Georgi Georgiev

Space X Falcon 9 First Stage Landing Prediction Project

Outline

- Executive Summary
- Introduction
- Results
- Conclusions

Excutive Summary

Methodologies

- Data Collection through API and Web Scraping
- Data Wrangling
- Exploratory Data Analysis with SQL and Data Visualization
- Interactive Visual Analytics using Folium and Plotly Dash
- Predictive Analysis(Classification)

Results

- Exploratory Data Analysis
- Visual Analytics
- Predictive Analystics

Introduction

- Space X is a spacecraft manifacturer founded 2002 in California. The company offers rocket lauches specially Falcon on lower cost than the competition. Reason for that is the re-usage of re-land rockets for the next mission. Goal this Project is to establish a machine learning pipeline to predict the landing outcome.
- Answers we search in this study:
- Factors that have influence of the landing outcome
- Finding the best condicions for successful landing

Methodology

- Data collection
- Data wrangling
- Exploratory Data analysis(EDA)using visualisation und SQL
- Interactive visual analytics using Folium and Plotty Dash
- Predictive analysis using classification models

Data Collection

- Data sets from Wikipedia: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922“

Data Collection

- Using request to collect data and creating Beautifull Soup object
- Extract the column names und converting into a panda Dataframe
- Link to notebook:https://github.com/Gonzinho09/coursera/blob/main/Web%20scraping%20Falcon%209%20and%20Falcon%20Heavy%20Launches%20Records%20from%20Wikipedia.ipynb

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of

response = requests.get(static_url)

BeautifulSoup = BeautifulSoup(response.text, "html.parser")

BeautifulSoup.find_all('title')

[<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</
```

```
column_names = []

temp = BeautifulSoup.find_all('th')

for i in range(len(temp)):
    try:
        name = extract_column_from_header(temp[i])
        if (name is not None and len(name) > 0 ):
            column_names.append(name)
    except:
        pass
```

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

Data Wrangling

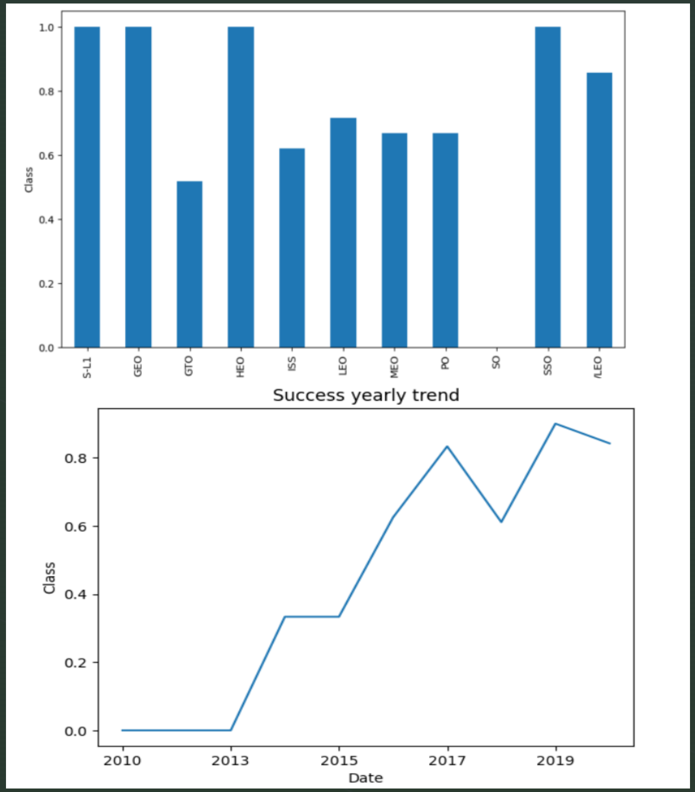
- Cleaning and converting the data in a propriate format for easy access and Exploratory Data Analysis(EDA)
- Calculation the number of lauches for each site
- Calculation of number and occurence of mission outcome for each orbit type
- Creating of an Outcome column
- Data Set: https://github.com/Gonzinho09/coursera/blob/main/Data%20wrangling.ipynb

EDA with SQL

- Following Queries were performed:
- Names of the unique launch sites in the space mission
- Five records where the launch sites begin with the string `CCA`
- Total payload mass carried by boosters launched bei NASA(CRS)
- Average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pas was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 and lass than 6000
- Total number of successful and failure mission outcomes
- Name of the booster versions which have carried the maximum payload mass
- 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.
- 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.
- Data Set: https://github.com/Gonzinho09/coursera/blob/main/EDA%20with%20SQL.ipynb

EDA with Visualisation

- Using a scatter plot to visualize the relationship between the components
- With a bar graph we showed the relationship between success rate of each orbit type
- With a line graph we visualize the success rate between 2013 and 2020
- Creating dummy variables that will be used in success prediction in the future module
- Data Set: https://github.com/Gonzinho09/coursera/blob/main/EDA%20with%20Visualisation.ipynb



Interative map with folium

- Creating marker indicators and circels for each launch site
- Creating markers for all launch records with color green for success and red for failure
- Calculating distance between launch sites and nearby coastline and city
- Data Set: https://github.com/Gonzinho09/coursera/blob/main/Launch%20Sites%20Locations%20Analysis%20with%20Folium.ipynb

Dashboard with Plotly Dash

- Create a Dropdown list allowing the user to select launch site
- Building Pie chart showing the total launches by a certain site
- Establish a Slider allowing the user to select payload mass range
- Create a Scatter chart showing the relationship betweeen Payload Mass in Kg and Outcome for a different booster version
- Data Set: https://github.com/Gonzinho09/coursera/blob/main/Plothy%20Dash%20Code.docx

Predictive Analysis

- Create a NumPy array from a Class column
- Standardaize and transform the data
- Split the set into training and test data
- Create a GridSearchCV object with parameter cv=10
- Using GridSeachCV on different algorithms: Logistic Regression, Support Vector Machine, Decision Tree and K-Nearest Neighbor
- Calculate accuracy of the test data using the score() method and access the confusion matrix for all models
- Identify the best model
- Data set: „https://github.com/Gonzinho09/coursera/blob/main/Predictive%20Analytics.ipynb“

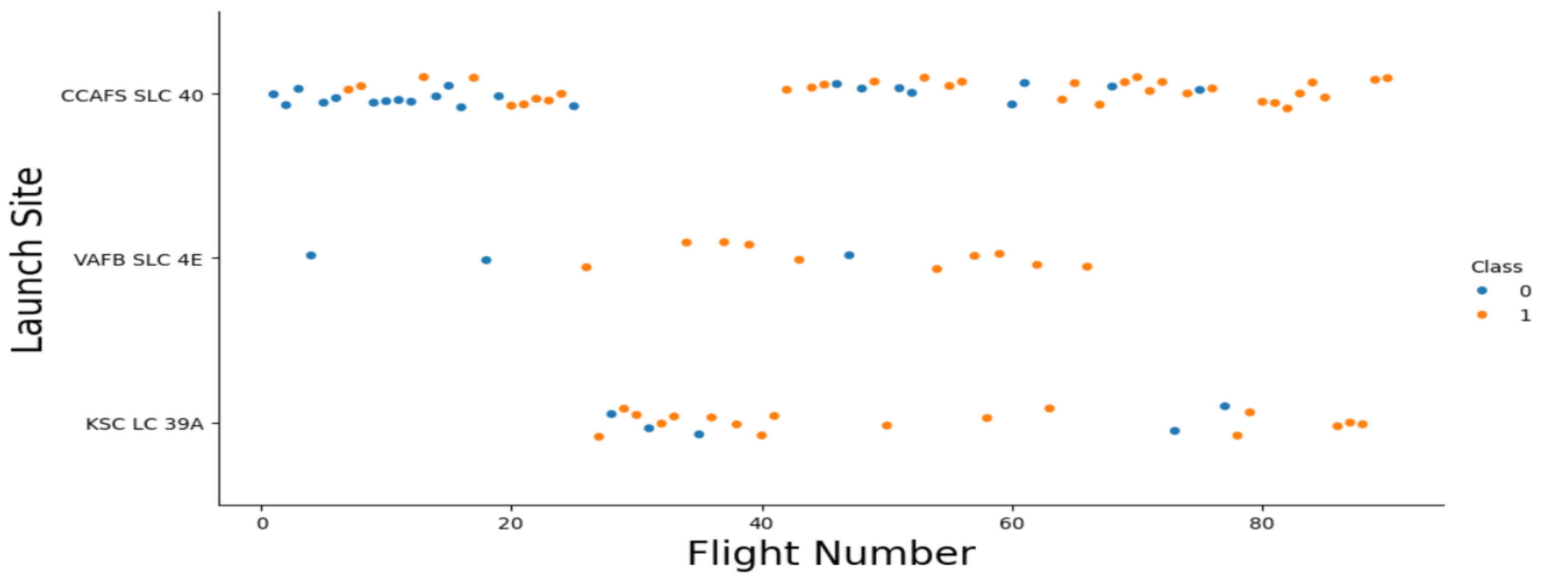
Results

- Exploratory data analysis
- Visual Analytics
- Predictive Analytics

Exploratory data analysis

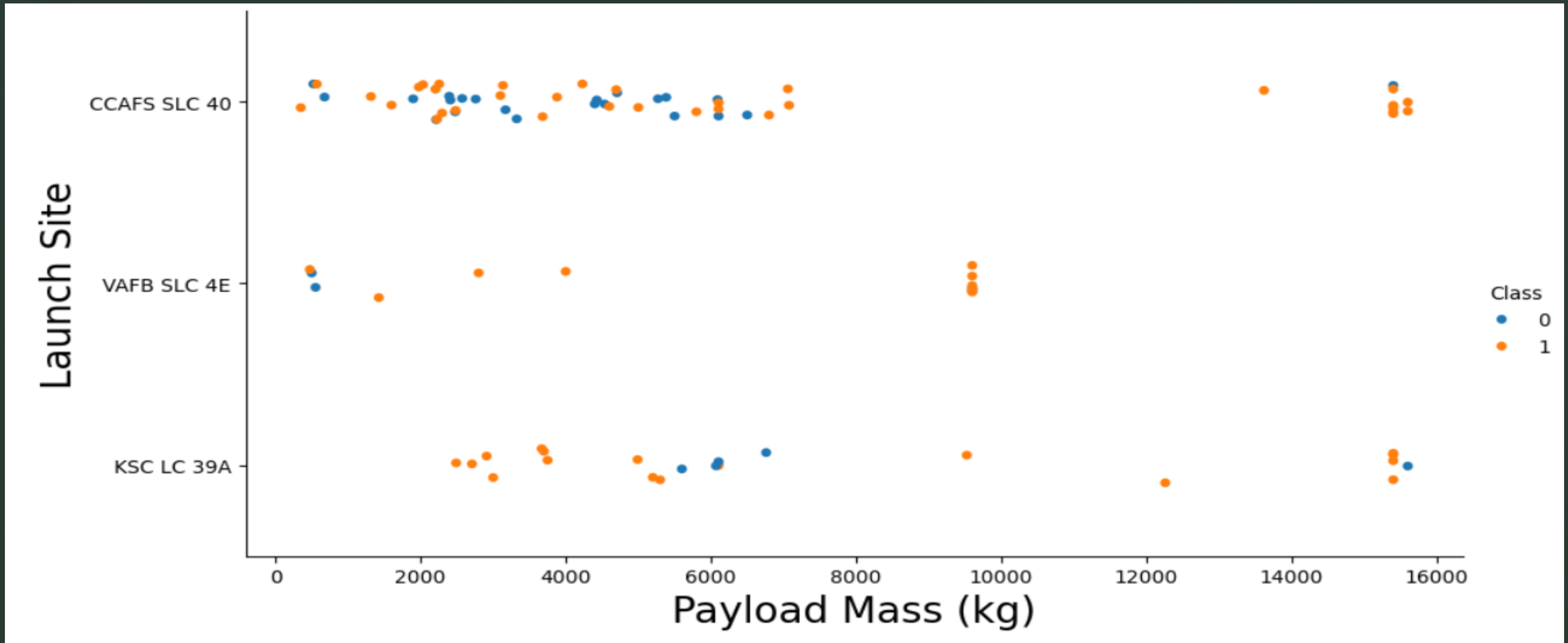
Flight Number vs.Launch Site

- Earlier flights have lower success rate (blue shows failed launches)
- Later flights have higher success rate (orange shows successed lauches)



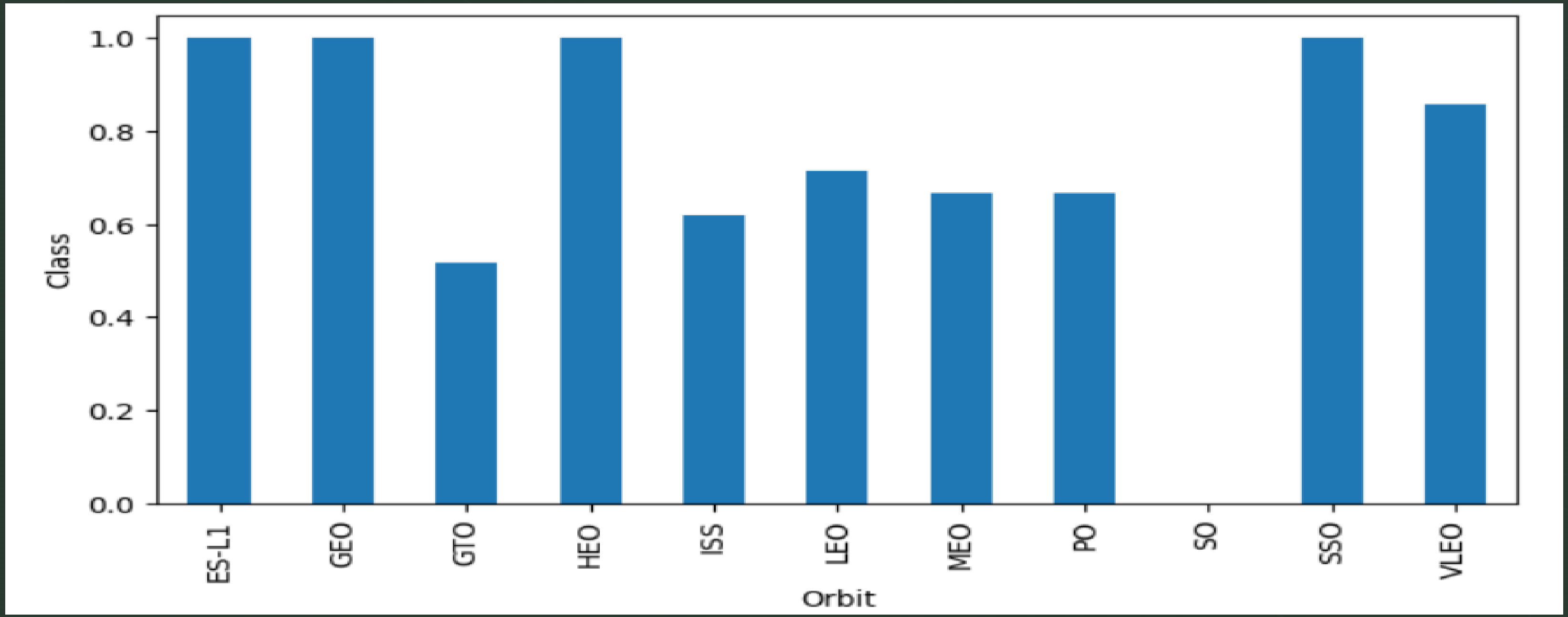
Payload vs. Launch Site

- Success rate is higher with a bigger payload mass (kg)
- Most of the launches with payload mass greater than 7000km are successful
- Launch Site KS LC 39A has 100% success rate with Payload Mass less than 5500kg



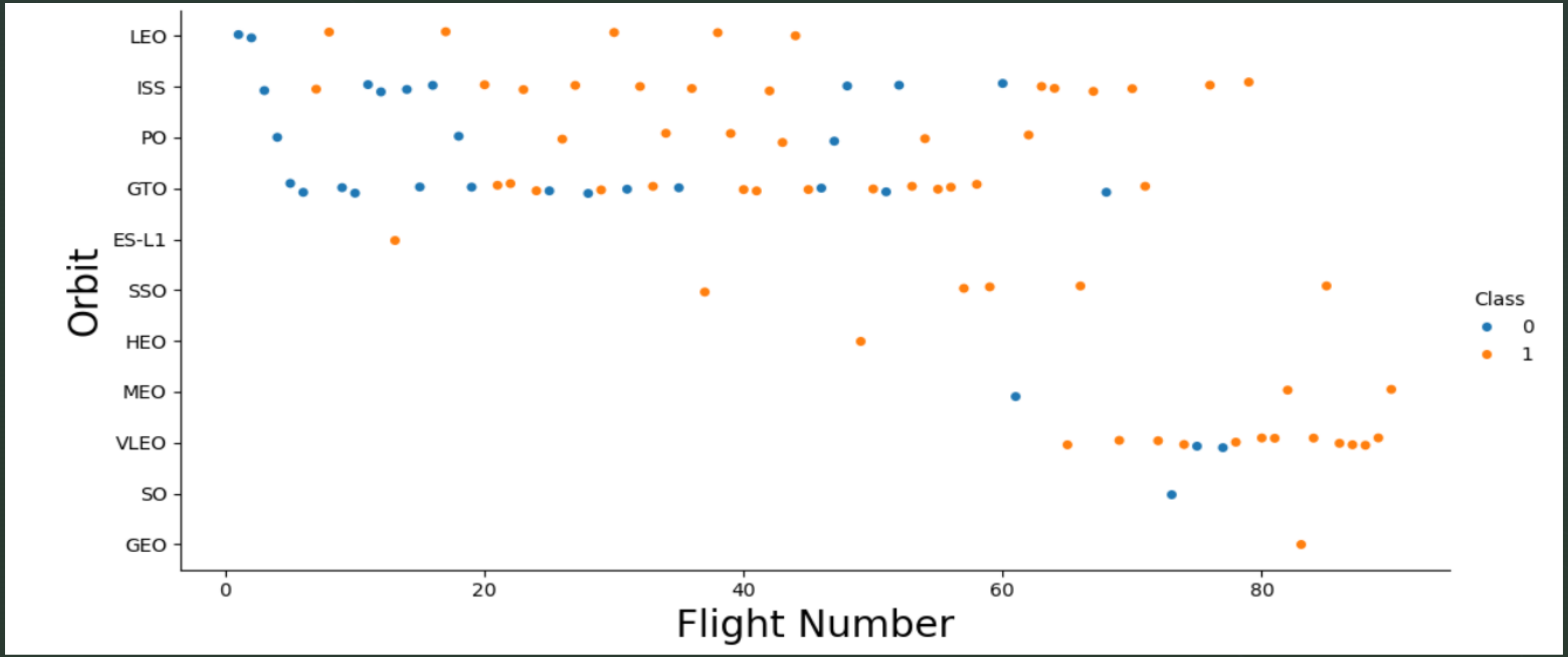
Success Rate by orbit

- ES-L1, GEO, HEO and SSO have 100% success rate
- GTO, ISS, LEO, MEO, PO have 50-80% Success rate
- SO have 0% success rate



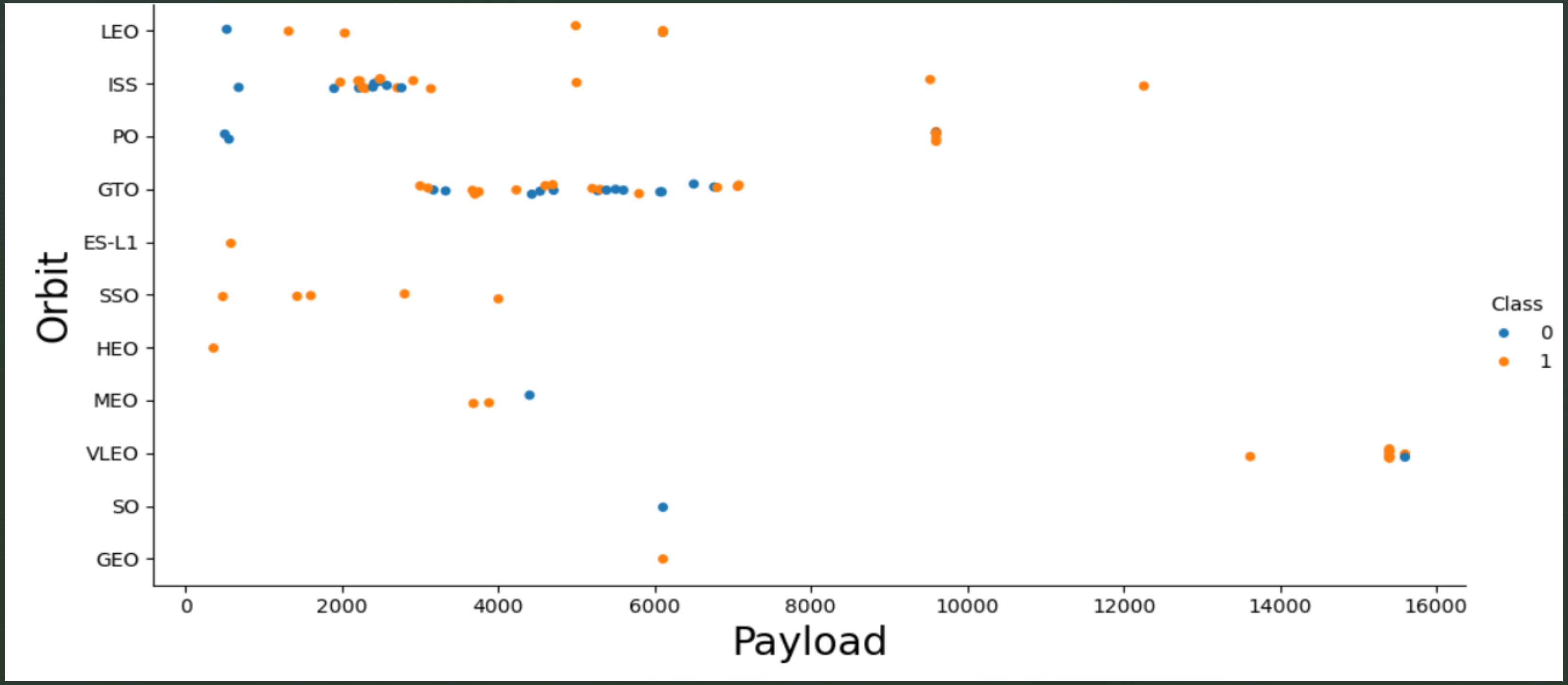
Flight Number vs. Orbit

- Success rate increases with the number of flights except LEO orbit



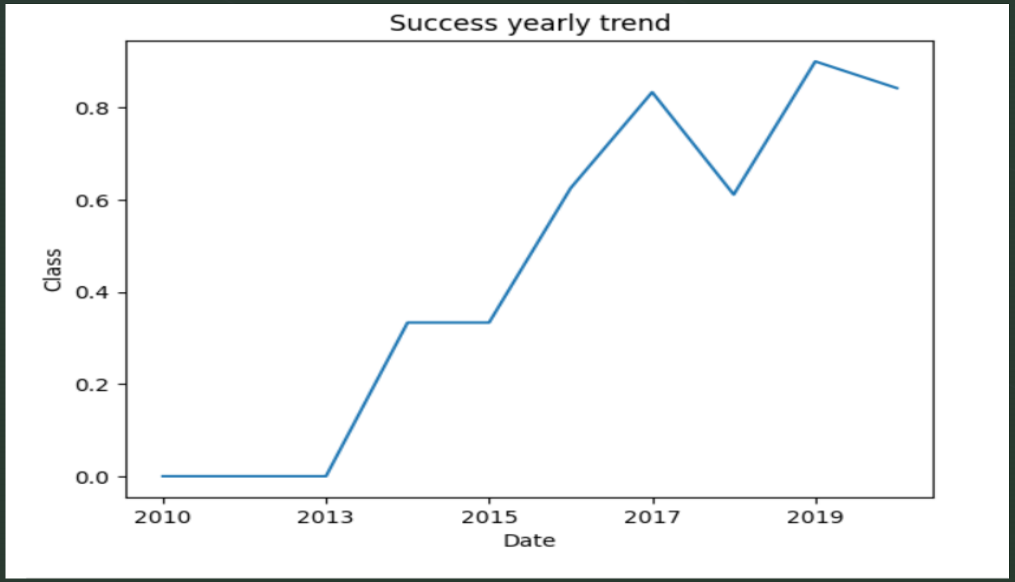
Payload vs. Orbit

- LEO, ISS and PO orbits have more success with heavier Payload



Success Yearly Trend

- The overview of the successful launches show continuous improvement from year 2013 to 2019 with exeption of 2017-2018
- Decline in 2019-2020



All Launch Site Names

```
%sql select distinct Launch_Site from SPACEXTBL;
```

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

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

Launch Site Names beginning with ‚CCA‘

```
%sql select * from SPACEXTBL 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
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer like 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version like'F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First successful Ground Landing Date

```
%sql select min(Date) from SPACEXTBL where Landing_Outcome like'Success (ground pad)'
```

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

min(Date)
2015-12-22

Successful Drone Ship with Payload between 4000 and 6000 kg

```
distinct Booster_Version from SPACEXTBL where Landing_Outcome like 'Success%(drone_ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000
```

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

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

Successful Drone Ship Landing with Payload

- %sql select count (Mission_Outcome) AS Successful_Outcome, (select count (Mission_Outcome) from SPACEXTBL where Mission_Outcome like 'Failure%') AS Failure_Outcome from SPACEXTBL where Mission_Outcome like 'Success%'

```
utcome) from SPACEXTBL where Mission_Outcome like 'Failure%') AS Failure_Outcome  from SPACEXTBL where Mission_Outcome like 'Success%'
```

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

Successful_Outcome	Failure_Outcome
100	1

Boosters Carried Maximum Payload

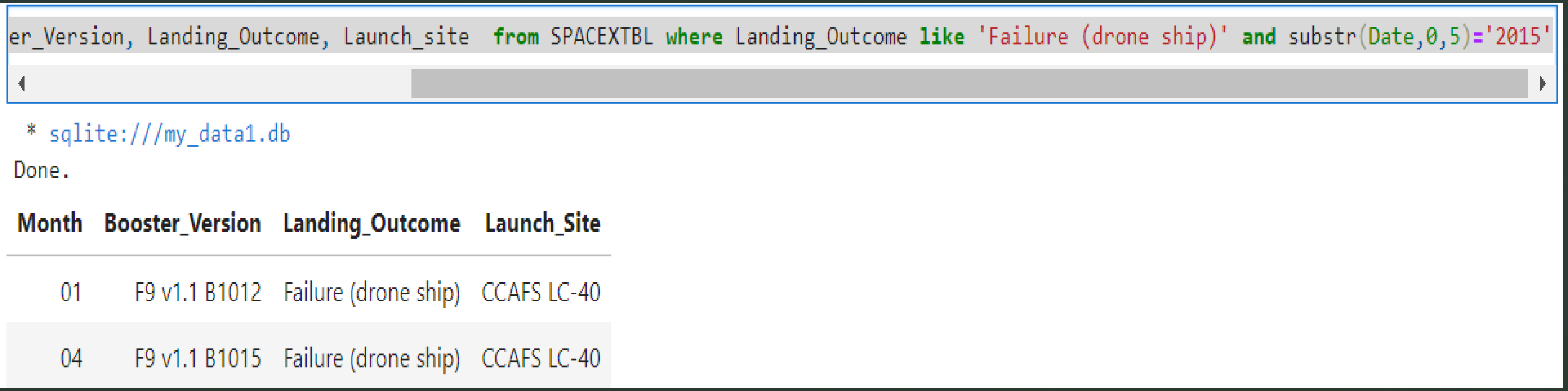
```
%sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ == (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

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 Records

- %sql select substr(Date, 6,2) as Month, Booster_Version, Landing_Outcome, Launch_site from SPACEXTBL where Landing_Outcome like 'Failure (drone ship)' and substr(Date,0,5)='2015'



```
er_Version, Landing_Outcome, Launch_site  from SPACEXTBL where Landing_Outcome like 'Failure (drone ship)' and substr(Date,0,5)='2015'
```

* sqlite:///my_data1.db
Done.

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

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

- %sql select Landing_Outcome, count(Landing_Outcome) from SPACEXTBL where Date between '2010-06-04' and '2017-03-20' group by Landing_Outcome order by count(Landing_Outcome) desc

```
utcome) from SPACEXTBL  where Date between '2010-06-04' and '2017-03-20' group by Landing_Outcome order by count(Landing_Outcome) des
```

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

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

Launch Site Analysis

Location of all Launch Sites

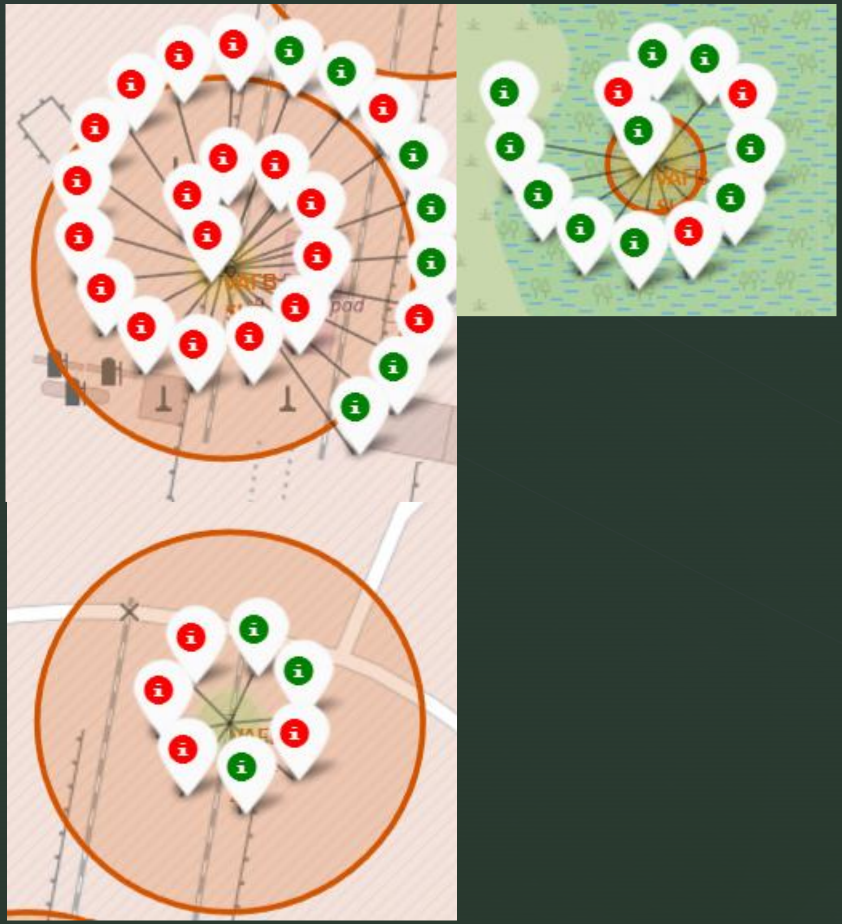
- Launch Sites in the USA

Launch Outcomes

- Green markers for successful Launches
- Red markers for unsuccessful Lauches

- California Launch Site

Launch Outcomes

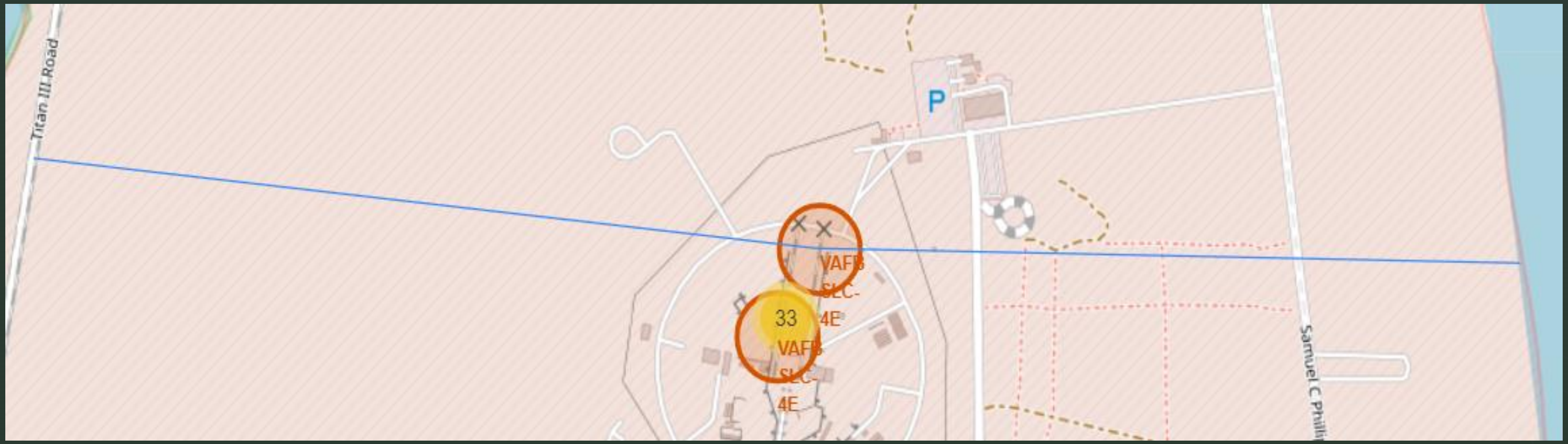


- Miami Launch Sites

Launch Sites Distance to Landmarks

VAFB SLC –4E

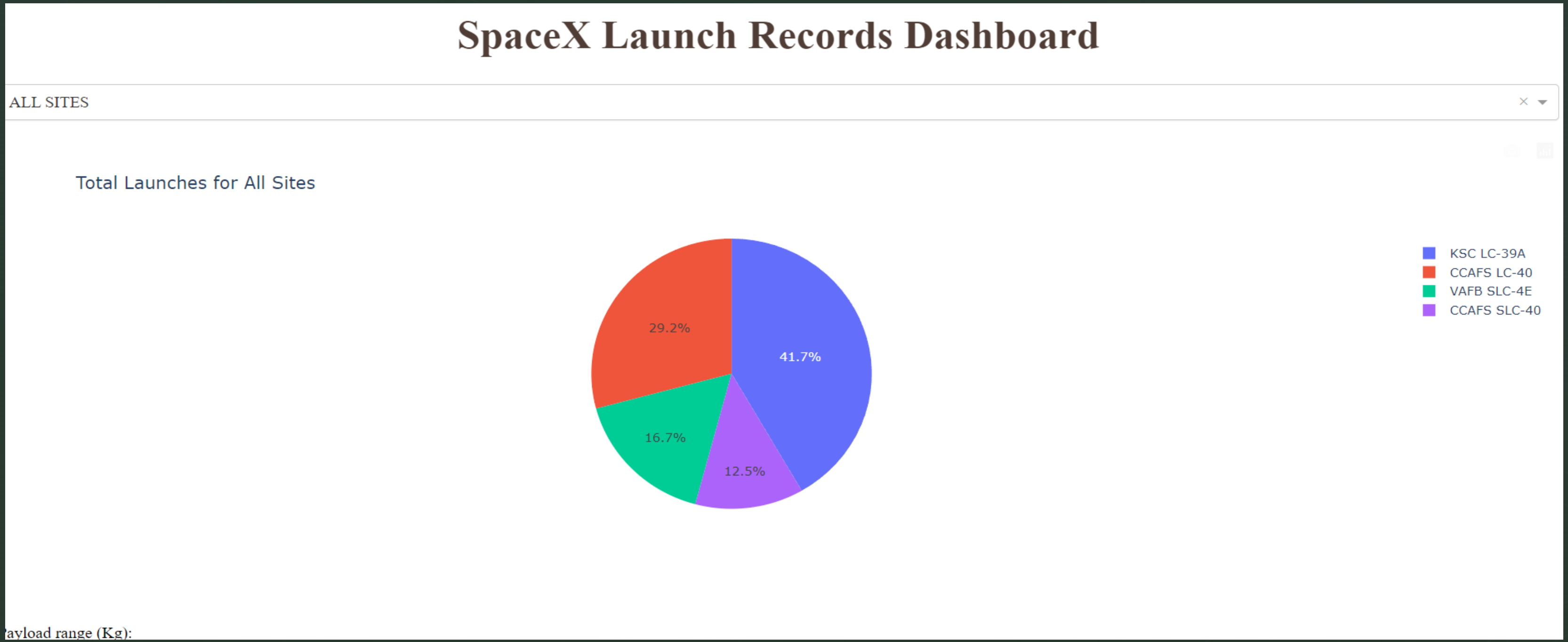
- Distance to nearest coastline
- Distance to nearest railway



Dashboard with Plotly Dash

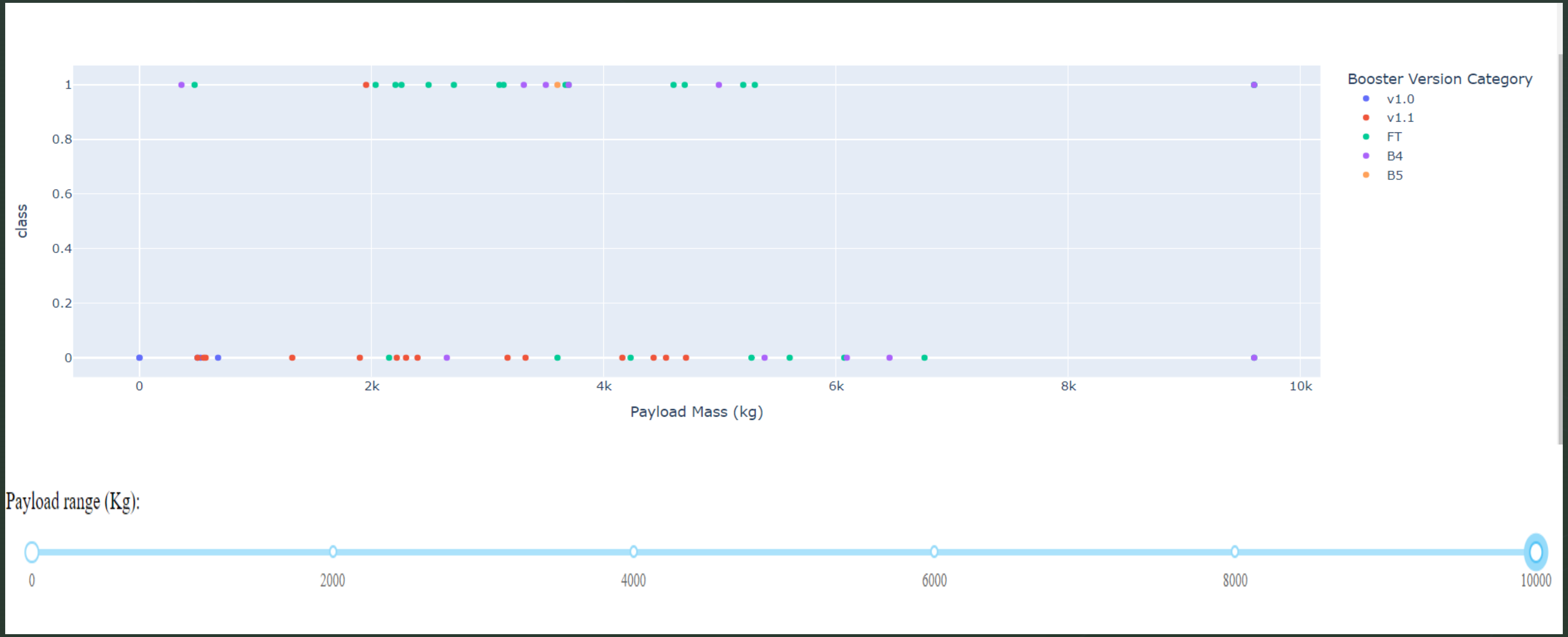
Launch Success by Site

- KSC LC-39 Site has the most successful Launches with 41.7%



Payload Mass and Success

- 1 indicates success and 0 failure
- Payloads between 2000 and 5000kg have the highest success rate



Predictive Analysis

Classification Accuracy

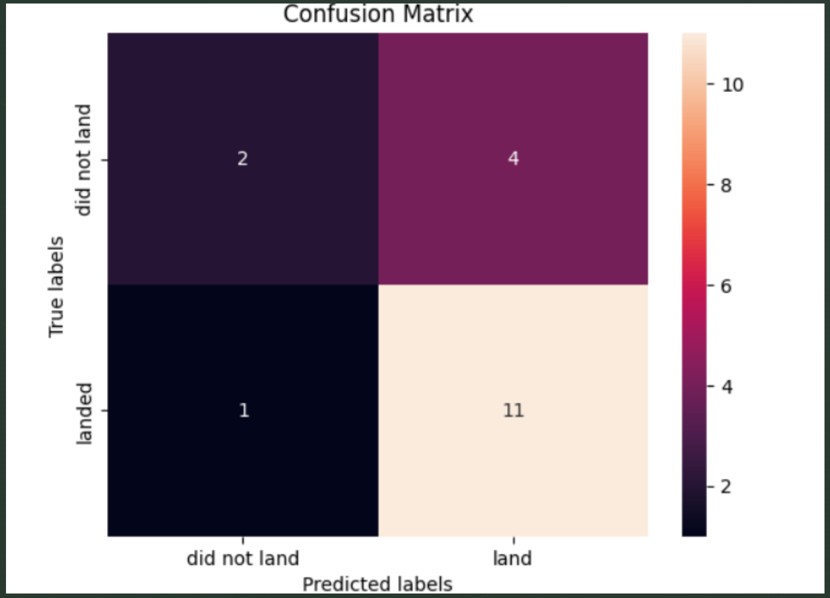
- The best Algorithm is the decision tree with accuracy of 0.8785714285714284

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 6, 'max
lit': 2, 'splitter': 'best'}
accuracy : 0.8785714285714284
```

Confusion Matrix

- The confusion matrix of the decision tree shows the performance of a classification algorithm. The problem here are the 4 false positives, these are unsuccessful landings marked as successful by the classifier



Conclusions

- Launch success increases over time
- Decision Tree Algorithm is the best Machine Learning approach for this data set
- The Launch Site KSC LC-39 have the most successful launches
- Orbits ES-L1, GEO, HEO and SSO have 100% success rate

Titel Lorem Ipsum Dolor

Lorem Ipsum dolor sit amet. Maecenas porttitor congue massa. Proin pharetra nonummy pede. Mauris et orci.

Lorem Ipsum dolor sit amet. Proin pharetra nonummy pede. Mauris et orci.

2017

2018

2019

Lorem Ipsum dolor sit amet. Proin pharetra nonummy pede. Mauris et orci.