Outline

- Executive Summary

- Introduction

- Methodology

- Results

- Conclusion

- Appendix

Executive Summary

- This Capstone Project implemented methods for Data Collection, Data Scraping, Exploratory Data Analysis, Folium Maps, Making Dashboard and finally implementing various classifier models for prediction on SpaceX test data

- We observed that with fine tuning of hyper-parameters we can improve the overall accuracy score of classification models.

- Machine Learning models could accurately predict the launch outcomes displaying its capabilities which may help in growing business in various important aspects.

Introduction

- In this capstone, we will predict if the Falcon 9 first stage will land successfully. 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

- This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. Instead of using rocket science to determine if the first stage will land successfully, we will train a machine learning model and use public information to predict if SpaceX will reuse the first stage.

Section 1

Methodology

Methodology

- Data collection methodology:

In this capstone project, we will be working with SpaceX launch data that is gathered from an API, specifically the SpaceX REST API. This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome. The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/.

- Perform data wrangling:

We used to get request using requests library to obtain launch data from API. Our response will be in json format. To convert this JSON to a dataframe, we can use the json_normalize function. This function will allow us to "normalize" the structured json data into a flat table.

- Perform exploratory data analysis (EDA) using visualization and SQL

- Perform interactive visual analytics using Folium and Plotly Dash

- Perform predictive analysis using classification models.

6

Data Collection

- The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/. We used it to target a specific endpoint of the API to get past launch data. We used get request using requests library to obtain launch data from API. Our response will be in json format. To convert this JSON to a dataframe, we can use the json_normalize function. This function will allow us to "normalize" the structured json data into a flat table form.

- Another popular data source for obtaining Falcon 9 Launch data is web scraping related Wiki pages. We implemented the Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records. Further we parsed the data from those tables and converted them into a Pandas data frame for further visualization and analysis
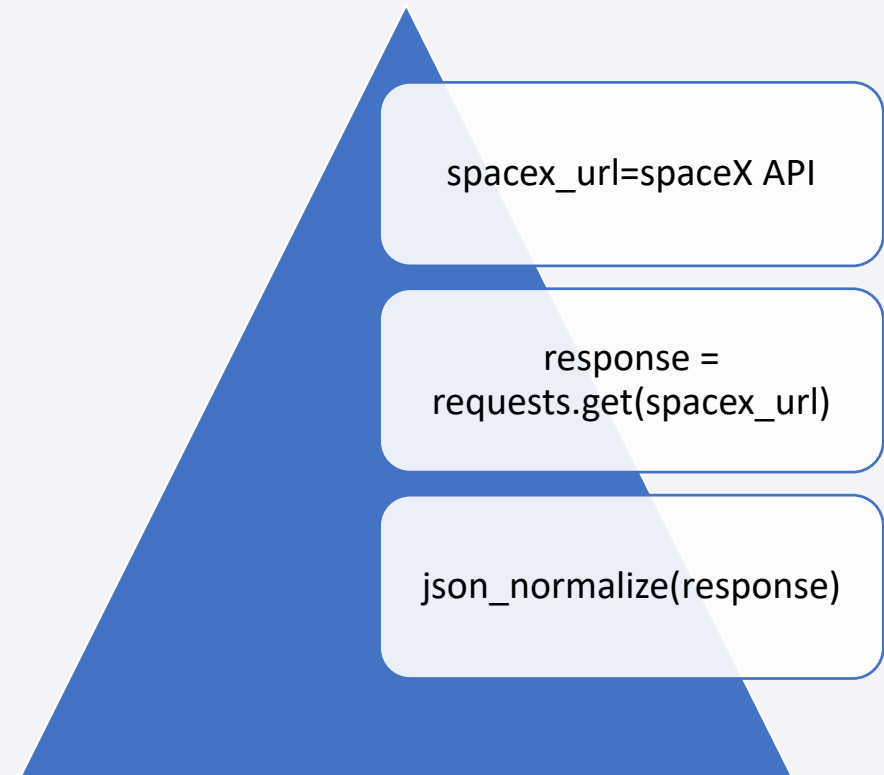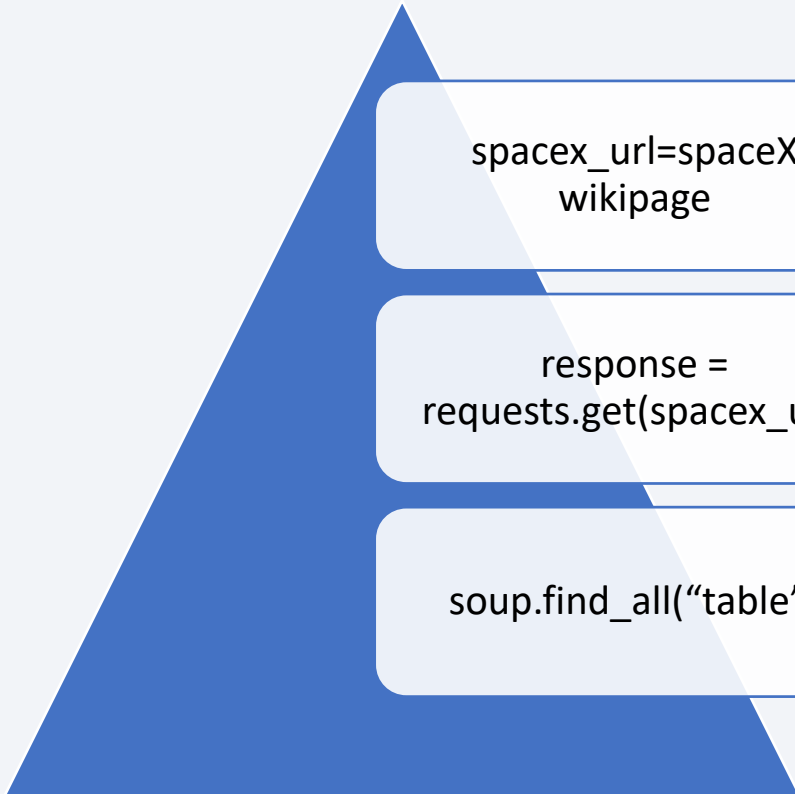
Data Collection – SpaceX API

1.SpaceX API to get past launch data

2. Import requests library to parse data from API or URL which will be in json format.

3. Use json_normalize() to convert json format response into flat tables

GitHub –

https://github.com/santo-mantras/IBM_CapstoneProject.git

Flowchart of SpaceX API calls

spacex_url=spaceX API

response = requests.get(spacex_url)

json_normalize(response)

Data Collection - Scraping

- We implemented the Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records. Further we parsed the data from those tables and converted them into a Pandas data frame for further visualization and analysis

GitHub –

https://github.com/santo-mantras/IBM_CapstoneProject.git

Flowchart of Web Scraping

spacex_url=spaceX wikipage

response = requests.get(spacex_url)

soup.find_all("table")

Data Wrangling

1. Identify and calculate the percentage of the missing values in each attribute.

2. Calculate the number of launches on each site.

3. Calculate the number and occurrence of mission outcome per orbit type.

4. Create a landing outcome label from Outcome column i.e., 1 for success and 0 for failure.

GitHub -

https://github.com/santo-mantras/IBM_CapstoneProject.git

Flowchart of Data Wrangling

df.isnull().sum()/df.shape[0]*100

df['LaunchSite'].value_counts()

df['Outcome'].value_counts()
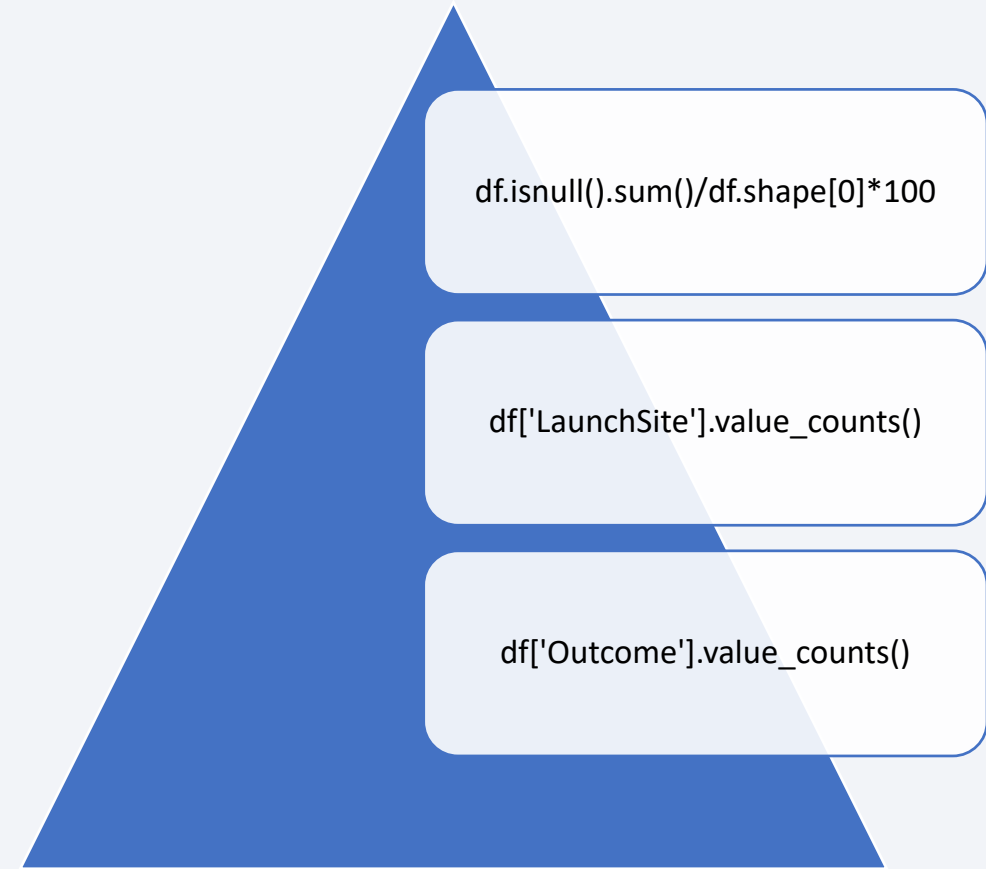
EDA with Data Visualization

- Graphs Plotted using seaborn as sns and matplot.pyplot as plt library

1. Scatter Plot (sns.catplot) : to observe if there is any relationship between launch sites and their payload mass.

2. Bar Chart (plt.bar) : to visually check if there are any relationship between success rate and orbit type.

3. Line Plot (sns.lineplot) : x axis as 'Year' and y axis to be average success rate, to get the average launch success trend.

- GitHub - https://github.com/santo-mantras/IBM_CapstoneProject.git

EDA with SQL

- SQL queries implemented –

1. 'Select' query to fetch the required rows from the table.

2. Select sum(Payload_mass) and avg(Payload_mass) to display total payload mass and avg payload mass.

3. Select Date query for displaying Dates from SpaceX table.

4. Select Count query to list the total number of successful and failure mission outcomes.

5. Using Sub queries to perform complex function like listing names of booster versions which have carried the maximum payload mass.

- GitHub - https://github.com/santo-mantras/IBM_CapstoneProject.git

Build an Interactive Map with Folium

- We used Folium for Interactive visual analytics following are some key attributes –

1. folium.Map : with an initial center location to be NASA Johnson Space Center at Houston.

2. folium.Circle : to add a highlighted circle area with a text label on a specific coordinate.

3. folium.marker : to Mark all launch sites on a map.

4. folium.icon : customize the Marker's icon property to indicate if launch was success or failed.

5. folium.PolyLine : to create a marker with distance to a closest city, railway, highway etc. and draw a line between the marker to the launch site.

- GitHub - https://github.com/santo-mantras/IBM_CapstoneProject.git

Build a Dashboard with Plotly Dash

- We have implemented Plotly Dash application to create Dashboard and perform interactive visual analytics on SpaceX launch data in real-time.

- This dashboard application contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart.

- Following are the important elements of Dashboard –

  1. Launch Site Drop-down Input Component.

  2. A callback function to render success pie chart based on selected site dropdown.

  3. A Range Slider to Select Payload.

  4. A callback function to render the scatter plot.

- GitHub - https://github.com/santomantras/IBM_CapstoneProject.git

Predictive Analysis (Classification)

1. Building the Model -

❖ Performed exploratory Data Analysis to determine training labels.

❖ Created a column for the class as 'O' for fail and '1' for successful launch.

❖ Standardize the data and Split into training data and test data.

2. Evaluation and Improvement -

• Finding best Hyperparameter for SVM, Classification Trees and Logistic Regression and the method which performs best using test data.

• GitHub - https://github.com/santo-mantras/IBM_CapstoneProject.git

Flowchart of Model Improvement

Finding Best Hyperparameters

Fine tune the model using hyperparameters

Evaluate Model on test data
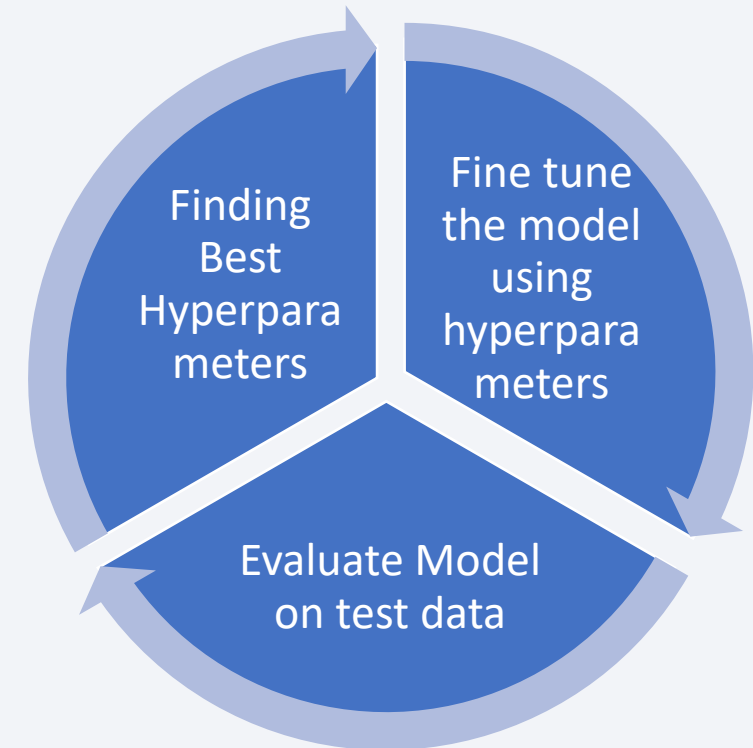
Results

- Exploratory Analysis Results comprises of different plots and screenshots which are mentioned from Section - 2, Section-3 and Section-4 of this presentation.

- Interactive analytics demo in screenshots is also provided from Section-2 onwards.

- Predictive analysis results are provided in the final section i.e., Section-5 of this presentation that consist of prediction accuracy, confusion matrix of different models that are implemented on our Space X dataset.
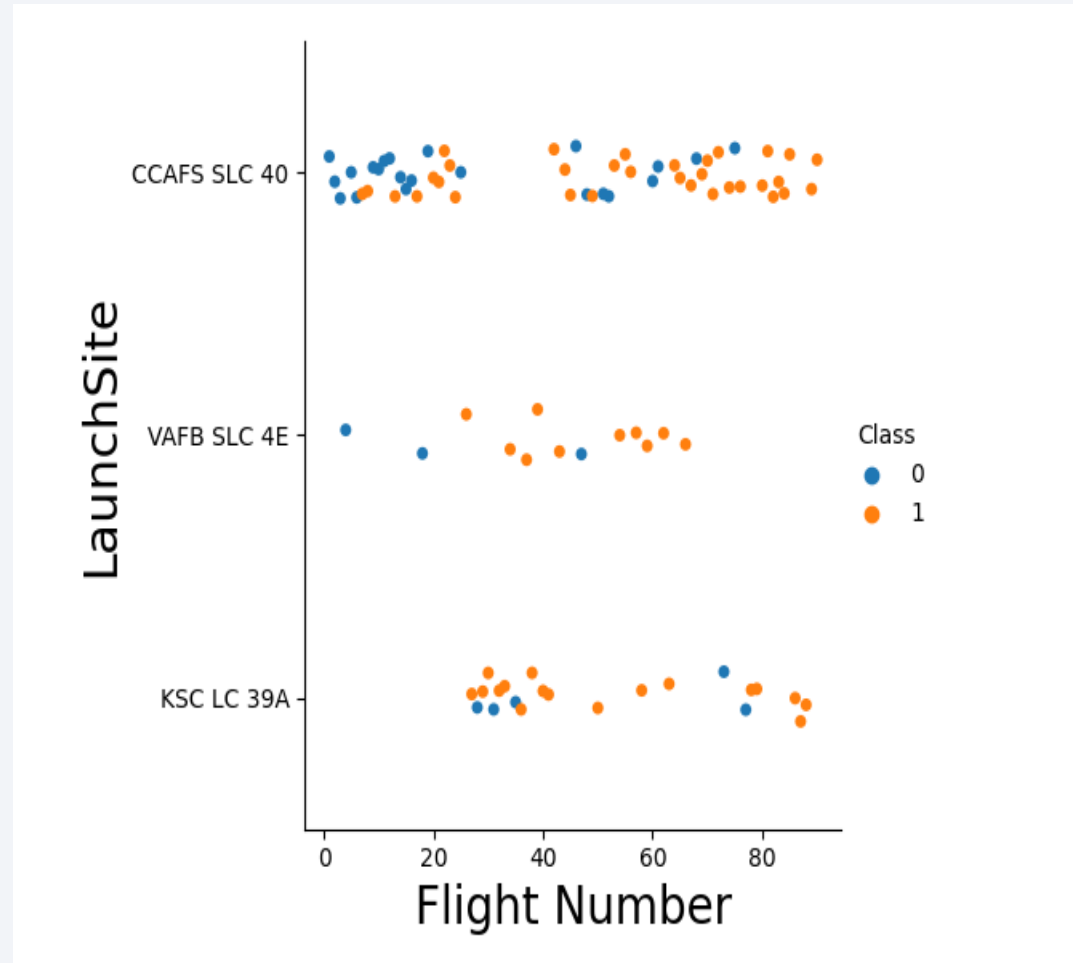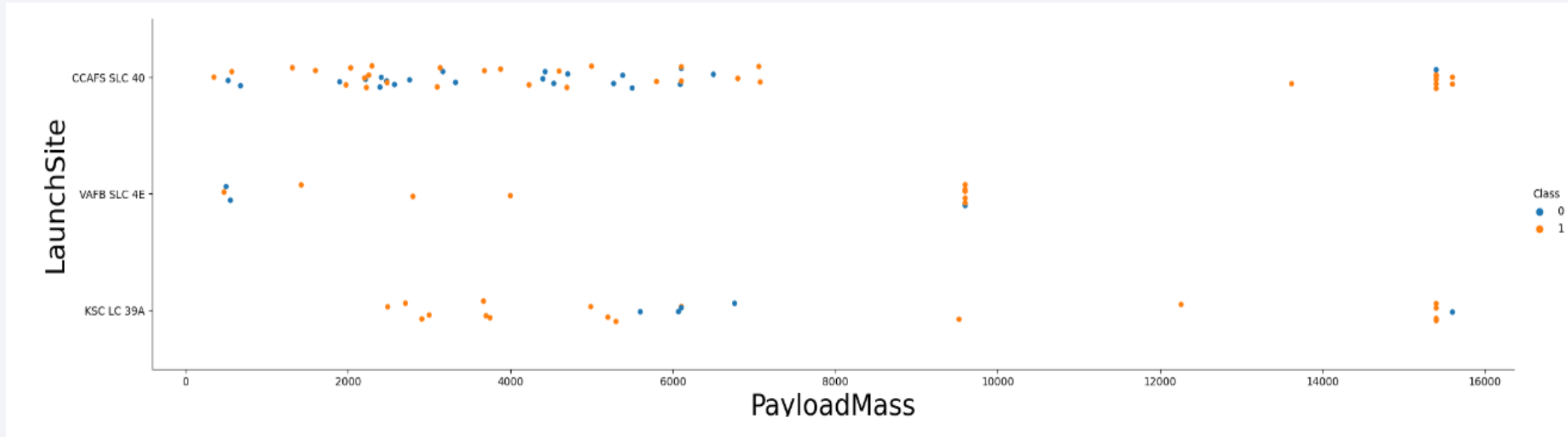
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site is shown.

- Class 0 indicates a failed launch and 1 indicates a successful launch.

- We can observe that there are much more launches from launch site CCAFS SLC 40 followed by launch site KSC LC 39A. However, the launch site VAFB SLC 4E has highest successful launches when compared to other launch sites.

Payload vs. Launch Site (Scatter Plot)

- Class 0 indicates a failed launch and 1 indicates a successful launch.

- We observed that for VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000 kg).

Success Rate vs. Orbit Type (Bar Chart)

- Bar chart for the success rate of each orbit type is shown.

- Class on Y axis here indicates mean value of class for each orbit type respectively.

- We observed that launch sites – ES-L1 , GEO, HEO, SSO have highest success rate while GTO has lowest.
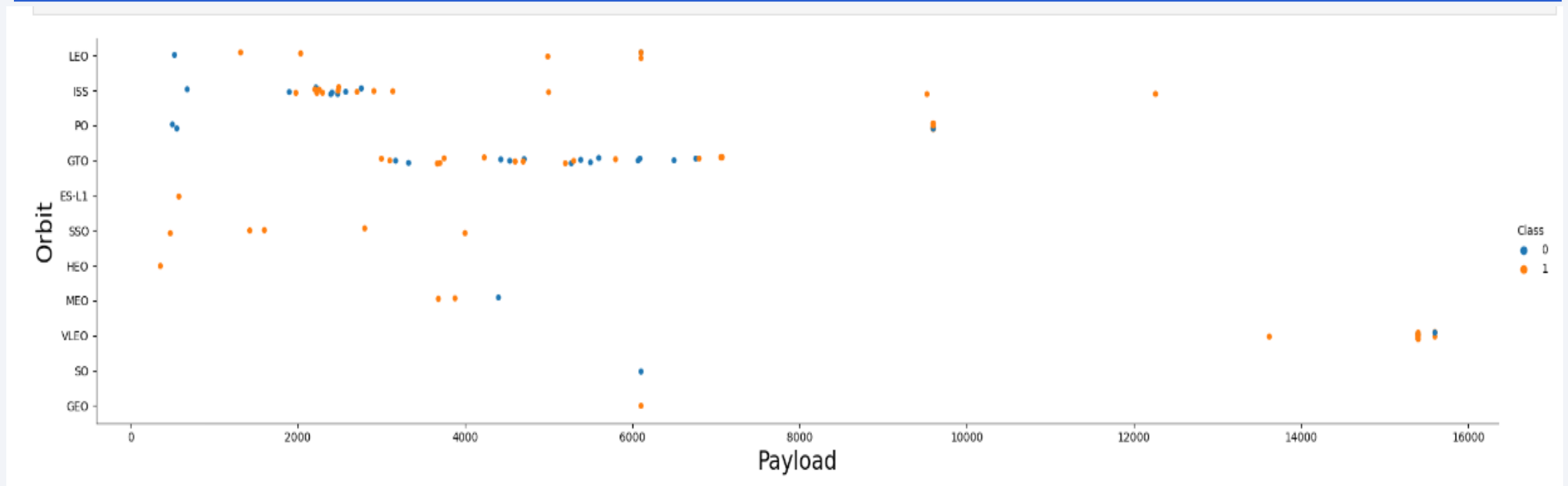
- Orbit SO has no successful launch.
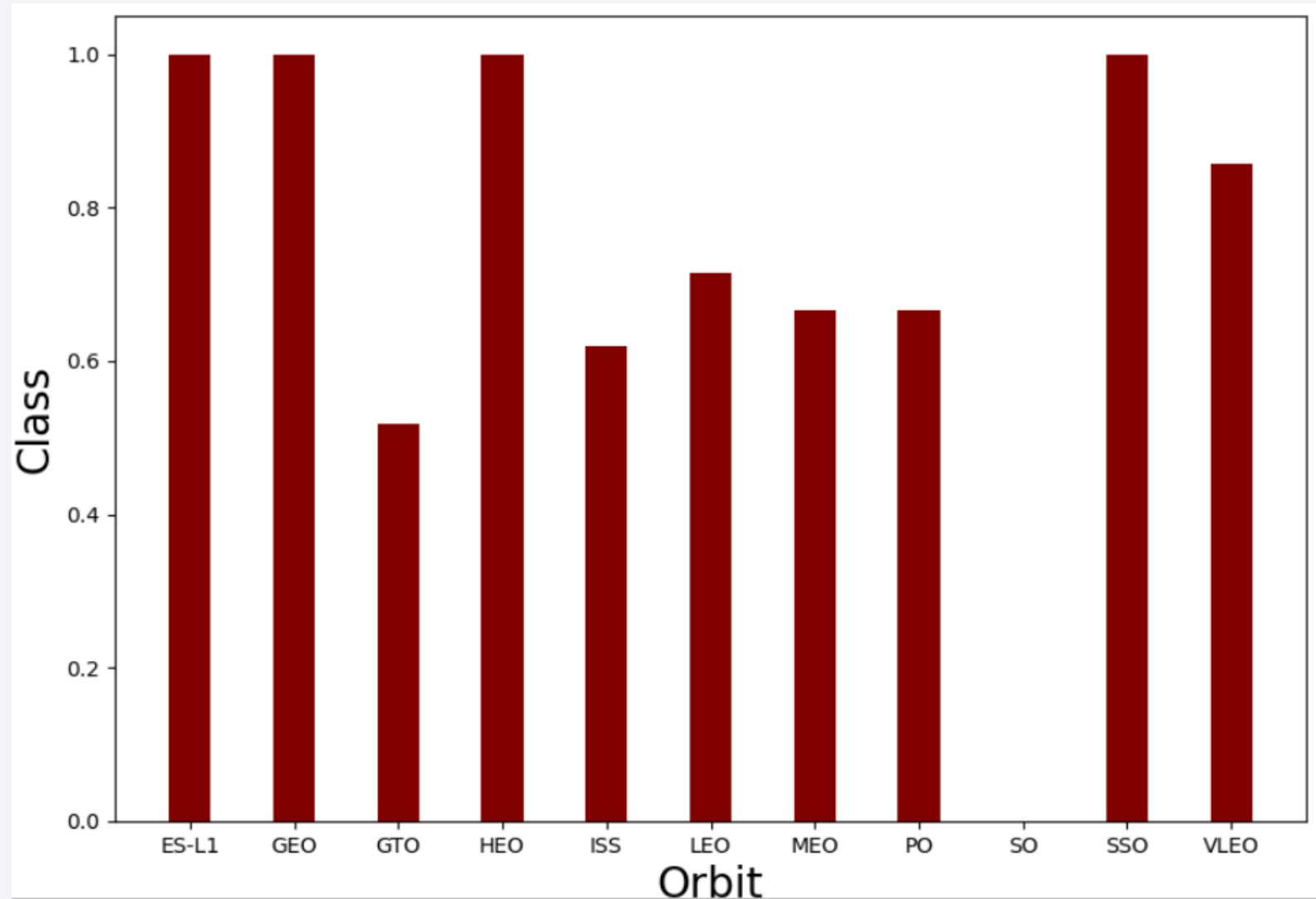
Flight Number vs. Orbit Type
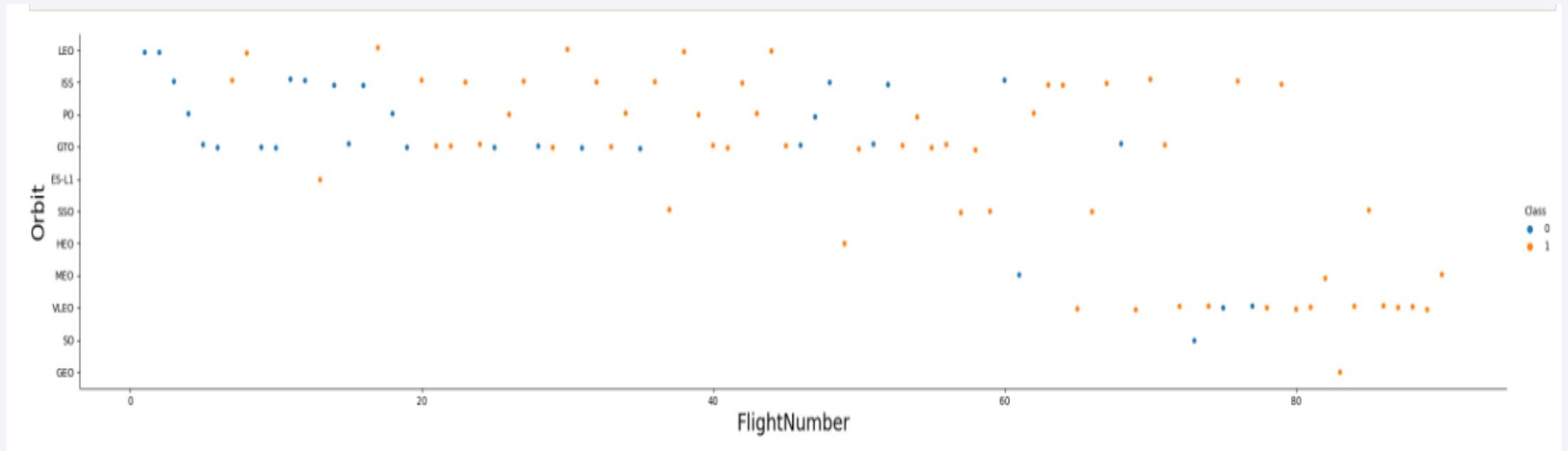
- Scatter plot of Flight number vs. Orbit type is shown.

- We observed 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.
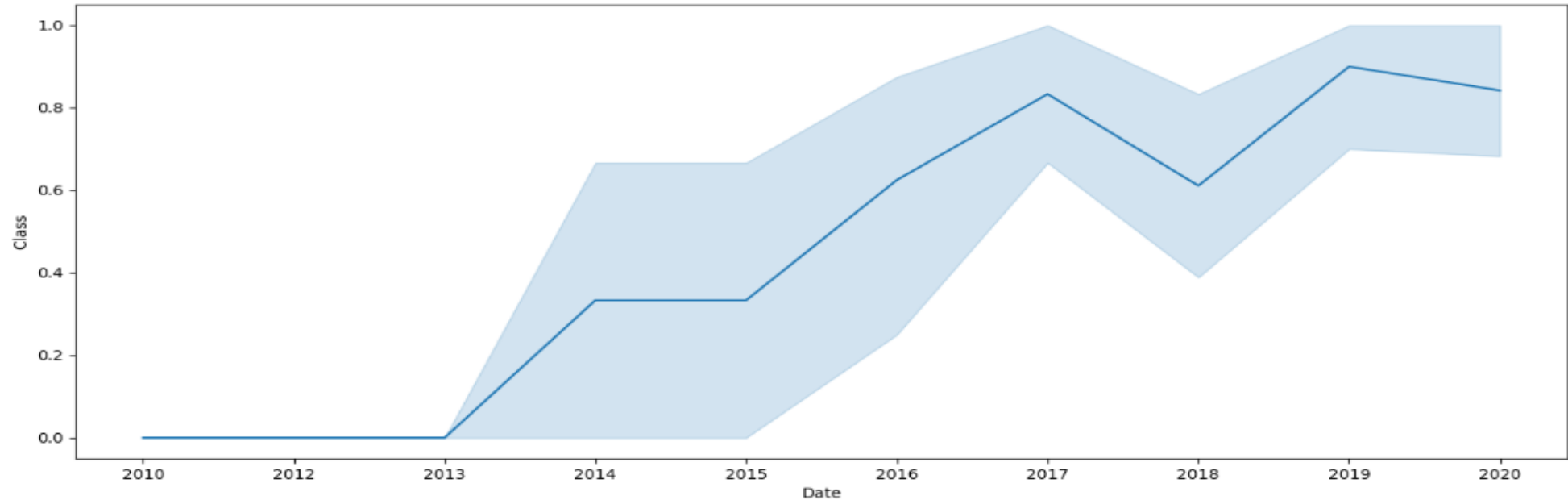
Payload vs. Orbit Type

- Scatter plot of Flight number vs. Orbit type is shown.

- We observed that 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

- A line chart of yearly average success rate is shown.

- We can observe that the success rate since 2013 kept increasing till 2020.

All Launch Site Names

- Below is the SQL query and its result for listing out names of all unique launch sites –

```
%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

- We observed that there are 4 unique launch sites named as CCAFS LC-40, VAFB SLC-4E, KSC LC-39A and CCAFS SLC-40 respectively.

Launch Site Names Begin with 'CCA'

- Below is the SQL query to display 5 records where launch sites name begin with 'CCA'

Display 5 records where launch sites begin with the string '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
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
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 (parachute)
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 attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Below is the SQL query and its result for finding out the total payload mass carried by boosters launched by NASA. We observed that the total payload mass values is 45596.0 kg.

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

```
%sql select sum(PAYLOAD_MASS__KG_) as Total_PAYLOAD_MASS from SPACEXTBL where Customer = 'NASA (CRS)';
```

 * sqlite:///my_data1.db
Done.

Total_PAYLOAD_MASS
45596.0

Average Payload Mass by F9 v1.1

- Below is the SQL query and its result for finding out the average payload mass carried by booster version F9 v1.1. We observed that the average payload mass values is 2534.67 kg.

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

```
]: %sql select round(AVG(PAYLOAD_MASS__KG_),2) as Avg_PAYLOAD_MASS from SPACEXTBL where Booster_Version like '%F9 v1.1%';

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

```
]:   Avg_PAYLOAD_MASS

              2534.67
```

First Successful Ground Landing Date

- Below is the SQL query and its result for finding out dates of the first successful landing outcome on ground pad. We observed that the date is - 01/07/2020

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

Hint:Use min function

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

* sqlite:///my_data1.db
Done.

min(Date)
01/07/2020

Successful Drone Ship Landing with Payload between 4000 and 6000

- Below is SQL query and its result for displaying the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 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

```
]: %sql select Booster_Version,SPACEXTBL.'Landing_Outcome',PAYLOAD_MASS__KG_ from SPACEXTBL where SPACEXTBL.'Landing_Outcome' like
   and PAYLOAD_MASS__KG_ between 4000 and 6000
```

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

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

- From the SQL query and its results, we can clearly observe that there are 100 successful mission outcomes and 1 Failed mission outcome.

List the total number of successful and failure mission outcomes

```
[27]: %sql select count(Mission_Outcome) as Success from SPACEXTBL where Mission_Outcome like '%Success%'
```

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

[27]:

Success
100

```
[22]: %sql select count(Mission_Outcome) as Fail from SPACEXTBL where Mission_Outcome like '%Fail%'
```

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

[22]:

Fail
1

Boosters Carried Maximum Payload

- Below is the SQL query and its result for the names of the booster which have carried the maximum payload mass

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

```
%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

- Below is the SQL query and its result to find List of failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015. We observed that there are two failure in month of April (04) and October (10) respectively.

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.

```
[30]: %sql select substr(Date, 4,2) as Month_No, SPACEXTBL.'Landing_Outcome', SPACEXTBL.'Booster_Version', SPACEXTBL.'Launch_Site'
      from SPACEXTBL where substr(Date,7,4)='2015' and SPACEXTBL.'Landing_Outcome' like '%fail%drone%'

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

[30]:

Month_No	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

- Below is the SQL query and its result to for Ranking the count of landing outcomes Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

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

```
%sql select count(SPACEXTBL.'Landing_Outcome') from SPACEXTBL where SPACEXTBL.'Landing_Outcome' like '%Success%'
and Date between '04-06-2010' and '20-03-2017'
```

 * sqlite:///my_data1.db
Done.

count(SPACEXTBL.'Landing_Outcome')
35

Section 3

Launch Sites Proximities Analysis

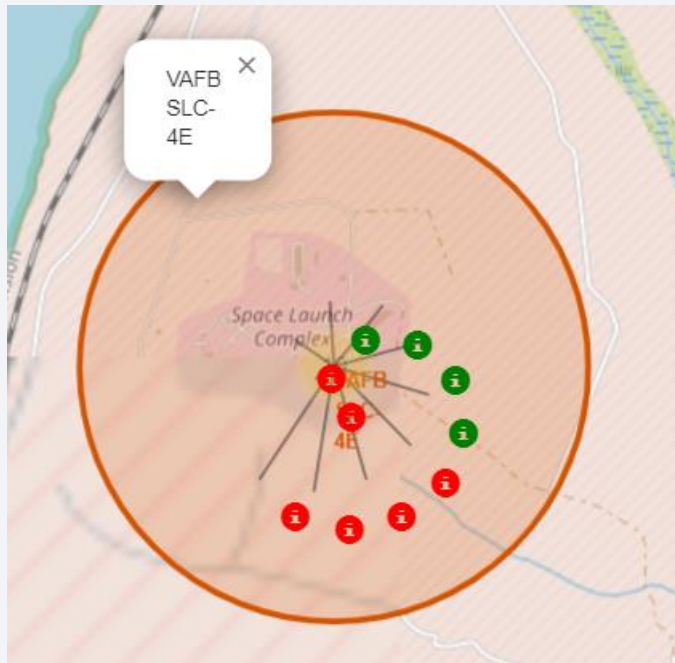# Folium Map Launch Site Location

- Below is the screenshot of USA map displaying launch site locations in red marker locations. We can observe that these locations are near coastline one at east and other one at west border respectively.
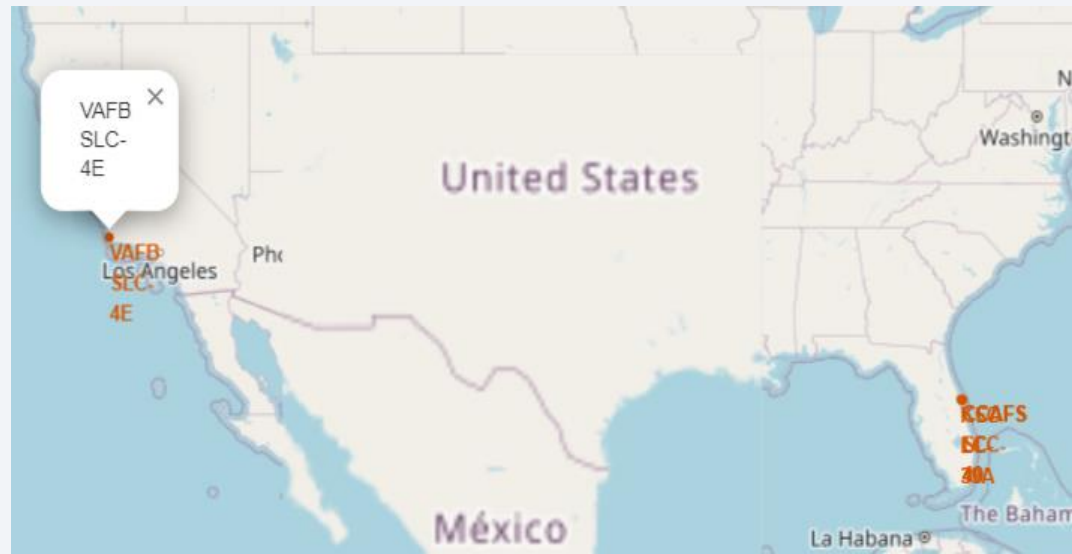
Folium Map Launch Outcomes

- Below is the screenshot of Folium Map Launch Outcomes in red and green markers as displayed for different launch sites. Please note that Successful launches are marked with green color while red color markers are for failed launch outcomes.
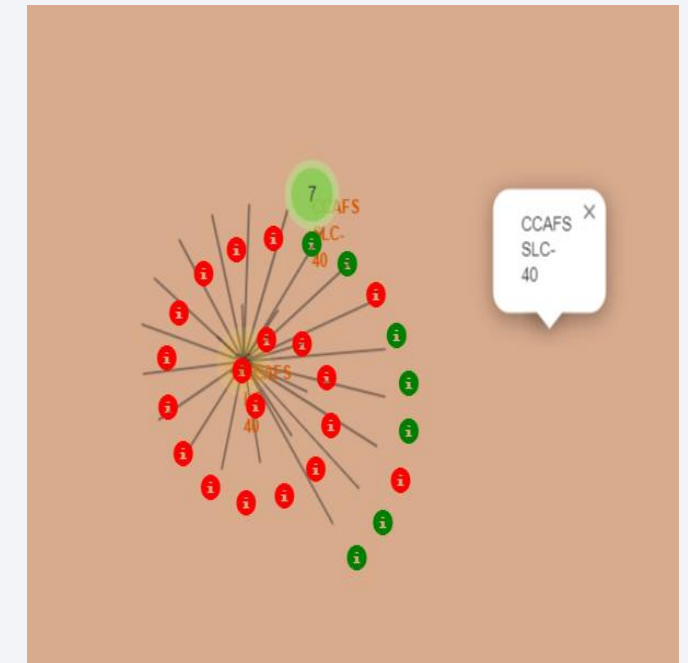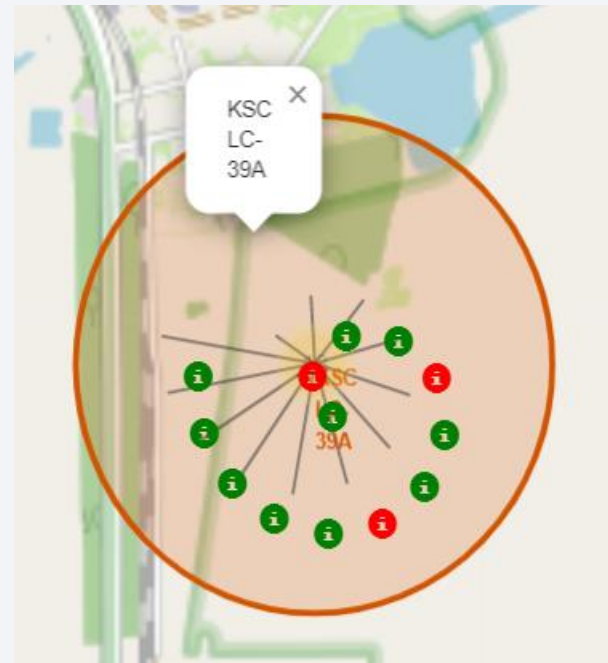
Folium Map Launch Site Proximities

1. Displayed screenshot of Folium Map Launch Site VAFB SLC - 4E close distance to railway line. We can observe that the distance is around 1.30 KM.

2. Displayed screenshot of Folium Map Launch Site CCAFS SLC - 40 close distance to coastline. We can observe that the distance is around 0.90 KM.
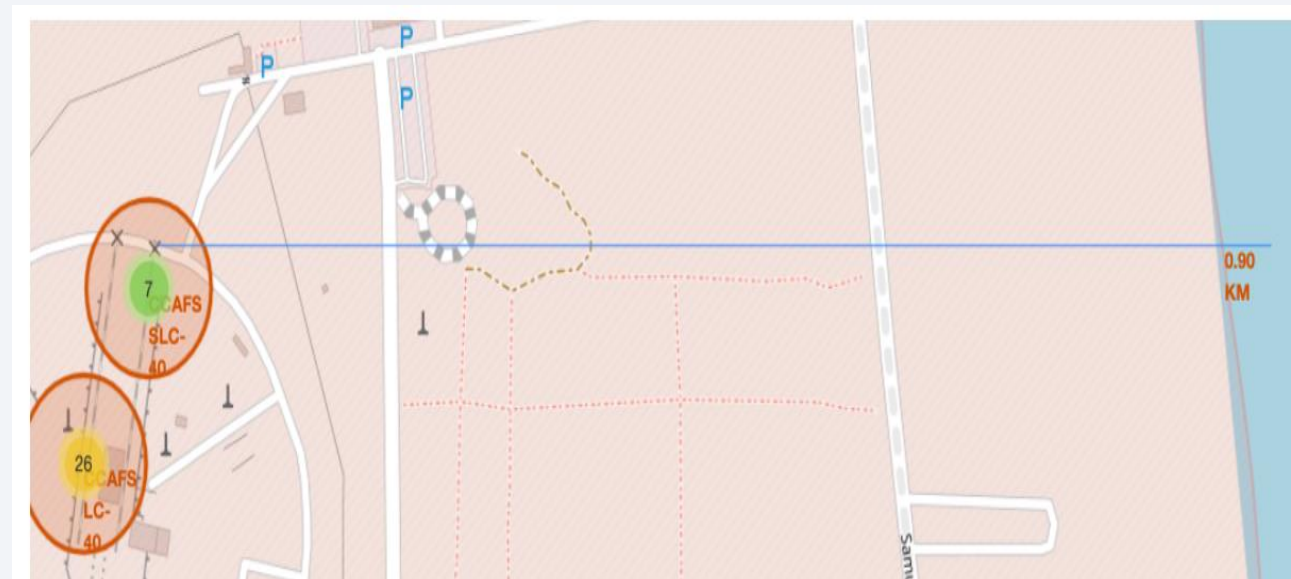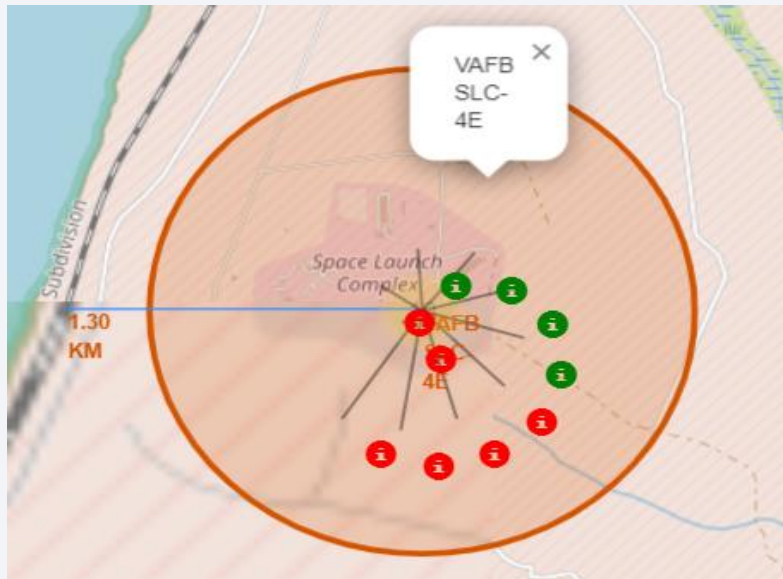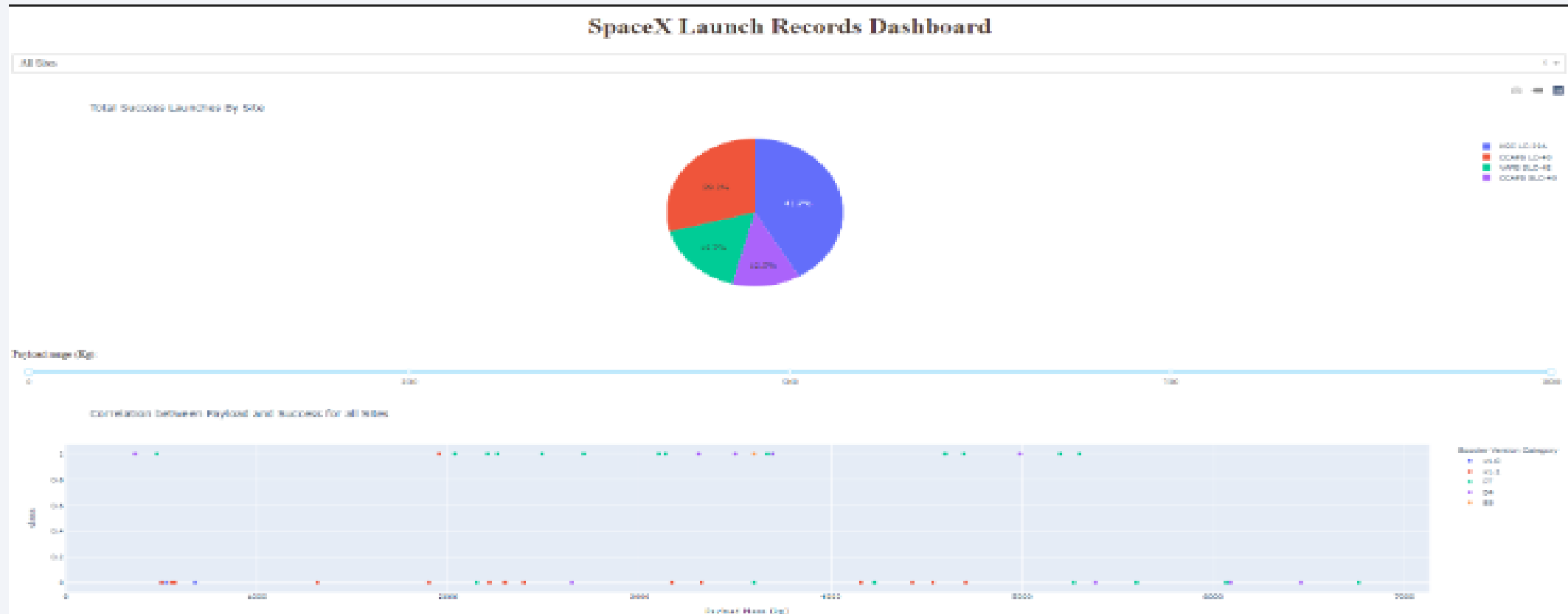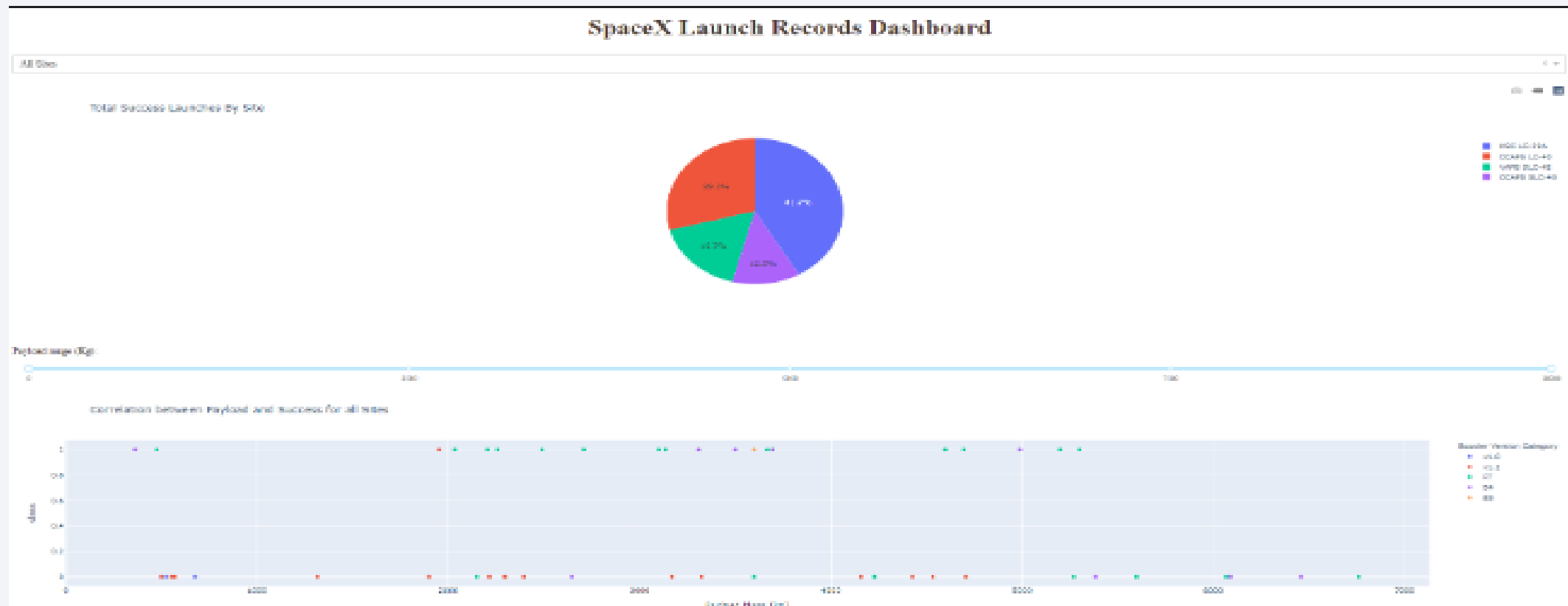
Build a Dashboard with Plotly Dash

Dashboard Pie Chart of success launch count

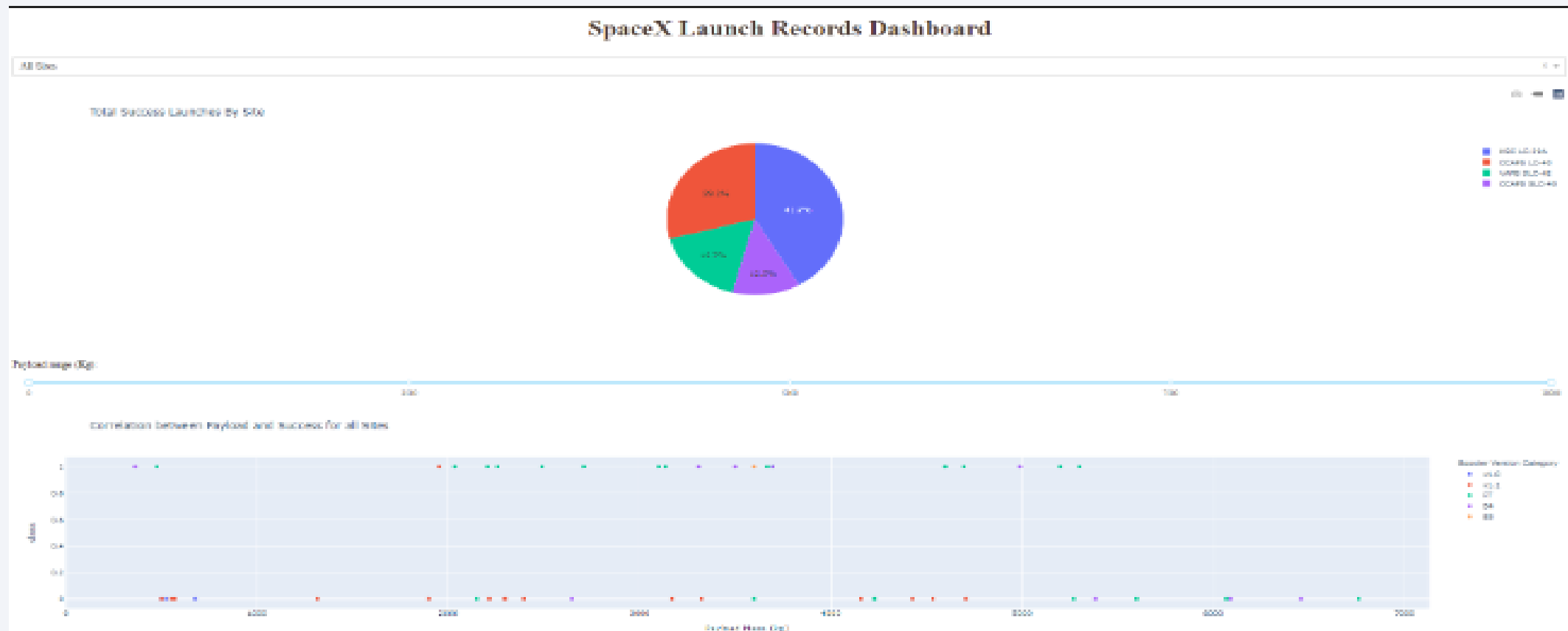- Below is the screenshot of Dashboard Pie Chart displaying successful launch counts.

Dashboard Screenshot of Pie chart

- Below is the screenshot of Dashboard Pie Chart displaying successful launch counts.

Dashboard Screenshot of Pie chart

- Below is the screenshot of Dashboard Pie Chart displaying successful launch counts.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- In this Capstone project we have implemented 4 models for classification tasks – LR (Logistic Regression), SVM (Support Vector Machine), DTC (Decision Tree Classifier) and KNN (K-Nearest Neighbor).

- Bar chart that represents prediction accuracy of all the 4 models is shown.

- Kindly Note Standard Scaling method has been used for preprocessing the data.
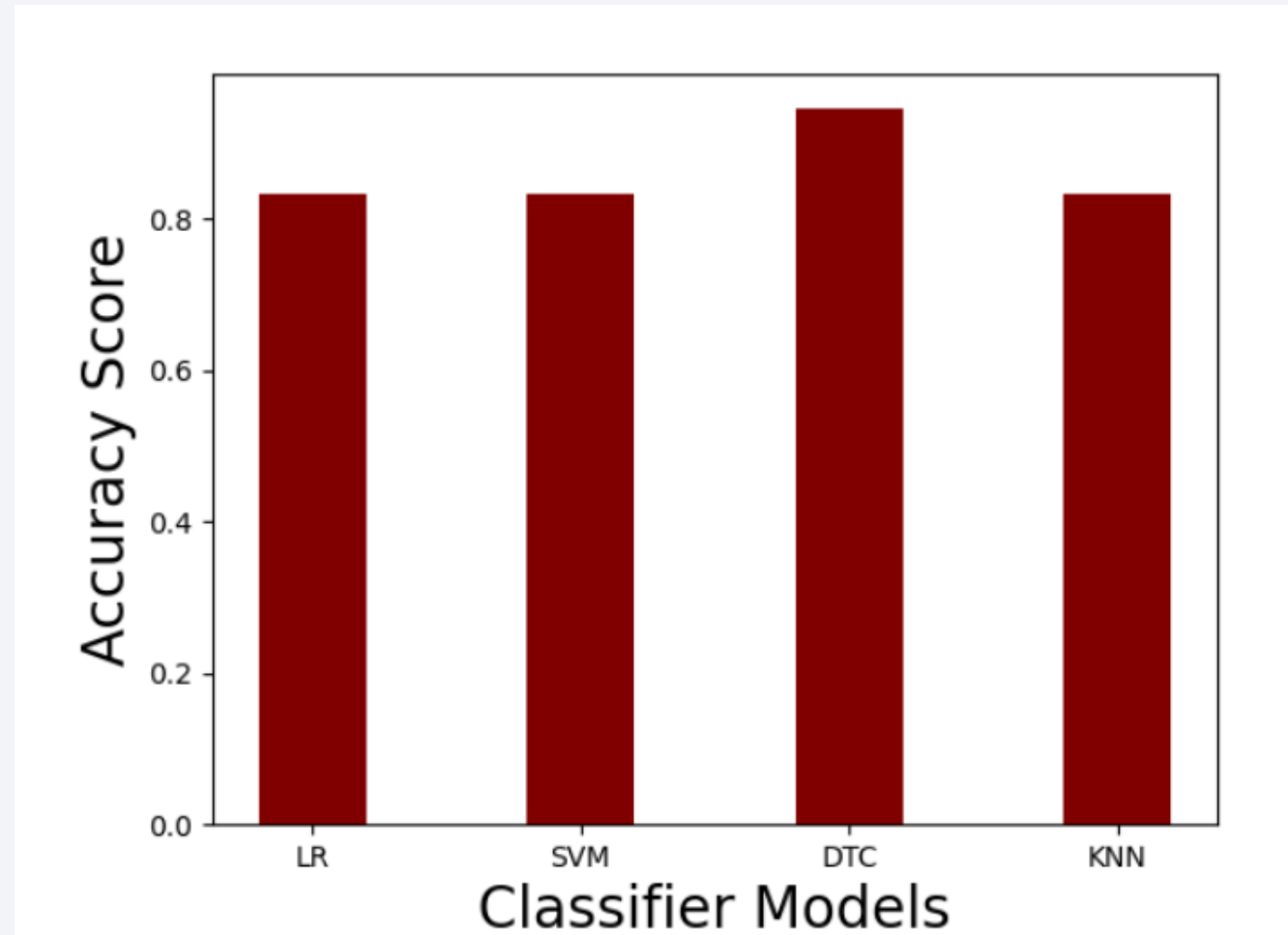
Confusion Matrix

- From the Bar chart in previous slide we can observe that DTC (Decision Tree Classifier) Model has highest accuracy score of 0.94 among other classification models.

- Confusion Matrix for DTC model is shown.

- True Positive – 12 , False Positive – 1, True Negative – 5 , False Negative – 0

- Recall Score – 1

- Precision Score – 0.92

- F1 Score – 0.9583
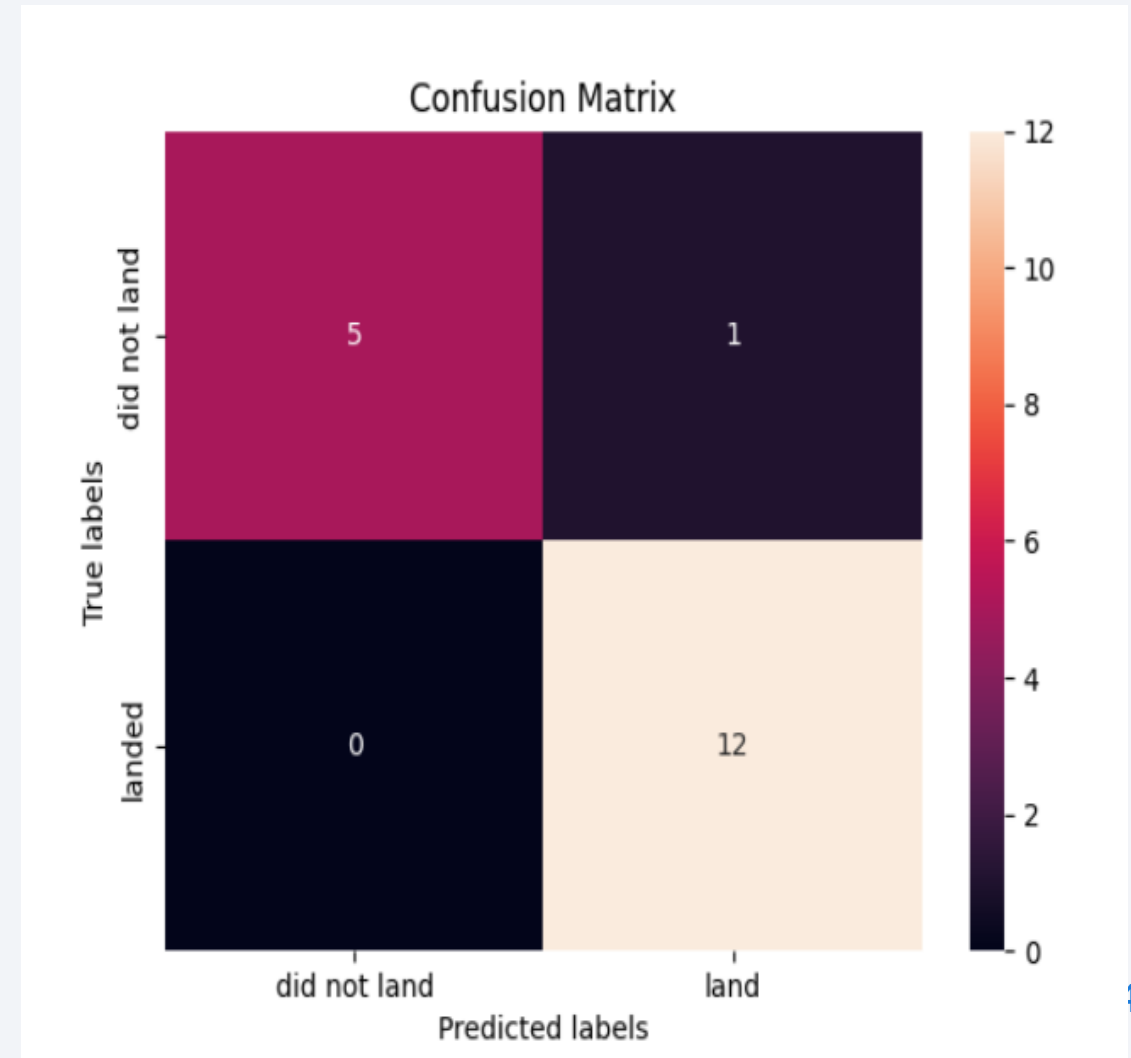
Conclusions

- In this capstone project of SpaceX, we predicted if the Falcon 9 first stage will land successfully.

- This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- From our data analysis and application of machine learning models we observed that Classifier models such as Logistic regression, SVM and KNN gave score of 83% accuracy which is a good score to build up the model further.

- However, 83% accuracy is still not good enough in rocket launch business where heavy amount of money is involved. Hence, We tried another model and observed that DTC (Decision Tree Classifier) model is most effective in terms of predicting whether a landing will successful or not. The model has accuracy of 94 % on test data which is most promising and helpful in making important decisions in the business.

Appendix

- GitHub Repository - https://github.com/santo-mantras/IBM_CapstoneProject.git

  Other Resources -

- TowardsDataScience - https://towardsdatascience.com

- Analytics Vidhya - https://www.analyticsvidhya.com/blog/2021/05/in-depth-understanding-of-confusion-matrix/

Thank you!