone ship) | 04 | 2015 |

SQLite is not support “MonthName” so we use sub-string for this

GitHub

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

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

```
%sql select count ("Landing _Outcome") as "Landind_Success between the date 04-06-2010 between 20-03-2017" from spacextbl where
("Landing _Outcome" LIKE '%Success%') and (Date >'04-06-2010') and (Date < '20-03-2017')
```

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

| Landind_Success between the date 04-06-2010 between 20-03-2017 |
|---|
| 34 |

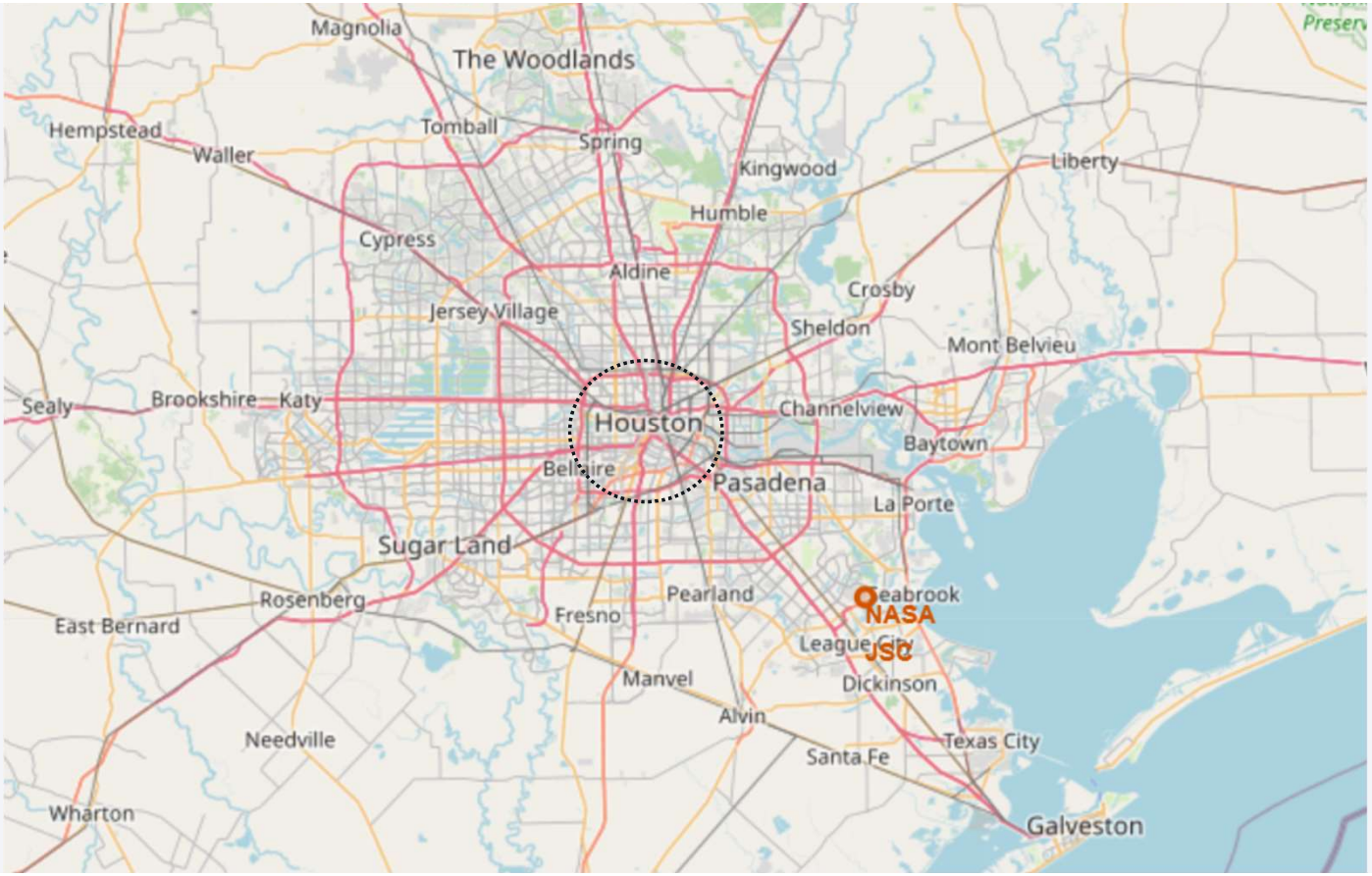
"LIKE" for string finding use "AND" to verify the condition must be true

GitHub


Section 4
Launch Sites
Proximities Analysis

# Initial the Folium map

We first need to create a folium Map object, with an initial center location to be NASA Johnson Space Center at Houston, Texas

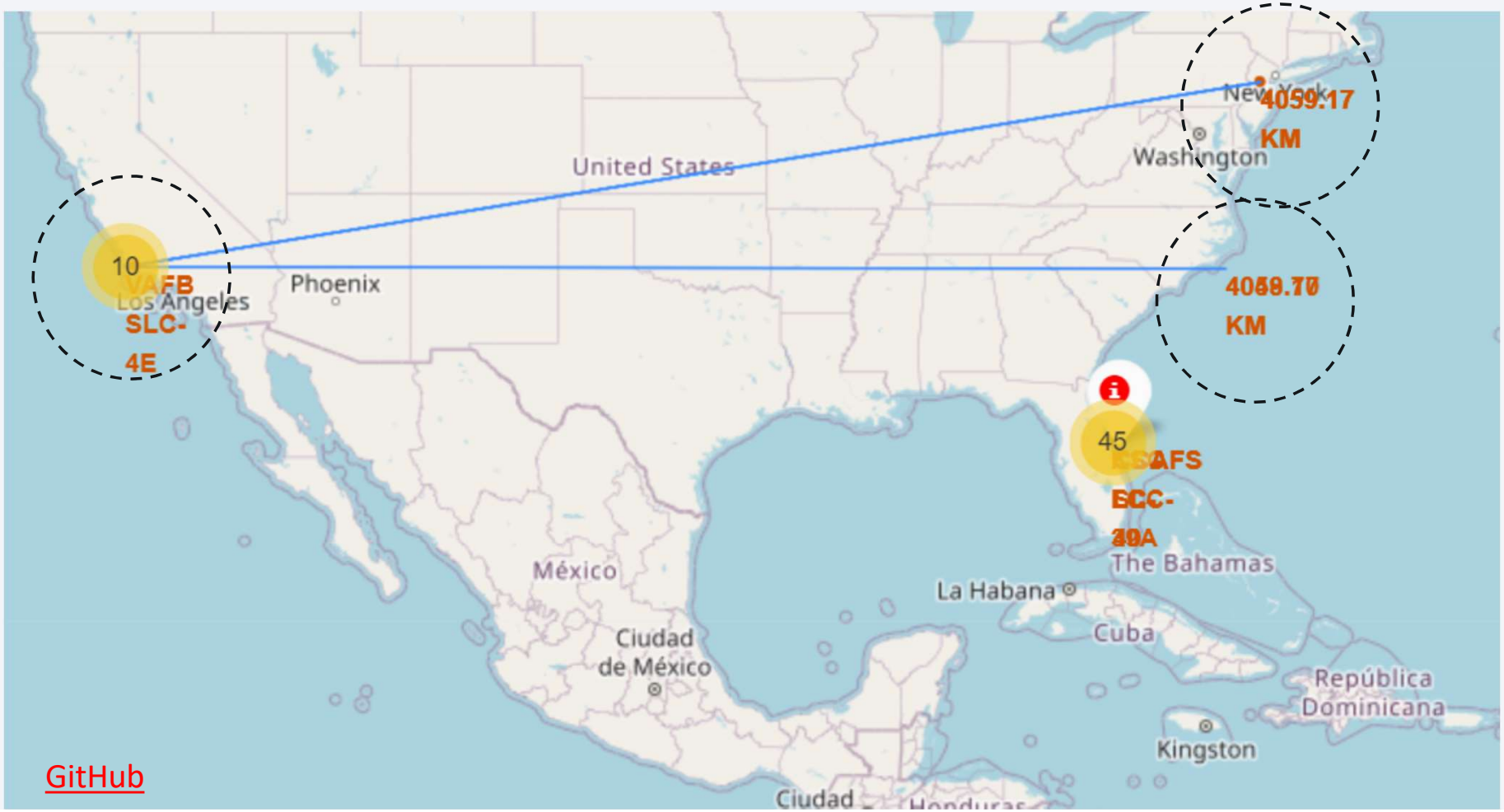


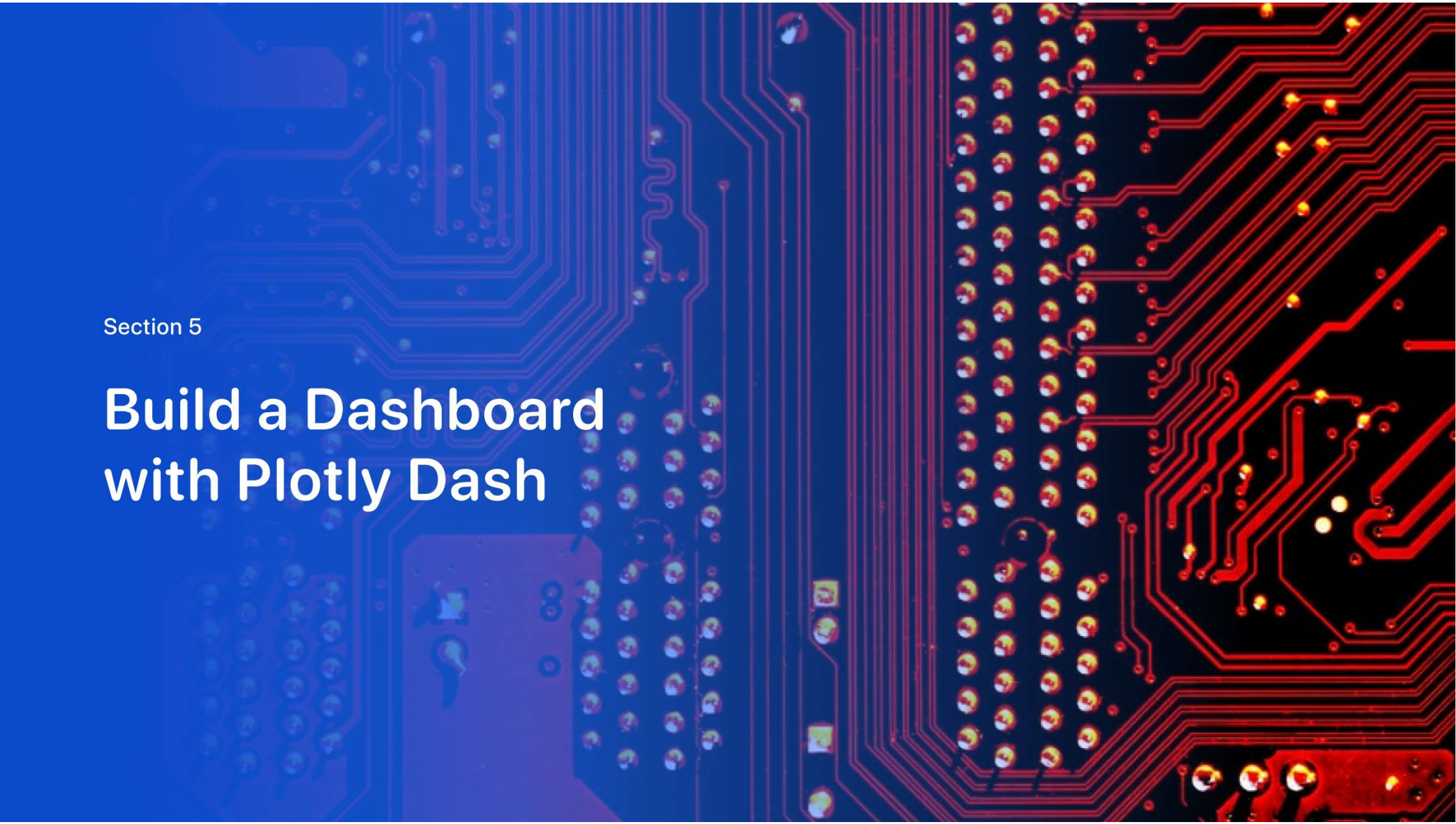

GitHub

# Create and add Circle, Marker for Launch Site


GitHub

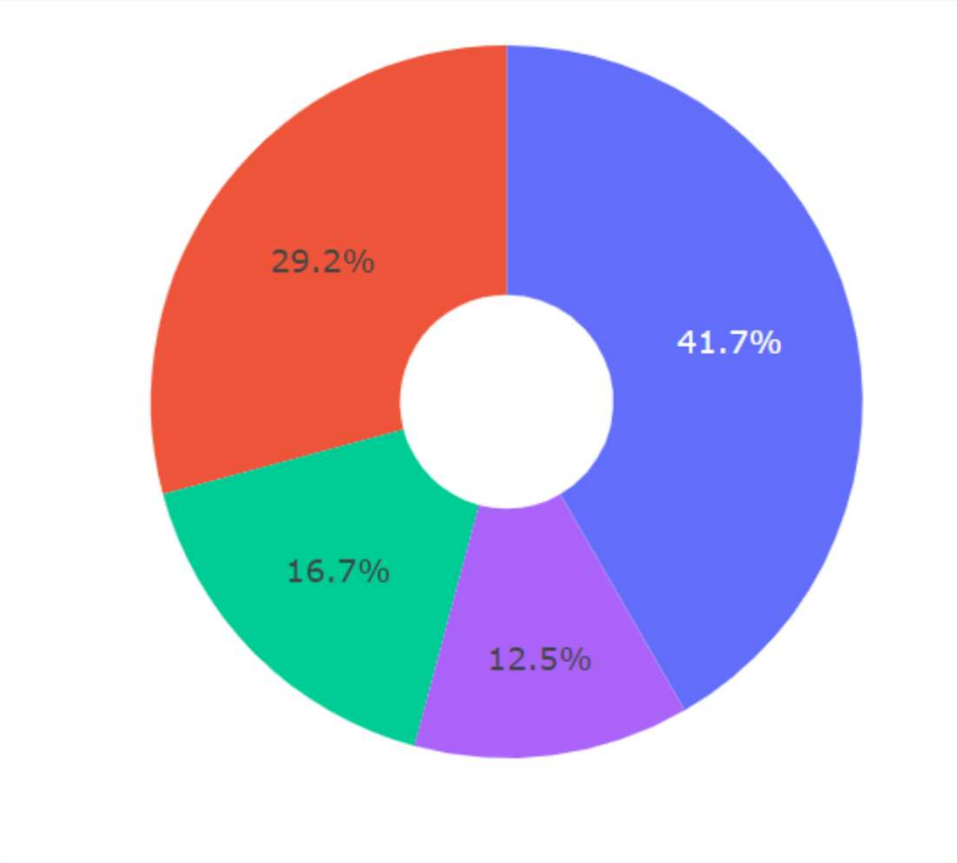
# Apply the Folium map to find distance




Section 5
Build a Dashboard with Plotly Dash

# SpaceX Launch Record Dashborad

**Total Success Launch By all sites**



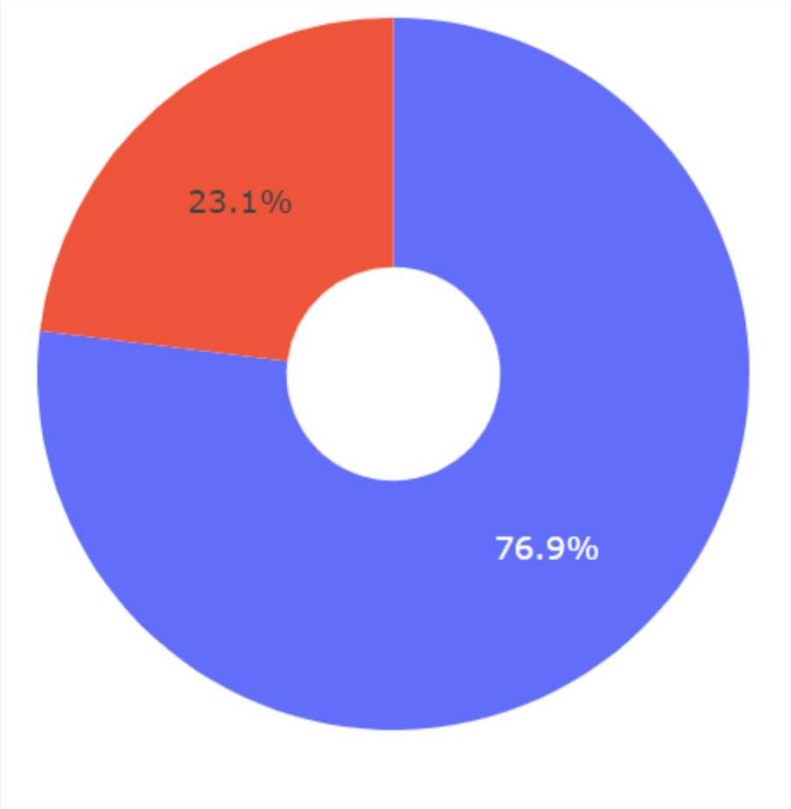

This chart is showing the success at all sites. Why we need to consider. Because on each in different wealth, size of rocket and also orbit for landing. It means the historical data must be analyze.



GitHub

# SpaceX Launch Record Dashborad

**Highest Launch success at site KSC LA 39A**




KSC LA 39A why this site has highest success rate. This question Should be find out more detail.

GitHub

# SpaceX Launch “Biggest, Risk and Cost”


My comment is “ “Biggest is High Risk and Cost”
So in my view they try to compromise 3 things

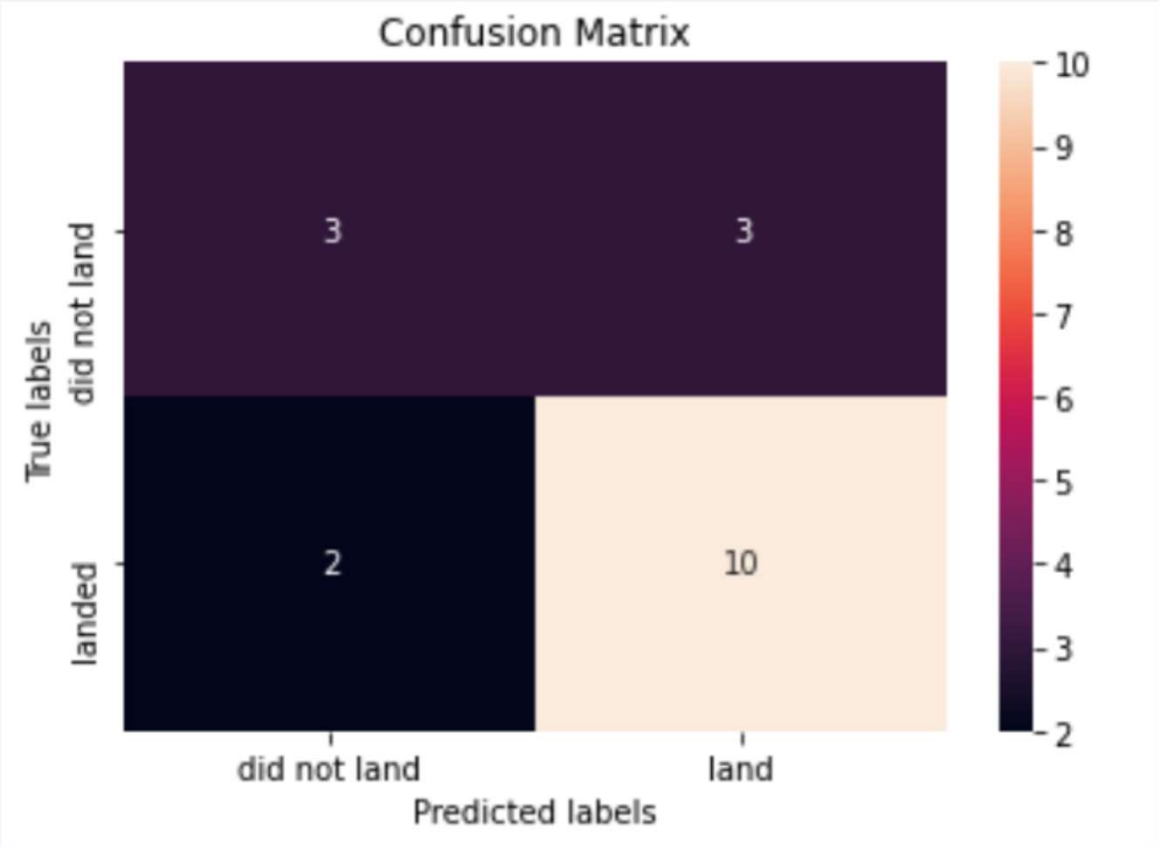
Booster Version
- F9 FT B1020
- F9 FT B1029.1
- F9 FT B1036.1
- F9 B4 B1041.1
- F9 FT B1036.2
- F9 B4 B1041.2
- F9 B4 B1043.2
- F9 FT B1030
- F9 FT B1021.2
- F9 FT B1034
- F9 FT B1037
- F9 FT B1031.2
- F9 B4 B1044
- F9 B4 B1040.2

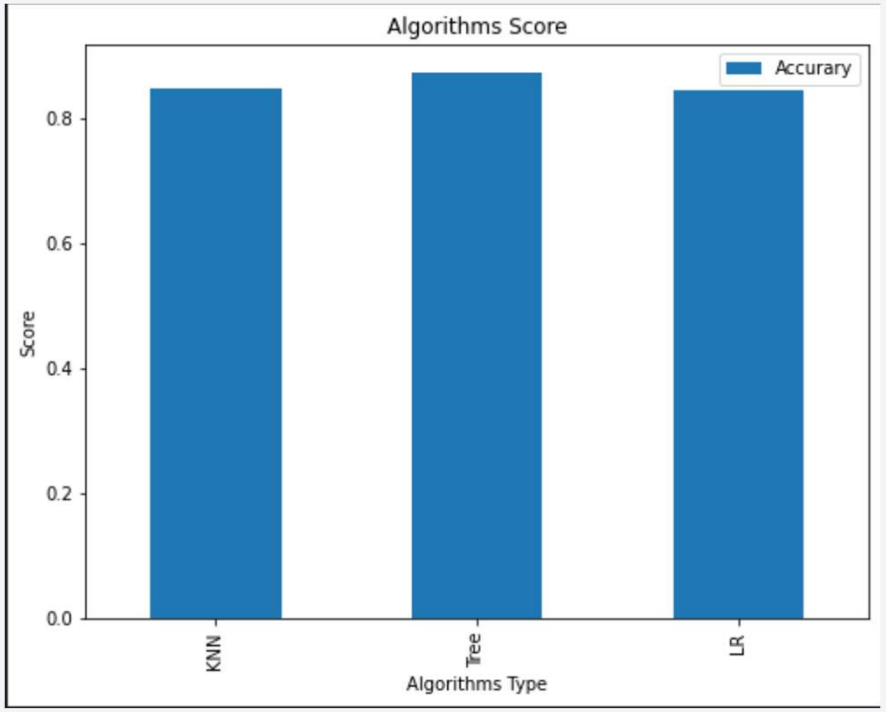
GitHub

Section 6

# Predictive Analysis (Classification)

# Confusion Matrix (Decision Tree Model)




**Accuracy:**0.8795

The accuracy is very near between 3 algorithms.
In my view if we have the sample space more this
It may know the differentiate of them

GitHub

# Conclusions

- Orbit should be GEO,HEO,SSO,ES-L1 are the success rate
- Success Rate is direct impact feature launch more high improvement
- Tree is the best model to Machine Learning (Unsupervised model)
- Real time dashboard will be helpful for retrieve or compare the historical information

# Appendix

- Python Tutorial (w3schools.com)
- pandas - Python Data Analysis Library (pydata.org)
- pandas.DataFrame.plot.bar — pandas 1.3.4 documentation (pydata.org)
- Anaconda | Individual Edition
- Hello World - GitHub Docs
- SQLite Python (sqlitetutorial.net)
- Geospatial Analysis using Folium in Python | Work with Location Data (analyticsvidhya.com)

# Thank you!