IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sedri OĞUR
2023-01-21

Outline

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

Executive Summary

- Used methodologies
  - Data Collection from SpaceX API
  - Clean and Prepare Data
  - EDA with SQL, Folium and Interactive DashBoard
- Results
  - Decision Tree is the best method for using prediction of success landing of Falcon 9.

Introduction

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. Therefore if we can determine if the first stage will land, we can determine the cost of a launch.

In this project we will predict if the Falcon 9 first stage will land successfully. For these prediction ***we will try to find best machine-learning algorithm.***

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - These SpaceX APIs are used to collect data
    - https://api.spacexdata.com/v4/launchpads
    - https://api.spacexdata.com/v4/payloads
- Perform data wrangling
  - Missing values replace with mean() value of each column.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data sets collecting in two ways
  - Using SpaceX APIs
    - https://api.spacexdata.com/v4/launches/past
  - Web Scrabing
    - Wikipedia website used:
      - https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

Data Collection – SpaceX API

```
[6]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
[7]: response = requests.get(spacex_url)
```

```
[11]: # Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

```
[13]: # Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Github:
https://github.com/franz-ogur/SpaceX-Falcon-9-Landing-Prediction/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

```
[4]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
[5]: # use requests.get() method with the provided static_url
     # assign the response to a object
     response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

```
[11]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
      soup = BeautifulSoup(response.content)
```

```
[15]: # Use the find_all function in the BeautifulSoup object, with element type `table`
      # Assign the result to a list called `html_tables`
      html_tables = soup.find_all('table')
```

Github:
https://github.com/franz-ogur/SpaceX-Falcon-9-Landing-Prediction/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

Data set which is collected from APIs had some *null* values (`PayLoadMass column`). The null values replaces with *mean()* value of relative column.

```
[34]: # Calculate the mean value of PayloadMass column
      mean_payLoads = data_falcon9['PayloadMass'].mean()
      data_falcon9['PayloadMass'].replace(np.nan, mean_payLoads, inplace=True)
      # Replace the np.nan values with its mean value
```

Github:
https://github.com/franz-ogur/SpaceX-Falcon-9-Landing-Prediction/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

EDA with Data Visualization

- The following charts has been created to show differentials betwwen success rate and year or orbit.
- Charts has been created:
  - Relationship between Flight Number and Launch Site
  - Relationship between Payload and Launch Site
  - Relationship between success rate of each orbit type
  - Relationship between FlightNumber and Orbit type
  - Relationship between Payload and Orbit type
  - The launch success yearly trend

Github:
https://github.com/franz-ogur/SpaceX-Falcon-9-Landing-Prediction/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Here are some SQL queries
    - SELECT DISTINCT Launch_Site FROM SPACEXTABLE
    - SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
    - SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)'
    - SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1'
    - SELECT MIN(DATE) FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)'
    - SELECT DISTINCT Landing_Outcome FROM SPACEXTABLE
    - SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
    - SELECT Mission_Outcome, COUNT(*) FROM SPACEXTABLE GROUP BY Mission_Outcome
    - SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
    - SELECT substr(Date, 6,2) as month, landing_Outcome, Booster_Version, launch_site FROM SPACEXTABLE WHERE substr(Date,0,5)='2015'
    - SELECT Landing_Outcome, COUNT(*) FROM SPACEXTABLE WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY COUNT(*) DESC

Github:
https://github.com/franz-ogur/SpaceX-Falcon-9-Landing-Prediction/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Markes and circles are used to launch sites and success flights. This signs are used to show that where are launch sites located as geo-strategies. It can be seen that launch sites are near by coast sites.

Github:
https://github.com/franz-ogur/SpaceX-Falcon-9-Landing-Prediction/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Interactions
  - Dropdown for sites
  - RangeSlider for payload range
- Plots/Graphs
  - Pie Chart for launch group by sites
  - Scatter chart for Payload mass by class and color for booster version

Github:
https://github.com/franz-ogur/SpaceX-Falcon-9-Landing-Prediction/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

1. X dataset is standardized with StandardScaler
2. Y dataset is created from ‘Class’ column
3. Train dataset is created with parameters as test_size = 0.2 and random_state = 2
4. Four different ML model (KNeighbors, SVM, Classification Trees and Logistic Regression) is trained with GridSearchCV by parameter cv = 10
5. best_score_ result is compared for all model to find the best model

Github:
https://github.com/franz-ogur/SpaceX-Falcon-9-Landing-Prediction/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

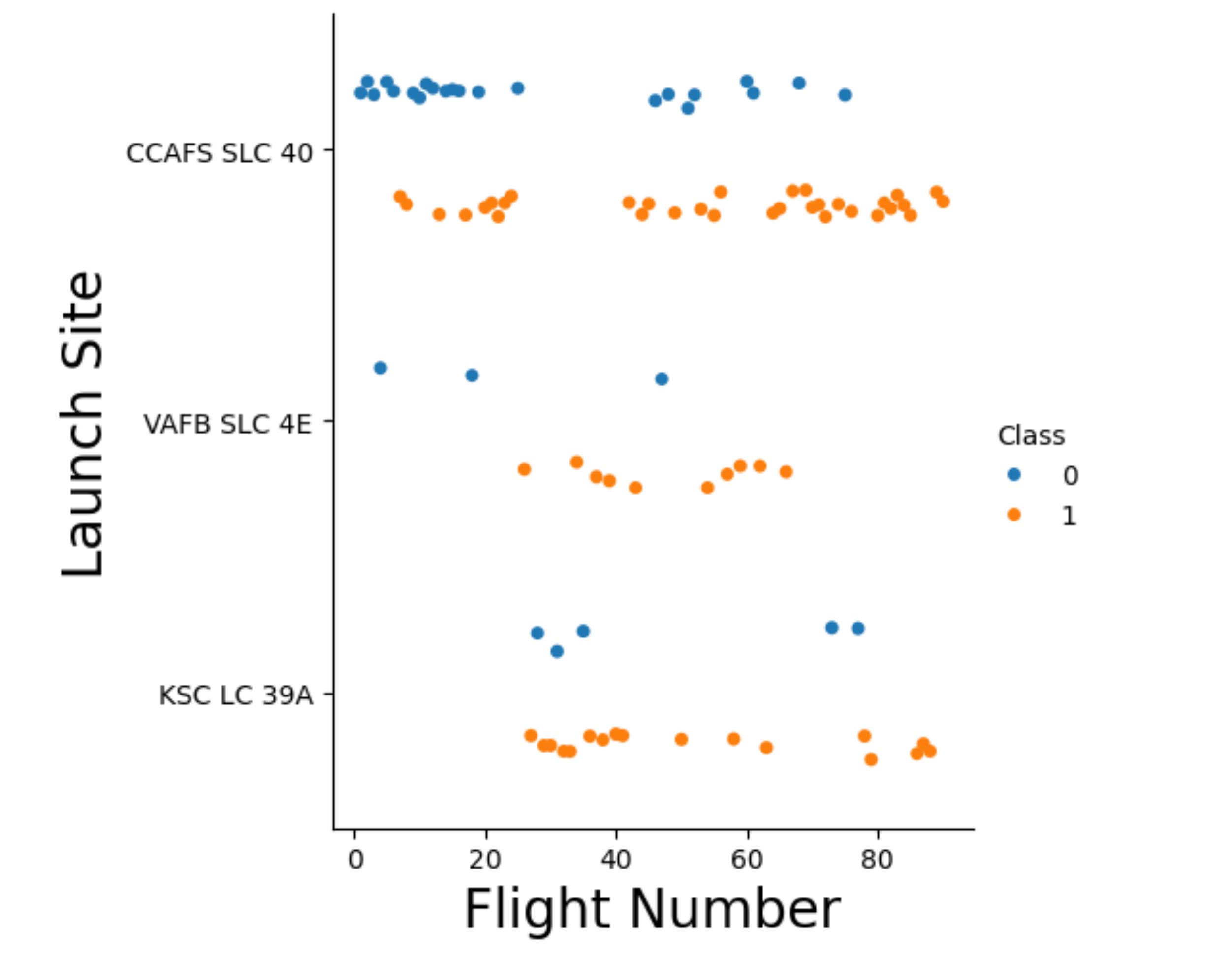
Results

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

Section 2

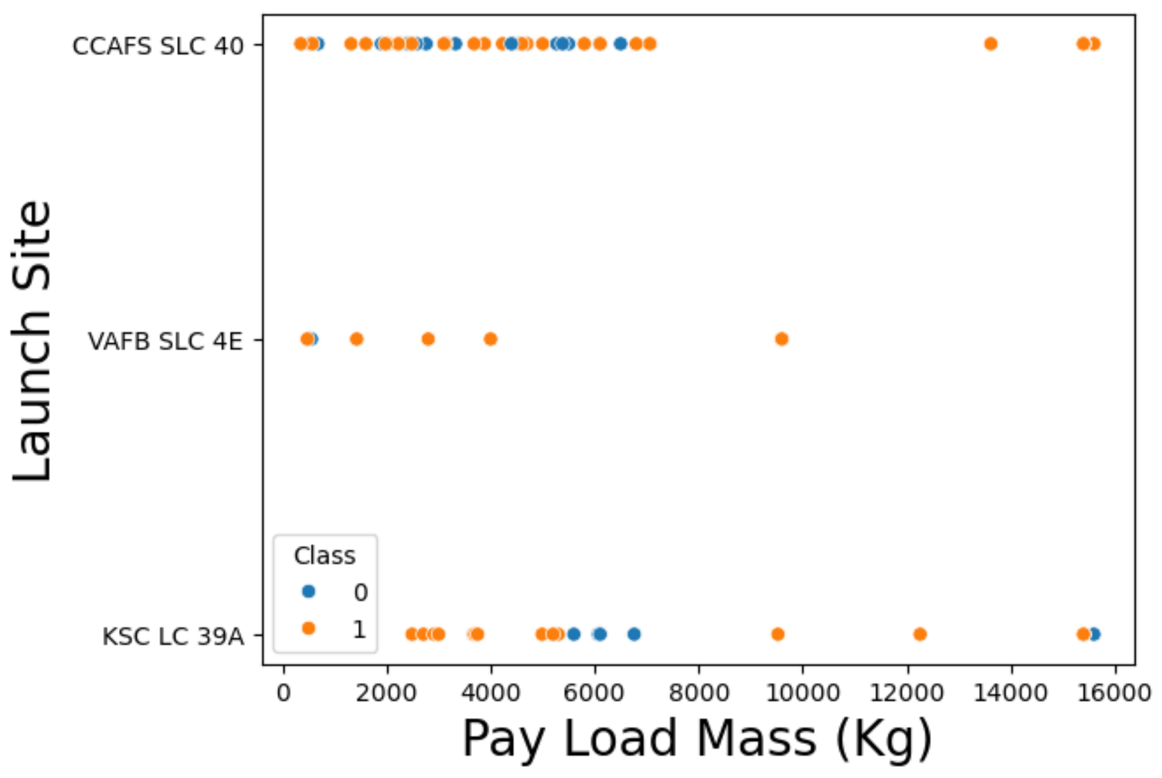
Insights drawn from EDA

Flight Number vs. Launch Site



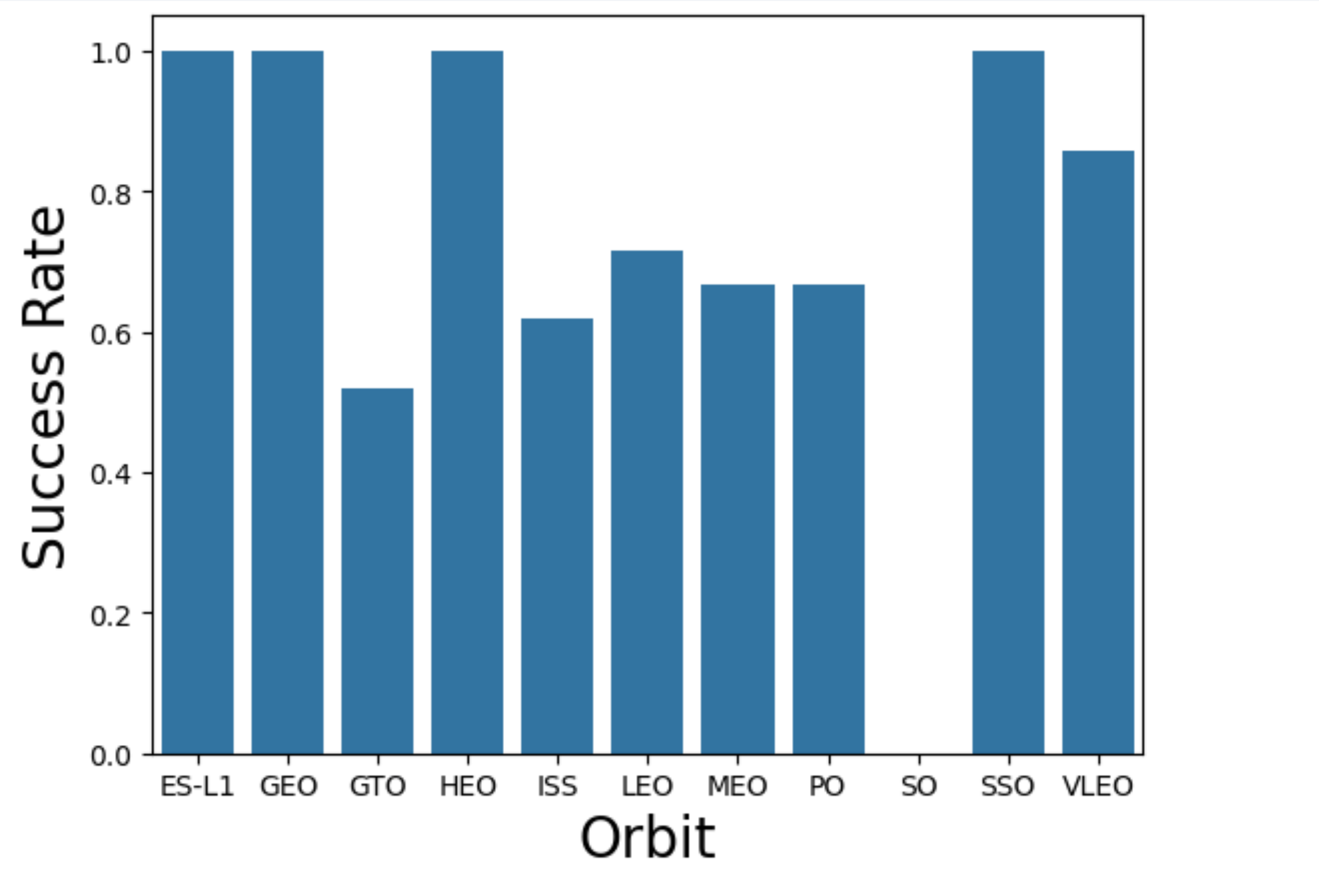
CCAFS LC-40 is the most used site at last and it has more success rate.

Payload vs. Launch Site



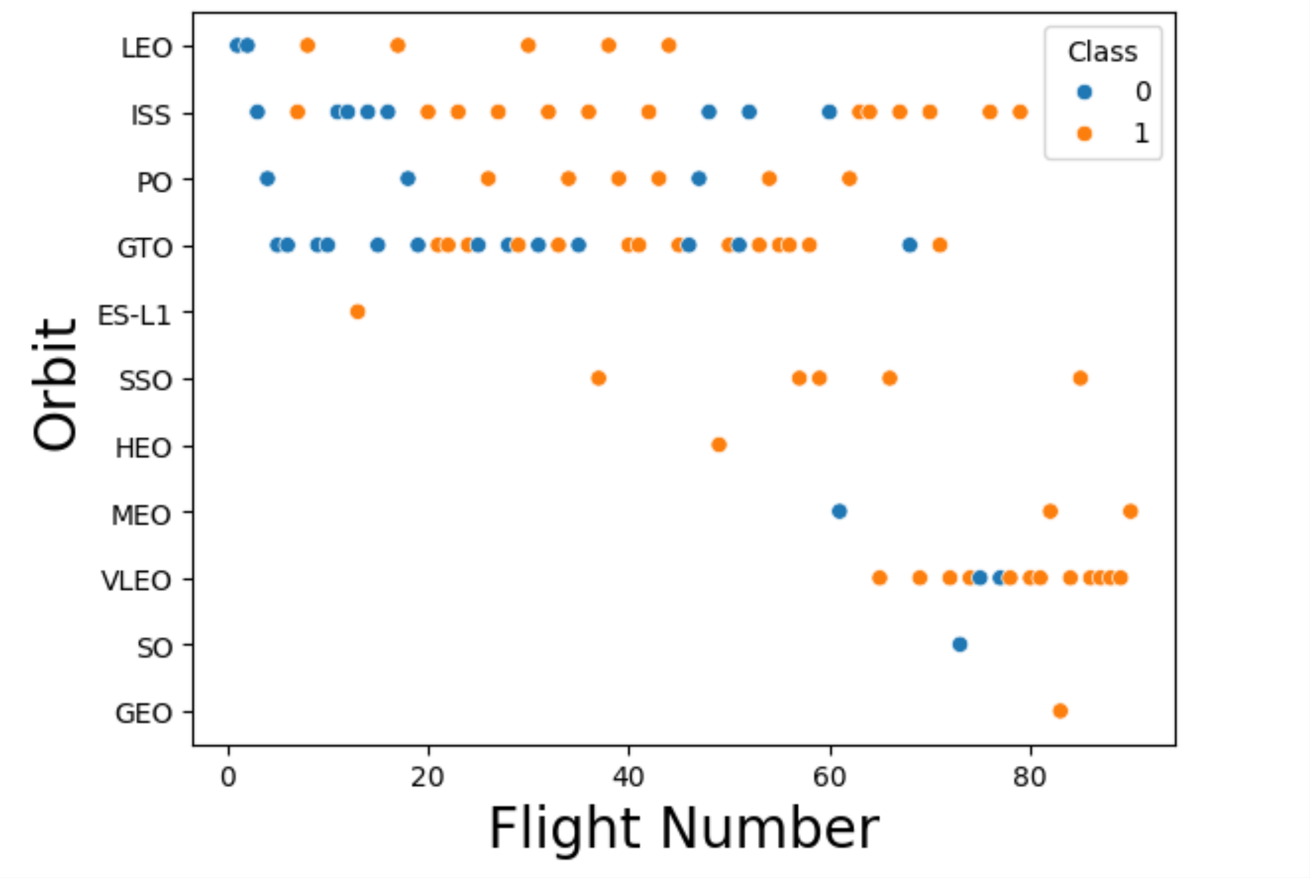
- VAFB-SLC launchsite there are no rockets launched for heavy payload mass(greater than 10000)
- KSC LC-39A is most used site for launches which payload mass more than 10000

Success Rate vs. Orbit Type



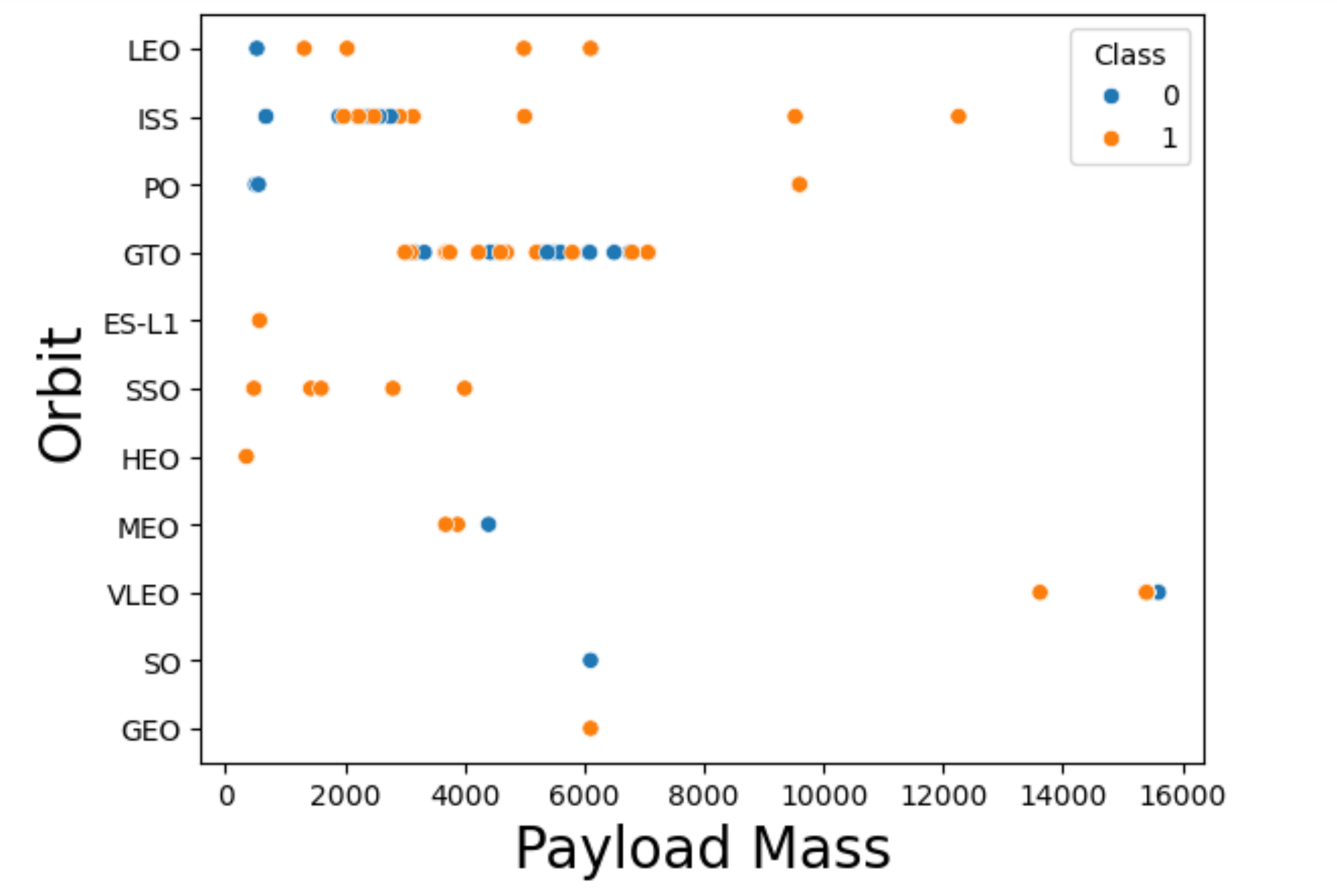
- Launches to ES-L1, GEO, HEO and SSO orbits have 100% success rate.
- Launches to GTO orbit have least success rate.

Flight Number vs. Orbit Type



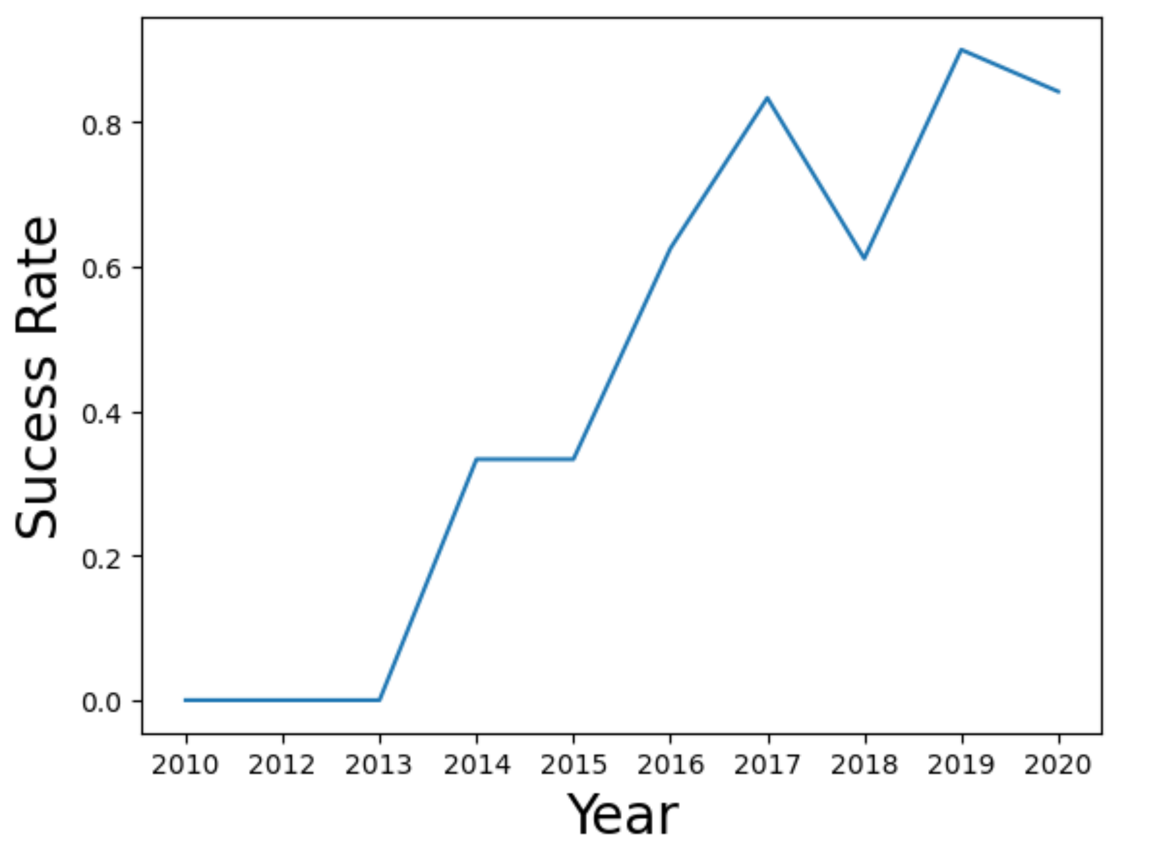
- The LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

Launch Success Yearly Trend



- The sucess rate since 2013 kept increasing till 2020

All Launch Site Names

There are four different Launch Site which are used in all launches.

```
[11]: %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE

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

[11]:

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

Launch Site Names Begin with 'CCA'

- Below you can see five records for launches from the sites which are named with CCA

```
[14]: %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5

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

[14]:

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

- Below you can see the total payload carried by boosters from NASA

```
[15]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)'

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

[15]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Below you can see the average payload mass carried by booster version F9 v1.1

```
[17]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1'

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

[17]:

SUM(PAYLOAD_MASS__KG_)
14642

First Successful Ground Landing Date

- Below you can see the dates of the first successful landing outcome on ground pad

```
[19]: %sql SELECT MIN(DATE) FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)'

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

[19]:

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Below you can see the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
[21]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000

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

[21]:

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

Total Number of Successful and Failure Mission Outcomes

- Below you can see the total number of successful and failure mission outcomes

```
[12]: %sql SELECT TRIM(Mission_Outcome), COUNT(*) FROM SPACEXTABLE GROUP BY TRIM(Mission_Outcome)

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

[12]:

TRIM(Mission_Outcome)	COUNT(*)
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Below you can see the names of the booster which have carried the maximum payload mass

```
[23]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE )

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

[23]:

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

- Below you can see the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

[26]:
```
%%sql SELECT substr(Date, 6,2) as month, landing_Outcome, Booster_Version, launch_site
FROM SPACEXTABLE WHERE substr(Date,0,5)='2015'
```

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

[26]:

month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
02	Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40
03	No attempt	F9 v1.1 B1014	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
04	No attempt	F9 v1.1 B1016	CCAFS LC-40
06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40
12	Success (ground pad)	F9 FT B1019	CCAFS LC-40

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

- Below you can see 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

```
%%sql SELECT Landing_Outcome, COUNT(*) FROM SPACEXTABLE
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome ORDER BY COUNT(*) DESC
```

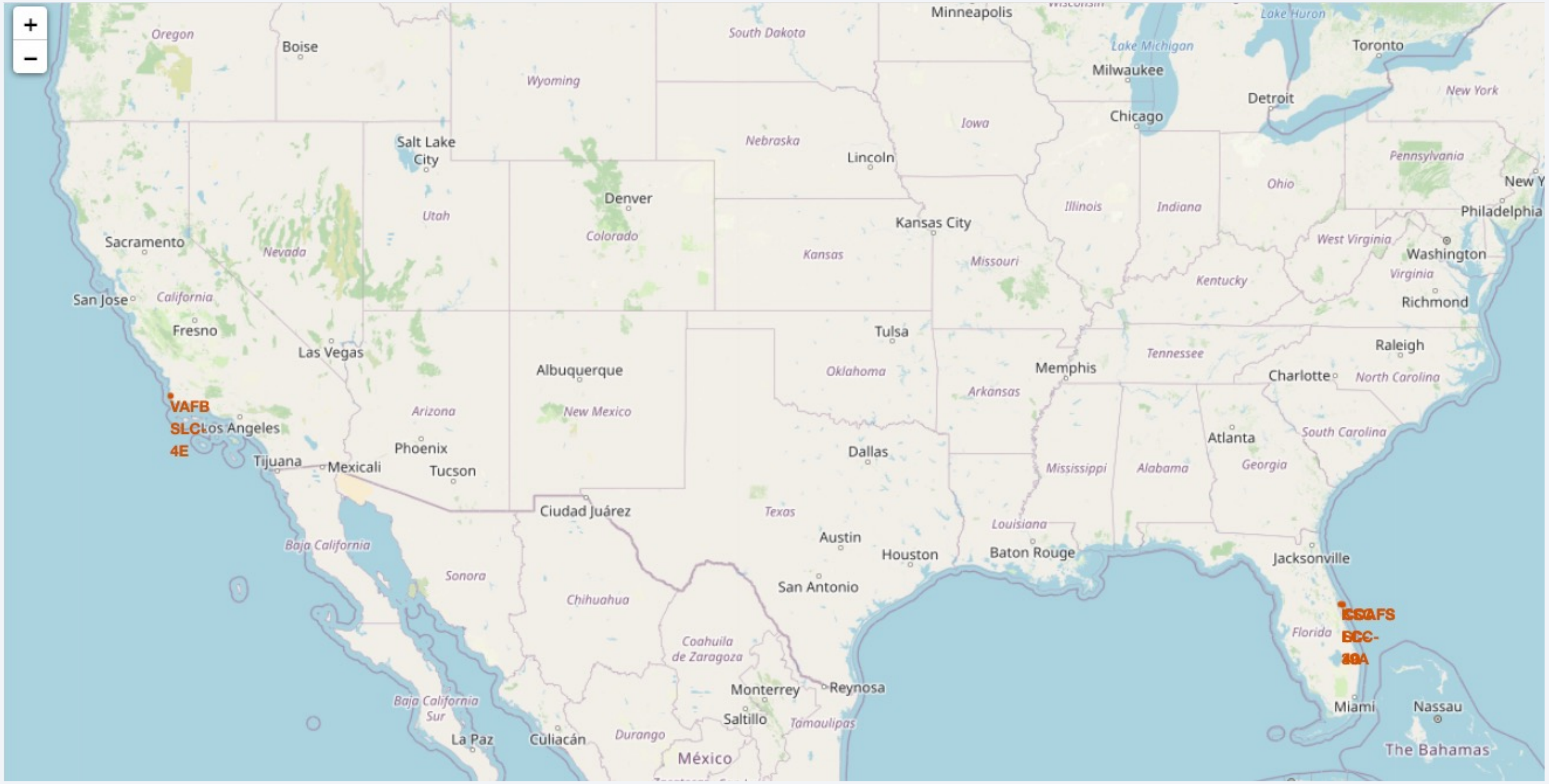
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	COUNT(*)
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

Section 3

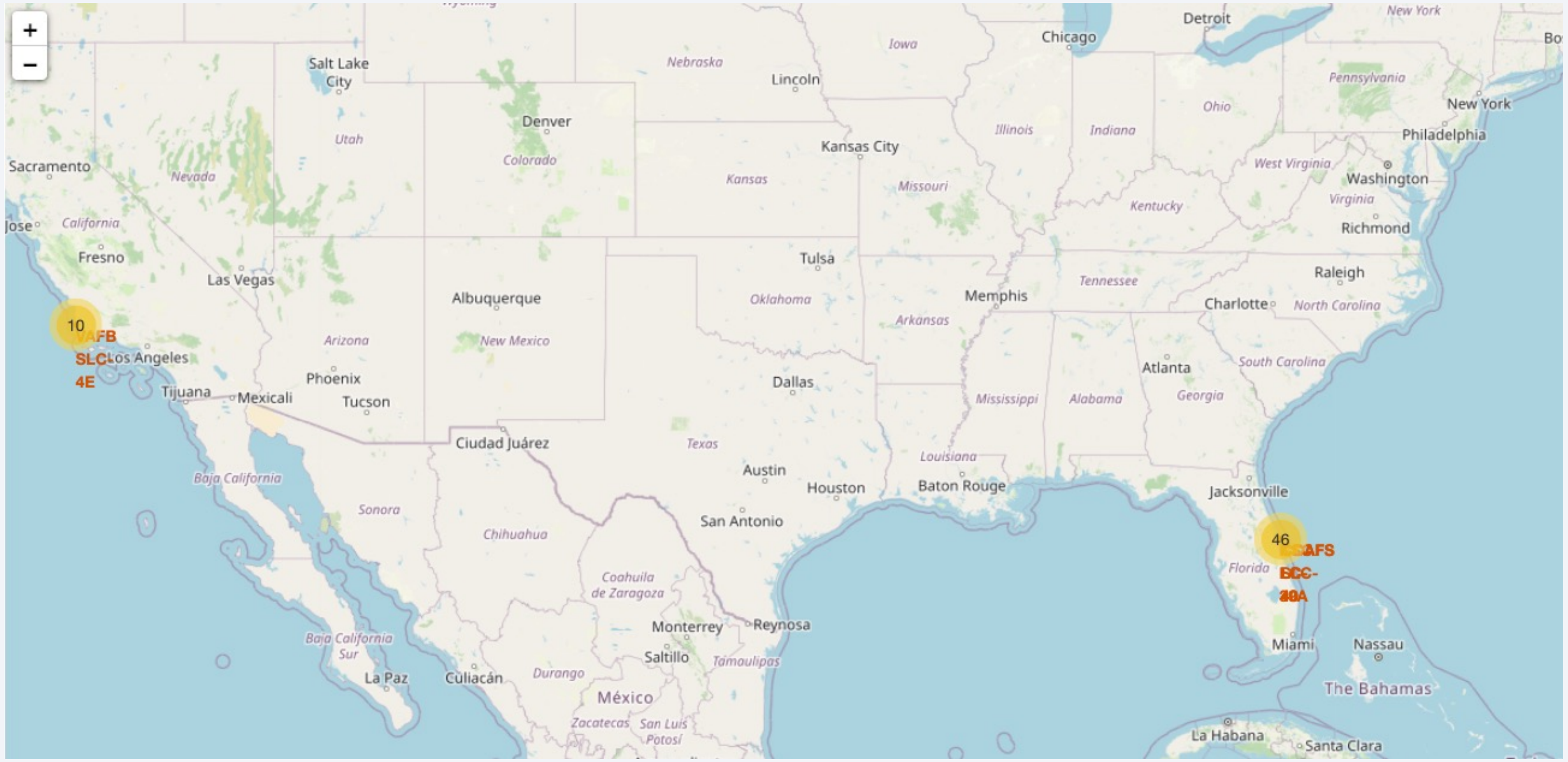
Launch Sites Proximities Analysis

All Launch Sites on Map



- All site are very close to the coast

Success Launches by Sites



<Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

- Explain the important elements and findings on the screenshot

Section 4

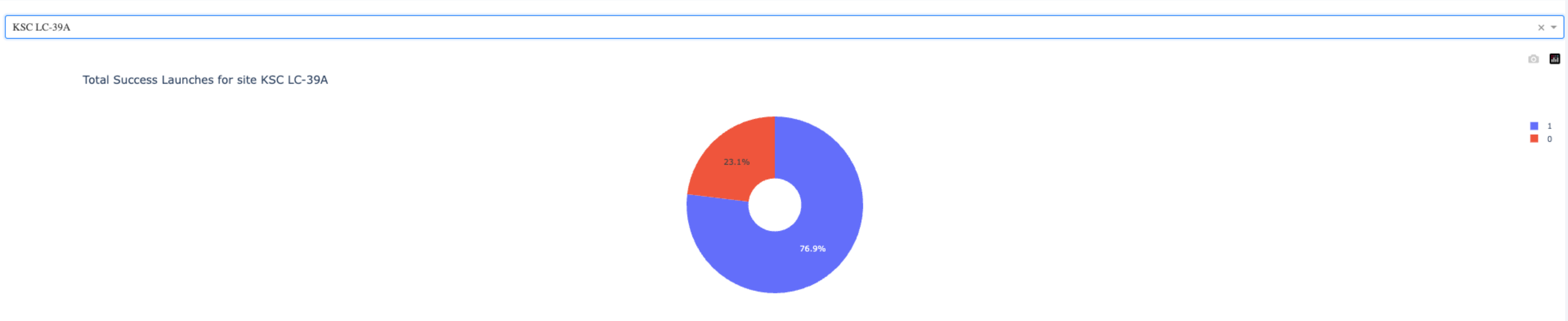
Build a Dashboard with Plotly Dash

Success count for all sites

- Site KSC LC-39A has most success rate with 41.7% rate, however VAFB SLC-4E has the least rate with 16.7% rate.

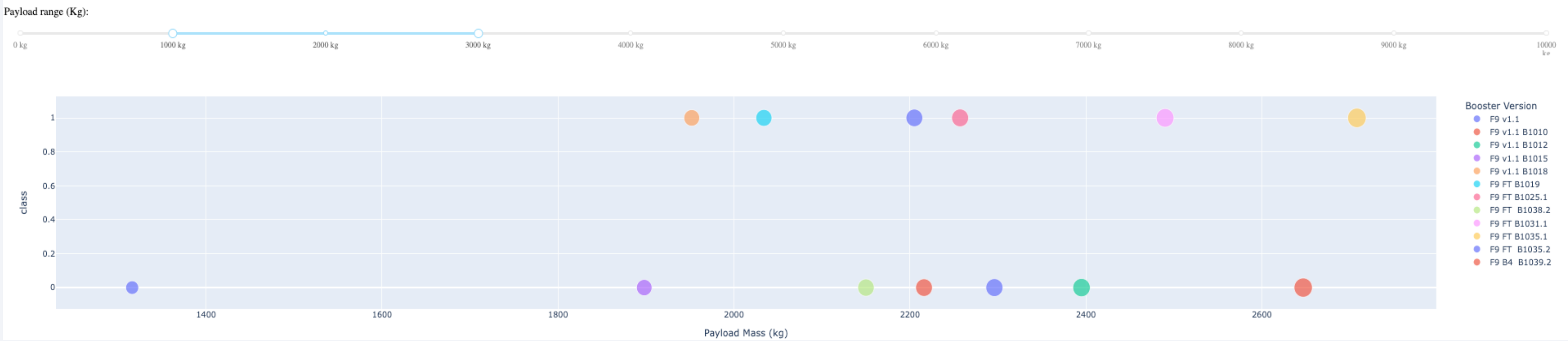
Most Successful Launch Site

- Most success launches achieved on KSC LC-39A site.



Relations between launch success and Payload Mass

Launches results which payload mass between 1000 and 3000 kg

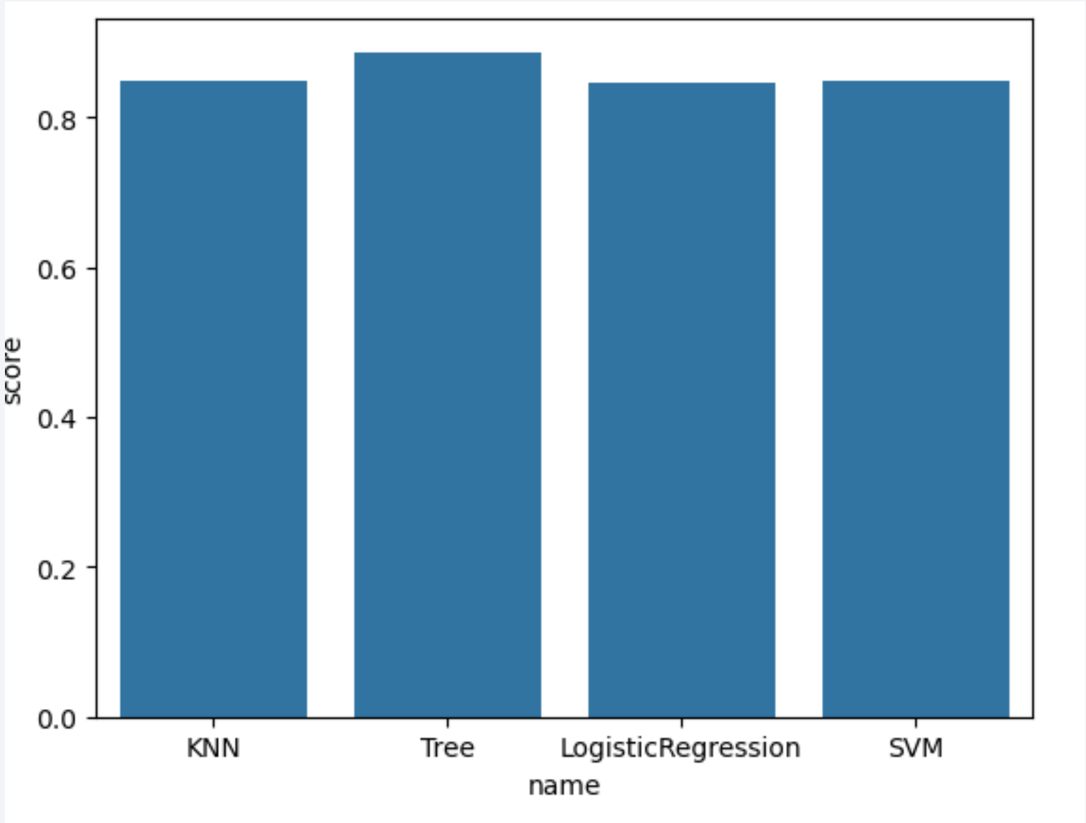


Launches results which payload mass greater than 9000 kg

Section 5

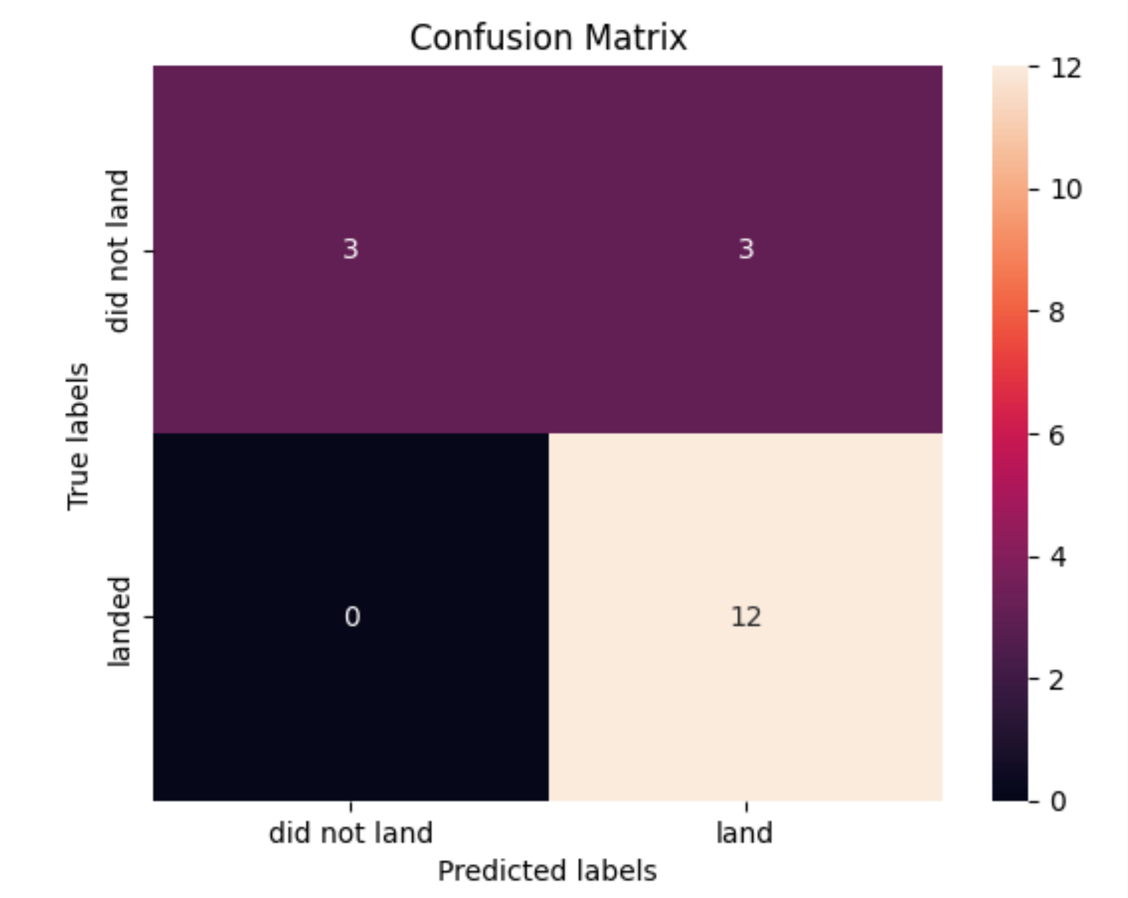
Predictive Analysis (Classification)

Classification Accuracy



- We see that Tree Classifier has the best score rate.

Confusion Matrix



- We see that Tree Classifier can distinguish between the different classes. And the major problem is false positives.

Conclusions

- KSC LC-39A have the most successful launches of any sites with rate of 76.9%
- As a technical information; Tree Classifier is better than other methods for this dataset.

Thank you!