IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sandeep Shashi Kumar
22 December 2022

Outline

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

Executive Summary

- Summary of methodologies
  - ❖ Data Collection through API
  - ❖ Data Collection with Web Scraping
  - ❖ Data Wrangling
  - ❖ Exploratory Data Analysis with SQL
  - ❖ Exploratory Data Analysis with Data Visualization
  - ❖ Interactive Visual Analytics with Folium
  - ❖ Machine Learning Prediction
- Summary of all results
  - ❖ Exploratory Data Analysis result
  - ❖ Interactive analytics in screenshots
  - ❖ Predictive Analytics result from Machine Learning Lab

Introduction

- Project background and context

  SpaceX is a company who has entered the space industry by providing rocket launches with Falcon 9 rocket for as low as 62 million dollars, while the cost of other providers are upward of 165 million dollar for each launch. The saving in the rocket launch is by reusing the first stage by re-landing the rocket which can be used for the next mission. By repeating this process there will be further decrease in the prices of the launch. As a data scientist, the goal of this project is to create the machine learning project to predict the landing outcome of the first stage in the future. This project will be crucial in identifying the right price to bid against SpaceX for a rocket launch.

- Problems you want to find answers

  - ❖ Identifying all factors that influence the landing outcome.
  - ❖ The relationship between each of the variables and effect of the outcome.
  - ❖ The best condition which is required to increase the probability of successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX REST API and web scrapping from Wikipedia.
- Perform data wrangling
  - Data was processed using one-hot encoding for categorical features.
- 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

Data collection is the process of gathering and analyzing accurate data from various sources to find answers to research problems, trends and probabilities to evaluate possible outcomes. The data set was collected by REST API and web scrapping from Wikipedia.

REST API was started by using the get request. The response content was decoded as json and then turned into a pandas data frame using json_normalize(). The data was then cleaned, checked for missing values and values were filled wherever required in the data.

BeautifulSoup was used for web scrapping to extract and launch the records as HTML table, parse the table and convert it to a pandas data frame for further analysis.

Data Collection – SpaceX API

Use get request for rocket launching data using API

Using json_normalize method to convert json result to dataframe

Perform data cleaning and input the missing values

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

```
response = requests.get(spacex_url)
```

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

```
# 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 roc
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)]
```

Link: https://github.com/Sandeep221085/Data-Science-Project/blob/main/Data%20Collection%20API.ipynb

Data Collection - Scraping

Request the Falcon9 launch Wikipedia page from URL

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
```

Create a BeautifulSoup from the HTML respsone

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, 'html.parser')
```

Extract all column/variable names from the HTML header

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            #print(flight_number)
            datatimelist=date_time(row[0])
```

Link: https://github.com/Sandeep221085/Data-Science-Project/blob/main/Data%20Collection%20with%20Web%20Scraping.ipynb

Data Wrangling

Data Wrangling is the process of removing errors and combining complex data sets to make them more accessible and easy to analyze.

First the number of launches on each site were calculated and then the number and occurrence of mission outcome per orbit was calculated.

Then the landing outcome label from the outcome column was created. The above will help in further analysis and visualization. Finally the result was exported as a CSV file.

Link: https://github.com/Sandeep221085/Data-Science-Project/blob/main/EDA.ipynb

EDA with Data Visualization

The following scatter plots were prepared between the attributes of the rocket launch

- Pay load mass (kg) and Flight Number
- Launch site and Flight Number
- Launch site and Payload Mass Kg
- Orbit and Flight Number
- Orbit and Payload Mass

Link: https://github.com/Sandeep221085/Data-Science-Project/blob/main/EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

The following SQL queries were performed on the data set

- 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 acheived.
- 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. Use a subquery
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for 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

Link: https://github.com/Sandeep221085/Data-Science-Project/blob/main/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- To visualize the launch data into an interactive map, the latitude and longitude coordinates at each launch site and a circle was added around each launch site with a label including the name of the launch site.
- The data frame launch outcomes (success or failure) was assigned with to classes 0 and 1 with Red and Green markers on the map in Marker Cluster().
- Folium circle was created and added to show the exact location of each of the launch site with name and Folium marker was created and added to show the result of the launch outcomes with green for success and red for failure.

Link: https://github.com/Sandeep221085/Data-Science-Project/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- An interactive dashboard with Plotly dash was built which allows the user to explore different pie charts and scatter plot showing the mission outcomes of different launch sites.
- Pie charts showing the total launches by each of the launch sites was plotted.
- Scatter plot showing the relationship between Payload Mass and Launch Outcomes was plotted for the different booster version.

Link: https://github.com/Sandeep221085/Data-Science-Project/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

Model Building

- The data set was loaded and a numpy array was created
- The data set was split into training and test data sets using train_test_split
- The models are trained and hyperparameters are selected using the function GridSearchCV

Model Evaluation

- The accuracy for each model was checked
- Tune hyperparameters for each type of algorithms
- Plot the confusion matrix

Model Improvement

- The model was improved using feature engineering and algorithm tuning

Best Model

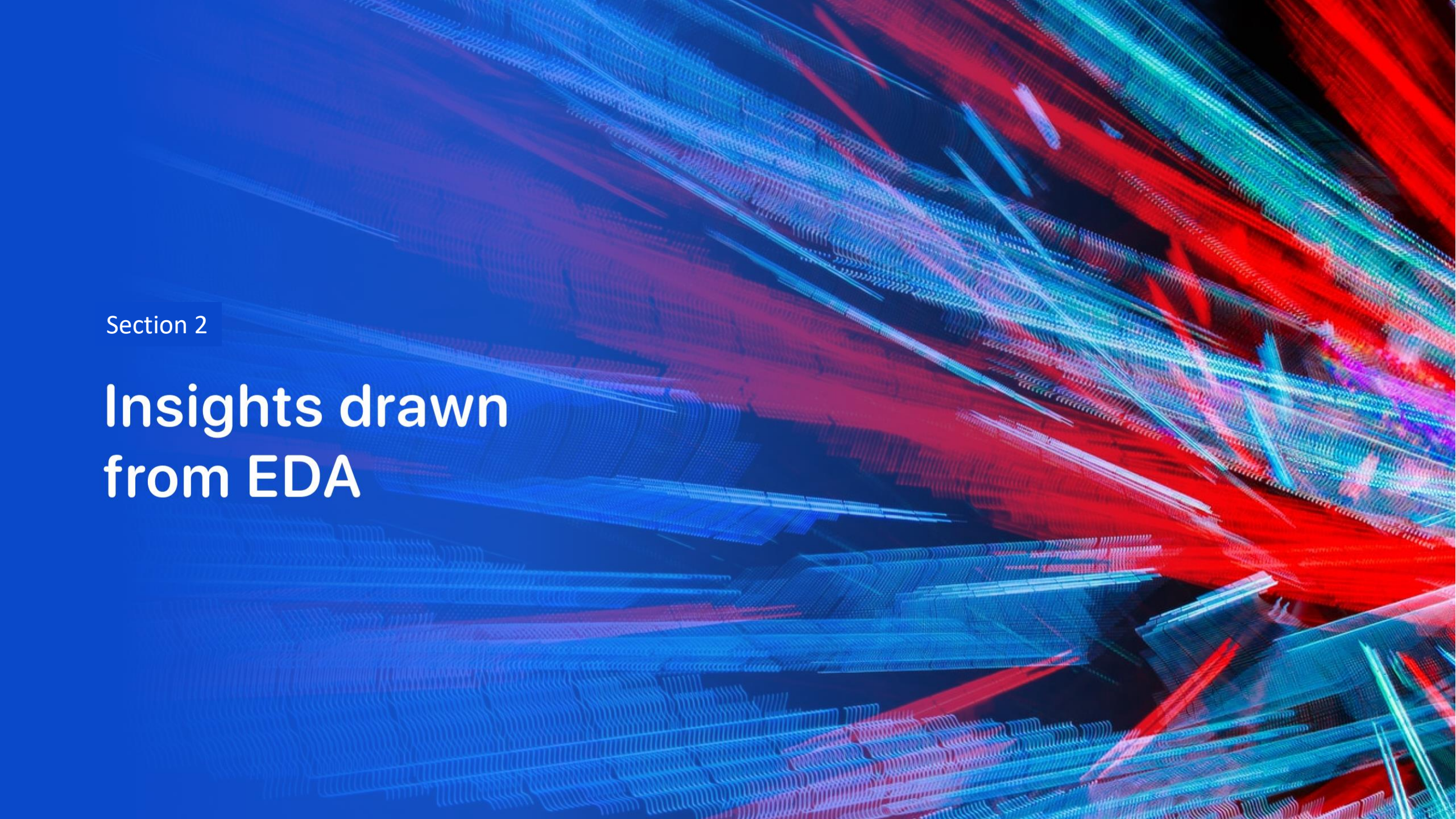
- The model with best accuracy is the best performing model

Link: https://github.com/Sandeep221085/Data-Science-Project/blob/main/Machine%20Learning%20Prediction.ipynb

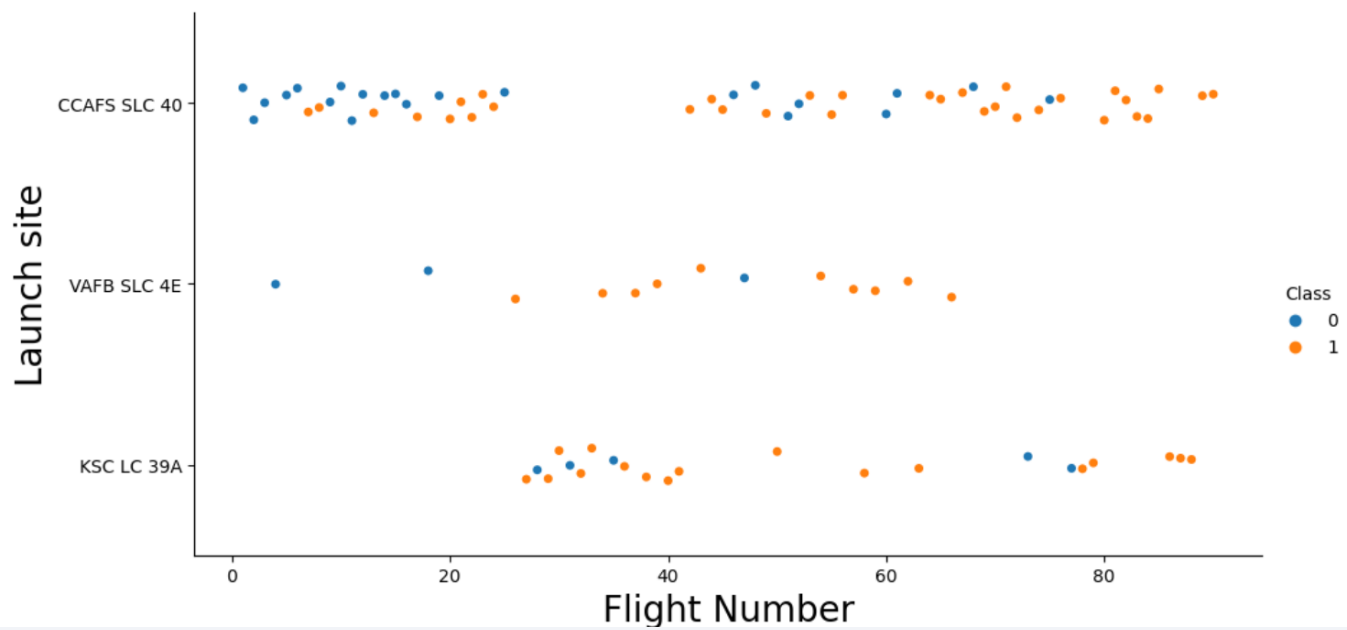
Results

The results obtained after analyzing the data set can be categorized into

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

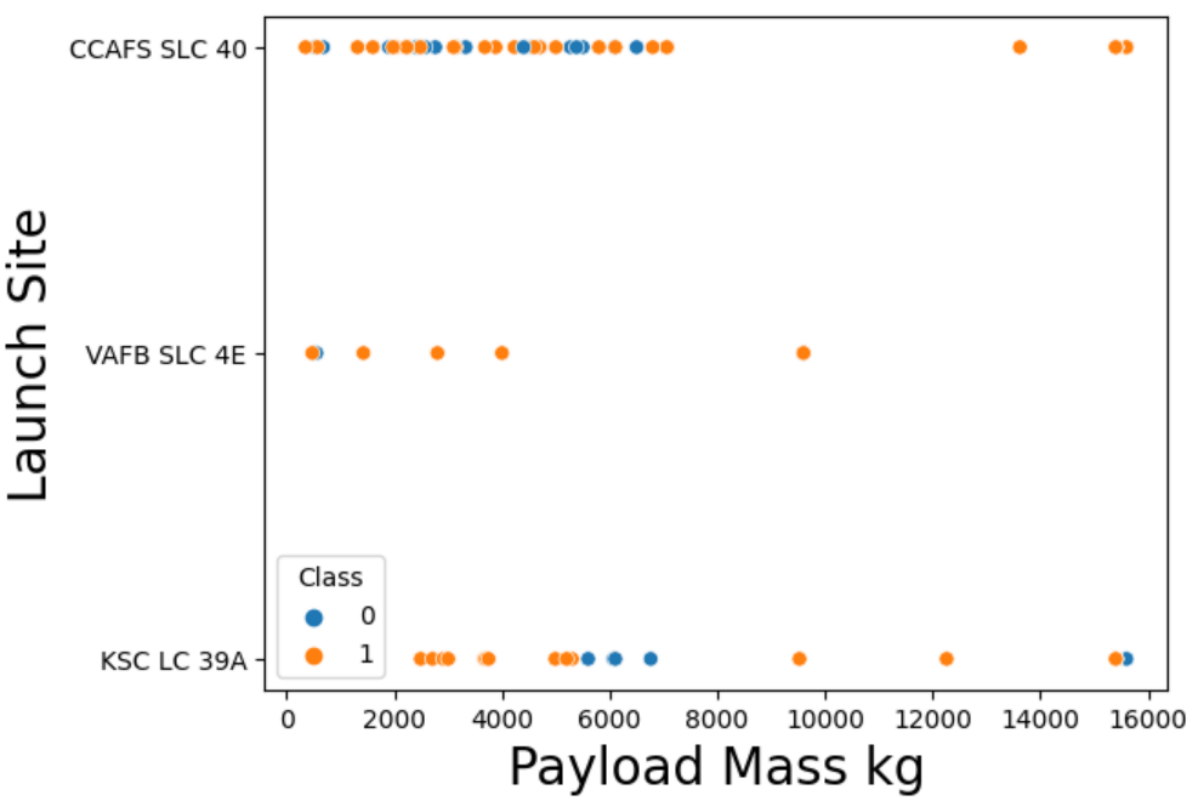

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



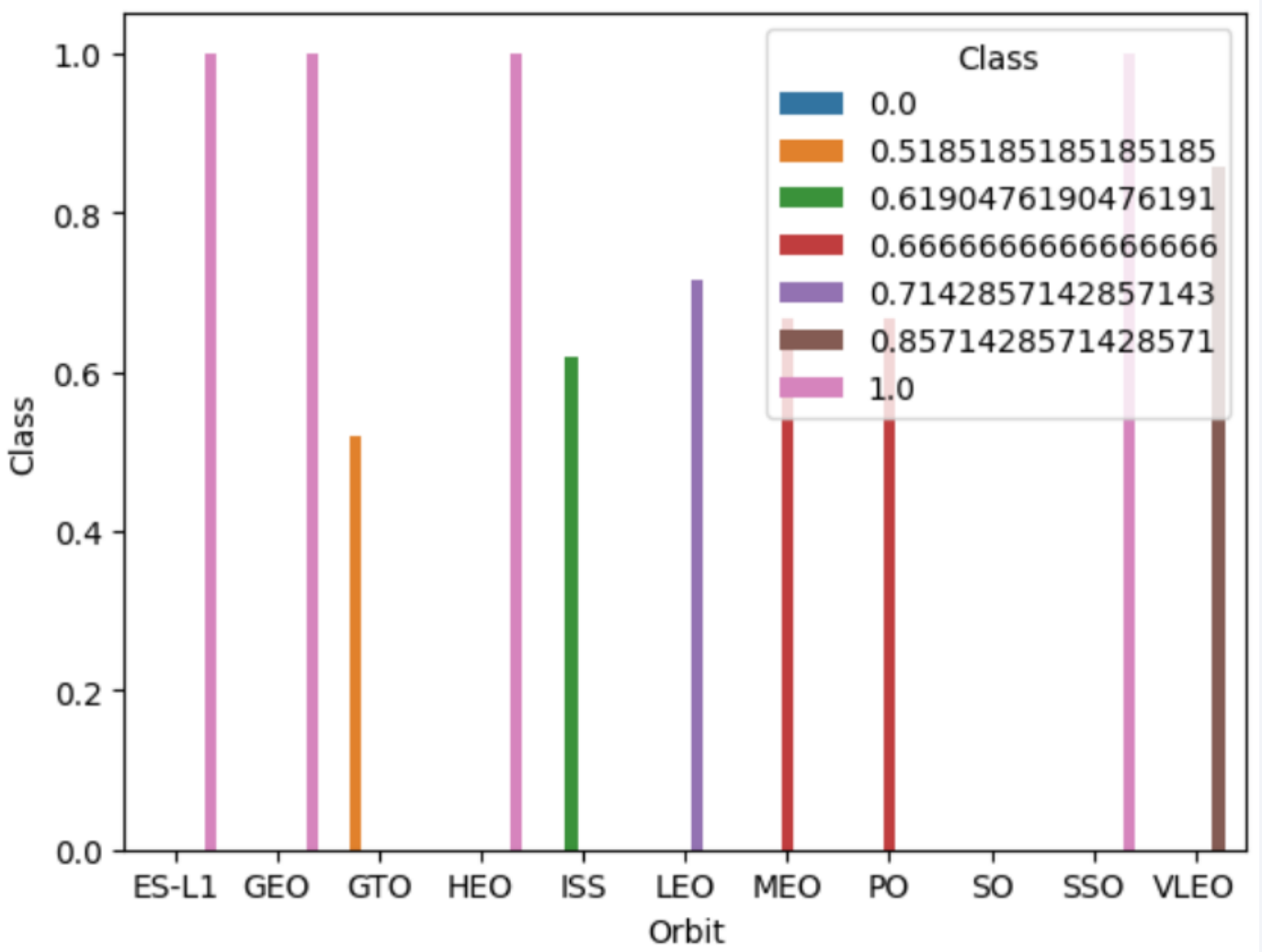
- The scatter plot shows when the number of flight increases in the launch station greater is the success rate.
- Launch site CCAFS SLC 40 has higher success rate when compared to the other launch sites.

Payload vs. Launch Site

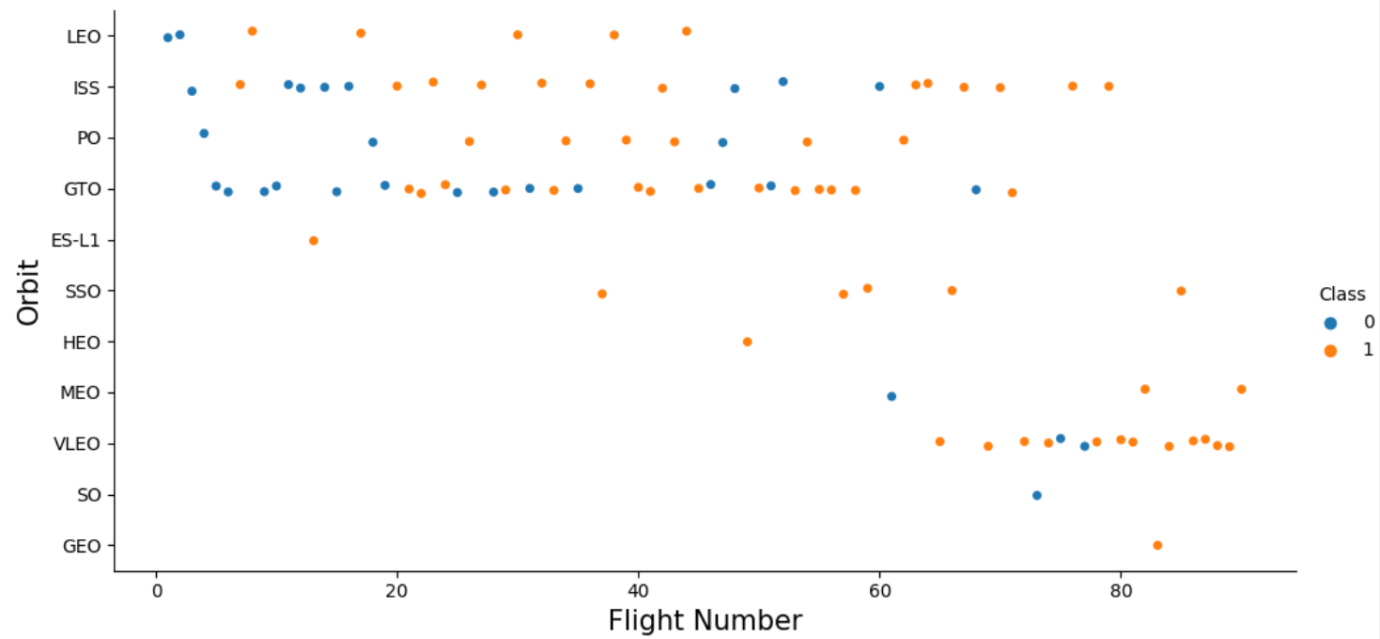


- The scatter plot shows that Launch Site VAFB SLC 4E has no rocket launches with payload mass of more than 10000kg.
- The plot also shows that the success rate of launch with payload mass more than 7000kg is high, however there is no definite pattern to show the dependency of payload mass on the launch outcomes.

Success Rate vs. Orbit Type

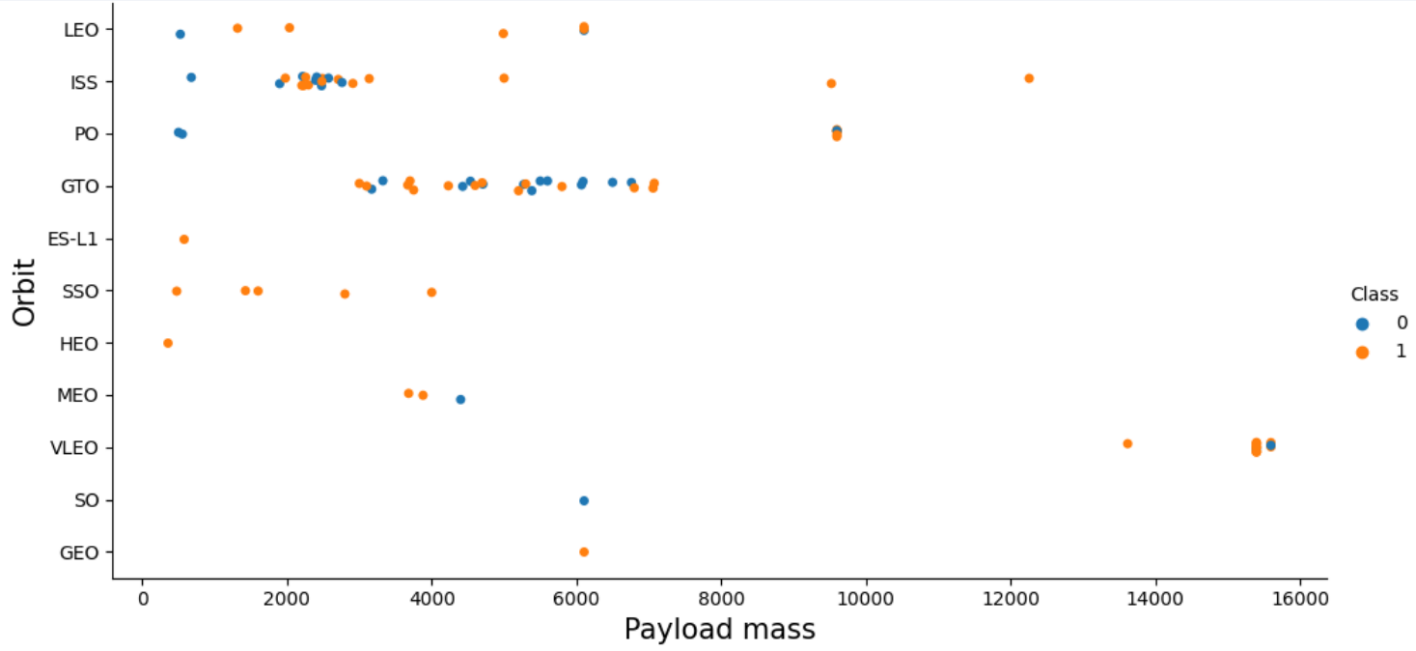


Flight Number vs. Orbit Type



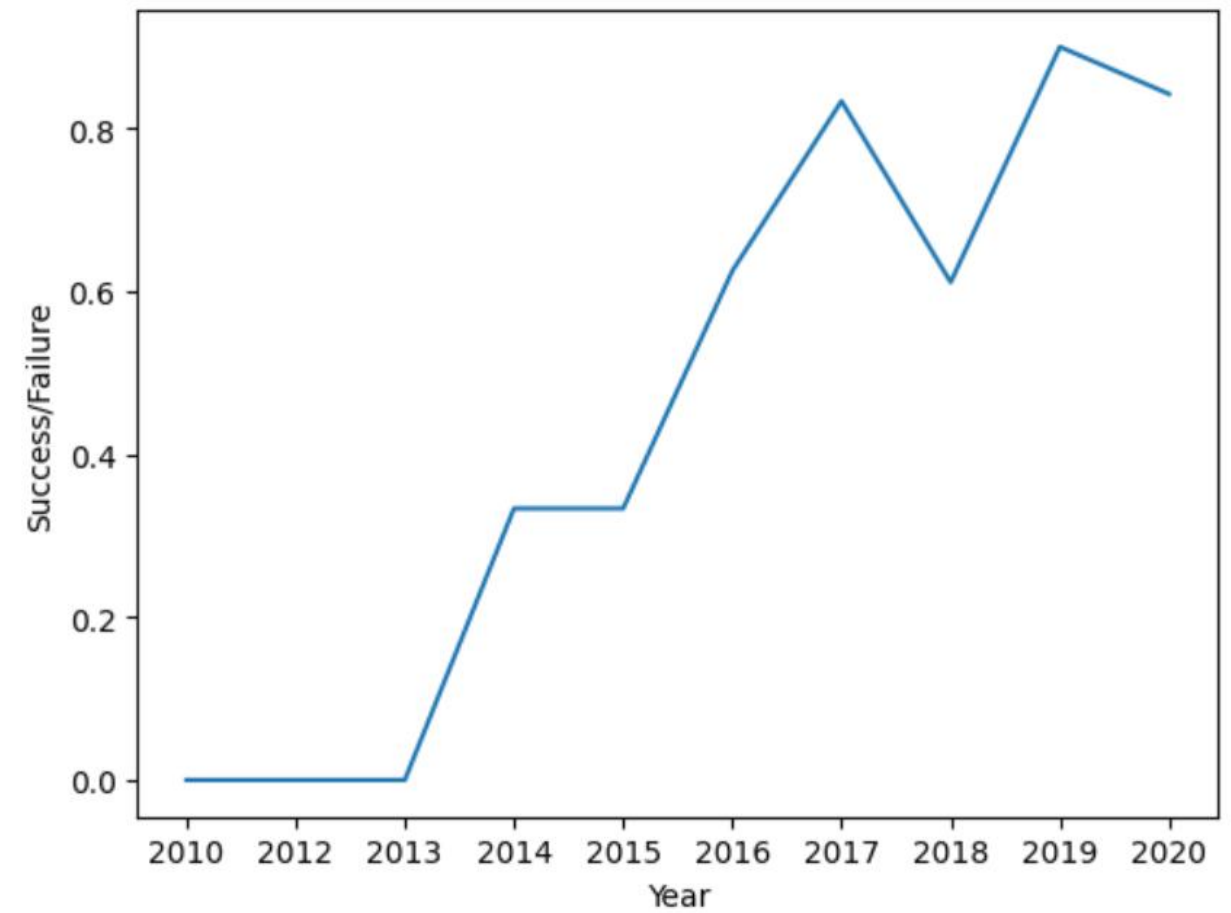
- The scatter plot shows that in the LEO orbit the Success appears related to the number of flights but on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



- The scatter plot shows that With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However for GTO we cannot distinguish between positive landing rate and negative landing (unsuccessful mission) with respect to payload mass.

Launch Success Yearly Trend



The graph shows a increasing trend in the success of launch from 2013 to 2020

All Launch Site Names

The key word “unique” was used to show only unique launch sites from the SpaceX data.

```
%sql select Unique(LAUNCH_SITE) from SPACEXTBL;
```

```
 * ibm_db_sa://qqj89844:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

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

Launch Site Names Begin with 'CCA'

The below query was used to display 5 records where launch sites begin with `CCA`

```
%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://qqj89844:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

The total payload carried by boosters from NASA was calculated as 45596 using the query below

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

```
 * ibm_db_sa://qqj89844:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

payloadmass
45596

Average Payload Mass by F9 v1.1

The average payload mass carried by booster version F9 v1.1 was calculated as 2928 using the below query.

```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL where Booster_Version = 'F9 v1.1';
```

```
 * ibm_db_sa://qqj89844:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

payloadmass
2928

First Successful Ground Landing Date

The min() function was used to find the result. We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

```
%sql select min(DATE) from SPACEXTBL where LANDING_OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://qqj89844:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

The WHERE clause to filter for boosters which have successfully landed on drone ship and then the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000 was used in the below query.

```
%sql select BOOSTER_VERSION from SPACEXTBL where LANDING_OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and
```

```
 * ibm_db_sa://qqj89844:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
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

The keyword count was used to count the no. of successful and failed Mission Outcomes.

```
%sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://qqj89844:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

missionoutcomes
1
99
1

Boosters Carried Maximum Payload

The booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function was used.

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTE
```

```
 * ibm_db_sa://qqj89844:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

boosterversion
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

The combinations of the WHERE clause was used to find out landing outcomes in drone ship, their booster versions, and launch site names for year 2015

```
%sql SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where EXTRACT(YEAR FROM DATE) = '2015';
```

```
 * ibm_db_sa://qqj89844:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

1	mission_outcome	booster_version	launch_site
1	Success	F9 v1.1 B1012	CCAFS LC-40
2	Success	F9 v1.1 B1013	CCAFS LC-40
3	Success	F9 v1.1 B1014	CCAFS LC-40
4	Success	F9 v1.1 B1015	CCAFS LC-40
4	Success	F9 v1.1 B1016	CCAFS LC-40
6	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	CCAFS LC-40

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

The Landing outcomes and the count of landing outcomes from the data was extracted and used where clause to filter for landing outcomes between 2010-06-04 to 2010-03-20. Group By clause was used to group and Order By clause was used to produce the data in descending order.

```
%sql SELECT LANDING_OUTCOME, COUNT(*) AS COUNT_LAUNCHES FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY
```

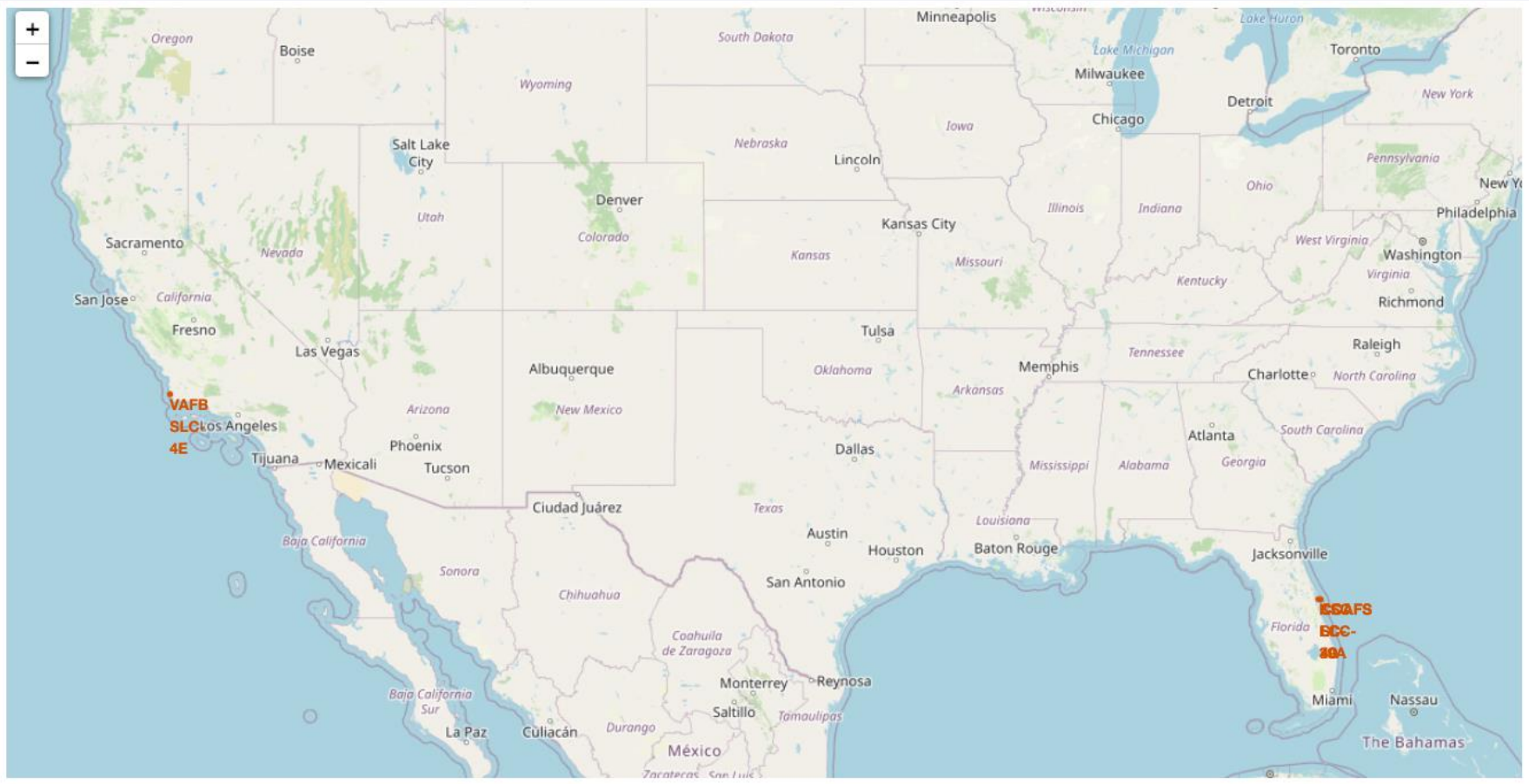
```
 * ibm_db_sa://qqj89844:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

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

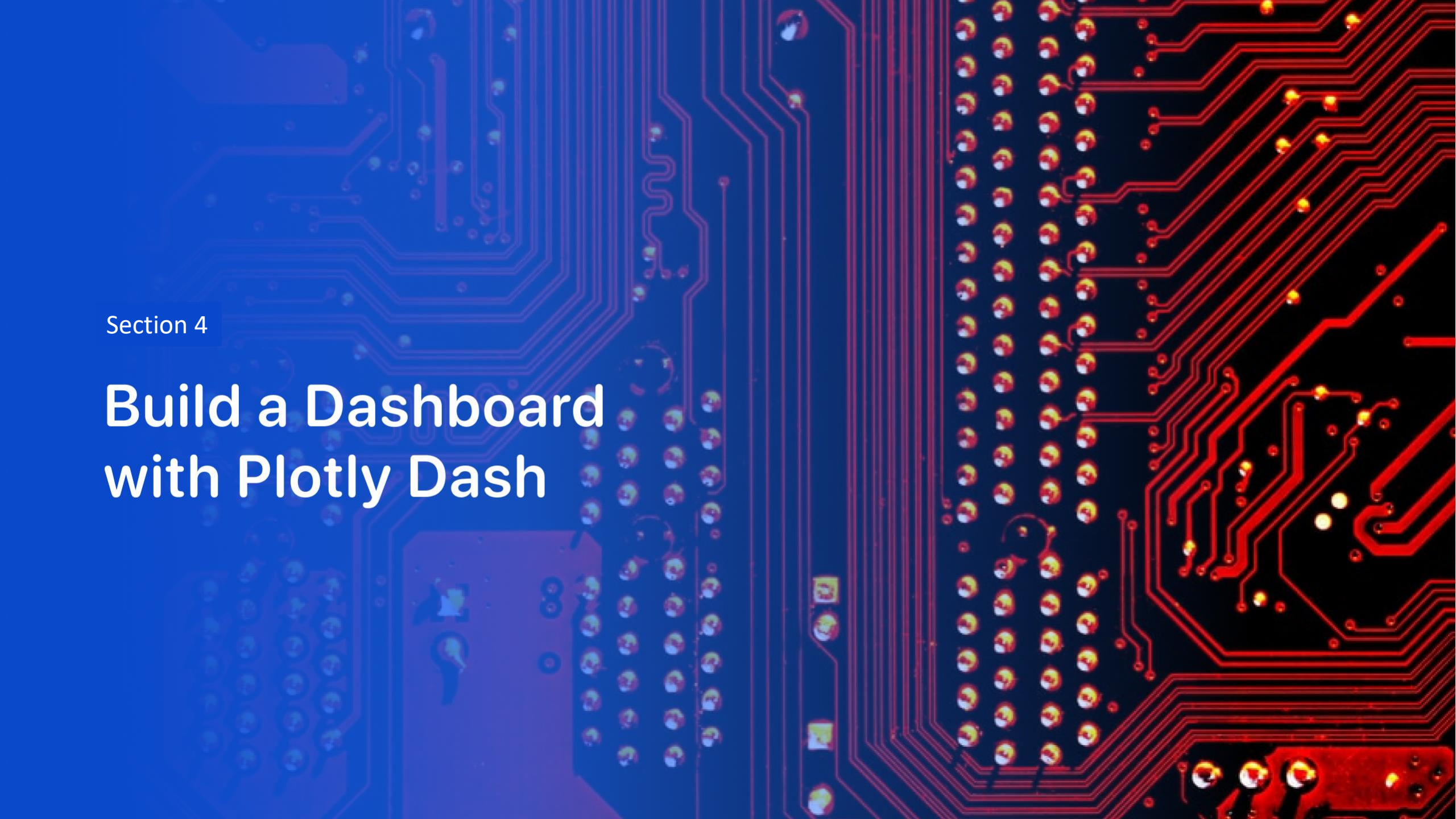
Section 3

Launch Sites Proximities Analysis

Location of all the Launch Sites

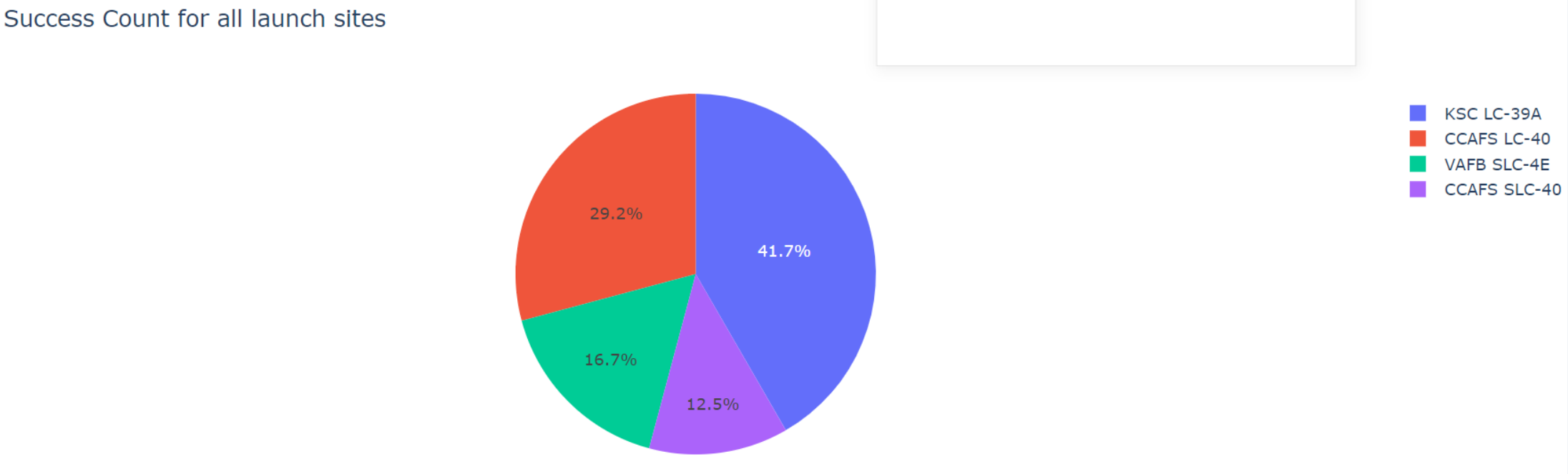


In the map we can see that all of the launch sites are located in the United states of America


Section 4
Build a Dashboard with Plotly Dash

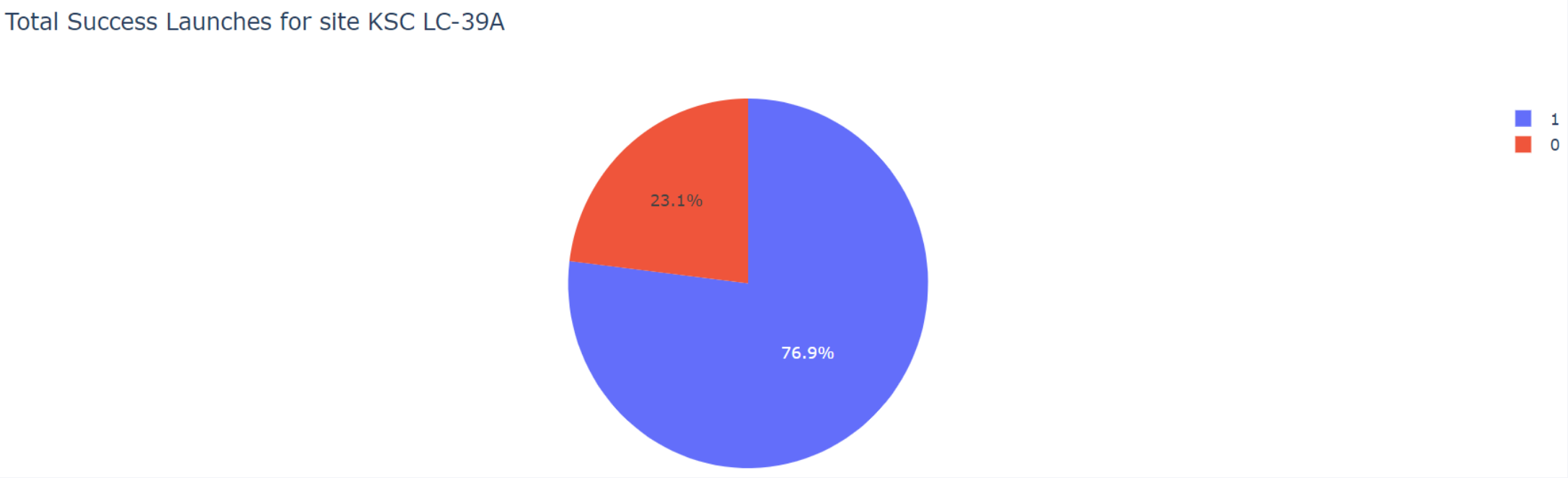
Success Percentage of Each Launch Site

The Pie Chart shows KSC LC-39A has the most number of successful launches amongst all the launch sites



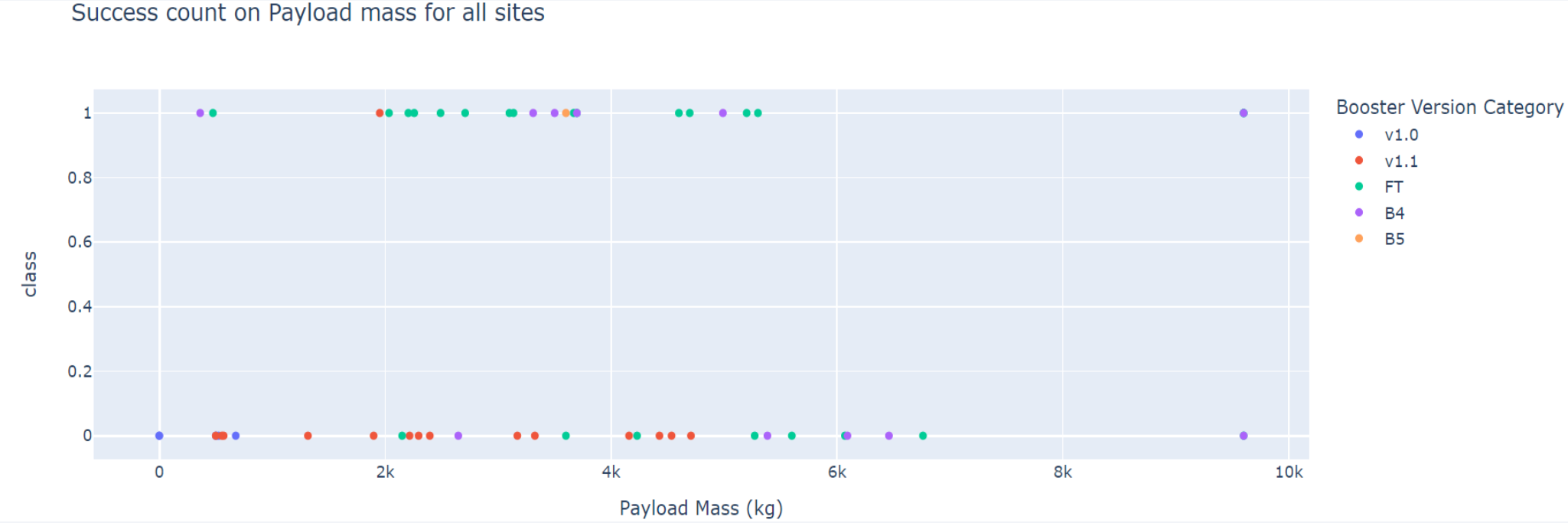
Launch Outcomes of KSC LC-39A (Success & Failure)

Launch Site KSC LC-39A has a success rate of 76.9% and failure rate of 23.1%



Payload Mass vs Launch Outcome Scatter Plot

The low weight payload has higher success rate than the heavy weight payload.



Section 5

Predictive Analysis (Classification)

Classification Accuracy

By using the below code, we can identify that the best algorithm is the Tree Algorithm which have the highest classification accuracy.

```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

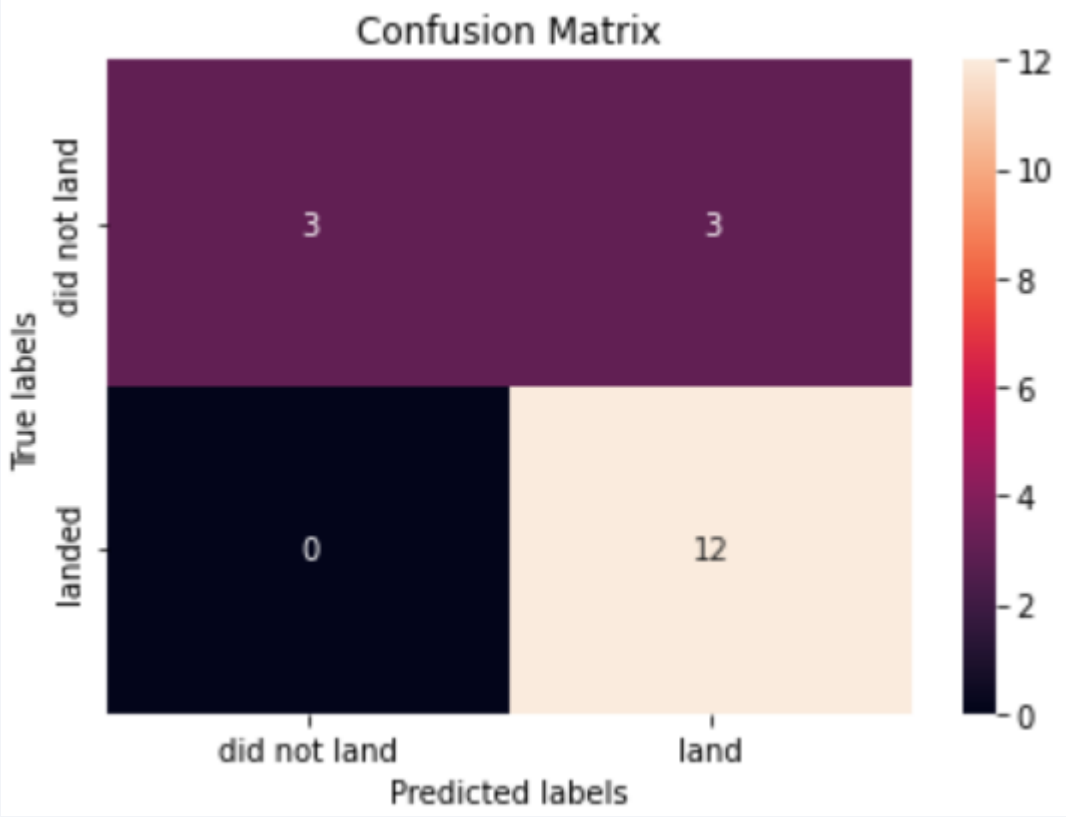
```
grid_search = GridSearchCV(tree, parameters, cv=10)
tree_cv = grid_search.fit(X_train, Y_train)
```

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 12, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 10, 's
plitter': 'best'}
accuracy : 0.8892857142857142
```

Confusion Matrix

The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is that unsuccessful landing is marked as successful landing by the classifier.



Conclusions

- The success rate for SpaceX launches has increased from 2013 to 2020. From this we can determine that there will be more successful launches.
- The launch with low weight payload which are 4000Kgs and below performed better than launch with heavy weight payload.
- Launch Site KSC LC-39A have the most successful launches with success rate of 76.9%
- SSO orbit have the most success rate of 100% and more than 1 occurrence.
- The Tree Classifier Algorithm is the best Machine Learning approach for this dataset.

Thank you!