IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Anish Bhalla
28th May 2023

Outline

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

Executive Summary

- This project involved collecting data from API calls and Web Scraping, followed by data cleaning to handle null values. The cleaned data was then subjected to Exploratory Data Analysis using Pandas and Seaborn Visualization. SQL queries were also utilized to understand the feature matrix variables.
- Considering the nature of the data, map visualization using the Folium library was employed to examine geographical proximities that impact site selection. Additionally, an interactive Plotly Dashboard App was developed to analyze the success rate of rockets across different sites and their boosters.
- Furthermore, four Machine Learning models were trained using the collected data. After evaluating their performance, the Decision Tree Model, with the highest accuracy, was selected as the preferred model for predicting launch outcomes.
- By leveraging these methodologies and tools, this project aimed to gain insights into the factors influencing rocket launch success rates and provide predictions for future launch outcomes.

Introduction

- In this project we'll discuss about SpaceX's Falcon 9 rocket, known for its cost-effectiveness compared to other rockets in the market. While Falcon 9 costs approximately $62 million, other rockets can be significantly more expensive, ranging up to $162 million. The key factor that makes Falcon 9 more affordable is its ability to reuse the first stage of the rocket. However, unsuccessful landings may hinder the recovery process.
- The objective of this project is to determine the price of each launch, which is directly influenced by the successful recovery of the first stage of the rocket. To achieve this, we will analyze the data from previous Falcon 9 launches to identify trends and patterns. By leveraging this historical data, we will develop a machine learning model that can be trained to predict the likelihood of a successful landing for future launches.
- Through this project, we aim to provide valuable insights into optimizing the recovery process and enhancing cost-efficiency for SpaceX's Falcon 9 rocket. By accurately predicting successful landings, we can contribute to informed decision-making regarding pricing strategies and overall operational effectiveness.

Section 1

Methodology

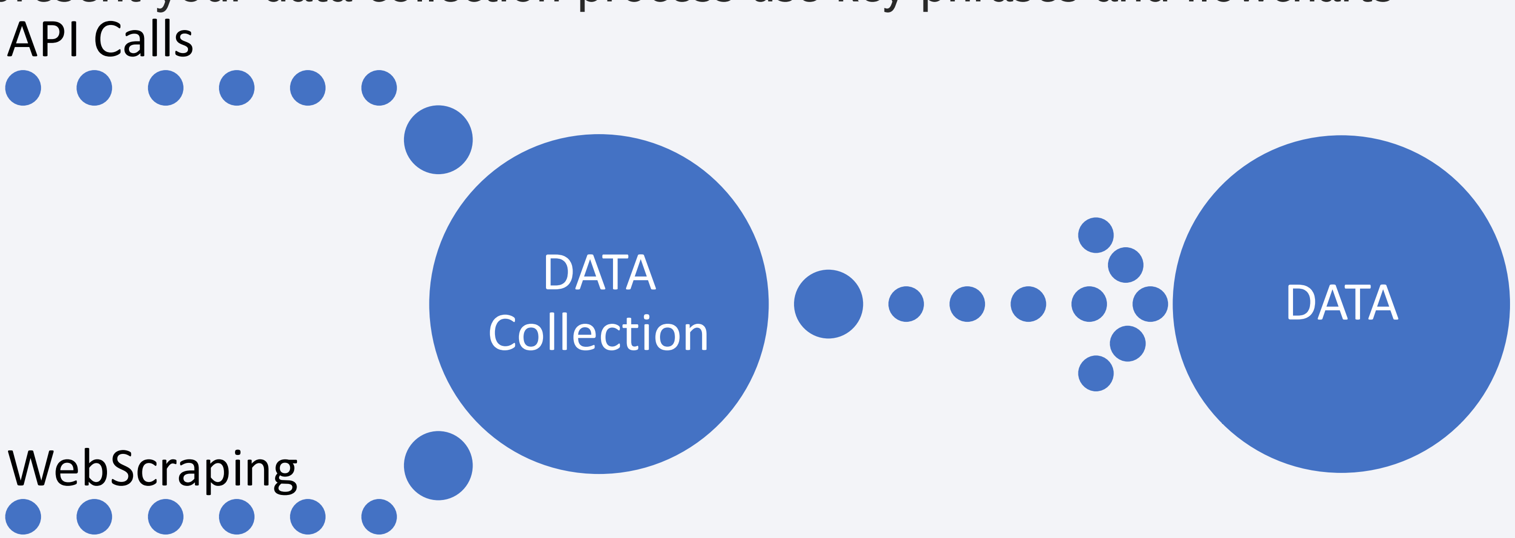
Methodology

Executive Summary

- Data collection methodology:
  - The data for SpaceX was collected using SpaceX API response and by WebScraping of Falcon 9 record from Wikipedia.
- Perform data wrangling
  - Checked for the null values and datatypes of each column, followed by exploring the number of Launch Sites and Orbits and Outcomes.
- 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

- The data for SpaceX Launch was collected using two methods:
    1. Data was collected by making a REST API call to SpaceX API.
    2. Data was extracted by Web Scraping from Wikipedia website
- You need to present your data collection process use key phrases and flowcharts



Data Collection – SpaceX API

- The methodology for data collection using API is depicted in the flowchart aside.
- Refer to Juptyer Notebook of Data Collection using the following link:
  [GitHub Link Jupyter Notebook: Data Collection Using SpaceX API]

Importing Libraries
- Requests
- Pandas
- Numpy
- Datetime

Defining Functions to Extract Data
- getBoosterVersion(data)
- getLaunchSite(data)
- getPayloadData(data)
- getCoreData(data)

Using Requests to make API call
- Requests.get(url)
- Using json_normalize method to get pandas dataframe named 'data'

Creating relevant DataFrame
- Removed unnecessary rows by using mapping function to columns like 'cores', 'payload' and converting 'datetime'. Data was restricted to date 13th Nov 2020.
- Created a new dataframe 'df' and populated data using the defined functions.

Filtering and Data Wrangling
- Dataframe was filtered to keep data for Booster version 'Falcon 9'
- Then, null values were checked and reported in Payload Mass which were replaced with mean values.

Data Collection - Scraping

- The methodology for data collection using Web Scrapping is depicted in the flowchart aside.
- Refer to Juptyer Notebook of Data Collection using the following link:
  [GitHub Link Jupyter Notebook: Data Collection Using Web Scrapping]

Importing & Installing Libraries
- Beautiful Soup
- Requests
- Pandas
- Unicodedata
- re

Defining Functions to Extract Data
- date_time(table_cells)
- booster_version(table_cells)
- landing_status(table_cells)
- get_mass(table_cells)
- extract_column_from_header(row)

Using Requests to get HTML page content
- Requests.get(url)
- Using beautiful soup to extract data from HTML response text
- Created a list of non-empty column names.

Creating relevant Dataframe
- Created a dictionary ‘launch_dict’ with keys from above column names and extracted data from response using the above defined functions
- Created a new dataframe ‘df’ using the dictionary ‘launch_dict’

Data Wrangling

- The methodology for Data Wrangling is depicted in the flowchart aside.
- Refer to Juptyer Notebook of Data Wrangling using the following link:
    - [GitHub Link Jupyter Notebook: Data Wrangling]

Importing Libraries
- Pandas
- Numpy
- Js
- Io

Fetching & Loading Data
- CSV Data file was fetched using js.fetch(URL) and downloaded using io
- Dataframe was created by loading the file using pandas

Dealing with Null Values
- Null values in the entire dataframe was checked and no null value was found

Launch Sites, Orbits & Outcomes
- Calculated the total count of Launch Sites
- Calculated the total count of Orbits
- Calculated the total count of Launch Outcomes
- Created a new column 'Class' for labelling the outcome of each launch site using 0 (Fail) and 1(Success)

EDA with Data Visualization

- Exploratory Data Analysis between variables to plot graphs and to check for trends of successful landing. The EDA were plotted for the following variables:
  1. Flight Number vs Payload Mass
  2. Flight Number vs Launch Site
  3. Payload Mass vs Launch Site
  4. Orbit vs Class
  5. Flight Number vs Orbit
  6. Payload vs Orbit
  7. Yearly Success Trend
- Refer to Juptyer Notebook of EDA using Data Visualization using the following link:
  - GitHub Link Jupyter Notebook: EDA using Data Visualization

EDA with SQL

- EDA was also performed using SQL-Lite and executed the following queries:
    1. %sql SELECT DISTINCT("Launch_Site") FROM SPACEXTBL
    2. %sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE '%CCA%' LIMIT 5
    3. %sql SELECT COUNT(PAYLOAD_MASS__KG_),Customer FROM SPACEXTBL WHERE Customer=='NASA (CRS)'
    4. %sql SELECT AVG(PAYLOAD_MASS__KG_),Booster_Version FROM SPACEXTBL WHERE Booster_Version LIKE '%F9 v1.1%'
    5. %sql SELECT Date,Landing_Outcome FROM SPACEXTBL WHERE Landing_Outcome == 'Success (ground pad)' ORDER BY Date DESC
    6. %sql SELECT Booster_Version,Landing_Outcome,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE (Landing_Outcome == 'Success (drone ship)' AND PAYLOAD_MASS__KG_>4000 AND PAYLOAD_MASS__KG_<6000)
    7. %sql SELECT COUNT(*),Mission_Outcome FROM SPACEXTBL GROUP BY Mission_Outcome

EDA with SQL

8. %sql SELECT Booster_Version,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
9. %sql SELECT SUBSTR(Date,4,2) AS MONTH,Landing_Outcome,Booster_Version,Launch_Site FROM SPACEXTBL WHERE (Landing_Outcome =='Failure (drone ship)' AND SUBSTR(Date,7,4)=='2015')
10. %sql SELECT Date,Landing_Outcome FROM SPACEXTBL WHERE (Landing_Outcome LIKE '%Success%'AND SUBSTR(Date,7,4)>'2010' AND SUBSTR(Date,7,4)<'2017') ORDER BY SUBSTR(Date,7,4) DESC

- Refer to Juptyer Notebook of EDA using SQL by using the following link:
  GitHub Link Jupyter Notebook: EDA using SQL

Build an Interactive Map with Folium

- Next, Geographical visualization using Folium library was done to depict the exact locations on map. To do this, various map objects were used to help visualize the locations.
  - To mark the locations on map, folium marker object was used and all the co-ordinates of sites were marked.
  - To highlight the marker on zoomed out map, circle object was used.
  - To distinguish between successful and unsuccessful launches, green and red colored markers were used depending on the class column.
  - To group these markers, marker cluster object was added to the map.
  - A mouse position object was added to display the map co-ordinates on hovering the mouse cursor to help determine the co-ordinates of point and later use it for measuring distance.
  - To mark the distance between site and coastline, a polyline object was used.
- Refer to Juptyer Notebook of Interactive Map using Folium by using the following link:

  [GitHub Link Jupyter Notebook: Interactive Map with Folium]

Build a Dashboard with Plotly Dash

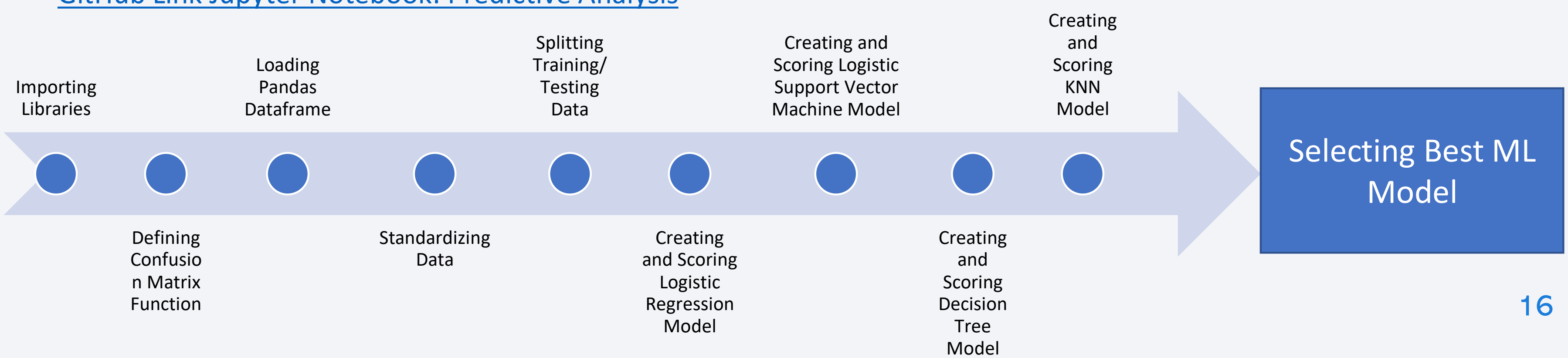
- An interactive Dashboard using Plotly Dash is also built to help understand the success rate from each Launch Site.
- A pie chart with drop down of Launch Sites was established showing the successful landing outcomes percentage for the selected launch site.
- A payload mass range slider was introduced to filter and select the payload range.
- A Scatter Plot between Payload Mass and Landing Outcome was plotted for the selected Launch Site from the dropdown.
- Additionally, in dropdown “All Sites” option was also added to show the pie plot and scatter plot considering all sites.
- Refer to Python Code of Plotly Dash App by using the following link:

  [GitHub Link : Plotly Dash App]

Predictive Analysis (Classification)

- The predictive analysis was done using several Machine Learning Models and Confusion Matrix for each model was plotted.
- The Flow Chart below shows the outline of doing Predictive Analysis of four models viz. Logistic Regression, Support Vector Machine (SVM), Decision Tree and K-Nearest Neighbor (KNN) Models.
- For each model, hyperparameters were optimized using GridSearchCV and accuracy was calculated for each ML model.
- Refer to Python Code of Plotly Dash App by using the following link:

  [GitHub Link Jupyter Notebook: Predictive Analysis]



Results

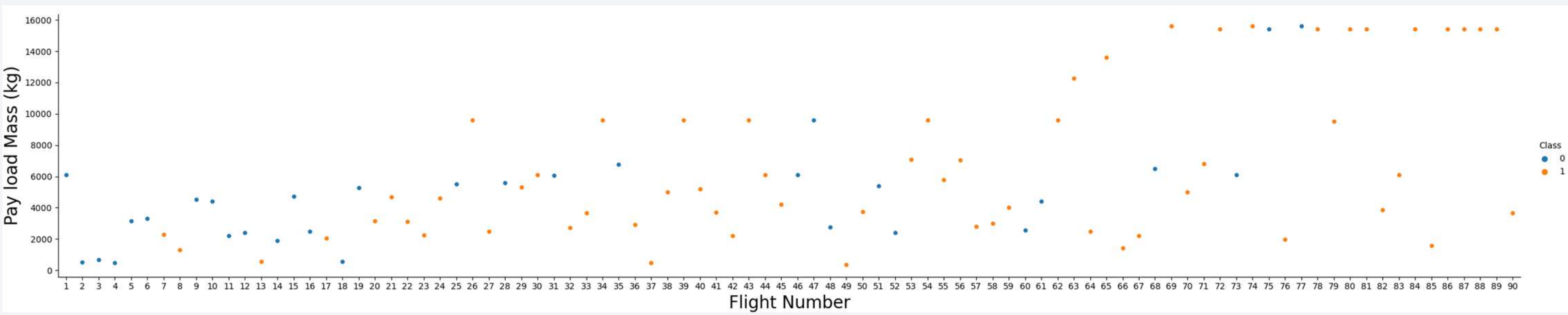
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

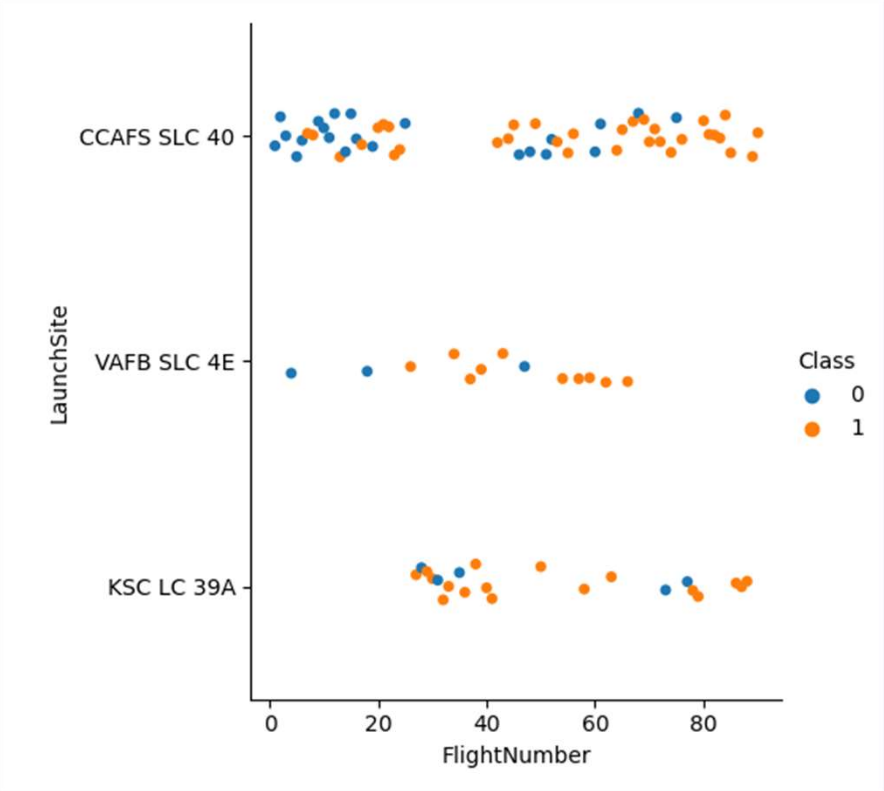
Flight Number vs. Payload Mass

- A Categorization Scatter Plot was plotted between Flight Number and Payload Mass using Seaborne library and depicting the launch outcomes.
- Here the Orange points corresponds to successful landing and blue corresponds to unsuccessful landing
- We can clearly see that as the Flight Number increases, the orange points increases i.e. as the flight number increases, the first stage is more likely to land successfully.



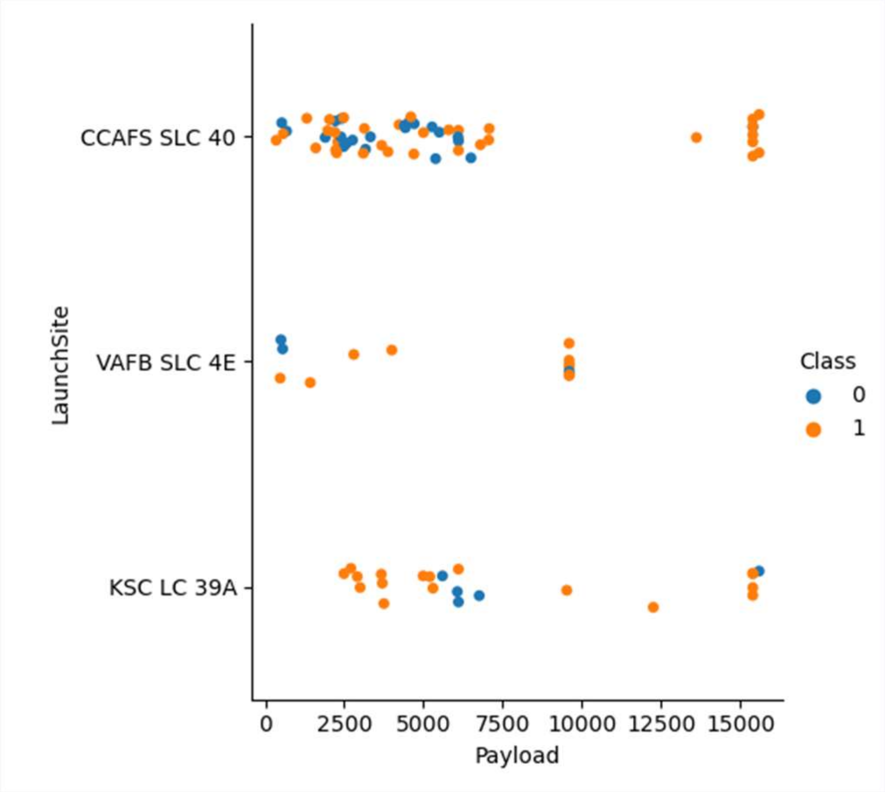
Flight Number vs. Launch Site

- A Categorization Scatter Plot was plotted between Flight Number and Launch Site using Seaborne library and depicting the launch outcomes.
- Here the Orange points corresponds to successful landing and blue corresponds to unsuccessful landing
- We can see that as the Flight Number increases, the orange points increases for the sites ‘VAFB’ and ‘KSC’ i.e. as the flight number increases, the first stage is more likely to land successfully. Whereas no such trend can be seen for site ‘CCAFS’



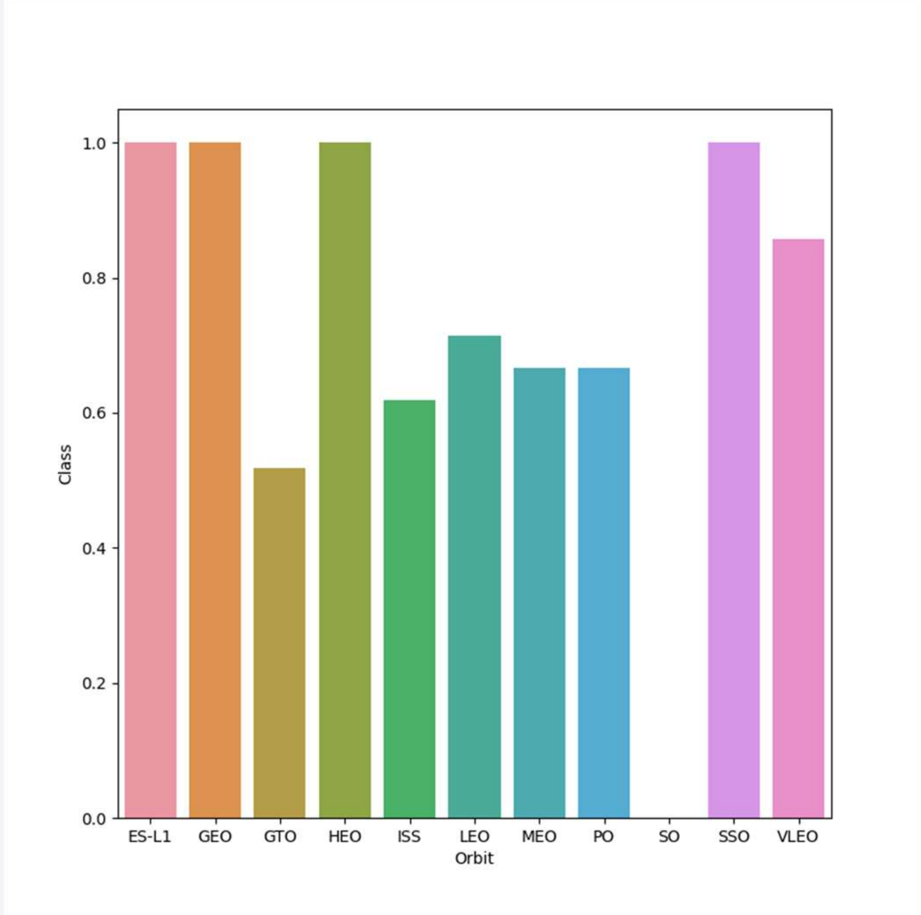
Payload vs. Launch Site

- A Categorization Scatter Plot was plotted between Payload Mass and Launch Site using Seaborne library and depicting the launch outcomes.
- Here the Orange points corresponds to successful landing and blue corresponds to unsuccessful landing
- We can see that as the Payload Mass increases, the orange points increases for the sites ‘VAFB’ and ‘KSC’ i.e. as the mass increases, the first stage is more likely to land successfully.
- Whereas for site ‘CCAFS’, for heavy load where mass >12500 Kg, the first stage is more likely to land successfully.



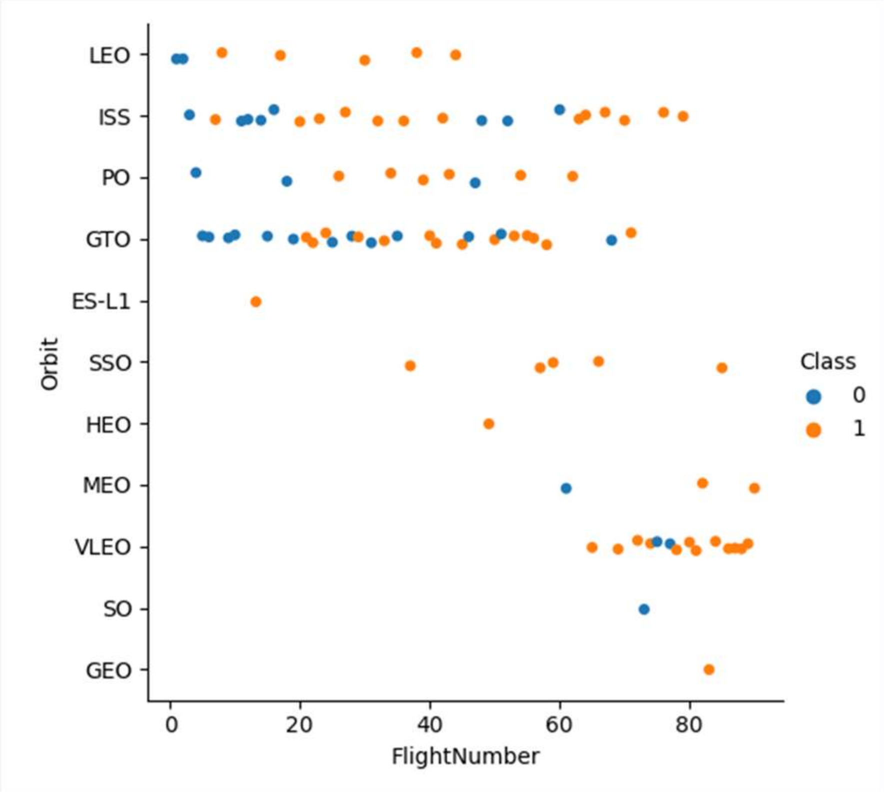
Success Rate vs. Orbit Type

- A Bar Chart was plotted between Orbit and Launch Outcome by grouping the orbits and taking mean values of dataframe.
- The orbits ES-L1, GEO, HEO, SSO have the highest success rate



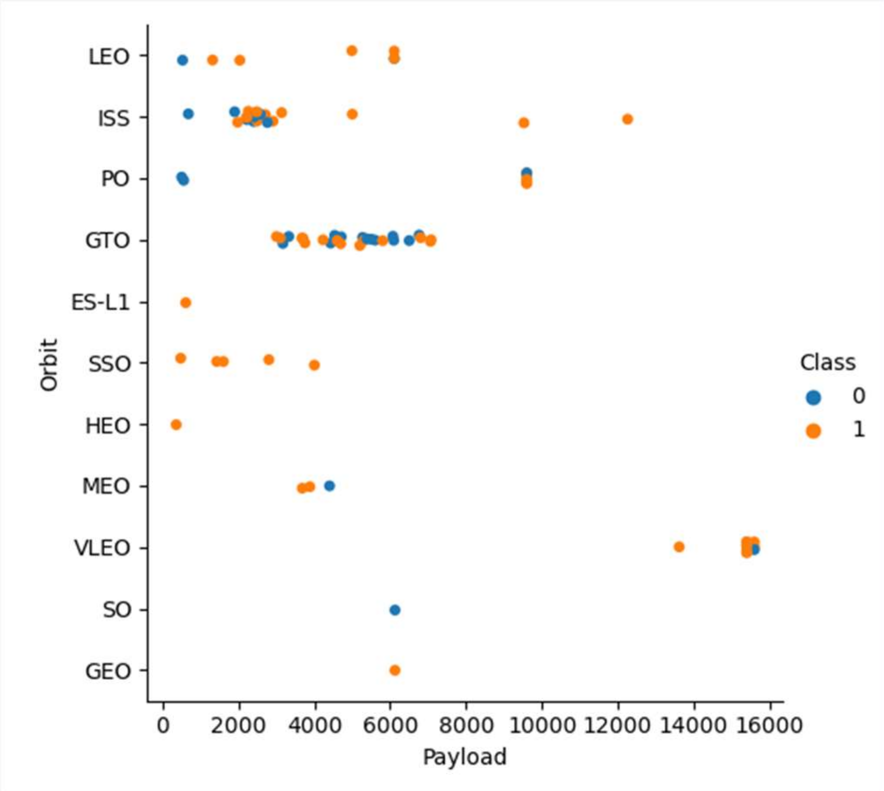
Flight Number vs. Orbit Type

- A Categorization Scatter Plot was plotted between Flight Number and various Orbits.
- Here the Orange points corresponds to successful landing and blue corresponds to unsuccessful landing
- We can see that for the LEO, MEO, VLEO orbits, Success Landing increases with increase in Flight Number whereas no such trend exists for GTO, ISS, PO orbits.
- Few orbits like ES-L1, SSO, HEO, GEO always have a successful landing and the orbit SO have an unsuccessful outcome.



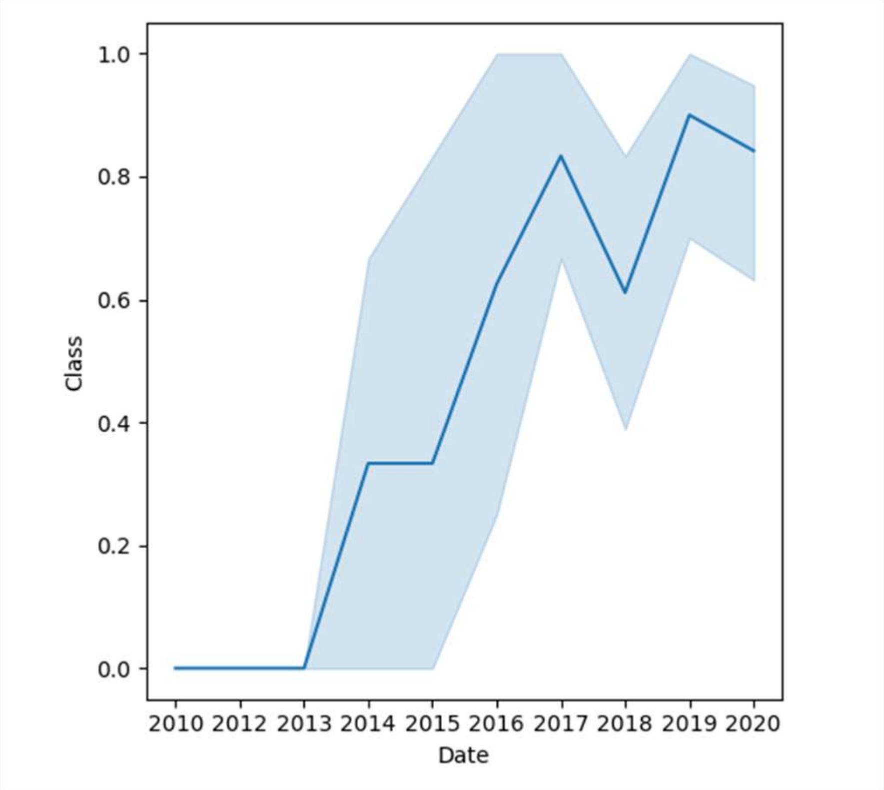
Payload vs. Orbit Type

- A Categorization Scatter Plot was plotted between Payload Mass and various Orbits.
- Here the Orange points corresponds to successful landing and blue corresponds to unsuccessful landing
- We can see that with heavy payloads the successful landing or positive landing rate are more for PO, LEO and ISS orbits.
- Few orbits like ES-L1, SSO, HEO, GEO always have a successful landing and the orbit SO have an unsuccessful outcome
- Orbits VLEO, MEO have successful outcomes for light payloads
- On the other hand, GTO has no trend with respect to Payload mass.



Launch Success Yearly Trend

- A Line Chart was plotted between the years of launch and Class to check for the outcomes of Successful Landing over the period.
- We can see that successful landing increased from the year 2013 to 2017, followed by a dip in year 2018 and again rising in the year 2020.



All Launch Site Names

- Unique Launch Sites was queried while doing EDA using SQL by the following command

%sql SELECT DISTINCT("Launch_Site") FROM SPACEXTBL

Task 1

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

In [7]:
```
%sql SELECT DISTINCT("Launch_Site") FROM SPACEXTBL
```

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

Out[7]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

- Launch Sites starting with ‘CCA’ was queried while doing EDA using SQL by the following command:

  %sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE '%CCA%' LIMIT 5

Task 2

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

In [8]:
```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE '%CCA%' LIMIT 5
```

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

Out[8]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (paracl
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (paracl
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No att
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No att
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No att

Total Payload Mass

- Total Payload Mass carried by NASA was queried while doing EDA using SQL by the following command:
  - %sql SELECT COUNT(PAYLOAD_MASS__KG_),Customer FROM SPACEXTBL WHERE Customer=='NASA (CRS)'

Task 3

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

In [9]:
```
%sql SELECT COUNT(PAYLOAD_MASS__KG_),Customer FROM SPACEXTBL  WHERE Customer=='NASA (CRS)'
```

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

Out[9]:

COUNT(PAYLOAD_MASS__KG_)	Customer
20	NASA (CRS)

Average Payload Mass by F9 v1.1

- Average Payload Mass carried by F9 v1.1 was queried while doing EDA using SQL by the following command:

  %sql SELECT AVG(PAYLOAD_MASS__KG_),Booster_Version FROM SPACEXTBL WHERE Booster_Version LIKE '%F9 v1.1%'

Task 4

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

```
[10]: %sql SELECT AVG(PAYLOAD_MASS__KG_),Booster_Version FROM SPACEXTBL  WHERE Booster_Version LIKE '%F9 v1.1%'

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

[10]:

AVG(PAYLOAD_MASS__KG_)	Booster_Version
2534.6666666666665	F9 v1.1 B1003

First Successful Ground Landing Date

- First Success Landing Outcome in ground pad was queried while doing EDA using SQL by the following command:

  %sql SELECT Date,Landing_Outcome FROM SPACEXTBL WHERE Landing_Outcome == 'Success (ground pad)' ORDER BY Date DESC

Task 5

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

Hint:Use min function

```
[11]: %sql SELECT Date,Landing_Outcome FROM SPACEXTBL  WHERE Landing_Outcome == 'Success (ground pad)' ORDER BY Date DESC
 * sqlite:///my_data1.db
Done.
```

[11]:

Date	Landing_Outcome
22/12/2015	Success (ground pad)
19/02/2017	Success (ground pad)
18/07/2016	Success (ground pad)
15/12/2017	Success (ground pad)
14/08/2017	Success (ground pad)
09/07/2017	Success (ground pad)
06/03/2017	Success (ground pad)
05/01/2017	Success (ground pad)
01/08/2018	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 was queried while doing EDA using SQL by the following command:
  - %sql SELECT Booster_Version,Landing_Outcome,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE (Landing_Outcome == 'Success (drone ship)' AND PAYLOAD_MASS__KG_>4000 AND PAYLOAD_MASS__KG_<6000)

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
[12]: %sql SELECT Booster_Version,Landing_Outcome,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.
```

[12]:

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696.0
F9 FT B1026	Success (drone ship)	4600.0
F9 FT B1021.2	Success (drone ship)	5300.0
F9 FT B1031.2	Success (drone ship)	5200.0

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes was queried while doing EDA using SQL by the following command:

%sql SELECT COUNT(*),Mission_Outcome FROM SPACEXTBL GROUP BY Mission_Outcome

Task 7

List the total number of successful and failure mission outcomes

```
[13]: %sql SELECT COUNT(*),Mission_Outcome FROM SPACEXTBL GROUP BY Mission_Outcome
 * sqlite:///my_data1.db
Done.
```

[13]:

COUNT(*)	Mission_Outcome
898	None
1	Failure (in flight)
98	Success
1	Success
1	Success (payload status unclear)

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass was queried while doing EDA using SQL by the following command:

  %sql SELECT Booster_Version,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

[14]:
```
%sql SELECT Booster_Version,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

[14]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600.0
F9 B5 B1049.4	15600.0
F9 B5 B1051.3	15600.0
F9 B5 B1056.4	15600.0
F9 B5 B1048.5	15600.0
F9 B5 B1051.4	15600.0
F9 B5 B1049.5	15600.0
F9 B5 B1060.2	15600.0
F9 B5 B1058.3	15600.0
F9 B5 B1051.6	15600.0
F9 B5 B1060.3	15600.0
F9 B5 B1049.7	15600.0

2015 Launch Records

- Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 was queried while doing EDA using SQL by the following command:
  - %sql SELECT SUBSTR(Date,4,2) AS MONTH,Landing_Outcome,Booster_Version,Launch_Site FROM SPACEXTBL WHERE (Landing_Outcome =='Failure (drone ship)' AND SUBSTR(Date,7,4)=='2015')

Task 9 ¶

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.

[15]:
```
%sql SELECT SUBSTR(Date,4,2) AS MONTH,Landing_Outcome,Booster_Version,Launch_Site FROM SPACEXTBL WHERE (Landing_Outcome =='Failure (drone ship)' AND SUBSTR(Date,7,4)=='2015')
```

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

[15]:

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

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

- Rank of 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 was queried while doing EDA using SQL by the following command:

  %sql SELECT Date,Landing_Outcome FROM SPACEXTBL WHERE (Landing_Outcome LIKE '%Success%'AND SUBSTR(Date,7,4)>'2010' AND SUBSTR(Date,7,4)<'2017') ORDER BY SUBSTR(Date,7,4) DESC

Task 10

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

In [16]:
```
%sql SELECT Date,Landing_Outcome FROM SPACEXTBL WHERE (Landing_Outcome LIKE '%Success%'AND SUBSTR(Date,7,4)>'2010' AND SUBSTR
```

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

Out[16]:

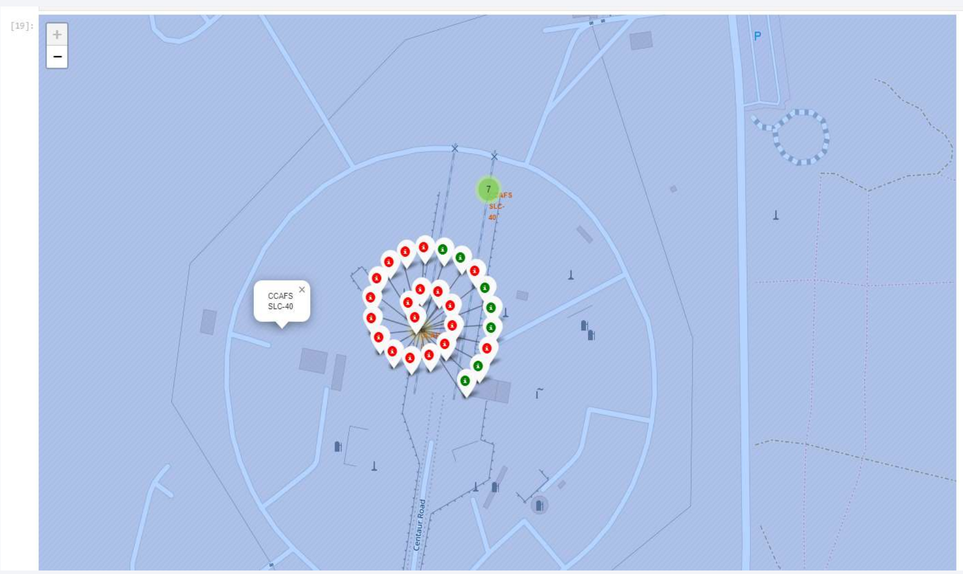
Date	Landing_Outcome
04/08/2016	Success (drone ship)
05/06/2016	Success (drone ship)
27/05/2016	Success (drone ship)
18/07/2016	Success (ground pad)
14/08/2016	Success (drone ship)
22/12/2015	Success (ground pad)

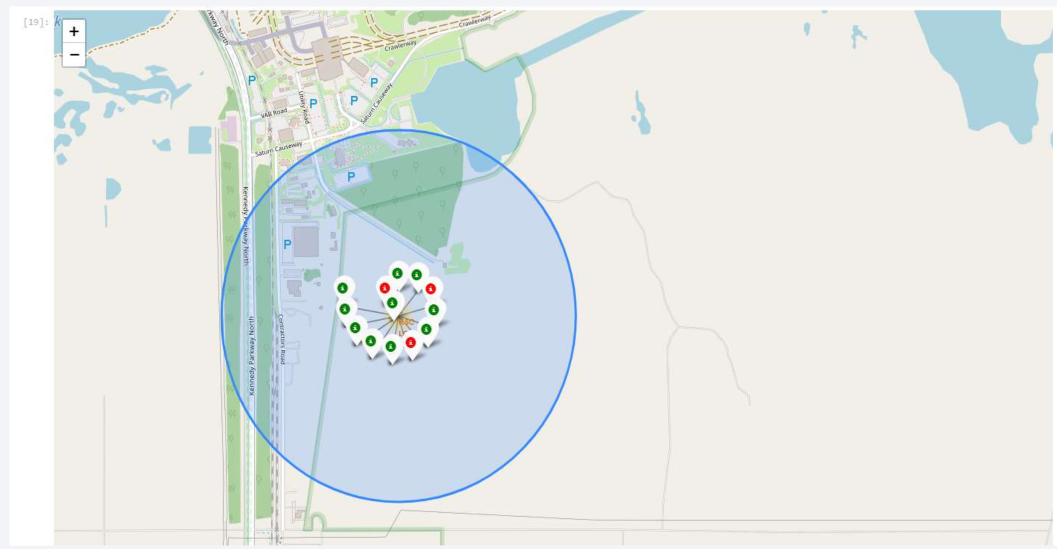
Section 3

Launch Sites Proximities Analysis

Map showing Successful/Failed Launches from each site

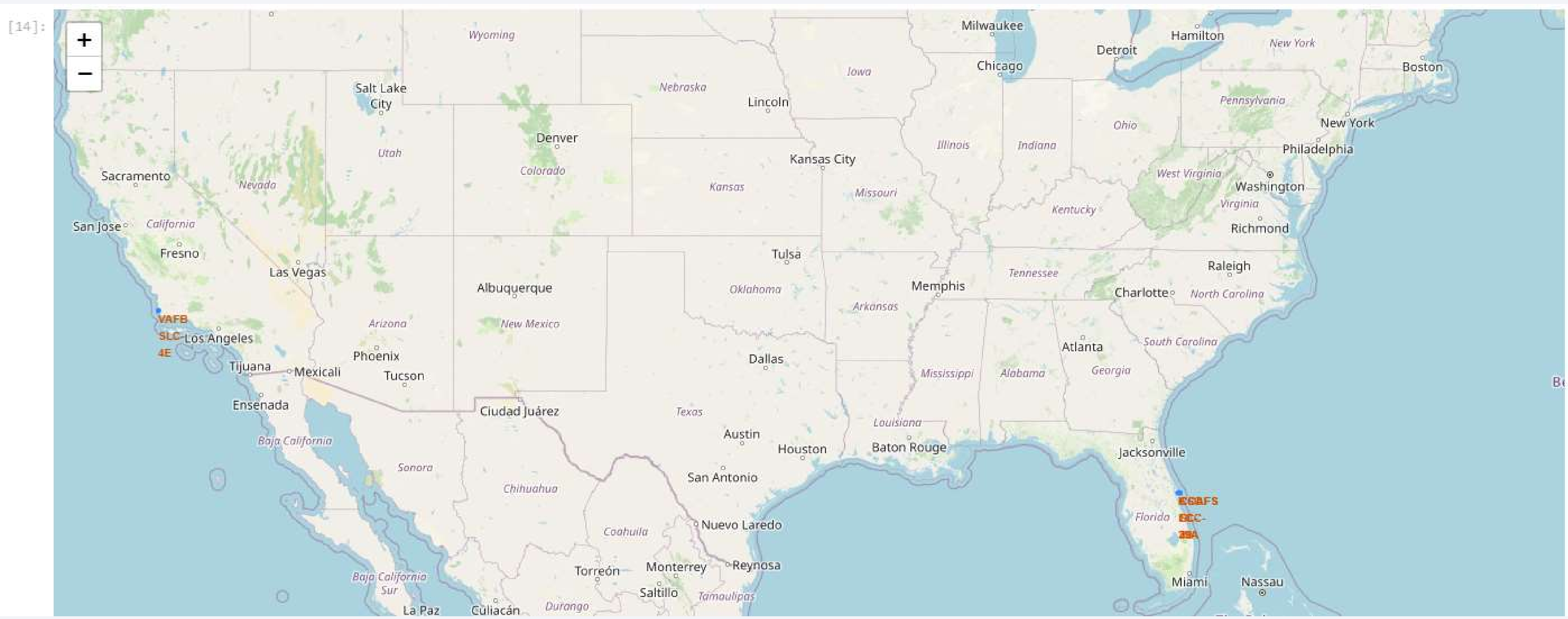
- From the screenshots attached below, we can visualize the number of successful and failed launches from the site where green markers show the successful launch and red markers depicting the failed launches





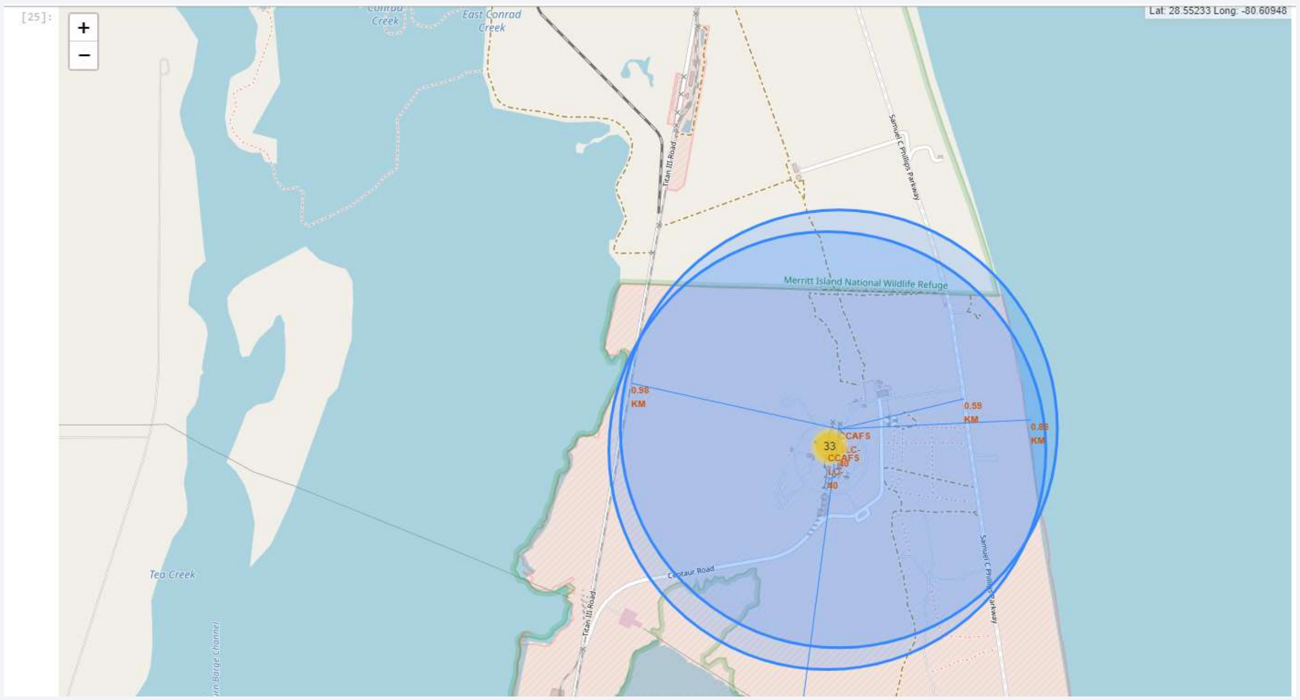
Map depicting all Launch Sites

- The location of site is shown in the screenshot below. It is clearly visible that the launch sites were chosen near the coastline line



Map showing the proximities from CCAFS Site

- Launch Site 'CCAFS' was in close proximity to Highway line (0.59 Km), followed by Coast line (0.88 Km), then Railway line (0.98 Km) and lastly the distance of nearest city (52Km).

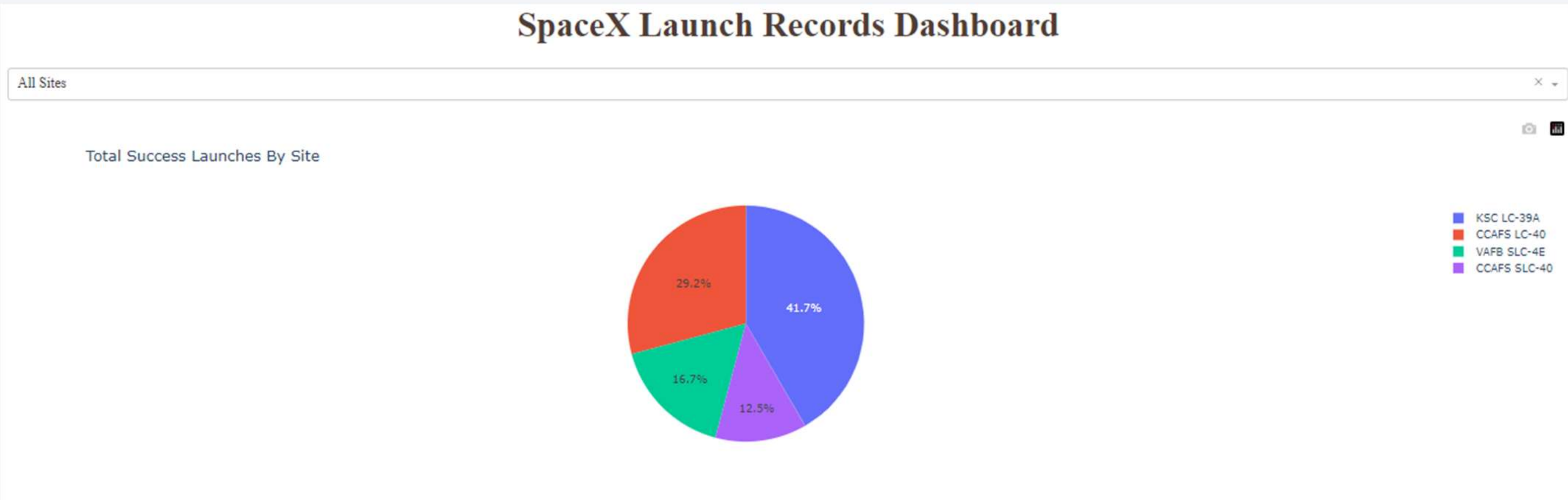


Section 4

Build a Dashboard with Plotly Dash

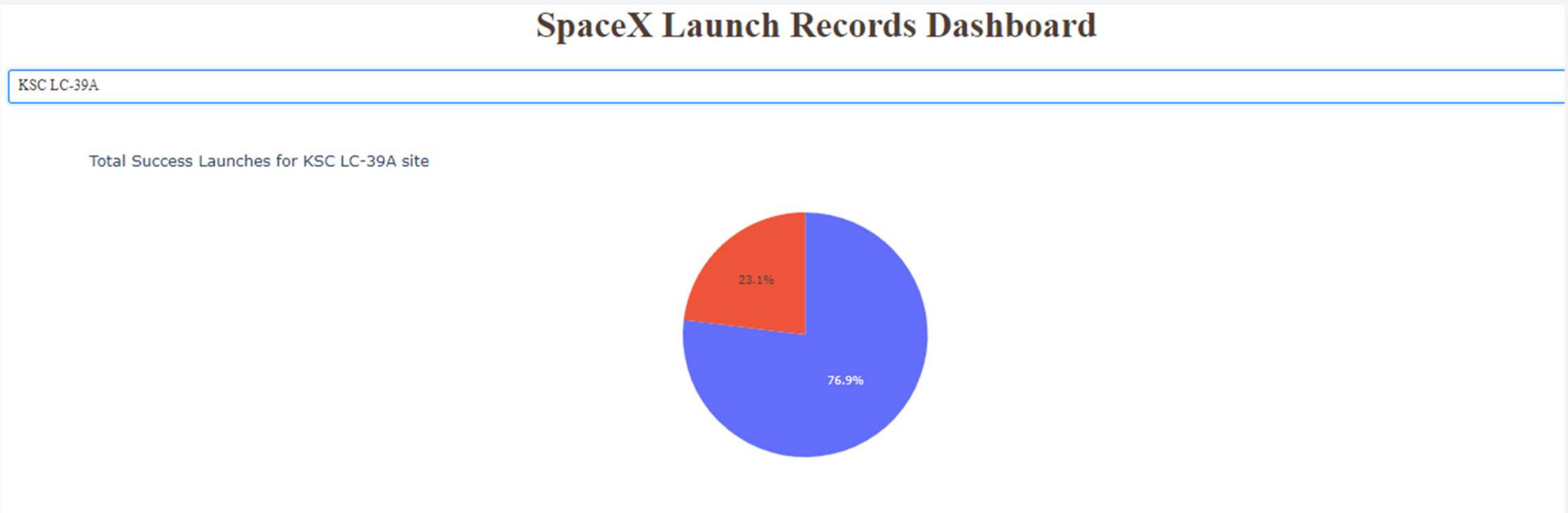
Plotly Dashboard showing Pie Chart Success rate plot for All Sites

- The pie chart illustrates the distribution of success rates among four areas. The KSC area has the highest success rate, accounting for 41.7% of the total. CCAFS-LC follows with a success rate of 29.2%, while VAFB and CCAFS-SLC have success rates of 16.7% and 12.5% respectively.



Plotly Dashboard showing Pie Chart plot for KSC LC-39A site

- The pie chart illustrates the success rate of 76.9% for launches from the site i.e. 76.9% launches from this site were successfully landed back whereas 23.1% stage-one failed while landing.

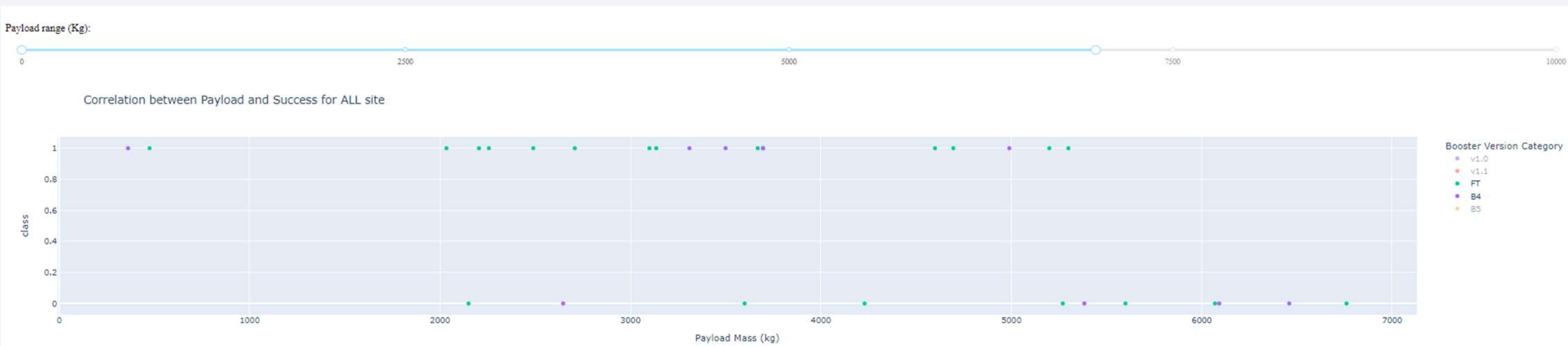


Plotly Dashboard showing Scatter plot for All Sites

- The Payload vs Launch Outcome scatter plot depicting different booster versions for a selected payload range is shown below.
- The plot describes mostly the success rate for booster versions is found for all versions except ‘v1.0’ booster over range of payload mass upto 10000 Kg

Plotly Dashboard showing Scatter plot for All Sites

- The Payload vs Launch Outcome scatter plot depicting different booster versions for a selected payload range is shown below.
- The plot also describes the booster version FT, B4 having the highest success rate for payload range between 0 and 5500 Kg mass.

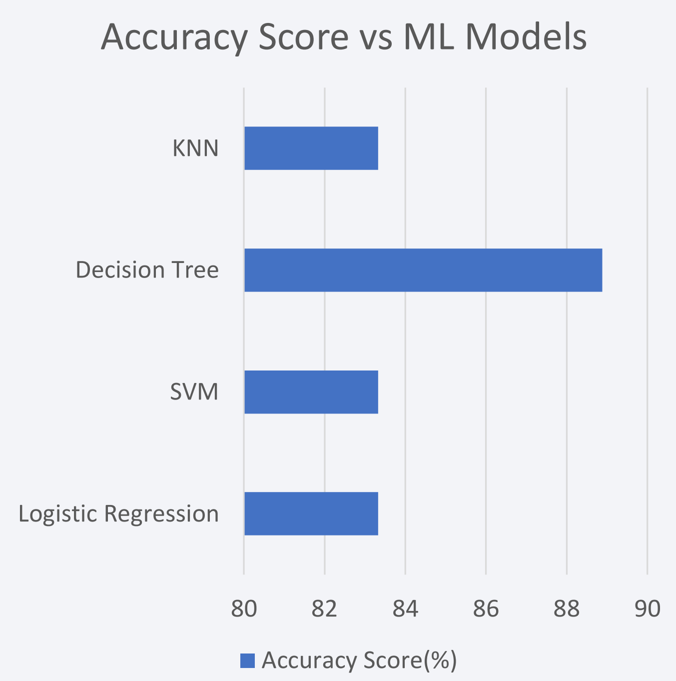


Section 5

Predictive Analysis (Classification)

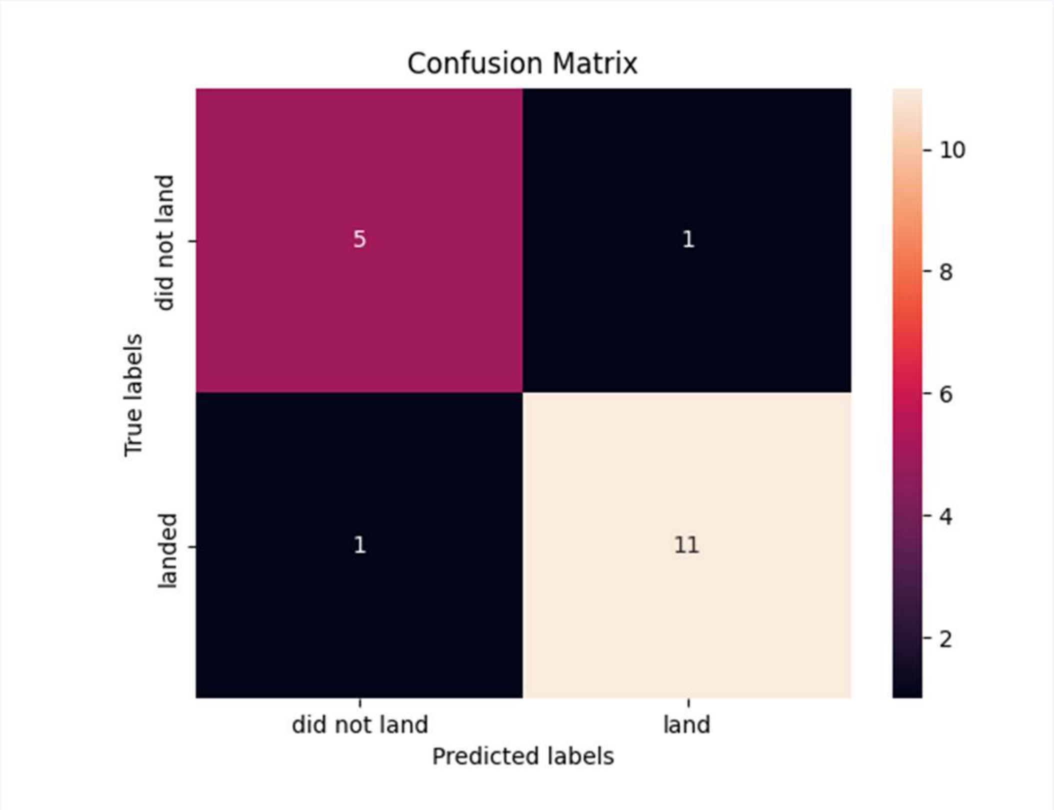
Classification Accuracy

- A bar chart is plotted for the accuracy percentage of the four Machine Learning Models viz. Logistic Regression, Decision Tree, Support Vector Machine and K-Nearest Neighbor
- Decision Tree Model showed the highest accuracy of 88.88% whereas left three models showed the same accuracy of 83.33% suggesting comparable performance.



Confusion Matrix

- The confusion matrix of Decision Tree Model is shown aside.
- Model correctly predicted landing outcome of 11 launches and failed landing of 5 launches whereas it incorrectly predicted 1 successful landing and 1 unsuccessful landing



Conclusions

- In conclusion, the analysis of the provided data offers valuable insights into the success rates and factors influencing rocket landings. These findings have significant implications for SpaceX and the broader space industry.
- Firstly, the relationship between flight number and successful first stage landings is evident. As the flight number increases, there is a clear upward trend in the success rate, indicating a continuous improvement in rocket design, manufacturing, and operational processes. This observation underscores SpaceX's commitment to iterative development and continuous refinement of their rockets.
- Secondly, the success rates across different launch sites reveal interesting trends. "VAFB" and "KSC" exhibit a positive correlation between flight number and successful landings, indicating that these sites are well-suited for achieving consistent and reliable results. However, for "CCAFS," no such trend is observed, suggesting potential site-specific challenges that impact landing outcomes. Further investigation into the unique characteristics of "CCAFS" can provide valuable insights for optimizing its operations and increasing success rates.
- The payload mass is another crucial factor influencing successful landings. For "VAFB" and "KSC," an increase in payload mass corresponds to a higher likelihood of successful landings. This observation suggests that the design and capabilities of rockets deployed from these sites are well-adapted to handle heavier payloads. On the other hand, for "CCAFS," the success rate is higher for payloads exceeding 12500 Kg. This finding indicates that specific considerations must be taken into account when launching heavier payloads from this site.

Conclusions

- Analyzing the success rates across different orbits provides further insights. Orbits such as ES-L1, GEO, HEO, and SSO consistently demonstrate high success rates, indicating their suitability for successful landings. Conversely, the SO orbit consistently yields unsuccessful landings. These findings can inform mission planning and payload allocation decisions, ensuring that the most appropriate orbits are selected to maximize the probability of successful landings.

- Examining the success rates over time, it is evident that there has been a positive trend from 2013 to 2017, indicating continuous advancements and improvements in rocket technology and operational procedures. However, a slight dip in success rates in 2018 followed by a subsequent rise in 2020 raises important questions. Further analysis and investigation into the underlying factors driving these fluctuations can provide valuable insights for mitigating risks and ensuring consistent success rates in the future.

- Overall, the analysis of the provided data highlights the importance of meticulous planning, continuous refinement, and the utilization of historical data for achieving successful rocket landings. The findings offer valuable guidance for SpaceX and other space agencies in optimizing their operational processes, reducing costs, and increasing overall mission success rates.

- In conclusion, the analysis provides valuable insights into the factors influencing successful rocket landings. These findings contribute to informed decision-making, optimization of operational processes, and continued advancements in rocket design and engineering. By leveraging these insights, SpaceX and other space agencies can improve success rates of landing, reduce costs, and enhance the overall efficiency and reliability of space missions. The quest for successful landings is crucial for pushing the boundaries of space exploration by determining price of rocket launch.


Thank you!