IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Stephen J Martins
3th of August, 2023

Outline

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

Executive Summary

- In this capstone project, our objective is to leverage SpaceX's comprehensive dataset to conduct a thorough analysis aimed at predicting the launch costs for Company SpaceX. Additionally, we delve into the intricate details of SpaceX's rocket launches. Our methodology encompasses a multi-faceted approach, combining data analysis, predictive modeling, and visual representation.

- Through an exhaustive examination of SpaceX's dataset, we have meticulously crafted interactive dashboards that unveil the intricate interplay between diverse variables and their impact on launch success rates. Moreover, we have harnessed the power of folium maps to provide insightful geospatial perspectives, revealing the spatial relationships between launch sites and prominent landmarks.

- By merging these analytical components, our capstone not only sheds light on the cost prediction for Company SpaceX's launches but also enhances our understanding of the critical elements influencing launch outcomes in the context of SpaceX's operations.

Introduction

- The modern space era has brought accessible space travel through companies like Blue Origin and SpaceX. Among these, SpaceX stands out for achievements like International Space Station missions and the Starlink satellite network. A key factor is their cost-effective approach, advertising Falcon 9 rocket launches for $62 million compared to competitors' $165 million. This is possible due to reusing the first stage. By predicting first-stage success, we can predict launch costs.
- Project Goals:

1. Launch Costs: Predicting the cost of each launch.
2. Success Factors: Understanding how variables like flights and payloads relate to success.
3. Launch Sites: Analyzing factors influencing rocket launch site choices.

Join us in uncovering the economics and dynamics of SpaceX's launches, revealing insights that shape the future of space travel.

Section 1

Methodology

Methodology

Executive Summary

- Data Collection Sources:
  - SPACEX API utilizing multiple endpoints, normalize using json_normalize() as the REST API was presented as an array of JSON Objects.
  - Our data also derives from launches of rockets of wikipedia: 'https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches'. Here it was applied BeautifulSoup to scrape the information from the existing table.
- Describe how data was processed
  - Data was focused on Falcon 9 launches, then data was integrated from both sources for a dataset. Was selected essential fields and also removing NULL values to clean the dataset
- 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 - Summary

- In this project, we're using data from two sources:
    - the SPACEX REST API and
    - Wikipedia via web scraping.

  This gives us detailed and comprehensive information for analysis.

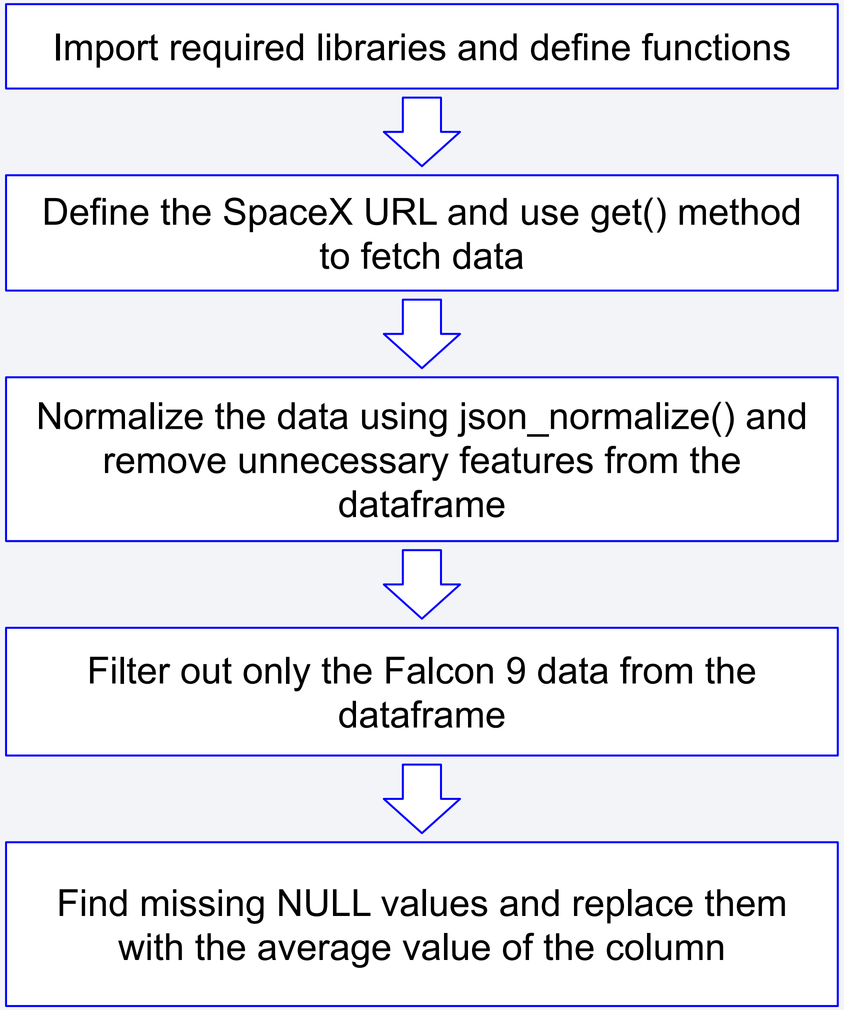
- The SPACEX REST API offers rich data, including launch details, booster versions, payloads, and more. Accessed at 'https://api.spacexdata.com/v4', its endpoints like /launchpads/, /rockets/, /payloads/, etc. provide unique datasets. We collect this data and organize this JSON data into a structured format, enhancing our analysis.

Data Collection – SpaceX API

- We used the below functions:
- **getBoosterVersion** takes the rocket column to call the API and append the data to a list, which will be the name of the rocket.
- **getLaunchSite** takes the launchpad column to call the API and append the data to a list, which we will get the latitude longitude and launch site name.
- **getPayloadData** takes the payloads column to call the API and append the data to a list, which we will get the Payload Mass in KG and Orbit.
- **getCoreData** takes the cores column to call the API and append the data to a list, which we will get Block core, ReusedCount, Serial, Outcome, Flights, GridFins, Legs, Landing Pad

- GitHub:

  https://github.com/Martinsconsulting/IBM-Space-X-/blob/main/IBM%20-%20Data%20Science%20-%20SpaceX%20-%201%20-%20Data%20Collecting.ipynb



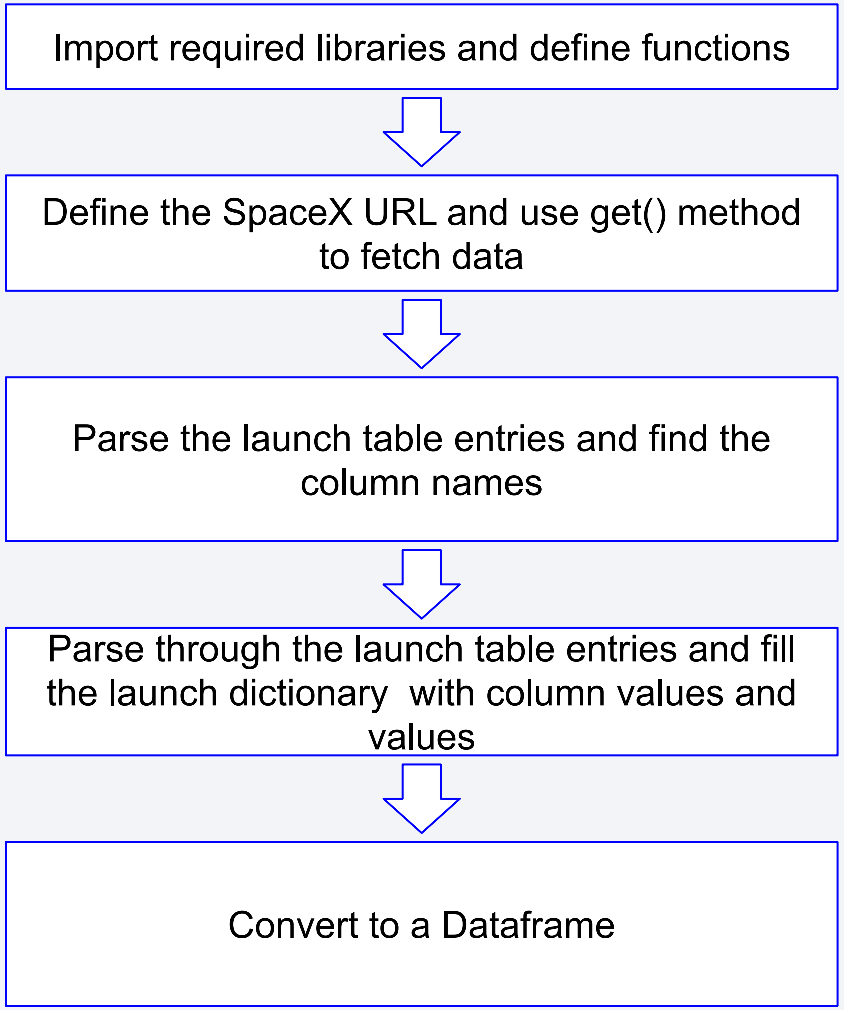
Data Collection - Scraping

- We used the below function:

- **date_time** this returns the date and time from the html table cell
- **booster_version** this returns the booster version of the html table cell
- **landing_status** this returns the landing status from the html table cell
- **get_mass** this returns the landing status from the html table cell
- **extract_column_from_header** this returns the column name from the html table cell.

- GitHub:

  https://github.com/Martinsconsulting/IBM-Space-X-/blob/main/IBM%20-%20Data%20Science%20-%20SpaceX%20-%202%20-%20Data%20Collecting%20with%20Web%20Scraping%20.ipynb



Data Wrangling

- Methods such as isnull,sum(),shape were used to calculate the percentage of missing values.

- value_counts method were used to determine how many flights were done per launch, how many flights for each orbit, and the type of outcome per orbit.

- With the filtered data of the outcome column we were able to create a list were the value 1 means a successful landing and 0 means fail landing. We then appended the list to the dataframe with the class

Import required libraries and define functions

⇩

Download dataset and apply it to a dataframe

⇩

Calculate the number of launches on each site, the number and occurrence of each orbit.

⇩

Create a new column called ‘class’ where the value 1 means successfully landed and 0 means landing fail

- GitHub:

  https://github.com/Martinsconsulting/IBM-Space-X-/blob/main/IBM%20-%20Data%20Science%20-%20SpaceX%20-%203%20-%20Data%20Wrangling.ipynb

EDA with Data Visualization

- The intent was to perform exploratory Data Analysis and Feature Engineering using pandas and Matplotlib. We draw the below charts in order to find insights if it affected the launch outcome Here are the charts we used:
    - Flight Number VS Payload
    - Flight Number VS Launch Site
    - Payload VS Launch Site
    - Orbit Type VS Average Success Rate
    - Flight number VS Orbit Type
    - Payload VS Orbit Type
    - Average Launch success VS Yearly Trend

- Completed Feature Engineering by creating dummy variables to categorical columns using get_dummies() from the pandas library

- GitHub:

  https://github.com/Martinsconsulting/IBM-Space-X-/blob/main/IBM%20-%20Data%20Science%20-%20SpaceX%20-%205%20-%20EDA%20with%20Visualization.ipynb

EDA with SQL

- SQL queries that were performed:
  - 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 v1.1
  - 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.
  - 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.

- GitHub:

  https://github.com/Martinsconsulting/IBM-Space-X-/blob/main/IBM%20-%20Data%20Science%20-%20SpaceX%20-%204%20-%20Complete%20the%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- The intent of using a Map is mark all launch sites, mark success/ failed launches anc calculate the distance between a launch site to its proximities.

- .For the markers we have used the methods:
    - Circle() to determine on the map the a radius of the location sites,
    - Marker() to determine on the map the exact location of the launch site
    - MarkerCluster() to determine on the map the success/failed launches per launch sites.
    - MousePosition() to determine the coordinates of proximities.
    - Marker() again to pinpoint the location of the proximities
    - Polyline() to trace a line from the launch site to the proximities.

- We also calculated the distance between the proximities, the proximities being the coastal line, highway, railroad and a city.

- GitHub:

  https://github.com/Martinsconsulting/IBM-Space-X-/blob/main/IBM%20-%20Data%20Science%20-%20SpaceX%20-%206%20-%20Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- The dashboard created was created with the following components:
  - A dropdown menu with the launch site(all sites and individually). This was display the success rate.
  - A Pie Chart of the successful/failed launches for the selected option
  - A Slider for payload range
  - A scatter plot to display correlation between success rate and payload.

- GitHub:

  https://github.com/Martinsconsulting/IBM-Space-X-/blob/main/IBM%20-%20Data%20Science%20-%20SpaceX%20-%207%20-%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.ipynb

Predictive Analysis (Classification)

Import required libraries, import data into a dataframe and standardize using Standard.fit_transfrom()

↓

Split the data into train and test sets using train_test_split() function

↓

Create models

↓

Find the accuracy of each model using GridSeachCV object, and print the tuned hyperparameters

↓

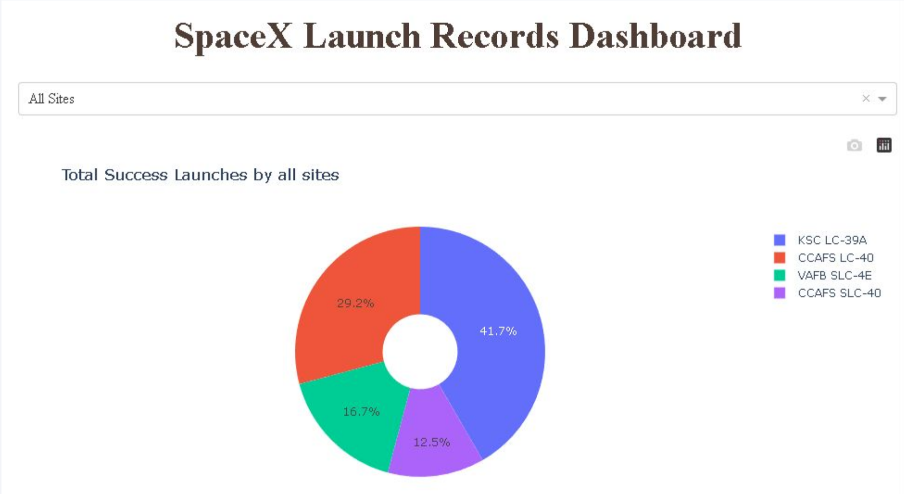
Display confusion matrix for all models and determine the best model

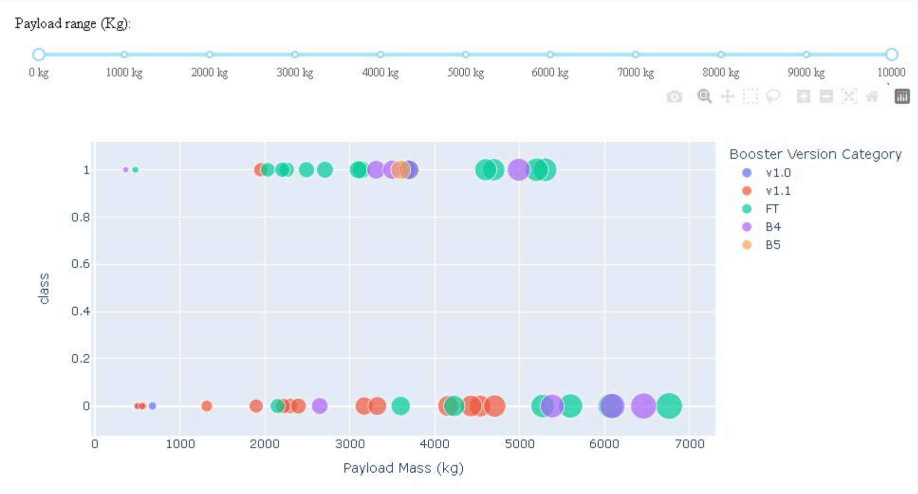
- We built the accuracy of each models using logistic regression, support vector machine, decision tree, and K-nearest neighbors. GridsearchCV object is used to find the tune hyperparameters and the accuracy of each model.
- The confusion matrix is built for each model using the auxiliary function plot_confusion_matrix(). The best model is decided based on the accuracy soccer and confusion matrix.

Results

- Exploratory data analysis results
  - The charts helped provide more insights of the different correlations between each parameters. The yearly success rate trend chart shows us that success rate has been better over the years when the best success rate was in the middle of 2019 and 2020.

- Interactive analytics demo in screenshots:



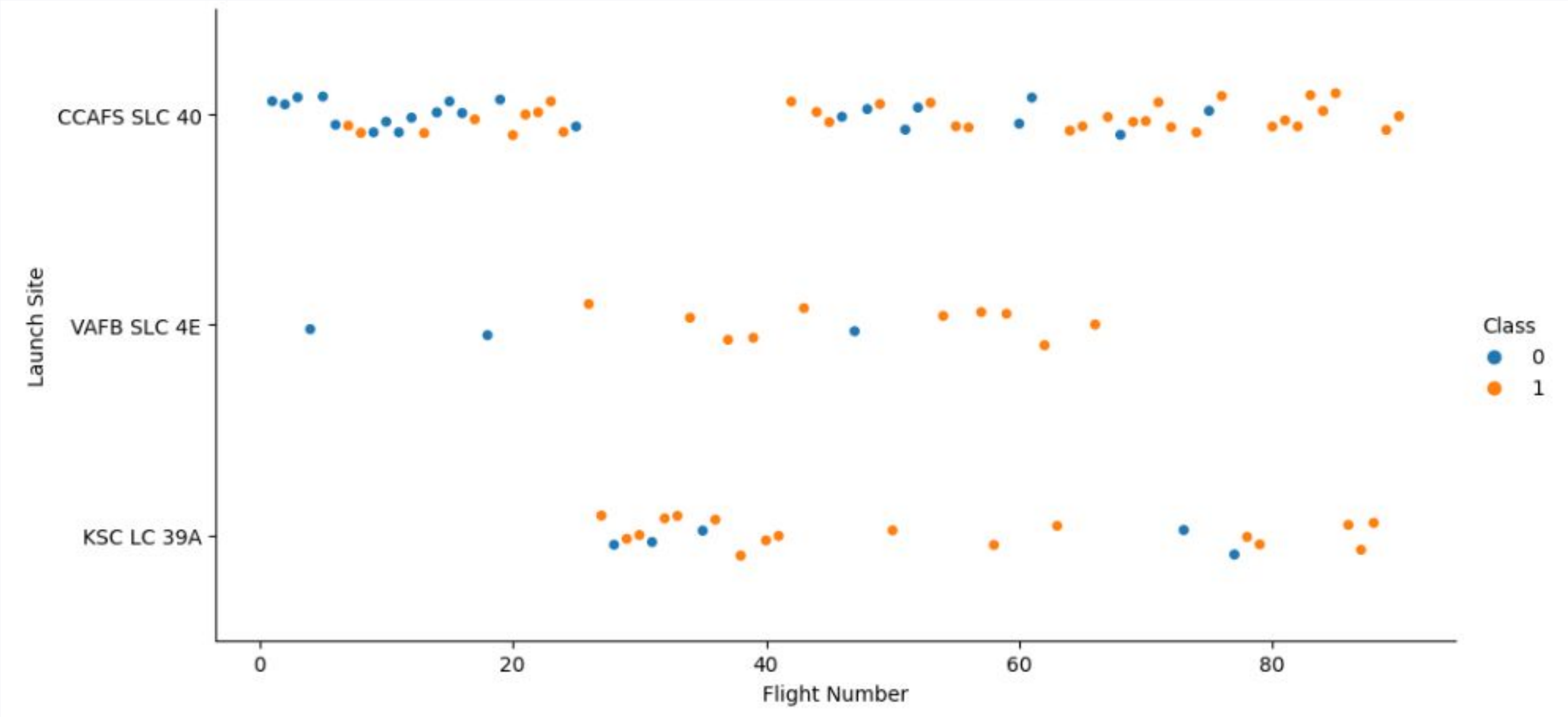


- Predictive analysis results:
  - The best performing model to use is Decision Tree with an accuracy of 0.88.

Section 2

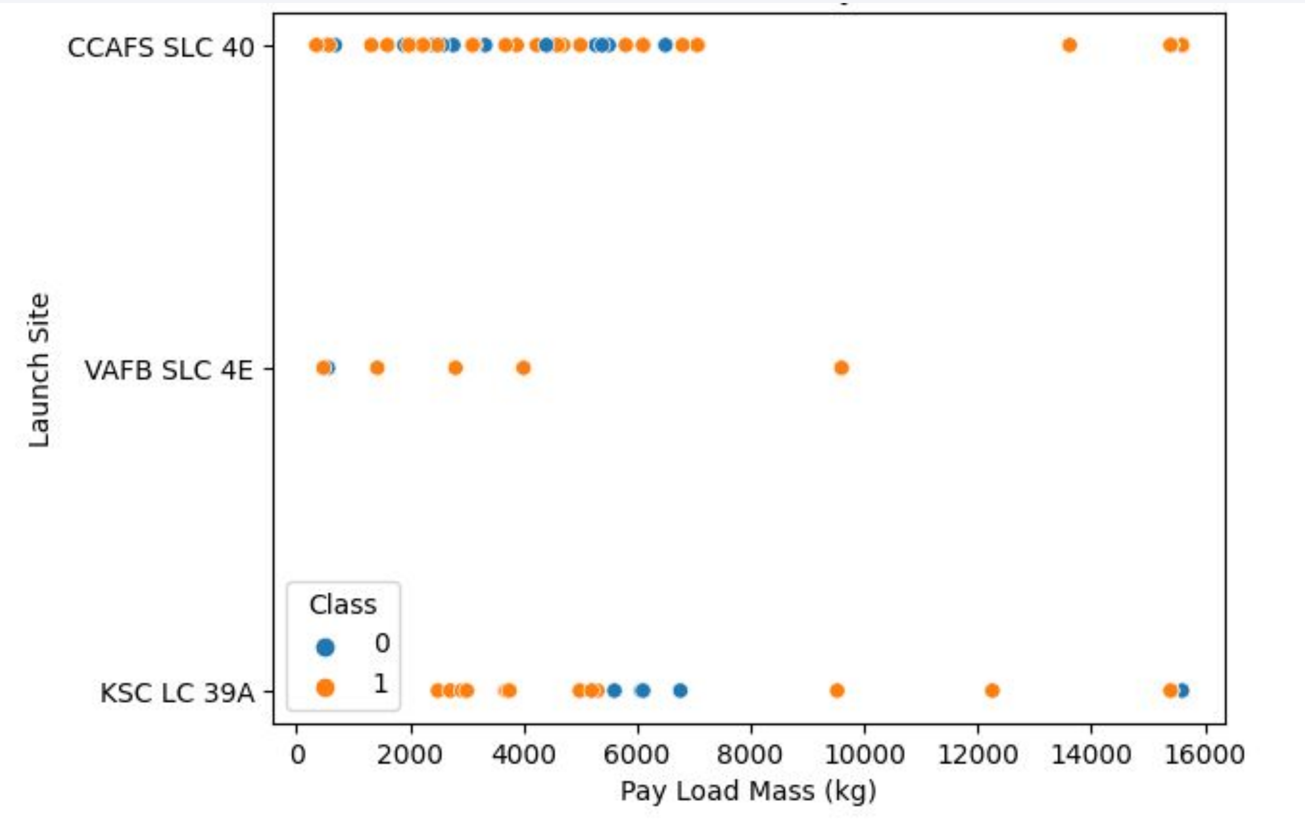
Insights drawn from EDA

Flight Number vs. Launch Site



On this chart we can visualize that the more flights are done the more success there is. There are more flights done at CCAFS SLC 40 then the other launch sites.

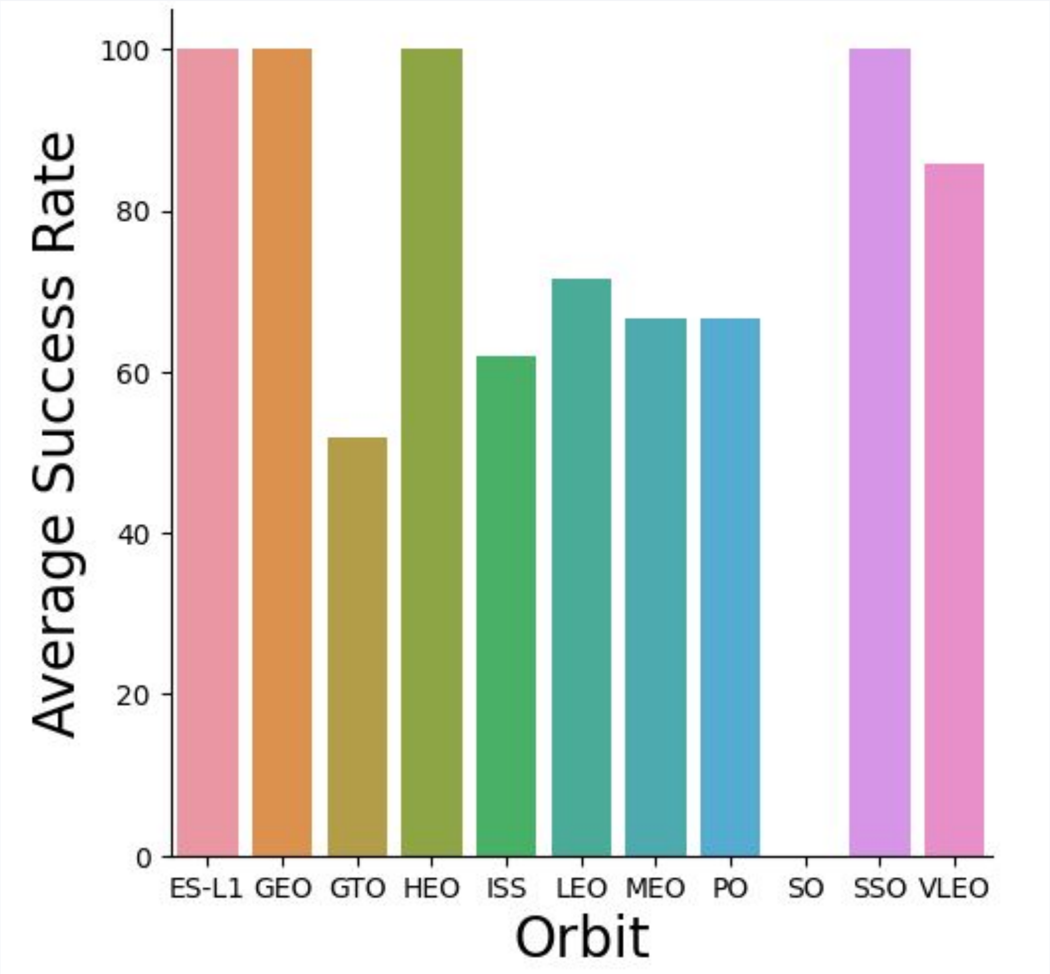
Payload vs. Launch Site



On this chart Payload Vs Launch site there seems to be no pattern here. We can conclude that there were none flights with more the 10000 (kg) in payload mass.

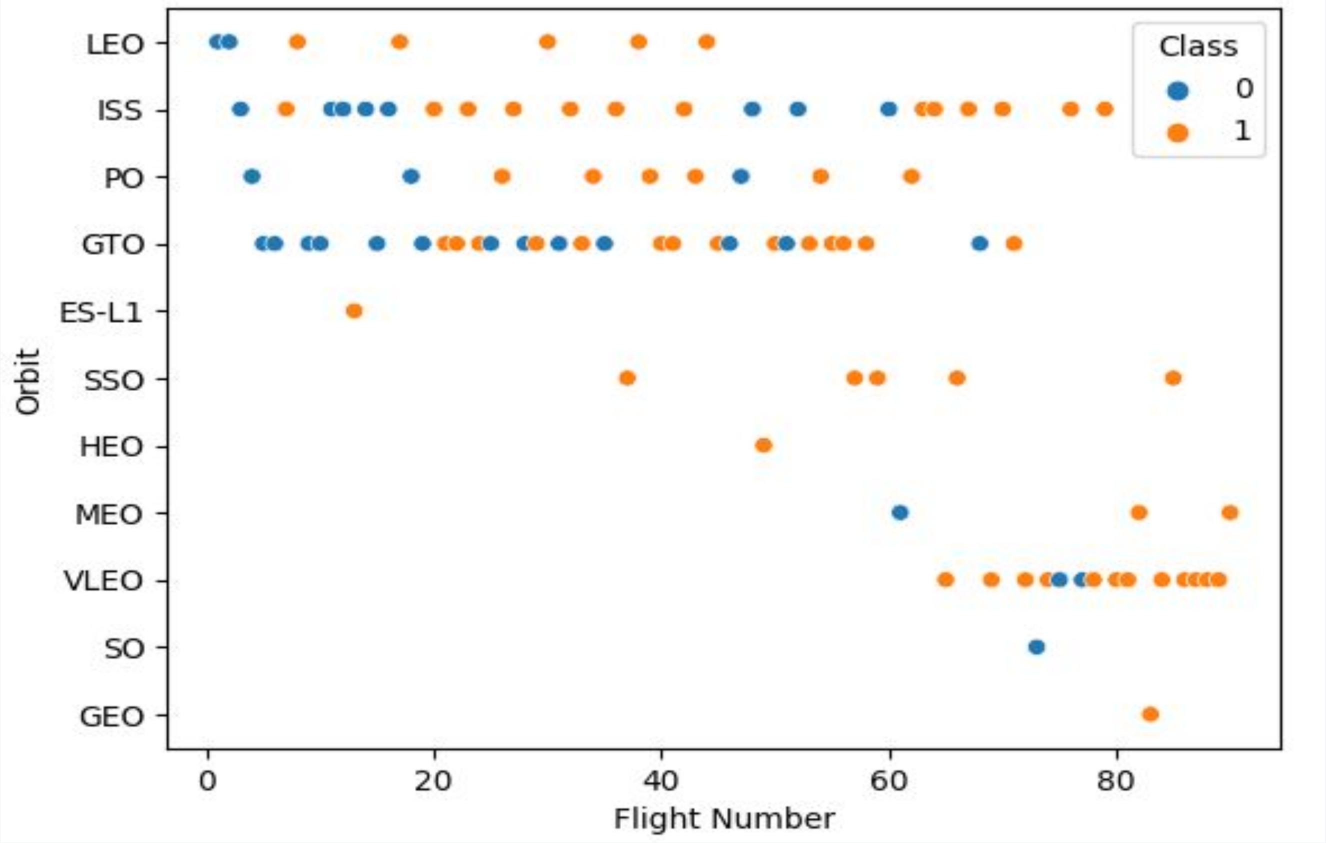
Success Rate vs. Orbit Type

- The orbits SSO, ES-L1, HEO and GEO have 100% success rate. VLEO has 85% success rate. The rest of the orbits have below 72% success rate.



Flight Number vs. Orbit Type

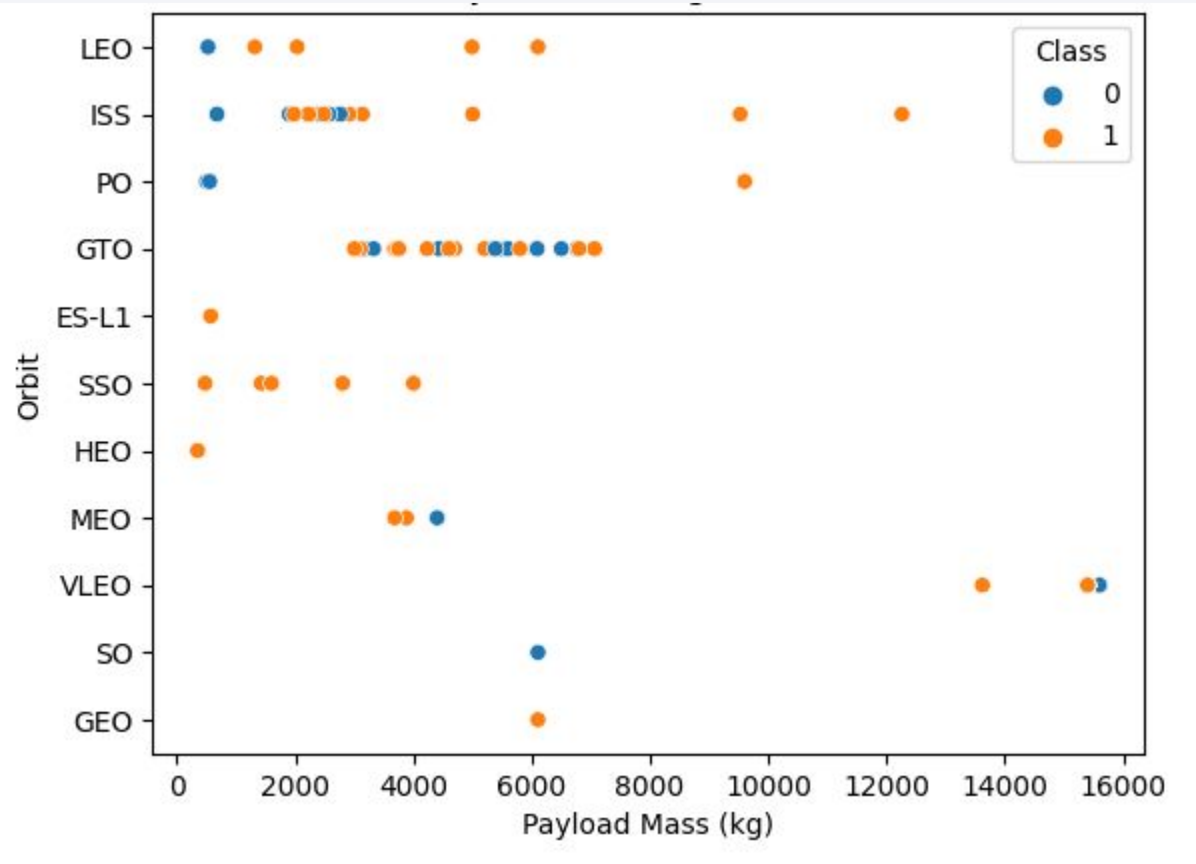
- You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



Payload vs. Orbit Type

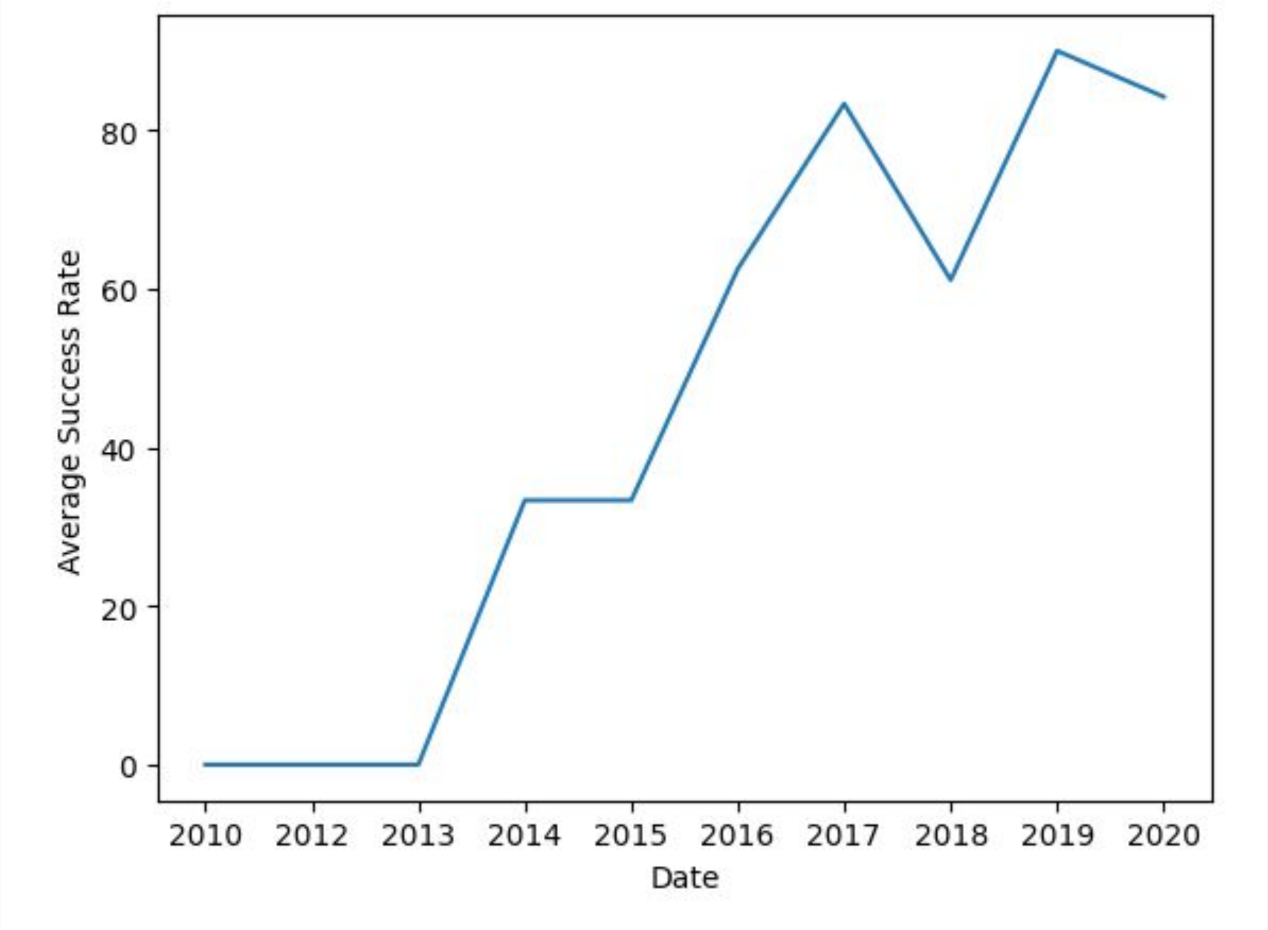
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

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



Launch Success Yearly Trend

- You can observe that the success rate since 2013 kept increasing till 2017 then dropped in 2018 then went higher, after 2019 it slightly dropped. In overall the success has been increasing over the years



All Launch Site Names

Task: Display the names of the unique launch sites in the space mission

```
%sql select DISTINCT Launch_site from SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

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

```
%%sql
select *
from SPACEXTABLE
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
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	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 MAX(PAYLOAD_MASS__KG_)
from SPACEXTABLE
where Customer = 'NASA (CRS)'
```

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

MAX(PAYLOAD_MASS__KG_)
3310

Average Payload Mass by F9 v1.1

```
%%sql

select AVG(PAYLOAD_MASS__KG_)
from SPACEXTABLE
where Booster_Version = 'F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

```
%%sql

select MIN(Date)
from SPACEXTABLE
where Mission_Outcome = 'Success'
```

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

MIN(Date)
2010-04-06

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql

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

Total Number of Successful and Failure Mission Outcomes

```
%%sql

select Mission_Outcome, COUNT(*)
from SPACEXTABLE
group by Mission_Outcome
```

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

Mission_Outcome	COUNT(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

```
%%sql

select booster_version
from SPACEXTABLE
where PAYLOAD_MASS__KG_ = (
      select MAX(PAYLOAD_MASS__KG_)
      from SPACEXTABLE
)
```

```
 * 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
    CASE substr(Date, 6, 2)
        WHEN '01' THEN 'January'
        WHEN '02' THEN 'February'
        WHEN '03' THEN 'March'
        WHEN '04' THEN 'April'
        WHEN '05' THEN 'May'
        WHEN '06' THEN 'June'
        WHEN '07' THEN 'July'
        WHEN '08' THEN 'August'
        WHEN '09' THEN 'September'
        WHEN '10' THEN 'October'
        WHEN '11' THEN 'November'
        WHEN '12' THEN 'December'
    END AS Month,
    Landing_Outcome,
    Booster_Version,
    Launch_Site,
    Date
FROM SPACEXTABLE
WHERE substr(Date, 1, 4) = '2015'
    AND Landing_Outcome = 'Failure (drone ship)'

```

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

Month	Landing_Outcome	Booster_Version	Launch_Site	Date
October	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-10-01
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

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

```
%%sql

SELECT
    COUNT(Landing_Outcome) AS outcome_count,
    Landing_Outcome,
    Date
FROM SPACEXTABLE
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
    AND (Landing_Outcome = 'Failure (drone ship)' OR Landing_Outcome = 'Success (ground pad)')
GROUP BY Landing_Outcome
ORDER BY Date DESC
```

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

outcome_count	Landing_Outcome	Date
5	Success (ground pad)	2015-12-22
5	Failure (drone ship)	2015-10-01

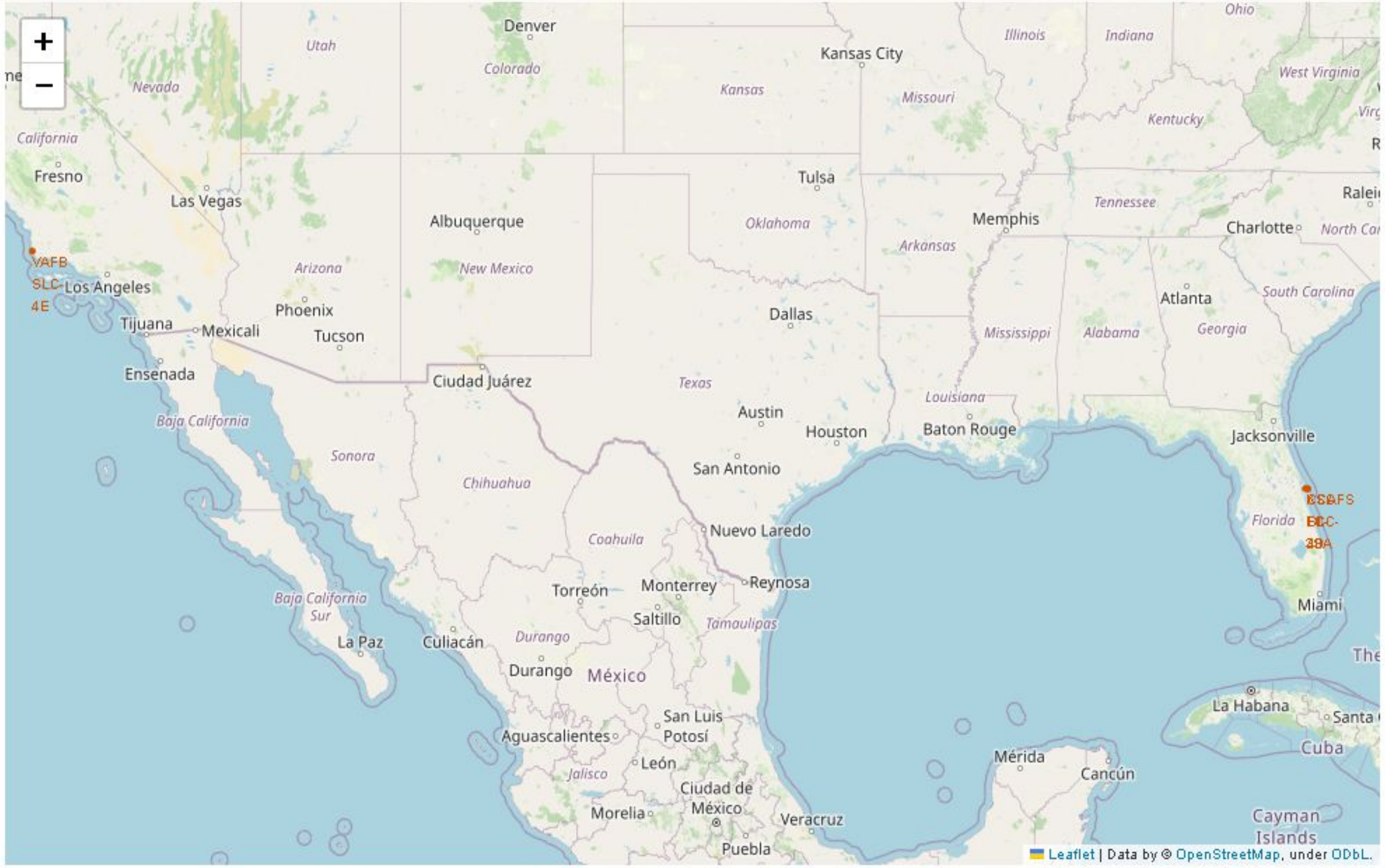
Section 3

Launch Sites Proximities Analysis

Folium map of all launch sites

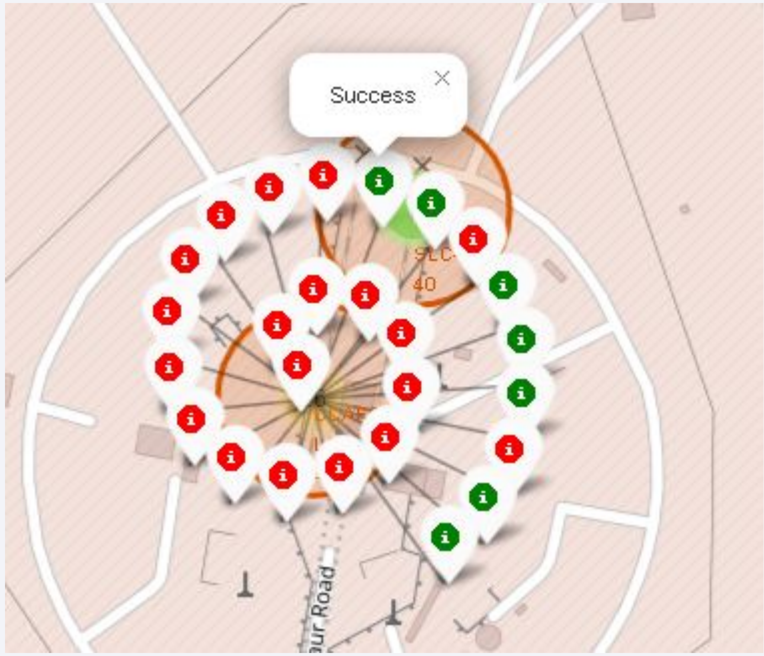
- The folium map shows all launch sites. We have used the function folium.map(). Through the latitude and longitude of each launch sites we were able to include a marker using .marker() function and with the circle() method we were able to include a radius of the launch sites.

- The map shows that 3 of the launch sites are in the state of Florida and the other site is in the California state.



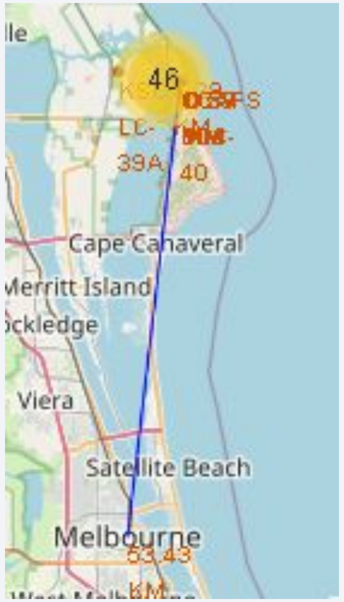
Folium map showing color-coded of outcomes for each launch site

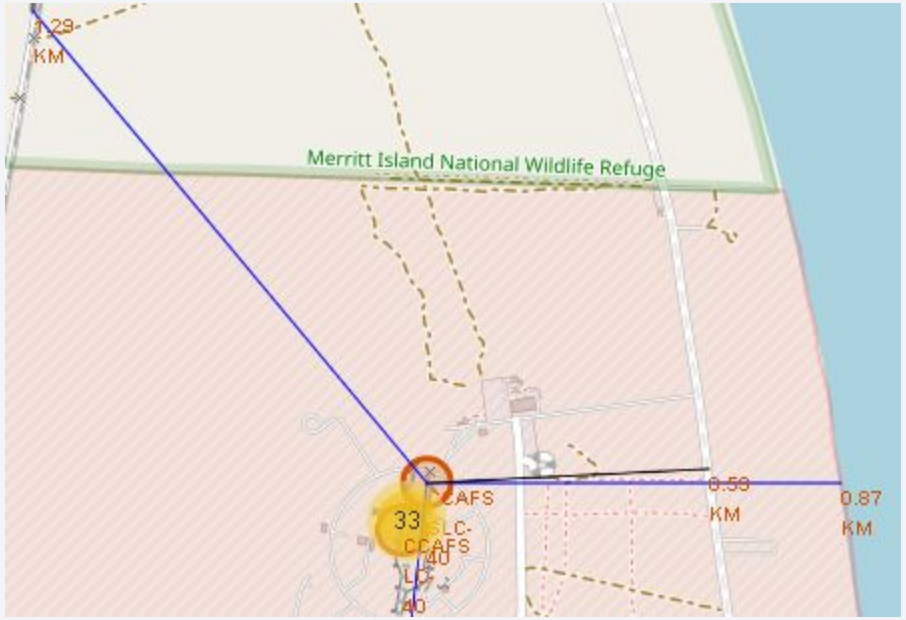
- On the left side screenshot we see the amount of launches per launch site this was done by applying the MarkerCluster() function. With this function we were able to mark multiple points on the map.
- On the right side screenshot we have applied the color green and a popup of ‘Success’ if a launch was successful and the color red and a popup of ‘Fail’ if the launch has failed.



Folium Map Proximities

- On this map we applied a MousePosition() function so we can better find the latitude and longitude of the nearest coastal line, highway, railroad, and city.
- We have included markers in this locations and traced a line with the PolyLine() function from the launch site to the proximities.
- We have also calculated the distance and added an icon with the calculated distance.





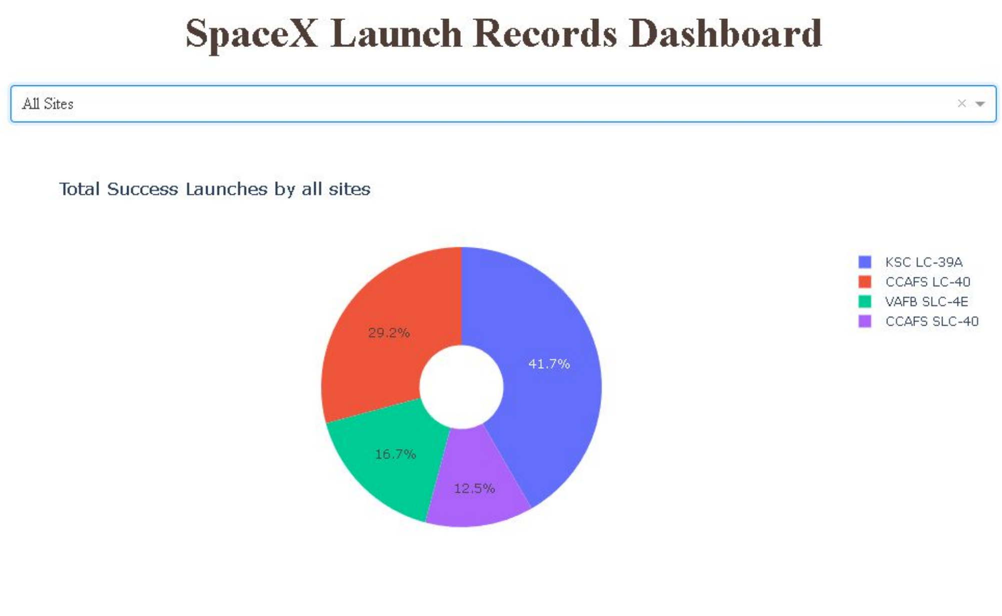
Section 4

Build a Dashboard with Plotly Dash

Dashboard - Pie Chart for Launches

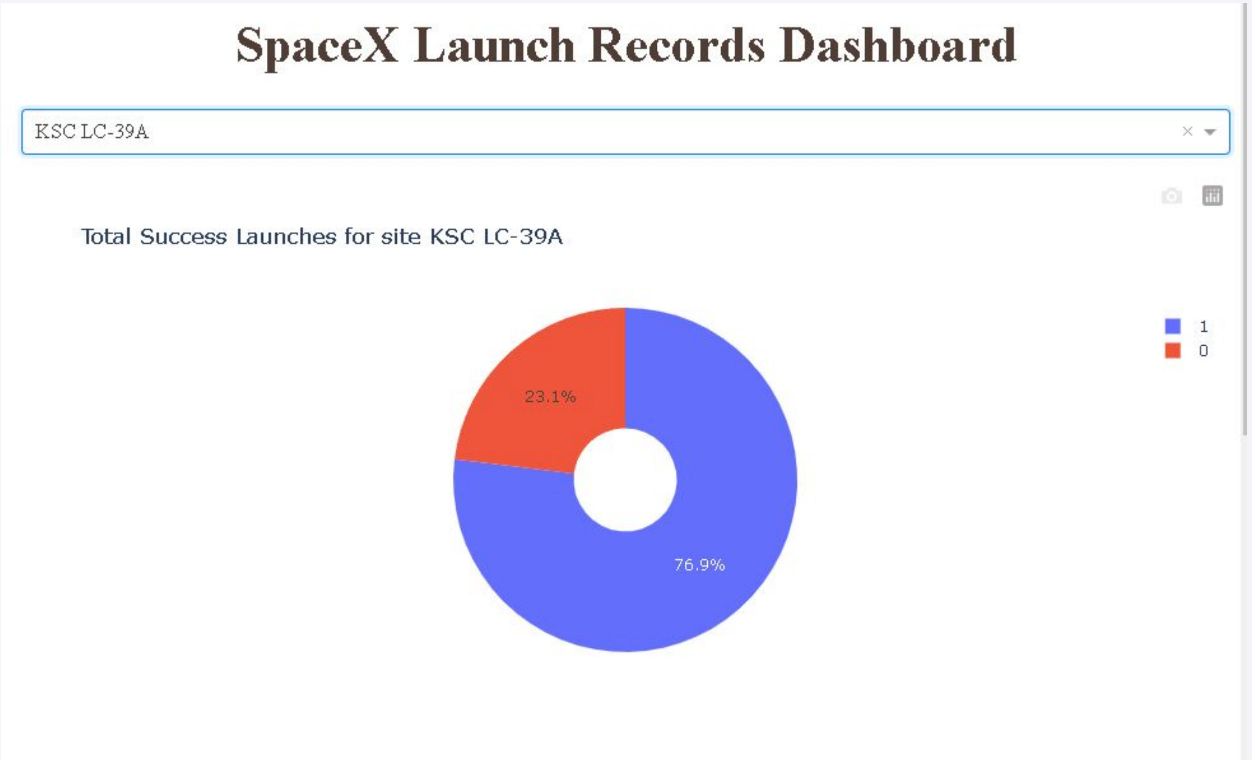
- Input is given in the form of a drop down box, containing options for all sites and each site individually. For each site selected, a pie chart is displayed showing the success and failure launch percentages.

- The graph for “All sites” option shows that the highest success rate is for the site KSC LC-39A and the least is for the site CCAFS SLC-40.



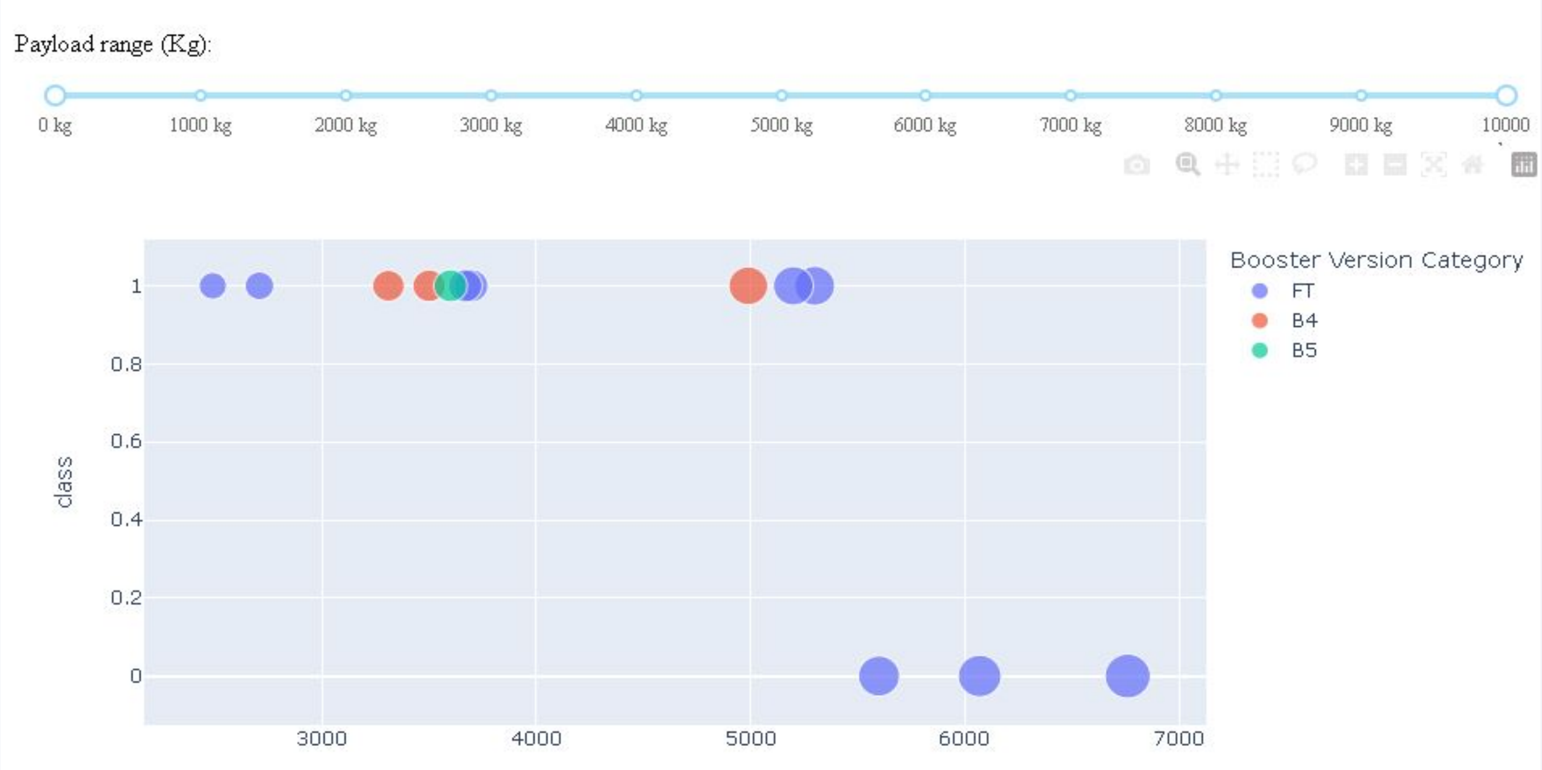
Dashboard showing success rate for each site

- The launch site with highest launch success ratio is KSC LC-39A, displayed in screenshot.
- The success percentage is shown in purple color in the pie chart and failure percentage in red. The legend on the side displays the “class” field (0=failure and 1=success).



Dashboard with slider and scatter plot

- The payload range that has the highest launch success rate is between 2000 kg and 4000kg.
- the payload range that has the lowest launch success rate is between 6000 kg and 7000kg
- The Booster version with the highest launch success rate is the Version FT.



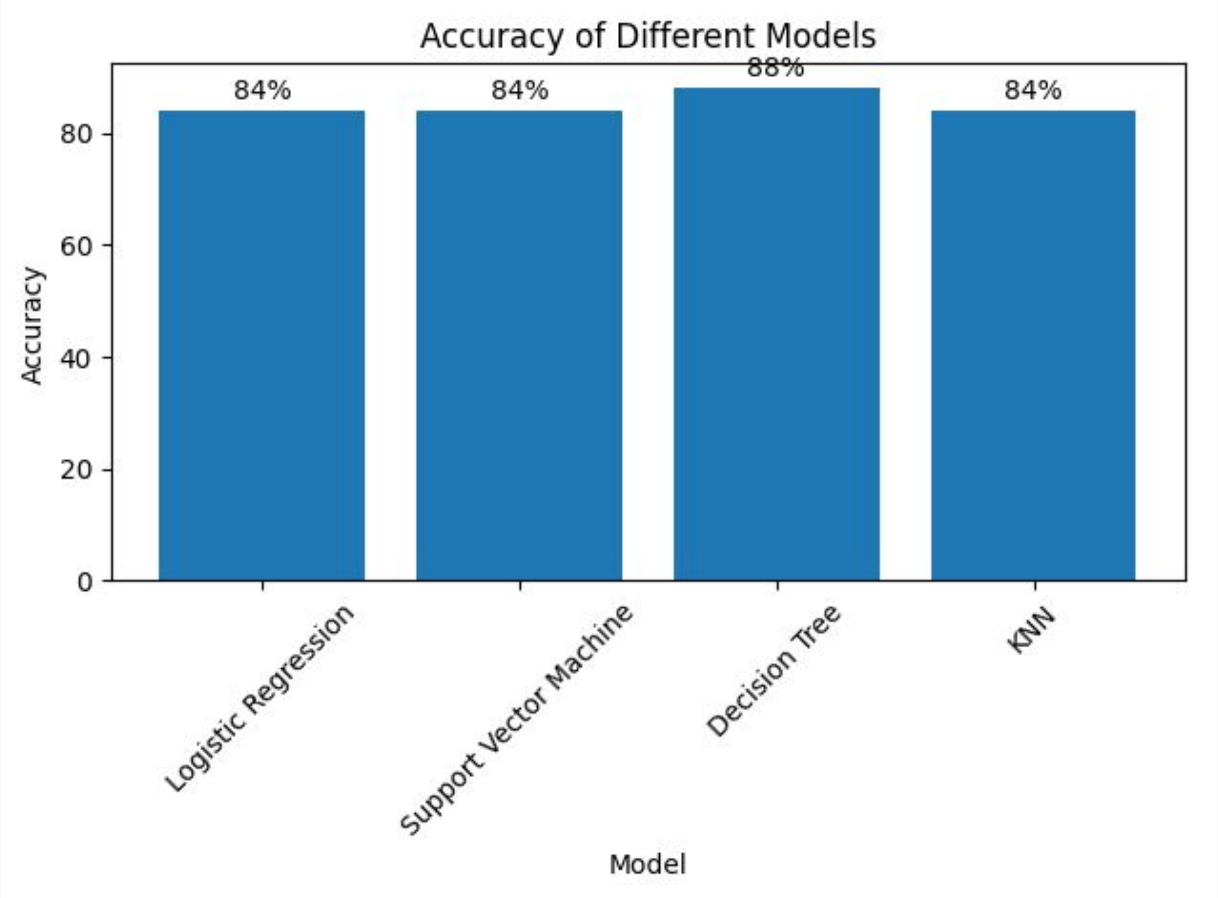
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- This bar chart shows the accuracy of each model built.

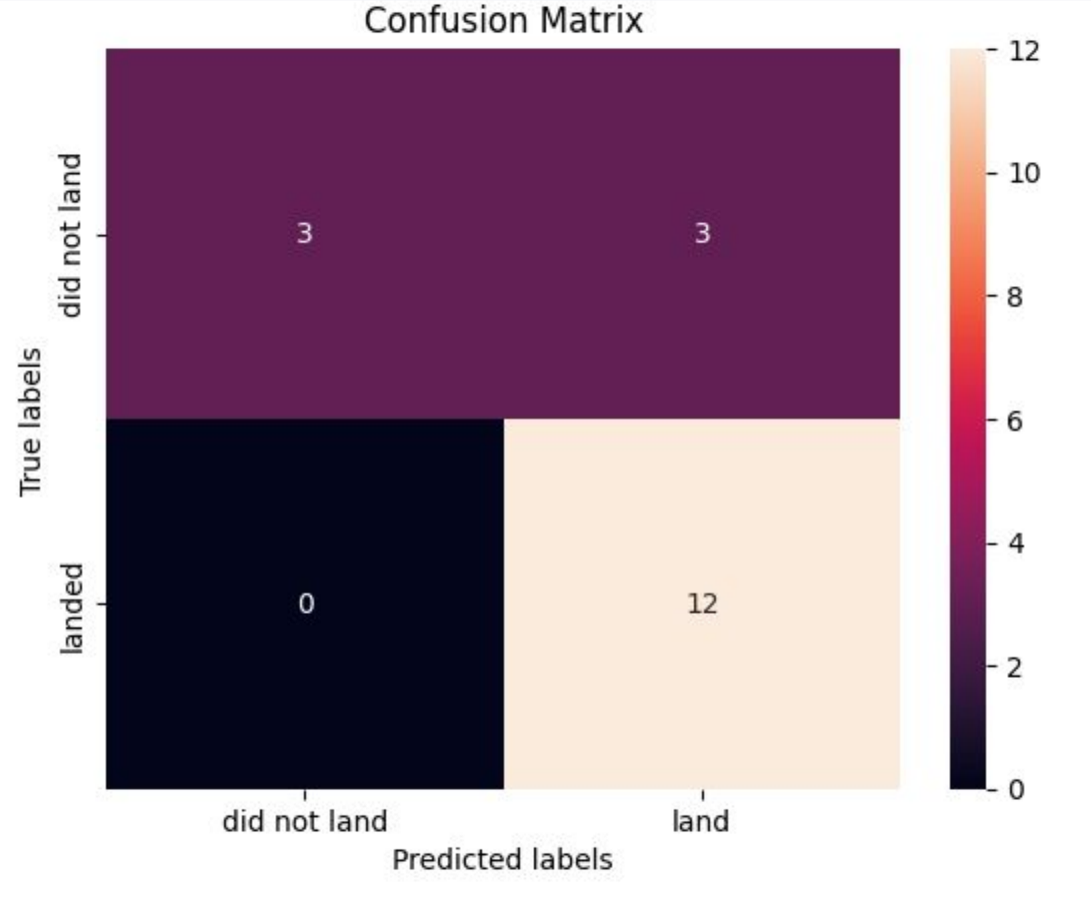
- The model having the highest classification accuracy is decision tree, with 88%. The other 3 models have the same accuracy of 84%.



Confusion Matrix

- This is the confusion matrix of decision tree based model. • It has 3 false positives(predicted=landed and actual=did not land) and 0 false negative (predicted=did not land and actual=landed).

• It has 3 true negatives (actual=did not land and predicted=did not land) and 12 true positives (actual=landed and predicted=landed).



Conclusions

- The best performing model is decision tree with an accuracy of 88%. The cost of each launch and the landing status can be predicted accurately using this model.
- The launch sites analysis show that the sites are in close proximity to coast lines and pretty far off from cities.
- The dashboard visualization of launch data gives the following conclusions:
    - Most successful year was 2020, and also the year with most launches.
    - Most successful site was KSC LC-39A, and most successful booster version was FT.
    - The least successful site was CCAFS SLC-40 and most unsuccessful booster version was V1.1.
    - The best payload mass range for success was between 2000kgs and 4000 kgs.

Thank you!