IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Satrio Nindito
28th August 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data Wrangling
  - Exploratory Data Analysis
  - Predictive Analysis
- Summary of all results
  - EDA Results
  - Interactive visual using Folium dan Dash Plotly
  - Predictive Analytics Result

Introduction

- Project background and context

  Designed and operated by SpaceX, **Falcon 9** is a partially reusable, human-rated, two-stage-to-orbit, medium-lift launch vehicle designed and manufactured in the United States by SpaceX. The first Falcon 9 launch was on 4 June 2010. In December 2015, Falcon 9 became the first rocket to land propulsively after delivering a payload into orbit. This reusability results in significantly reduced launch costs, as the cost of the first stage constitutes the majority of the cost of a new rocket. The goal is to determine the price of each launch. We will do this by gathering information about Space X and creating dashboards for the team. And also determine if SpaceX will reuse the first stage. We will train a machine learning model and use public information to predict if SpaceX will reuse the first stage.

- Problems you want to find answers
  - What are the factors that can affect the rocket success to landed
  - The correlation of each features with the success rate of landing
  - Which conditions of the operation need to be present to ensure the successful landing

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collection using a get request to the SpaceX API and web scraping from Wikipedia webpage.
- Perform data wrangling
  - Calculate number of launch on each site, calculate number and occurence of each site, calculate the number and occurrence of mission outcome per orbit type and create a landing outcome label.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification model
  - Using the best hyperparameter values, we will determine the model with the best accuracy using the training data.

Data Collection

- The data sets collecting sources:
  - The SpaceX REST API
  - Falcon 9 Past Launches from wikipedia page
- The data collecting methods:
  - A get request using the requests library to get the data from the API.
  - The response will be in the form of a JSON and convert it to a pandas dataframe using the json_normalize function.
  - Using the Python BeautifulSoup package to web scrape wikipedia page tables that contain valuable Falcon 9 launch records.
  - Then parse the data from the tables and convert them into a Pandas dataframe
  - Then clean the datasets and dealing with missing data

Data Collection – SpaceX API

- For data collecting from SpaceX REST API, we used a get request then turn it into pandas dataframe using json_normalize. Clean the data by eliminate data that are not we use and dealing with the missing data.

- GitHub URL: https://github.com/sat150/IBM_Data_Science_Capstone_SPACEX/blob/main/Data-Collection-API.ipynb

```
[10]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN
      response=requests.get(static_json_url)
```

We should see that the request was successfull with the 200 status response code

```
[11]: response.status_code

[11]: 200
```

Now we decode the response content as a Json using `.json()`

and turn it into a Pandas dataframe using `.json_normalize()`

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

```
[28]: # Calculate the mean value of PayloadMass column
      PayloadMass_Avg = data_falcon9['PayloadMass'].mean()

      # Replace the np.nan values with its mean value
      data_falcon9['PayloadMass'].replace(np.nan, PayloadMass_Avg, inplace=True)
      data_falcon9.isnull().sum()
```

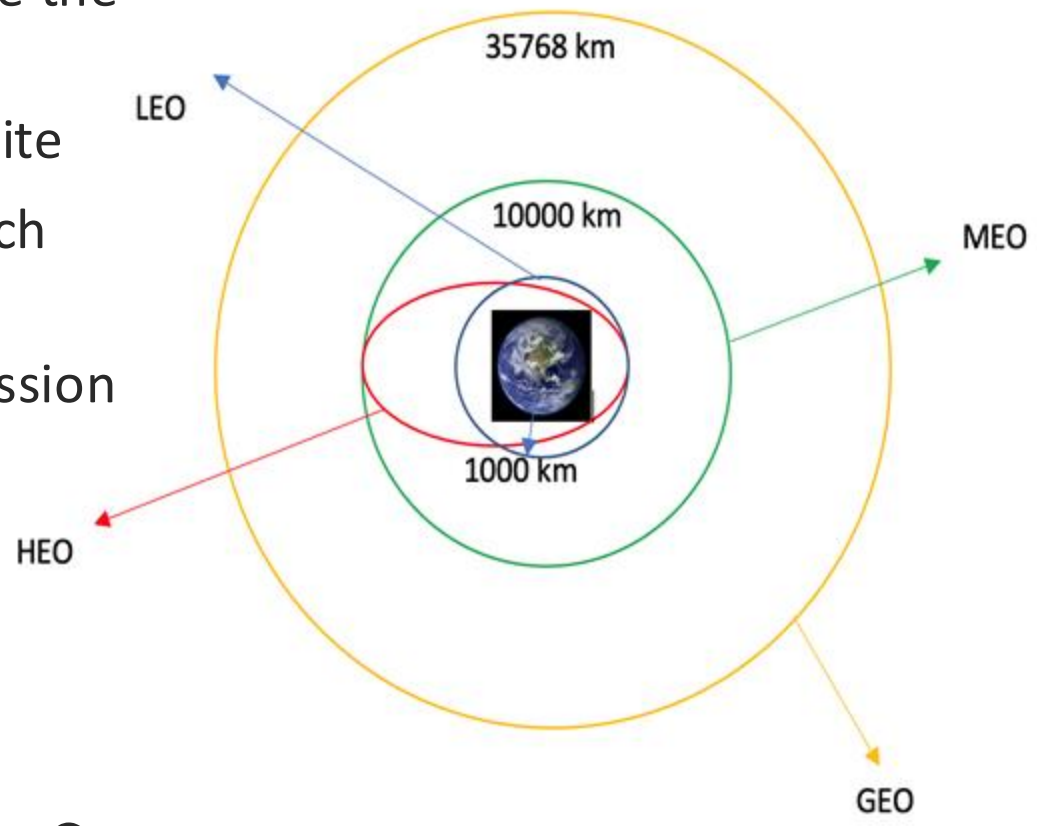
Data Collection - Scraping

- We use BeautifulSoup to web scraping the Falcon 9 from wikipedia page.
- After we extract the column name, we fill in the parsed launch record values into launch_dict, and create a dataframe from it.
- The GitHub URL: https://github.com/sat150/IBM_Data_Science_Capstone_SPACEX/blob/main/Data-Webscraping.ipynb

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

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

[16]: 200

      Create a BeautifulSoup object from the HTML response

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

      ...

[19]: # Use soup.title attribute
      soup.title

[19]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>

      ...

[22]: column_names = []

      # Apply find_all() function with `th` element on first_launch_table
      # Iterate each th element and apply the provided extract_column_from_header() to get a column name
      # Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
      headers =  first_launch_table.find_all('th')
      for i in headers:
          column = extract_column_from_header(i)
          if column is not None and len(column) > 0:
              column_names.append(column)
```

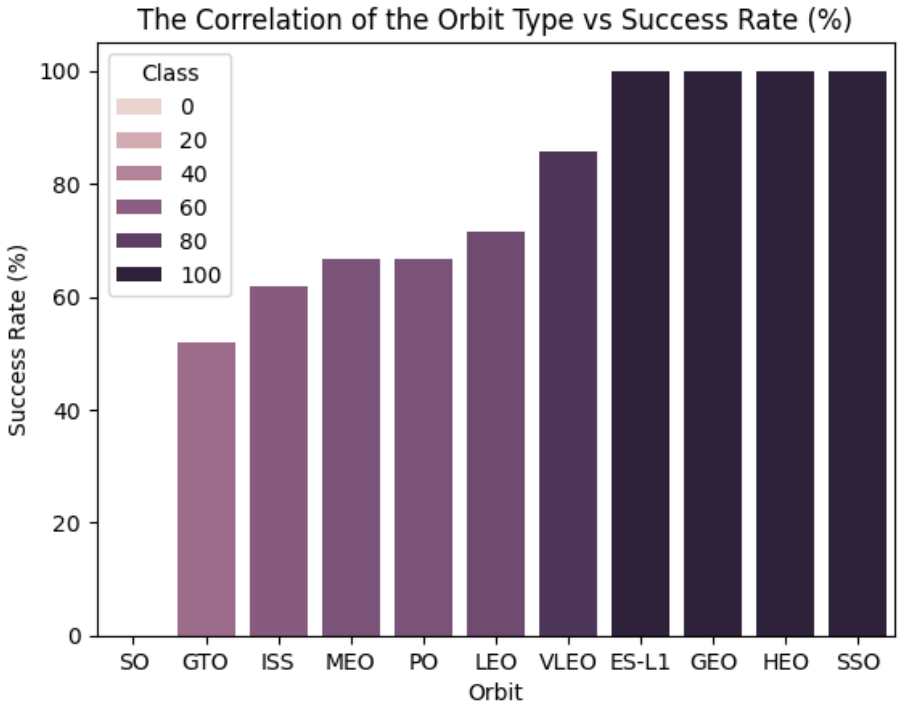
Data Wrangling

- We will perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine the label for training supervised models.
- We calculate the number of launches on each site
- We calculate the number and occurrence of each orbit
- We calculate the number and occurrence of mission outcome of the orbits
- Create a landing outcome label from Outcome column
- Export the result to csv
- The GitHub URL: https://github.com/sat150/IBM_Data_Science_Capstone_SPACEX/blob/main/Data%20wrangling.ipynb



EDA with Data Visualization

- The charts below on the left is a line chart of the average launch success rate trend from 2010 until 2020
- On the right is a bar chart that shows the correlation between Orbit Type and the success rate (%)
- The GitHub URL: https://github.com/sat150/IBM_Data_Science_Capstone_SPACEX/blob/main/EDA%20with%20Data%20Visualization.ipynb



EDA with SQL

- Using the python sql magic we load the SpaceX dataset, we execute the sql query to get answer for below questions:
  - Display the names of the unique launch sites in the space mission
  - 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 names of the booster_versions which have carried the maximum payload mass.
  - 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.

- The GitHub URL: https://github.com/sat150/IBM_Data_Science_Capstone_SPACEX/blob/main/EDA-SQL.ipynb

Build an Interactive Map with Folium

- We used marker and circle for all launch sites on map.
- We added colored mark to each launch site with the success and failed launch outcome where class 0 is red and class 1 is green.
- We used lines to calculate the distance between launch site and the proximities such as Highway, Railway and the closest City.
- The GitHub URL: https://github.com/sat150/IBM_Data_Science_Capstone_SPACEX/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- We plot a Pie chart that show the total success rate for all launch site and each launch site based on the choose dropdown site.
- We plot a Scatter chart that show the correlation between Payload Mass (kg) and the success rate for each booster version.
- There is also a Payload Mass (kg) slide bar to give a dynamic interaction between Payload Mass (kg) and the scatter chart result.

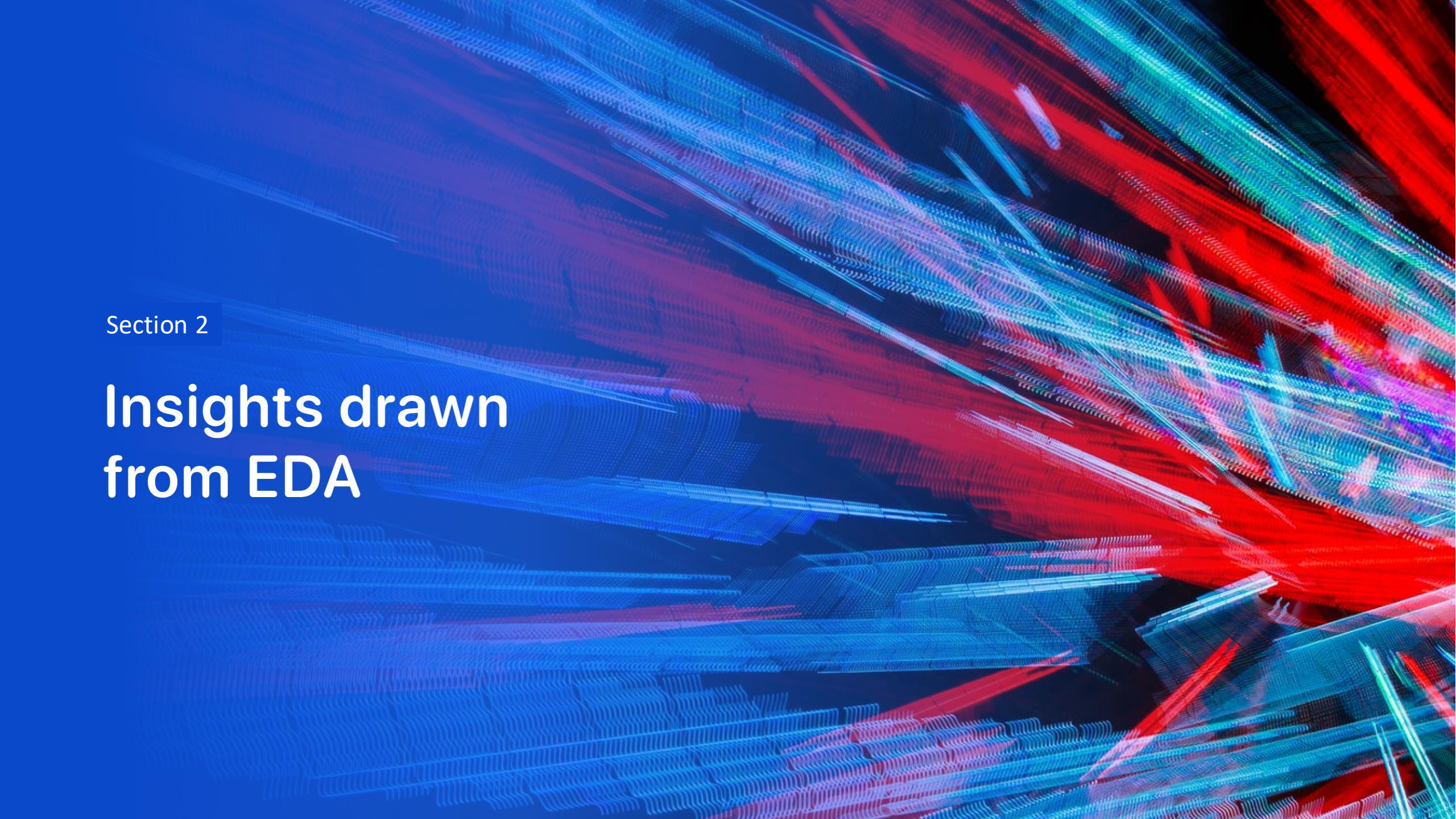
- Add the GitHub URL: https://github.com/sat150/IBM_Data_Science_Capstone_SPACEX/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

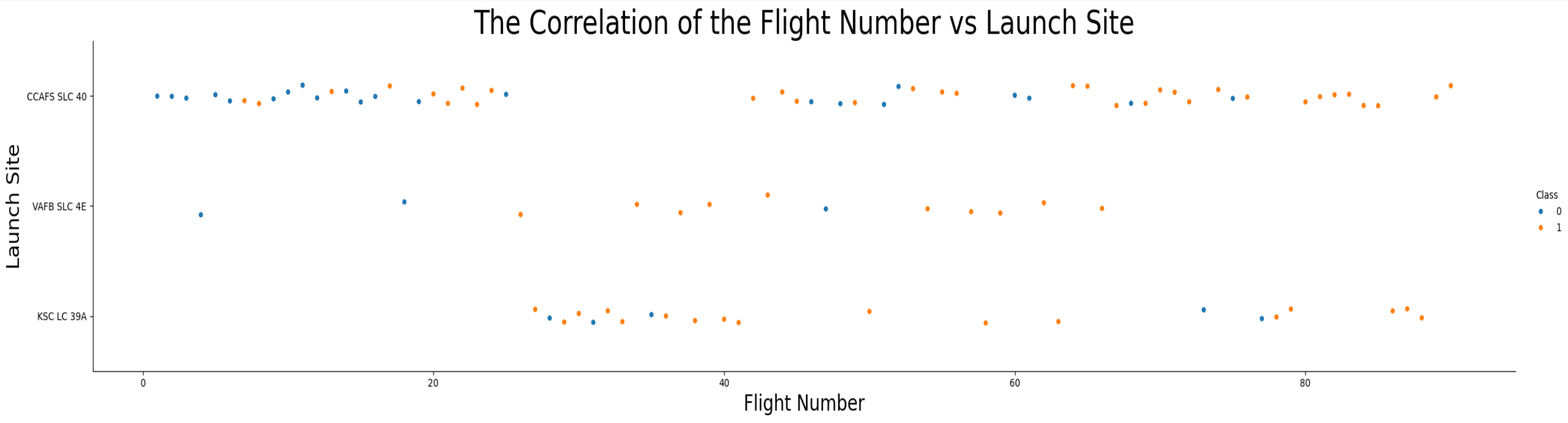
- We load the dataset into pandas dataframe
- By using numpy we create array from column "Class" in the data
- We standardize the data in X and then reassign it to the variable X using the transform
- We split the data into train (80%) and test (20%) dataset
- We use Logistics Regression, SVM, Decision Tree Classifier, and k nearest neighbors model. And tuning the hyperparameters using GridSearchCV
- Then after compare from each model, we got the best method that perform best of them all
- The GitHub URL: https://github.com/sat150/IBM_Data_Science_Capstone_SPACEX/blob/main/Machine%20Learning%20Prediction.ipynb

Results

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

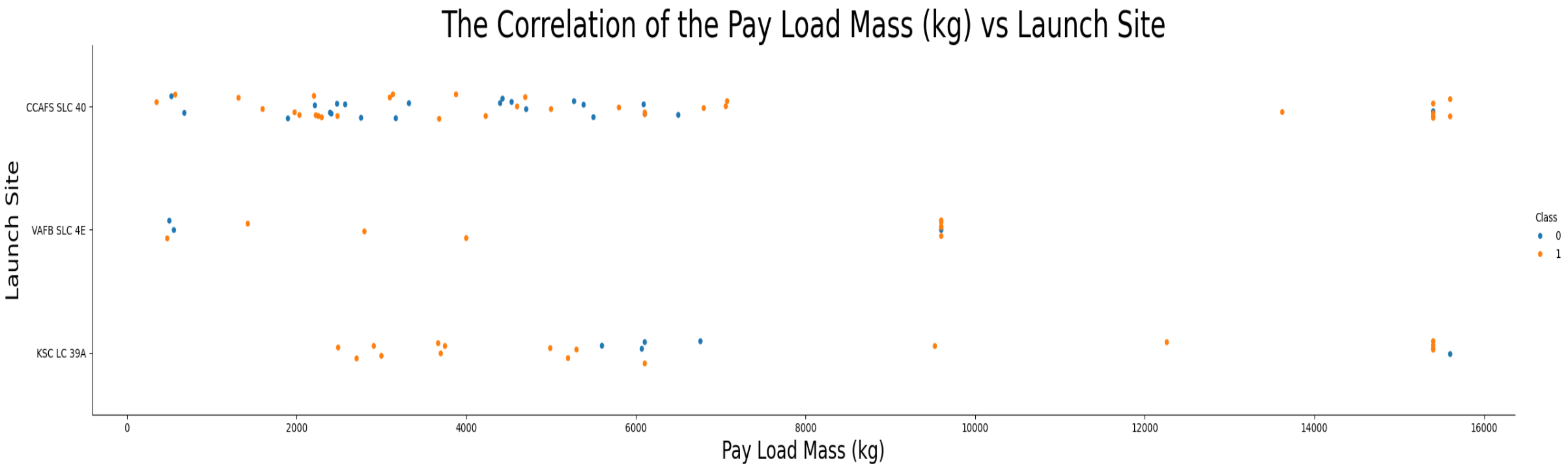

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



- From the scatter plot, we can see that the best success Launch Site is CCAFS SLC 40, because this Launch Site have more Flight Number then the others Launch Site

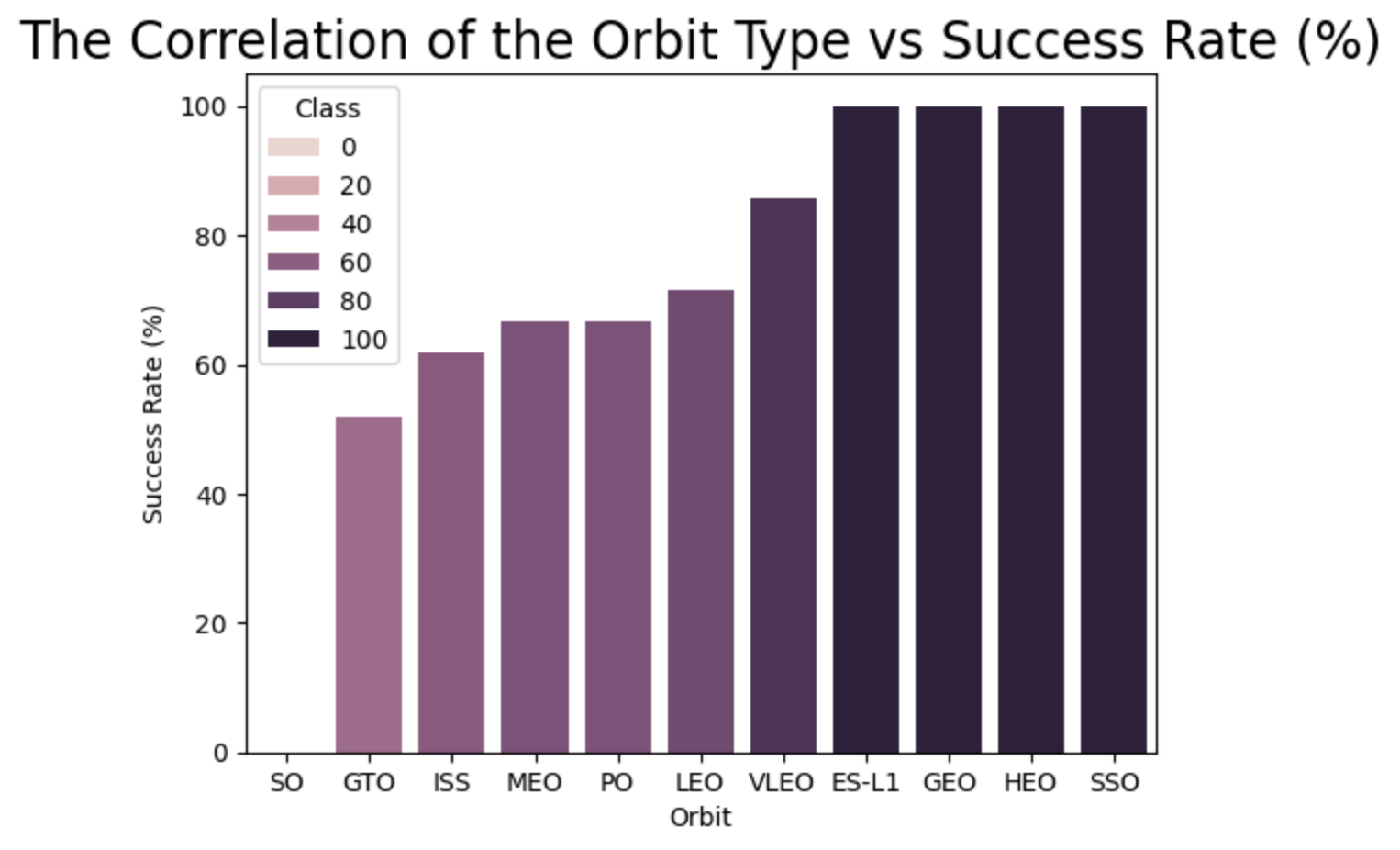
Payload vs. Launch Site



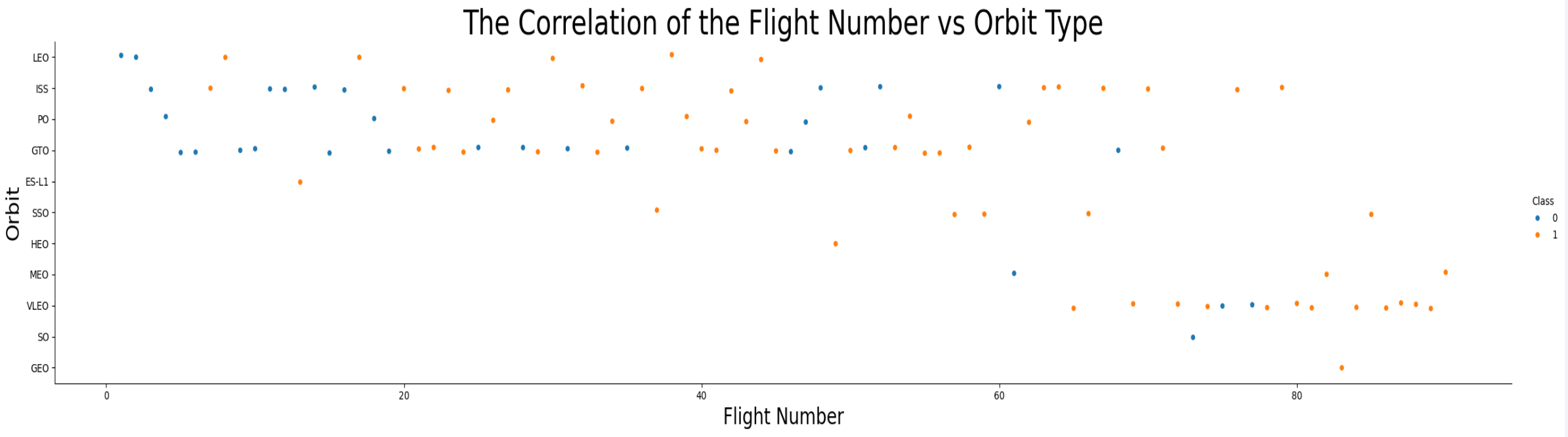
- From the scatter plot, it shows that with more Payload in the Flight it give more success rate rather then the lower Payload in the Flight

Success Rate vs. Orbit Type

- From the bar chart we can see that the most success rate are Orbit with these type: ES-L1, GEO, HEO, SSO and VLEO

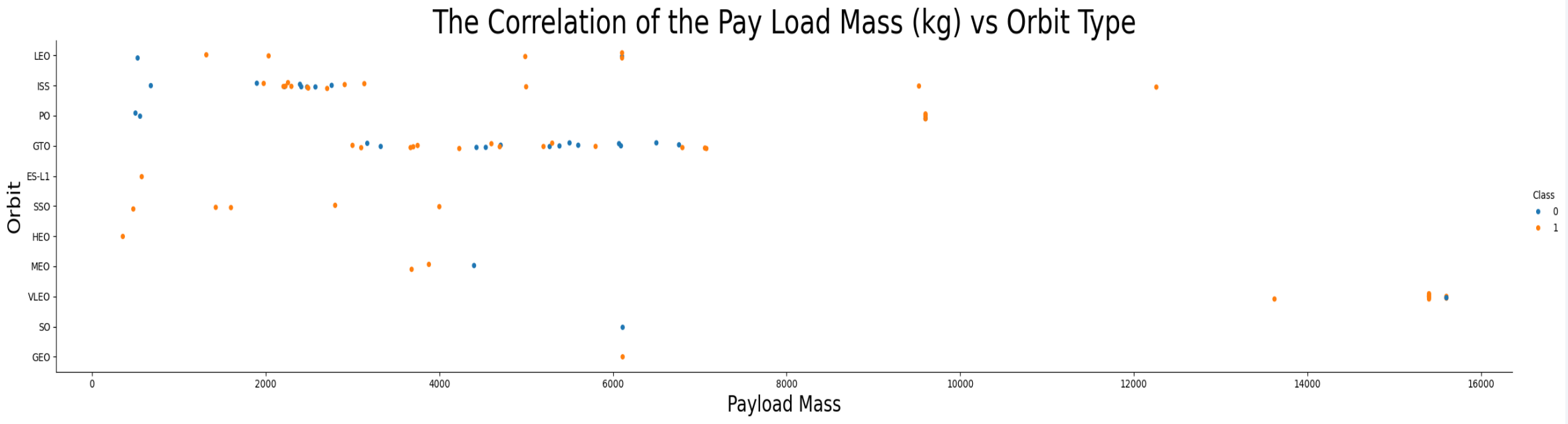


Flight Number vs. Orbit Type



- From the scatter plot, we can see that ISS, GTO, and VLEO orbit have a great number of success mission

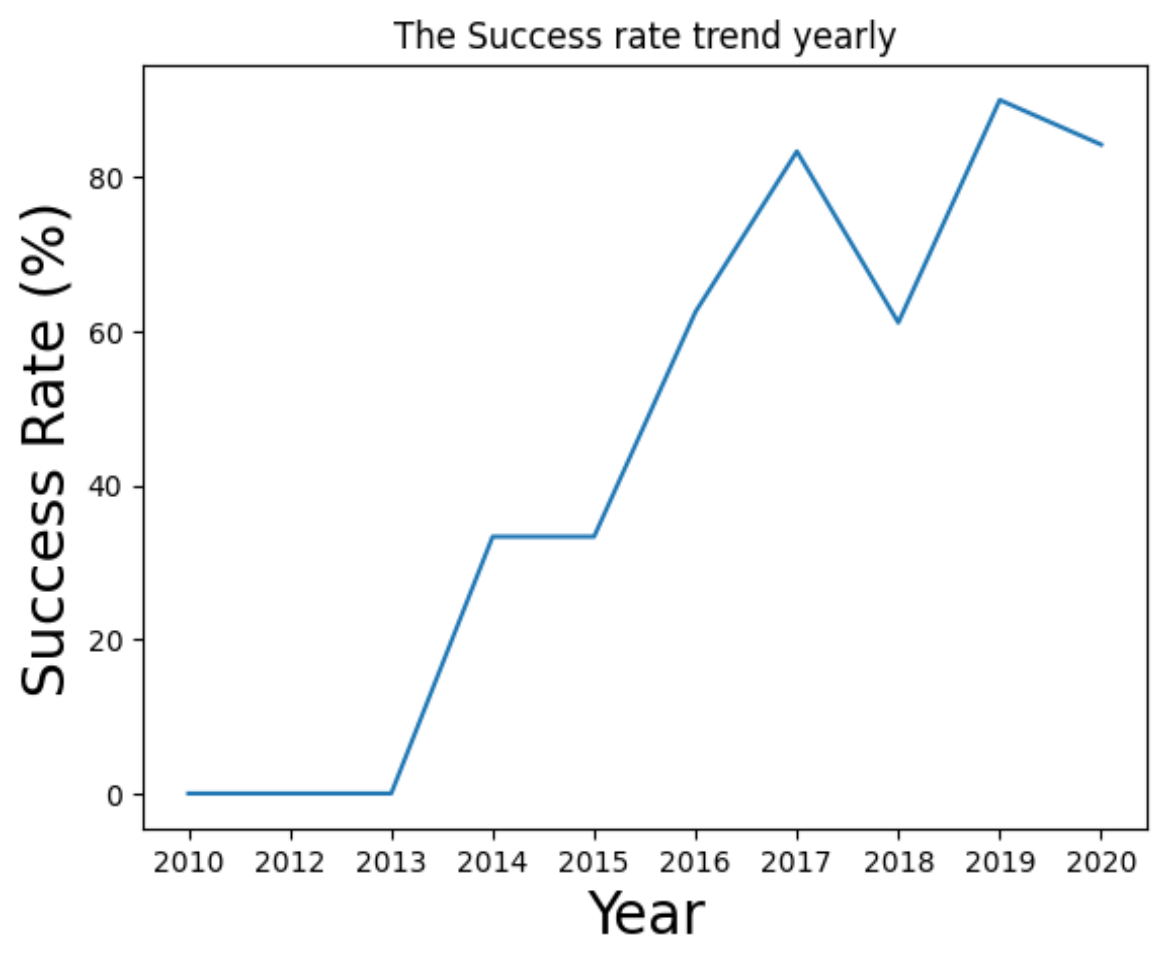
Payload vs. Orbit Type



- From the scatter plot we can see with more heavy payload especially at Orbit type: LEO, PO, and VLEO have many success mission

Launch Success Yearly Trend

- We can see based on the line chart the success rate is increasing from 2013 to 2020



All Launch Site Names

- At the right side are all Launch Site names, we can get the data using DISTINCT query

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

```
[27]: %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
 * sqlite:///my_data1.db
Done.
```

[27]:

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'

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

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

[29]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- By adding query " FROM LaunchSite LIKE 'CCA' LIMIT 5", we can search the Launch Site that have string 'CCA' and only show 5 rows from the data

Total Payload Mass

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

```
[35]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS Payload_Mass_Tot FROM SPACEXTABLE WHERE Customer LIKE "NASA (CRS)"
 * sqlite:///my_data1.db
Done.
```

[35]:

Payload_Mass_Tot
45596

- By using query " SUM(PayloadMassKG) AS Total_Payload " and " WHERE Customer LIKE 'NASA (CRS)' ", we calculated all Payload Mass that carried by boosters from NASA (CRS) and the result is 45596

Average Payload Mass by F9 v1.1

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

```
[37]: %sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_Payload_F9 FROM SPACEXTABLE WHERE Booster_Version LIKE "F9 V1.1"
```

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

[37]:

AVG_Payload_F9
2928.4

- By using this query we can calculate the average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date

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

Hint:Use min function

```
[38]: %sql SELECT MIN(Date) AS FIRST_SUCCESS_LAND FROM SPACEXTABLE WHERE Landing_Outcome LIKE "Success (ground pad)"
 * sqlite:///my_data1.db
Done.
[38]: FIRST_SUCCESS_LAND
2015-12-22
```

- By using syntax "MIN(DATE)", we can search the first date in the column DATE with the condition "Success (ground pad)"

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
[40]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome LIKE "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
 * sqlite:///my_data1.db
Done.
```

[40]:

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

- We use AND condition to get the payload mass number greater than 4000 and less than 6000

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
[50]: %sql SELECT COUNT(Mission_Outcome) AS Mission_Success FROM SPACEXTABLE WHERE Mission_Outcome LIKE "Success%"
 * sqlite:///my_data1.db
Done.
```

[50]:

Mission_Success
100

```
[46]: %sql SELECT COUNT(Mission_Outcome) AS Mission_Failure FROM SPACEXTABLE WHERE Mission_Outcome LIKE "Failure%"
 * sqlite:///my_data1.db
Done.
```

[46]:

Mission_Failure
1

- By using query WHERE MissionOutcome LIKE "Success%", we can get the data of the successful mission
- While query WHERE MissionOutcome LIKE "Failure%", we can get the data of the failure mission

Boosters Carried Maximum Payload

- By using subquery like the picture shown we can get the data of which booster that have maximum payload mass

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

```
[66]: %sql SELECT Booster_Version, PAYLOAD_MASS__KG_  FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ =  (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
 * sqlite:///my_data1.db
Done.
```

[66]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

```
[72]: %sql SELECT substr(Date,6,2) AS Month, Landing_Outcome,Booster_Version, Launch_Site FROM SPACEXTABLE WHERE Landing_Outcome LIKE "Failure (drone ship)" AND substr(Date,0,5) = '2015'
 * sqlite:///my_data1.db
Done.
```

[72]:

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- By selecting multiple column and the condition WHERE LandingOutcome LIKE 'Failure (drone ship)' AND DATE BETWEEN '2015-01-01' AND '2015-12-31', we can get the list of the failed landing_outcomes in drone ship

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

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.

```
[80]: %sql SELECT Landing_Outcome, COUNT(*) AS COUNT FROM SPACEXTABLE WHERE DATE >= "2010-06-04" AND DATE <= "2017-03-20" GROUP BY Landing_Outcome ORDER BY COUNT DESC
```

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

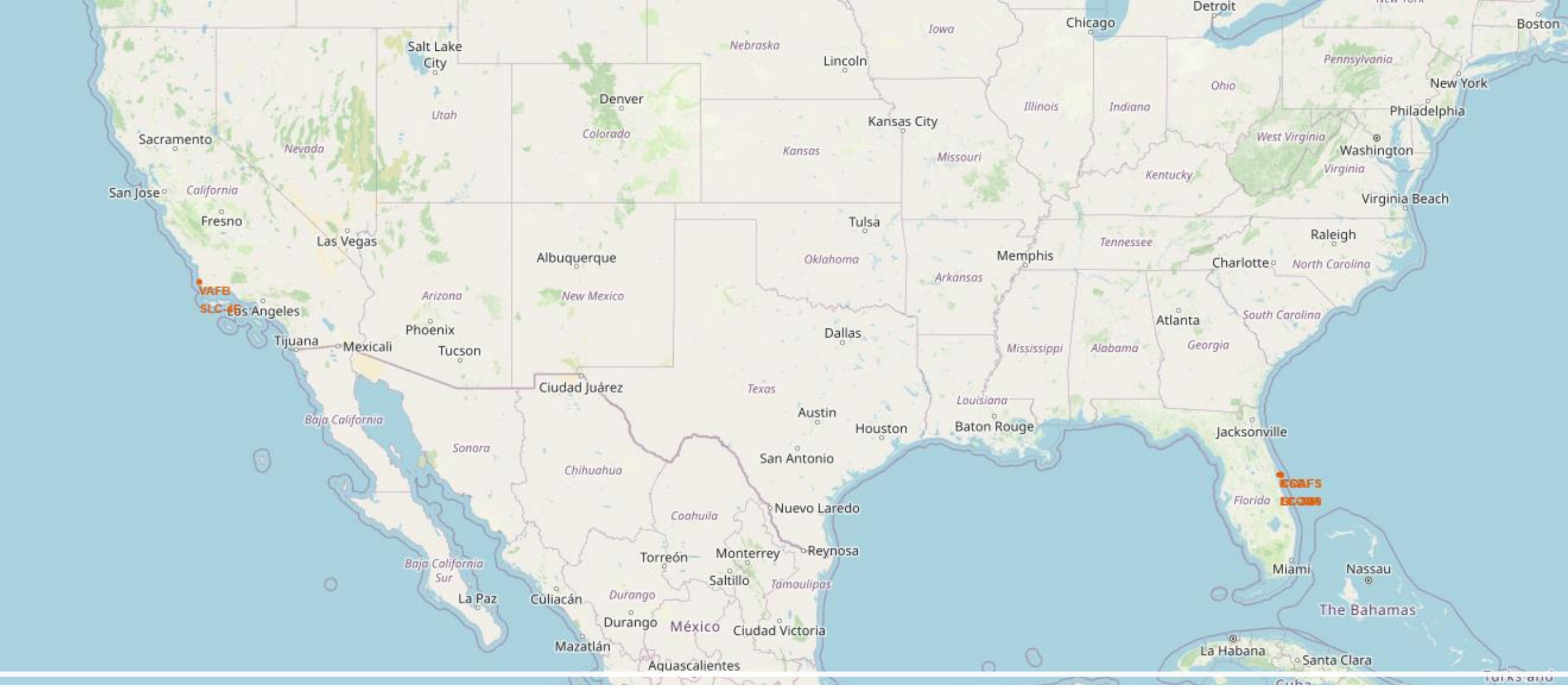
[80]:

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

- By using syntac COUNT we can get the rank of landing outcomes between 2010-06-04 and 2017-03-20 in descending order

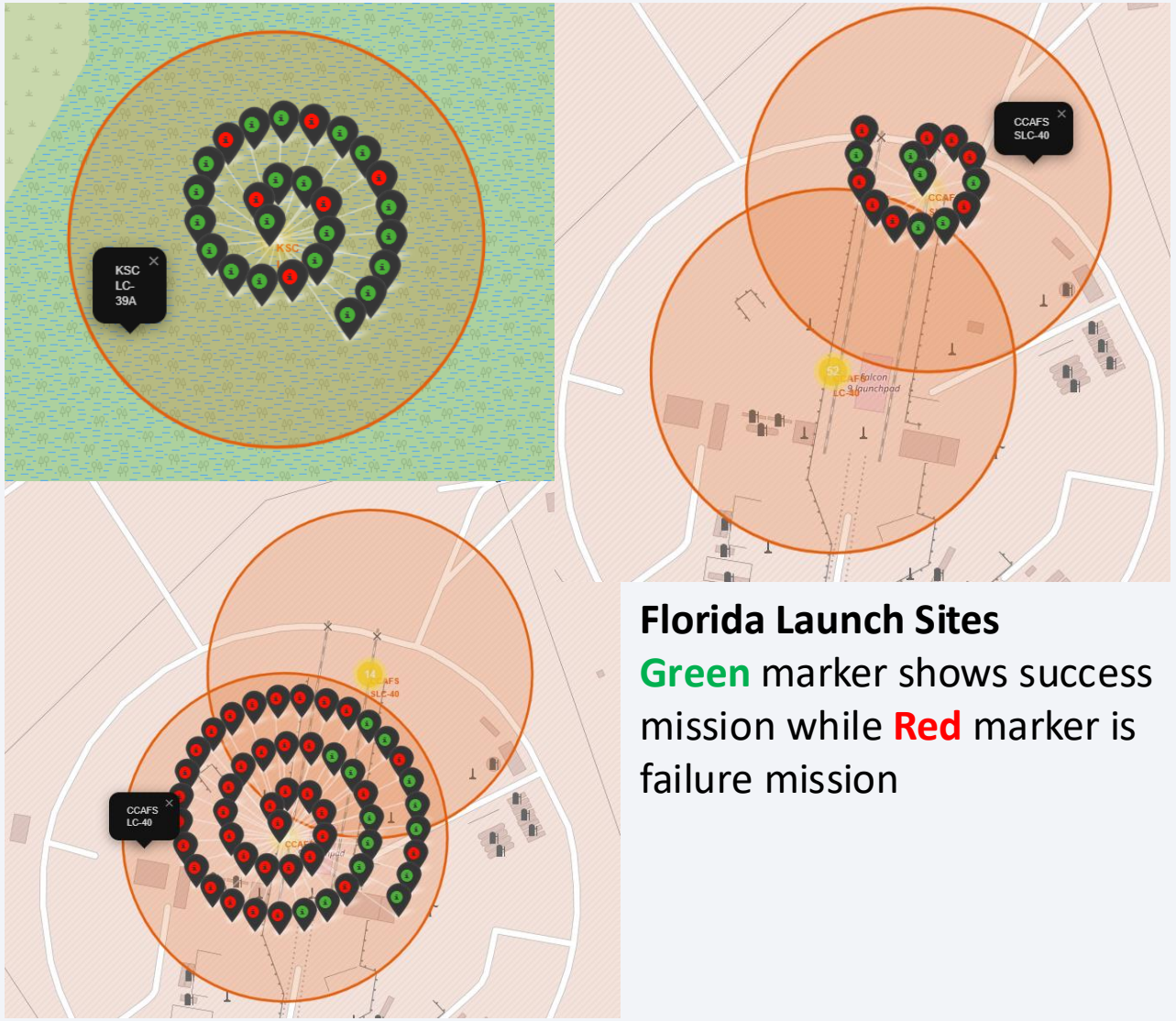
Section 3

Launch Sites Proximities Analysis



All Launch Site on Map

Mark the success/failed launches for each site on the map

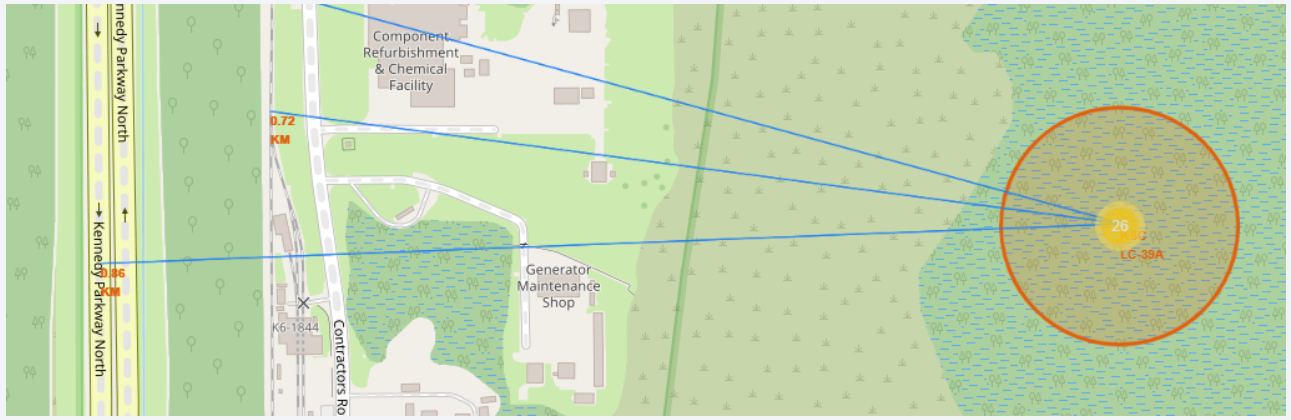


Florida Launch Sites

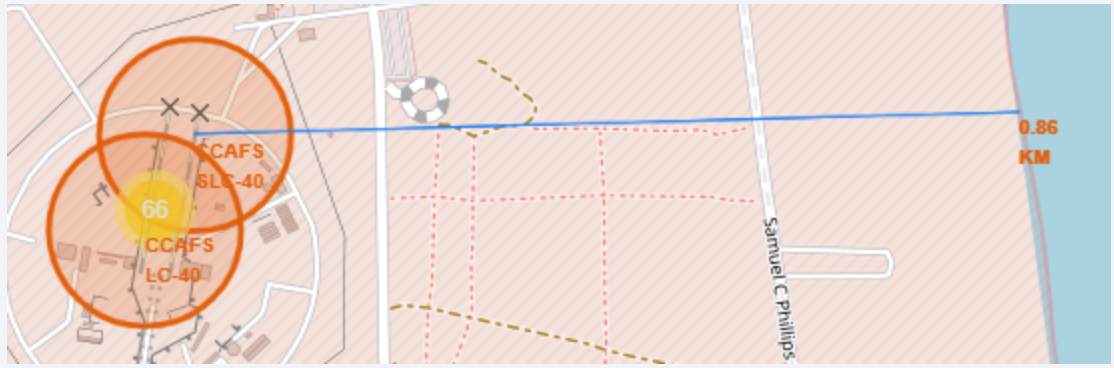
Green marker shows success mission while **Red** marker is failure mission

California Launch Sites

Calculate the distances between a launch site to its proximities

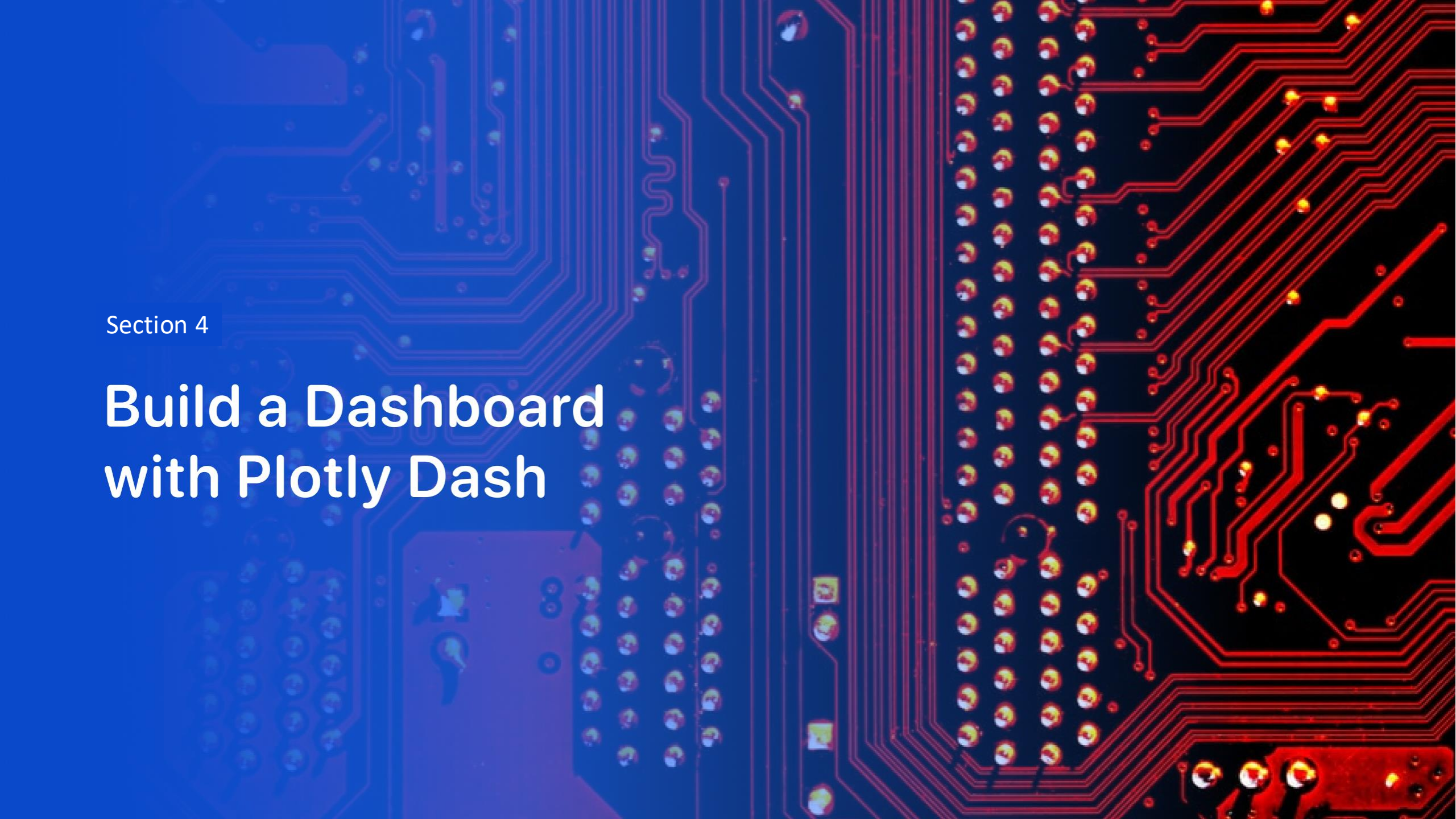


Distance to Railway and Highway

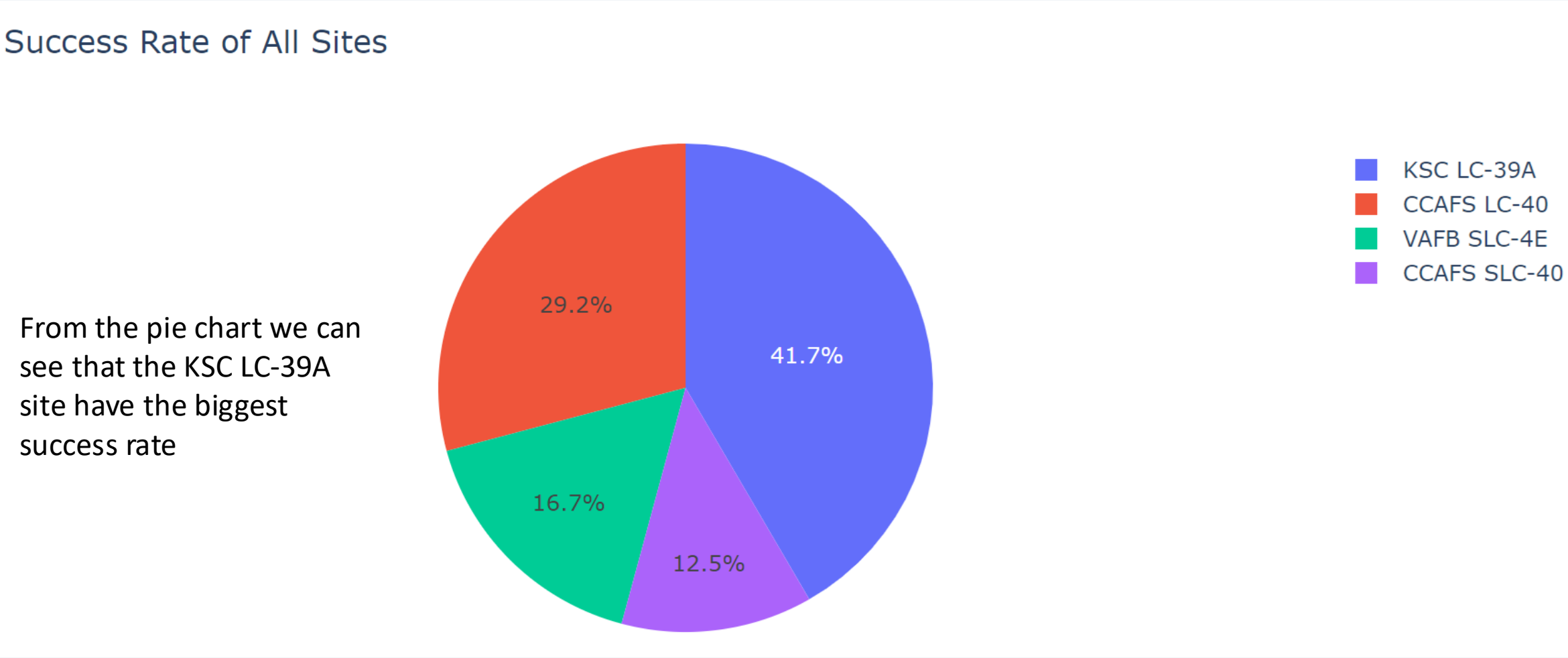


Distance to coastline

Distance to City

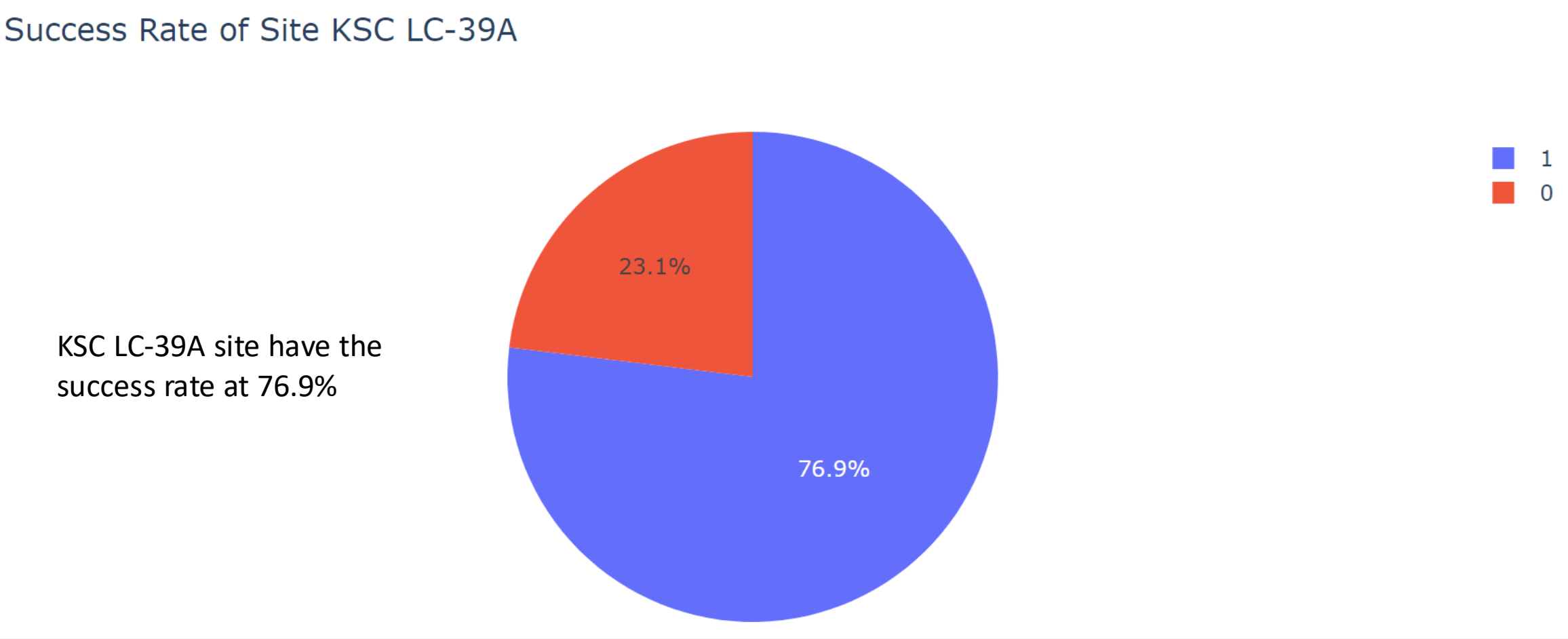

Section 4
Build a Dashboard with Plotly Dash

Pie Chart of the Success Rate for All Sites



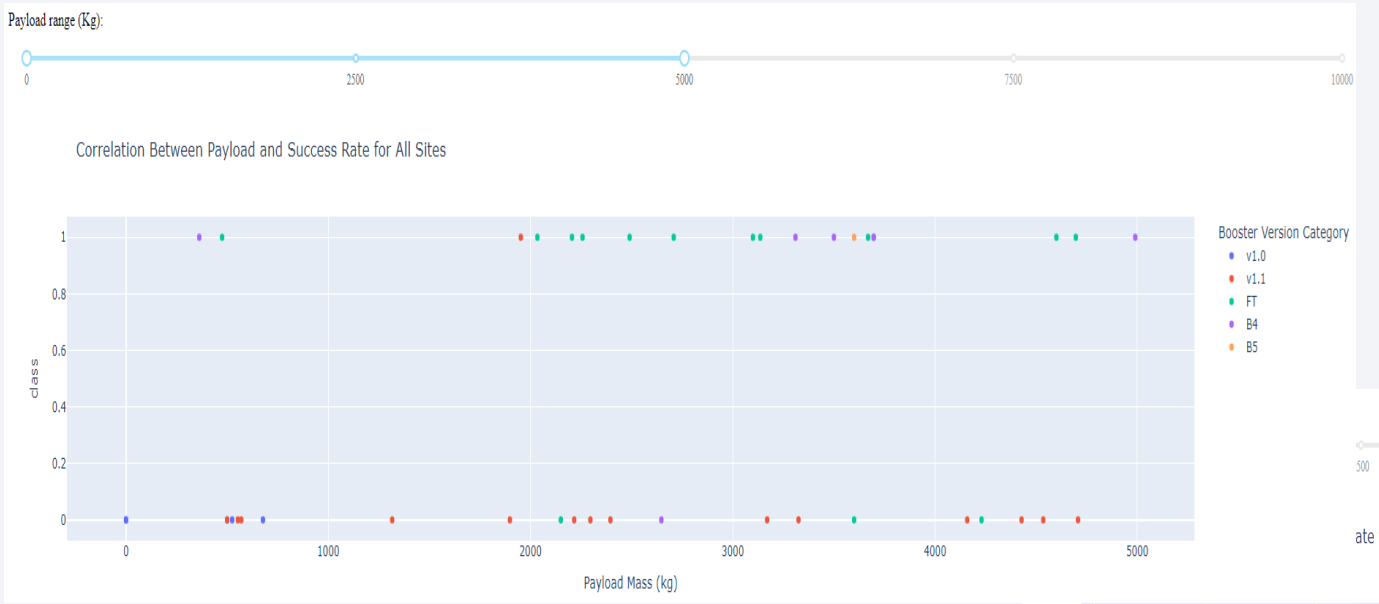
From the pie chart we can see that the KSC LC-39A site have the biggest success rate

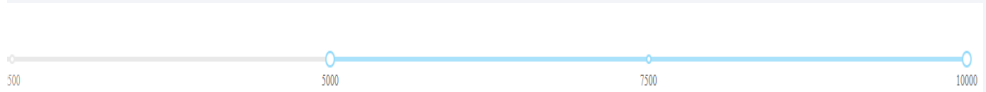
Pie Chart of the Success Rate for KSC LC-39A Site



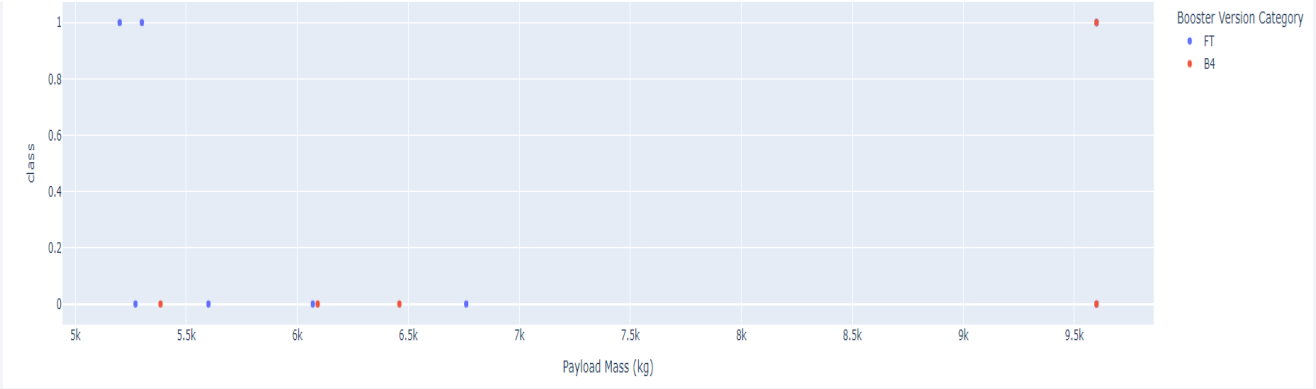
KSC LC-39A site have the success rate at 76.9%

Scatter plot of the correlation Payload Mass vs Booster Version





ate for All Sites



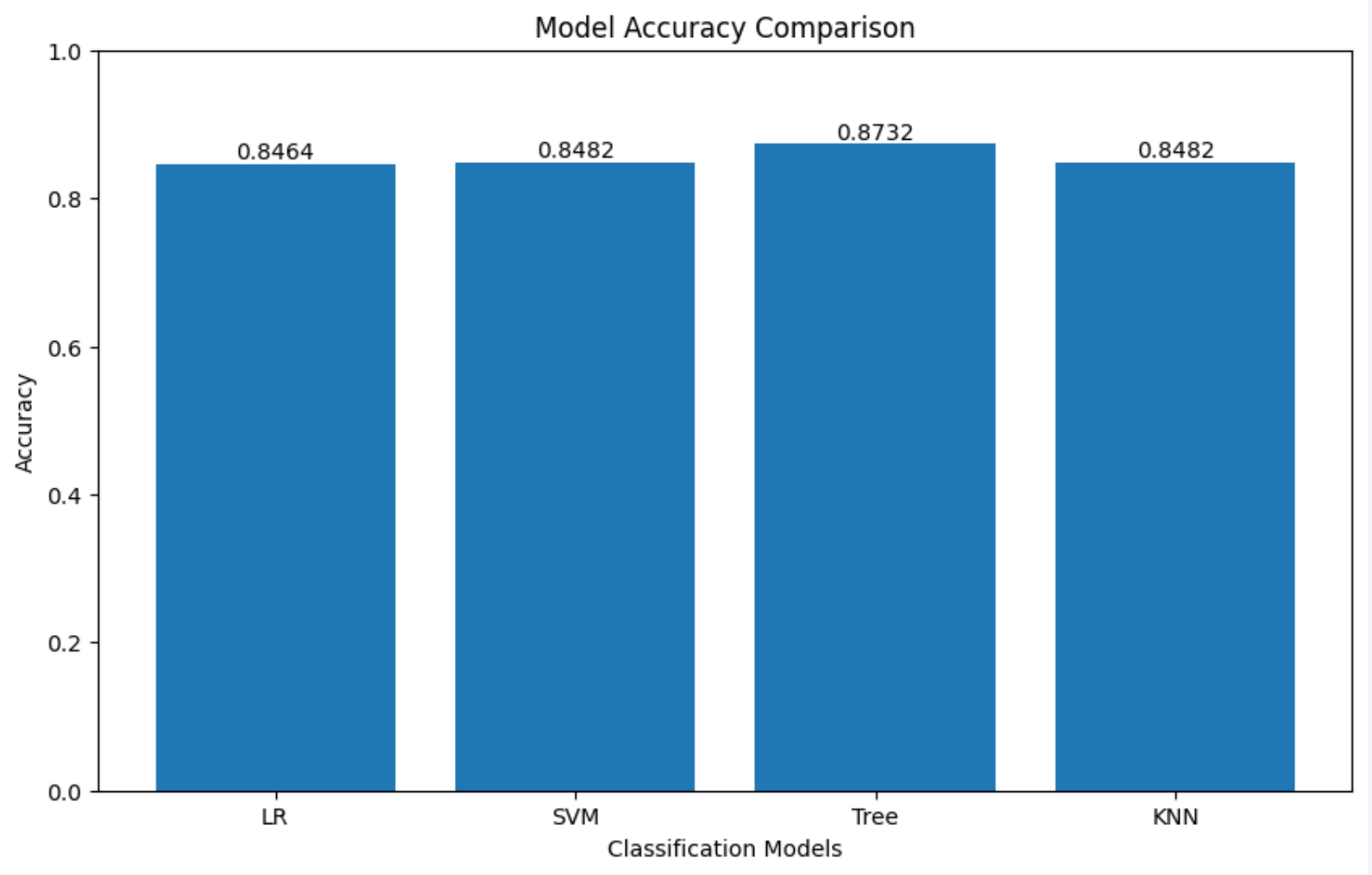
From the scatter plot we can see that the Booster Version have high success rate at payload mass lower than 5.000 Payload Mass (kg)

Section 5

Predictive Analysis (Classification)

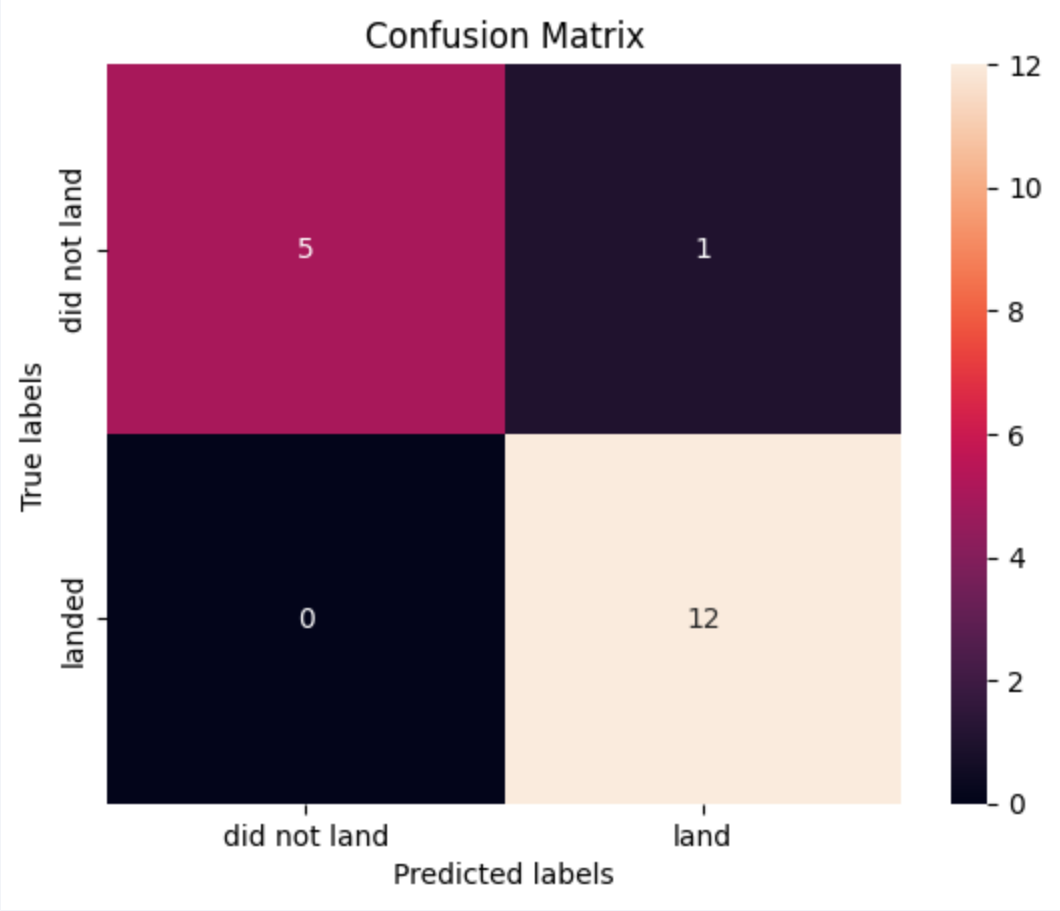
Classification Accuracy

- The DecisionTree model has the highest classification accuracy at 0.8732



Confusion Matrix

- a confusion matrix is an essential tool in evaluating the performance of a machine learning model. It provides insights into the accuracy and precision of predictions made by the model
- The DecisionTree Accuracy : 87%



Conclusions

- The more flight amount at the launch site, it give more success rate of the mission
- The more payload mass in flight it also increase the success rate
- The ES-L1, GEO, HEO, SSO and VLEO orbit type have the best success rate
- The Decision Tree algorithm is the best machine learning model for this case

Thank you!