IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sahar Merheb
Aug 2023

Outline

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

Executive Summary

- Summary of methodologies:
  - Data collection:
    - API
    - Web Scraping
  - Data wrangling
  - EDA with:
    - Data visualization
    - SQL
  - Building an interactive map with Folium
  - Predicitve analaysis (Classification)
- Summary of all results:
  - Optimum hyperparameters for LR, SVM, Decision Tree and KNN.
  - Best performing method.

Introduction

- Project background and context:
  - SpaceX advertises Falcon 9 rocket at a **competitive** price.
  - The key reason behind this relatively low price is the savings from the **reuse of the first stage**.

- Problems you want to find answers:
  - Is it possible to predict whether the first stage will land successfully?
    - Identify the **variables impacting** the success/failure of a landing.
    - Thus determine the actual price of the launch.

Section 1

Methodology

Methodology

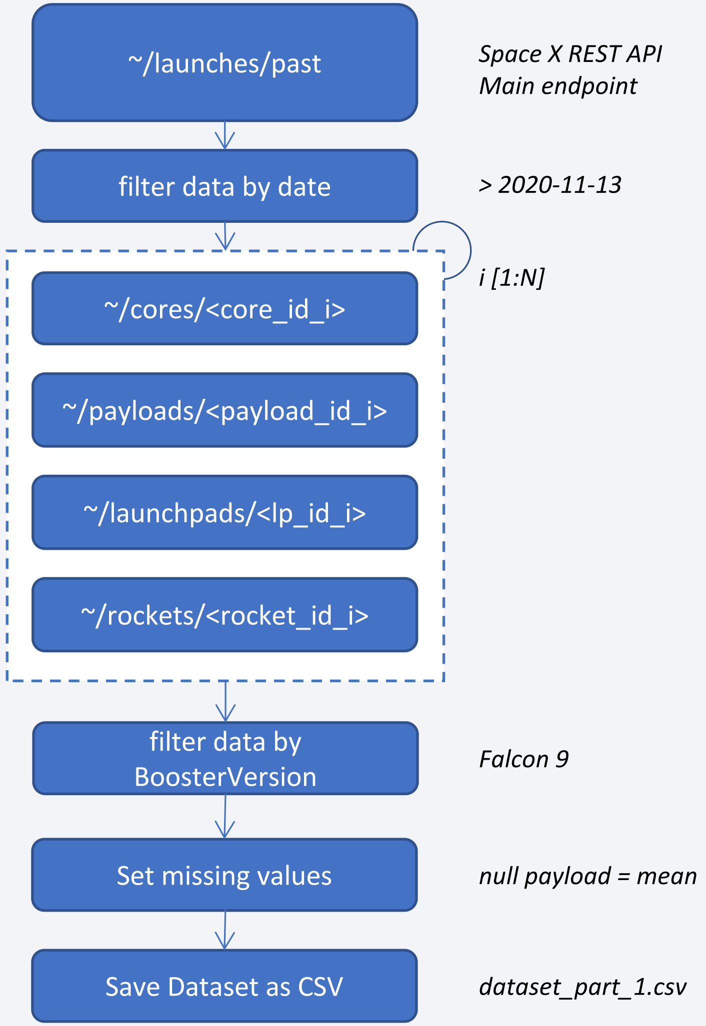
Executive Summary

- Data collection methodology - data sources:
  - SpaceX REST API
  - Wiki page (Web Scraping)
- Perform data wrangling
  - Converted different Landing Outcomes into binary landing-class:
    - 1 for Success, 0 for Failure
- 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 – SpaceX API

1. Fetch Rocket Launch json data from SpaceX API - *main endpoint*
2. Load json data into a dataframe using json_normalize function
3. Opt out launches that took place before Nov. 2020.
4. Fetch reference data, by iterating each record, using reference endpoints:
   - Fetch core-details by core id
   - Fetch payload-details by payload id
   - Fetch launchpads-details by launhpad id
   - Fetch rocket-details by rocket id
5. Merge main and reference data
6. Filter dataframe by Booster version, and keep only Falcon 9 launches
7. Set missing values of Payload Mass to the mean of all payload-mass.
8. Export data to csv file

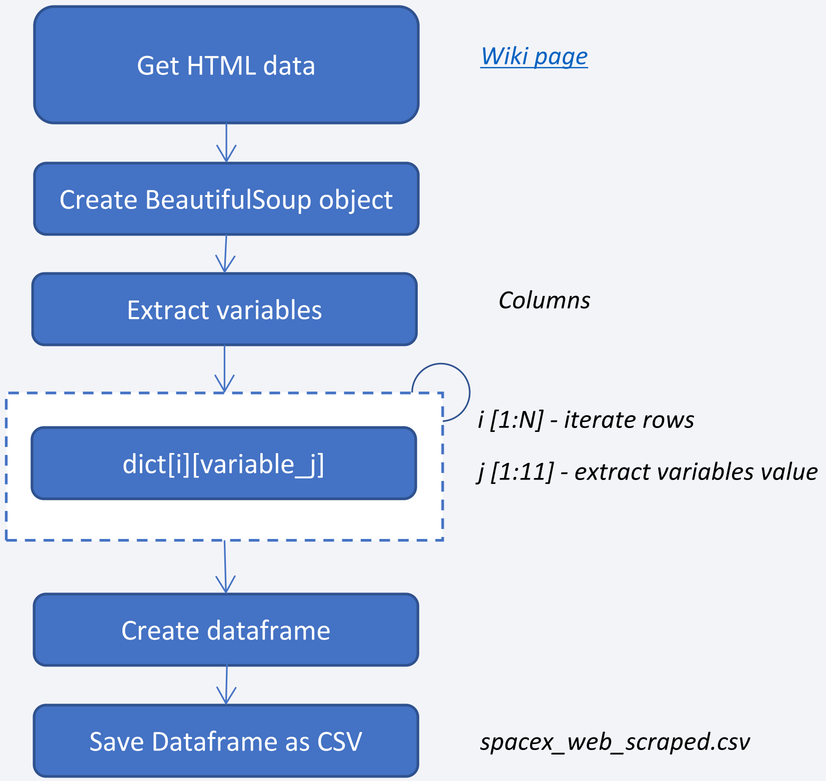
[Check Notebook 1 on GitHub]()



Data Collection – Web Scraping

1. Get Falcon 9 Launch HTML data from a Wiki page
2. Load HTML data into a BeautifulSoup object.
3. Extract variable names, by extracting all columns, from the table header.
4. Load data from relevant html table into a Dictionary, by iterating each row in the html table.
5. Create a dataframe from the dictionary
6. Export dataframe to csv file

[Check Notebook 2 on GitHub]()



Data Wrangling

1. Calculations:
    1. Number of launches/Site.
    2. Number,Occurence/Orbit
    3. Number,Occurence/Outcome/Orbit
2. Created Landing Outcome Label (Class)
    - Convert Outcome column from Categorical to Binary (Numerical)
        - 0 for Failure
        - 1 for Success
3. Calculated Success rate:
    - By calculating the mean of Class's values
4. Export dataframe to csv file (dataset_part_2.csv)

[Check Notebook 3 on GitHub]

EDA with Data Visualization

- Plotted charts:
  - Scatter Plots
    - Flight Number vs. Payload Mass
      - Motivation: Would flight no. and payload mass affect the launch outcome?
      - Observation:
        1. Flight Nb ↗ then probability of first stage to land successfully ↗
        2. Payload Mass ↗ then probability of first stage to land successfully ↘
    - Similarily we tried to visualise whethere there is a relationship between the following set of variables pairs:
      - Flight Number vs. Launch Site
      - Payload Mass vs. Launch Site
      - Payload Mass vs. Orbit

Check Notebook 4 on GitHub

EDA with SQL

- Queries:
  - Displayed the names of the unique launch sites in the space mission.
  - Displayed 5 records where launch sites begin with the string 'CCA' .
  - Calculated the total payload mass carried by boosters launched by NASA (CRS).
  - Calculated average payload mass carried by booster version F9 v1.1.
  - Found the date when the first succesful landing outcome in ground pad was acheived.
  - Listed the names of the boosters which have success in drone ship and have payload mass betwee 4000 and 6000.
  - Calculated the total number of successful and failed missions.
  - Listed the names of boosters which have carried the max payload mass.
  - Listed failed-landing-outcomes on drone ship, by booster version, launch site and month, that took place on 2015.
  - Calculated the count of landing outcomes between 2010-06-04 and 2017-03-20, and sorted them by descending order.

Check Notebook 5 on GitHub

Build an Interactive Map with Folium

- Added Markers for each Launch Site:
    - Highlighted Nasa Johnson Space Center with a blue circle
    - Highlighted Launch Sites with red circles
    - Purpose: To gain insights on the geoposition of Launch sites
        - Nearby, etc..
- Added Markers of Launch Outcome Class:
    - Green for success
    - Red for failure
    - Purpose: Visualize wich launch sites have a higher success rate.
- Added distance between launch site and proximities:
    - Nearest coastline, railway, etc.

[Check Notebook 6 on GitHub]

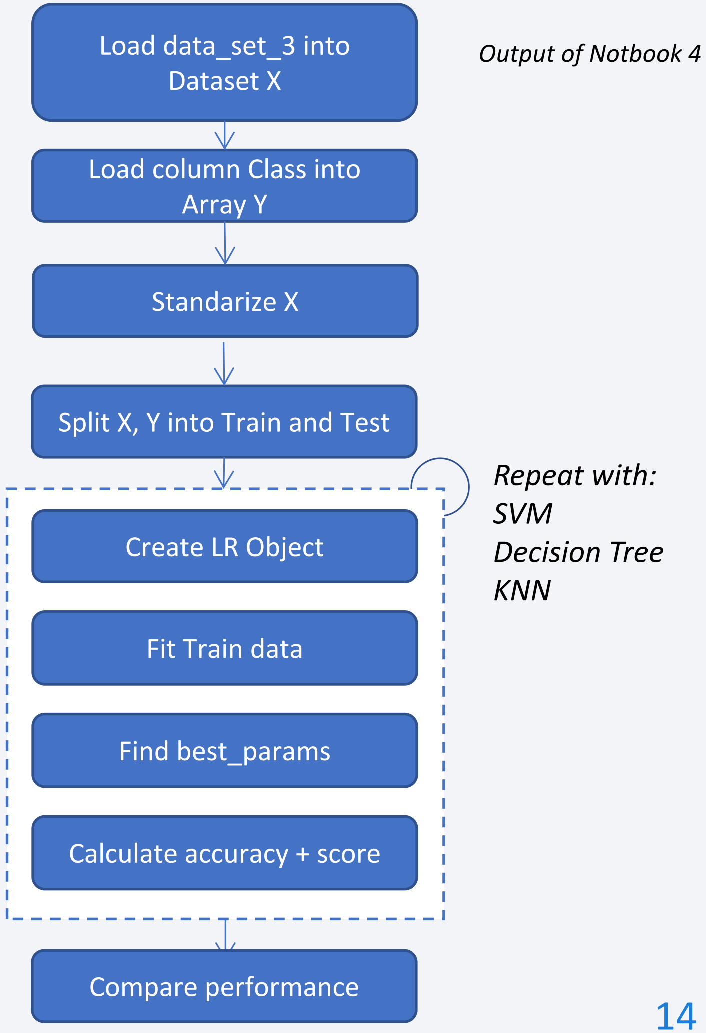
Build a Dashboard with Plotly Dash

- Dropdown list of Launch Sites
  - Enable user to select all or specific launch site
- Pie chart showcasing successful launches
  - Enable user to check the percentage of successful/unsuccessful launches
- Slider of payload mass range
  - Enable user to specify the range of payload mass
- Scatter plot visualizing Payload mass vs Success rate per Booster version
  - Enable user to visualize the correlation between payload mass and launch outcome

Predictive Analysis (Classification)

1. Independant Variables - X:
   - Load the data into dataframe X
2. Dependant variable - Class ~ Y:
   - Create an array, Y, out of Class column using Numpy
3. Standarize the data in X using standard-scalar transformation
4. Split the data in X and Y into training and test data.
   - Split using train_test_split method
   - Test data to be used as validation
5. Analysis:
   - **Logistic Regression**
     1. Create a LR object
     2. Fit the training set
     3. Find the hyperparameters using best_params
     4. Calculate the accuracy on test data using score method
     5. Plot confusion matrix
   - Repeat the same flow for ***Support Vector Machine***, ***Decision Tree*** and ***K nearest neighbors***

[Check Notebook 7 on GitHub]()



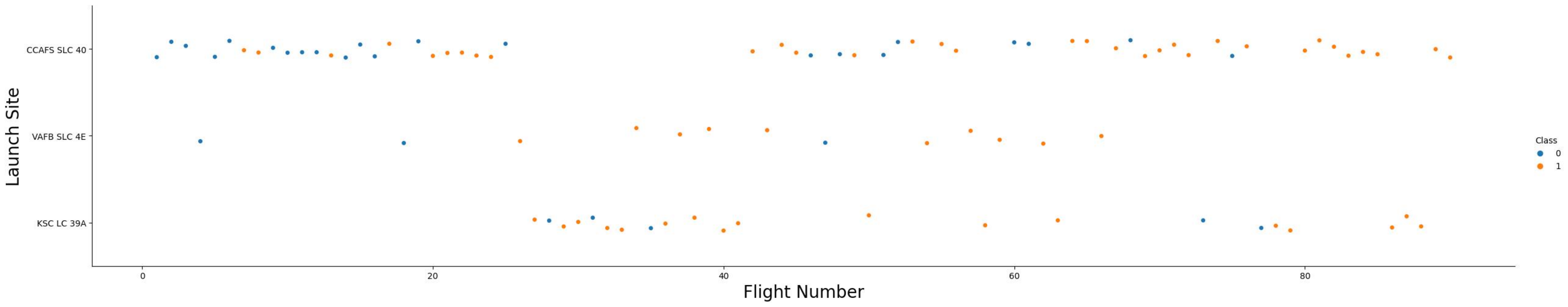
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

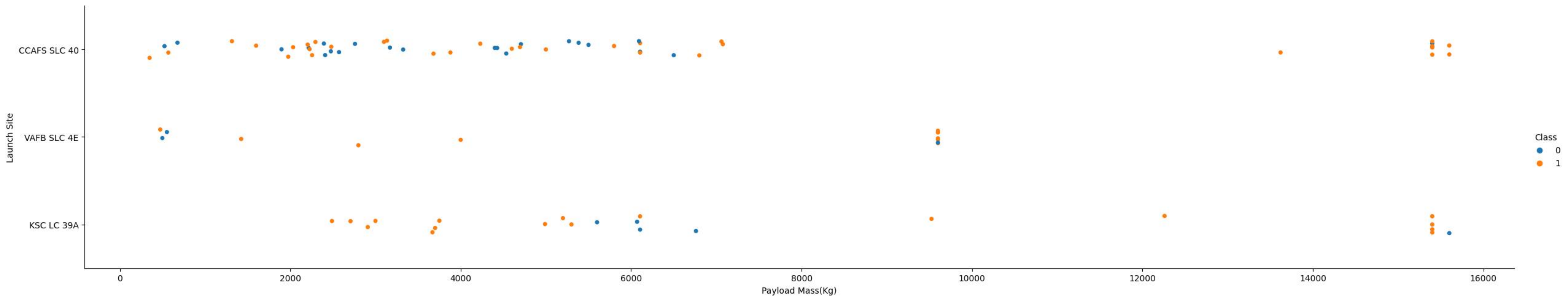
Insights drawn from EDA

Flight Number vs. Launch Site



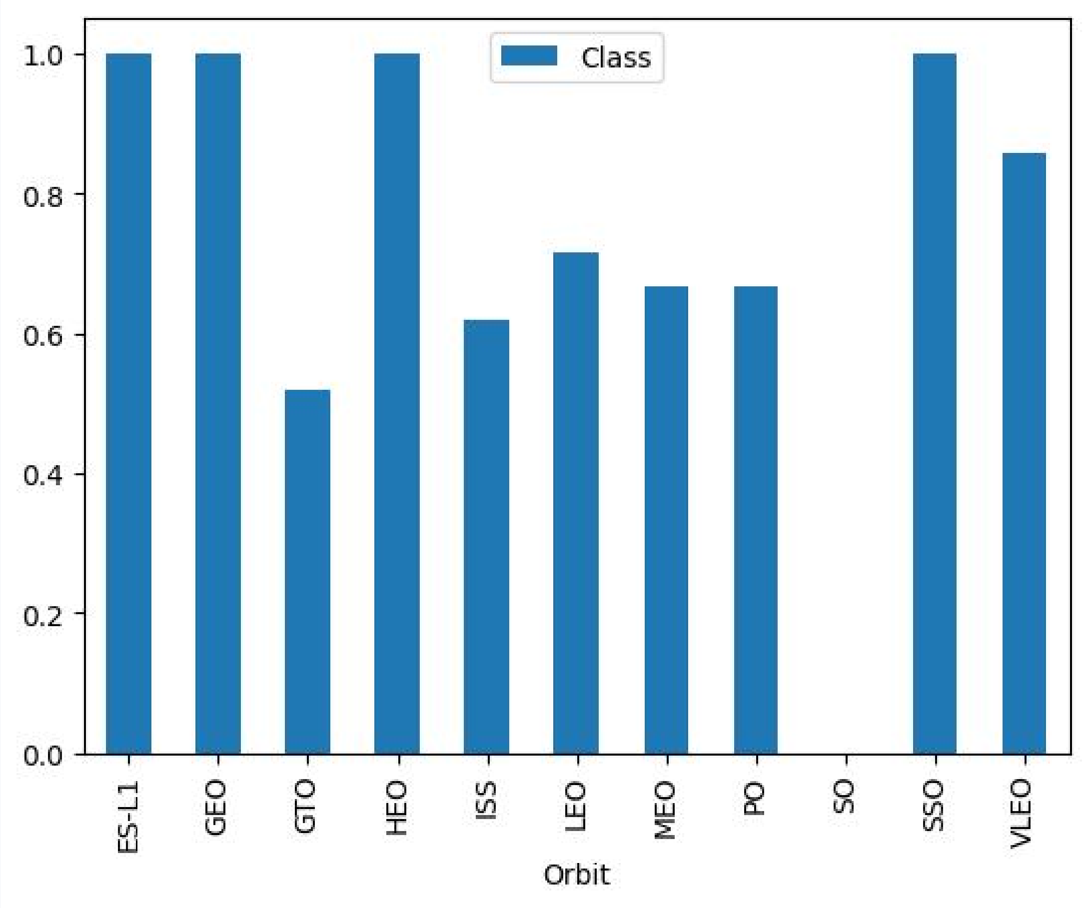
- **Observations**:
  - Flight Number , Success Rate ; This is noticed for all launch sites.
  - CCAFS SLC 40 has the majority of launches
  - KSC LC 39A and VAFB SLC 4E have a higher succees rate comparing to the third launch site

Payload vs. Launch Site



- **Observations**:
  - Payload mass ↗, Success Rate ↗.
  - For KSC LC 39A, and for payload < 5,750 kg -> ALL launches are successful
  - For VAFB SLC 4E, there are no rockets launched for heavy payloads (> 10k)

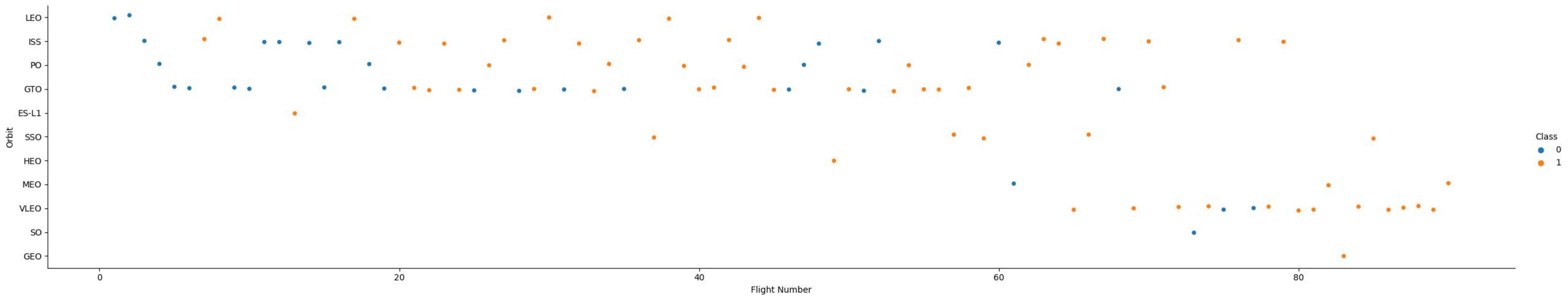
Success Rate by Orbit



Observations:

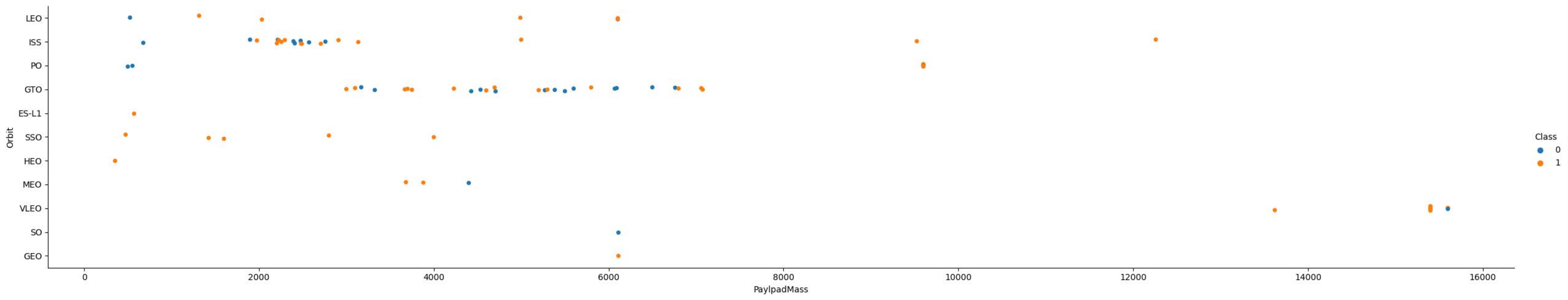
- 0 Sucess Rate: **SO**
- 100% Success Rate: **ES-L1, GEO, HEO and SSO**

Flight Number vs. Orbit Type



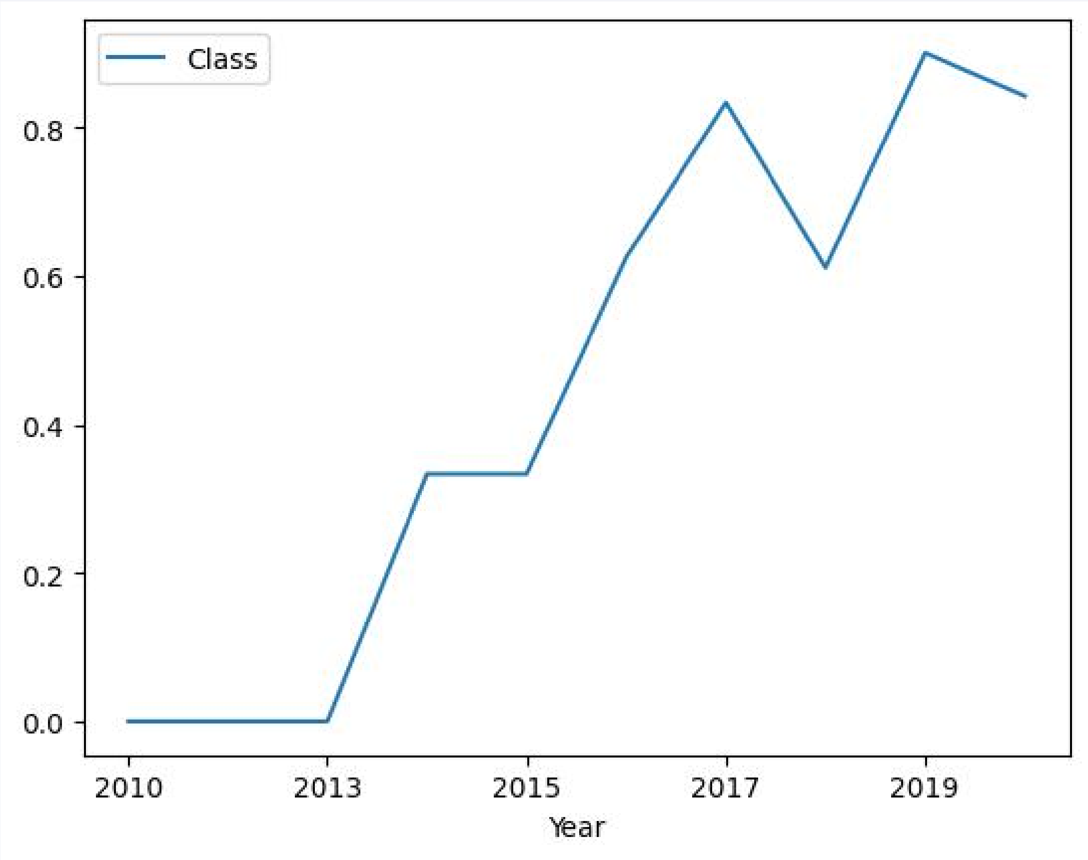
- **Observations**:
  - Flight Number , Success Rate ; This pattern applies for the majority of orbit types
    - This is noticeable for **LEO**
    - However this doesnt apply for **GTO**

Payload vs. Orbit Type



- **Observations**:
  - Payload mass , Success Rate , this is noticeable mainly for Polar, LEO and ISS
  - This is not applicable for GTO, no correlation is noticed.

Launch Success over Time



- Observations:
  - Overall, success rate is improving over time
  - There was a significant increase between 2013-2014 and between 2015-2017.
  - There was a drop in 2018.

All Launch Site Names

```
cur.execute("select distinct(Launch_Site) from  SPACEXTBL")
sites = cur.fetchall()
for site in sites:
    print(site[0])
```

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

Launch Site Names Begin with 'CCA'

```python
cur.execute("select * from  SPACEXTBL where Launch_Site like 'CCA%' limit 5")
records = cur.fetchall()
for record in records:
    print(record)
```

Python

```
('2010-04-06', '18:45:00', 'F9 v1.0  B0003', 'CCAFS LC-40', 'Dragon Spacecraft Qualification Unit', 0, 'L
('2010-08-12', '15:43:00', 'F9 v1.0  B0004', 'CCAFS LC-40', 'Dragon demo flight C1, two CubeSats, barrel
('2012-05-22', '07:44:00', 'F9 v1.0  B0005', 'CCAFS LC-40', 'Dragon demo flight C2', 525, 'LEO (ISS)', 'N
('2012-08-10', '00:35:00', 'F9 v1.0  B0006', 'CCAFS LC-40', 'SpaceX CRS-1', 500, 'LEO (ISS)', 'NASA (CRS)
('2013-01-03', '15:10:00', 'F9 v1.0  B0007', 'CCAFS LC-40', 'SpaceX CRS-2', 677, 'LEO (ISS)', 'NASA (CRS)
```

Total Payload Mass

- Total payload mass carried by boosters launched by NASA (CRS): **45,596**

```python
cur.execute("select SUM(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = 'NASA (CRS)'")
total = cur.fetchall()[0][0]
print ('the total payload mass carried by boosters launched by NASA (CRS):', total)
```

Python

```
the total payload mass carried by boosters launched by NASA (CRS): 45596
```

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1: **2,928.4**

```python
cur.execute("select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version is 'F9 v1.1'")
avg = cur.fetchone()[0]
print ('the average payload mass carried by booster version F9 v1.1:', avg)
```

Python

```
the average payload mass carried by booster version F9 v1.1: 2928.4
```

First Successful Ground Landing Date

- Date of the first successful landing outcome on ground pad: **2015-12-22**

```python
cur.execute("Select Min(Date) from SPACEXTBL Where Landing_Outcome is 'Success (ground pad)'")
first_success_data = cur.fetchone()[0]
print('the first successful landing in ground pad was on:', first_success_data)
```

Python

```
the first successful landing in ground pad was on: 2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- Boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000:
  - SKY Perfect JSAT Group
  - SES
  - SES EchoStar

```python
cur.execute("Select distinct(Customer) from SPACEXTBL where Landing_Outcome is 'Success (drone ship)'
boosters = cur.fetchall()
for b in boosters:
    print(b[0])
```

Python

```
SKY Perfect JSAT Group
SES
SES EchoStar
```

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes
    - **100 Success**
    - **1 Failure**

```
cur.execute("Select (Select count(*) from SPACEXTBL where Mission_Outcome like 'Success%') as Success,\n
            (Select count(*) from SPACEXTBL where Mission_Outcome like 'Failure%') as Failure")
mission_outcomes = cur.fetchone()
print('Success: ',mission_outcomes[0])
print('Failure: ',mission_outcomes[1])
```

```
Success:  100
Failure:  1
```

Boosters Carried Maximum Payload

```
cur.execute("Select distinct(Booster_Version) from SPACEXTBL where \n
            PAYLOAD_MASS__KG_=(Select max(PAYLOAD_MASS__KG_) from SPACEXTBL)")
booster_versions = cur.fetchall()
for i,v in enumerate(booster_versions):
    print(i+1,':',v[0])
```

```
1 : F9 B5 B1048.4
2 : F9 B5 B1049.4
3 : F9 B5 B1051.3
4 : F9 B5 B1056.4
5 : F9 B5 B1048.5
6 : F9 B5 B1051.4
7 : F9 B5 B1049.5
8 : F9 B5 B1060.2
9 : F9 B5 B1058.3
10 : F9 B5 B1051.6
11 : F9 B5 B1060.3
12 : F9 B5 B1049.7
```

2015 Launch Records

- Failed landing in drone ship + booster versions + launch site names + month in 2015

Month	Landing Outcome	Booster Version	Launch Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

```
cur.execute("Select substr(Date,6,2),Landing_Outcome,Booster_Version,Launch_Site from SPACEXTBL where Landing_Outcome is 'Failure (drone ship)' and substr(date,1,4)='2015'")
records = cur.fetchall()
for r in records:
    print(r[0],r[1], r[2], r[3])
```
✓ 0.0s

```
10 Failure (drone ship) F9 v1.1 B1012 CCAFS LC-40
04 Failure (drone ship) F9 v1.1 B1015 CCAFS LC-40
```

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

```
cur.execute("Select Landing_Outcome,count(*) from SPACEXTBL where Date between '2010-06-04' and '2017-03-20' group by Landing_Outcome order by count(*) desc")
outcomes = cur.fetchall()
for o in outcomes:
    print(o[0],o[1])
```

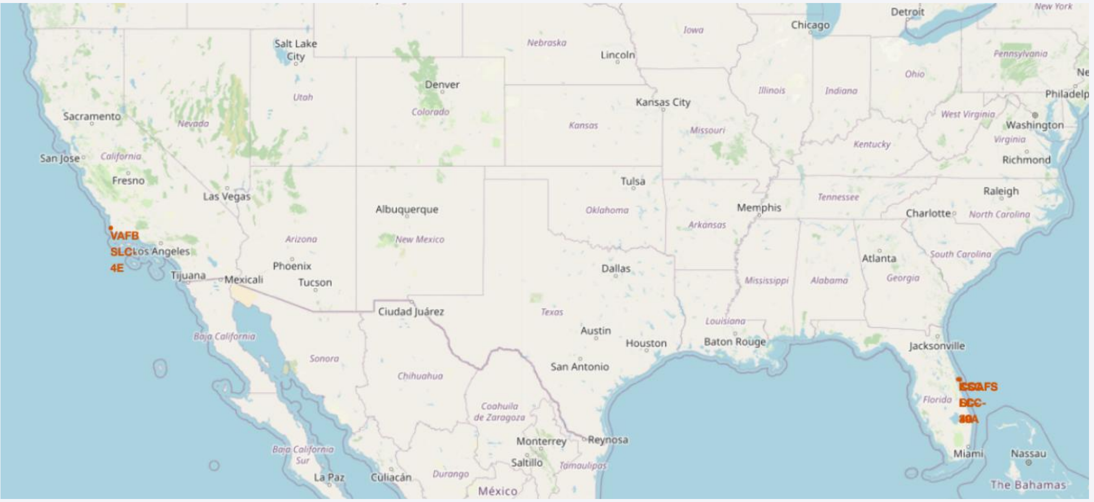
```
No attempt 10
Success (ground pad) 5
Success (drone ship) 5
Failure (drone ship) 5
Controlled (ocean) 3
Uncontrolled (ocean) 2
Precluded (drone ship) 1
Failure (parachute) 1
```

Section 3

Launch Sites Proximities Analysis

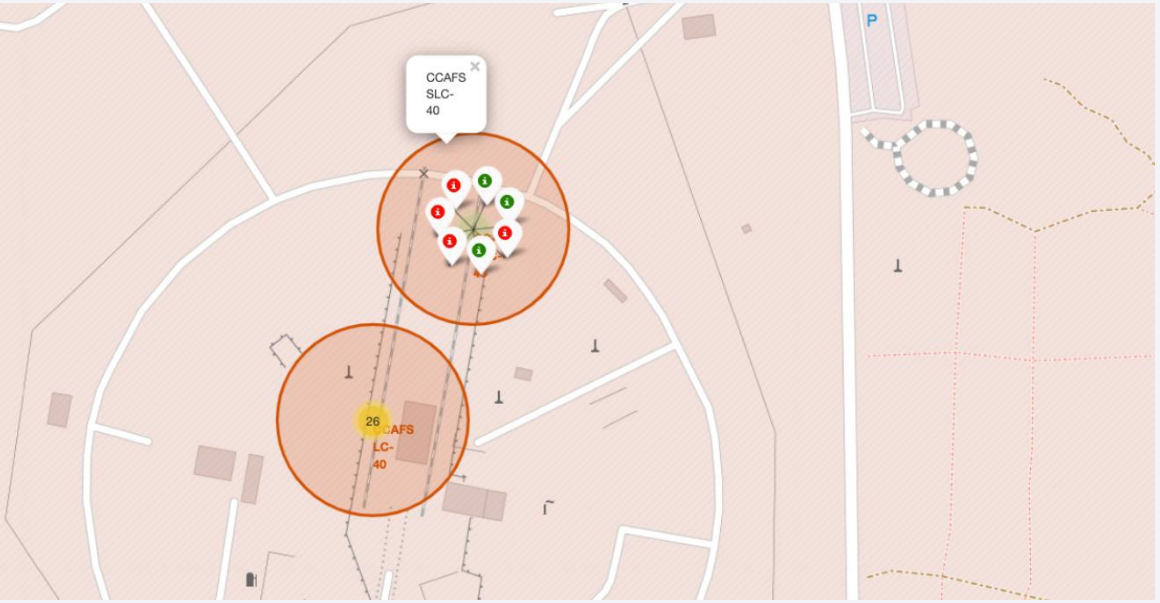
Launch Sites - Geo Positions

- Observations:
  - All launch sites in proximity to the Equator line
    - This is expected, since it’s easier to launch rockets as we approach the equator
  - All launch sites in very close proximity to the coast.



Launch Sites - Geo Positions

- Observations:
  - For each Launch Site, each green marker symbolize a successful launch, and red for failed ones.
  - For CCAFS SLC-40, we notice that we have:
    - 3 successful out of 7 (<50% success rate)



Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

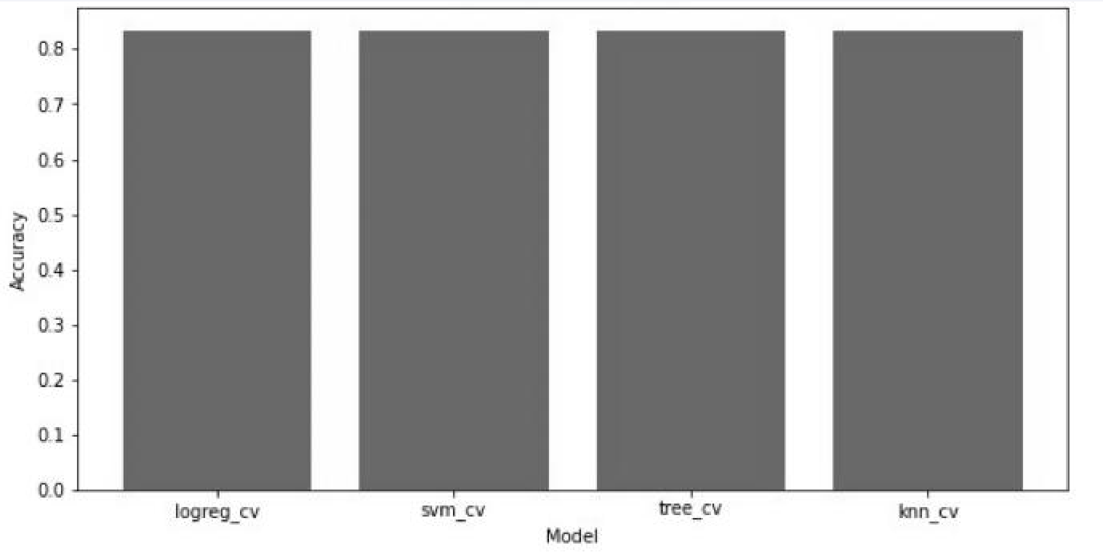
Section 5

Predictive Analysis (Classification)

Classification Accuracy

Observation:

All models performed similarily -> same accuracy

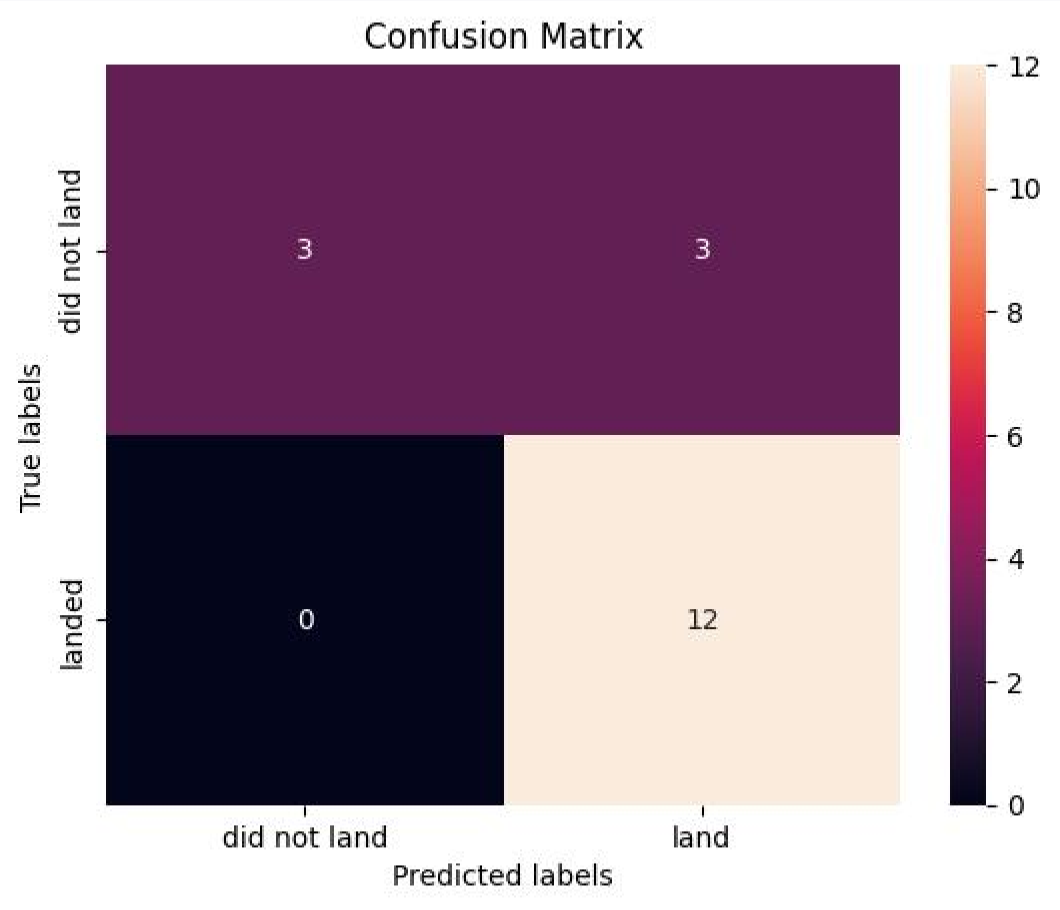


Confusion Matrix

Observation:

- All models shared the same confusion matrix.

- What would require attention is the False Positive: **3**
    - *(3 out of 18 is not negligeabale)*



Conclusions

- **Lessons learned**: Success rate is improving over time, this is regardless of any variable, this means that Space X is leveraging the lessons learned from failed launches.
- **Equator/Coast**: The choice of the location site is definetly not random
  - the closer to the equator/coast the better.
- **Payload Mass**: the higher the mass the higher the success rate.
- **Best model**: we couldnt conclude on what would be the best model, with the dataset we had for this study, all models have performed similarily
  - Perhaps with more data in hand, we will have better clarity

Appendix

- Please refer to the notebooks links provided in relevant slides

Thank you!