IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Raju Boddupalli
29-Dec-2024

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- In this project, I want to predict whether Falcon 9 will land successfully or not
- SpaceX advertises Falcon 9 rocket launches on its website at 62 million dollars; other providers charge upward of 165 million dollars each.
- Much of the savings are because SpaceX can reuse the first stage. Therefore, if we can determine whether 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.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Made REST call to SpaceX API (https://api.spacexdata.com/v4/) and retrieved rockets, launchpad, payload, cores, and finally rocket launch data

Perform data wrangling

  - Describe how data was processed

- 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

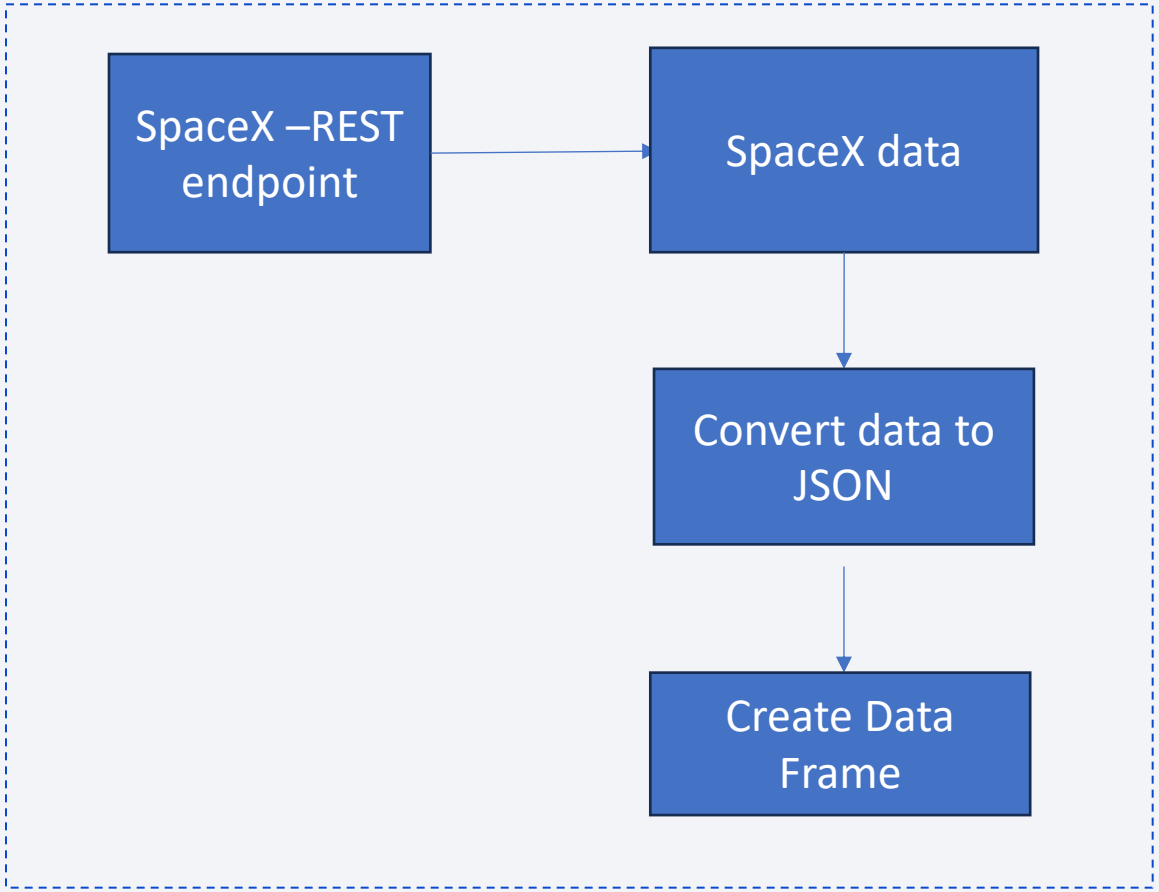
Data was collected using various methods

1. Made REST call to SpaceX API (https://api.spacexdata.com/v4/) and retrieved rockets, launchpad, payload, cores, and finally rocket launch data
2. Decoded and Normalized the JSON content and converted it into a data frame
3. Cleaned the data, and checked the data for missing values. Replaced the data with appropriate alternatives like mean
4. Also collected data from Wikipedia using beautifulSoup

Data Collection – SpaceX API

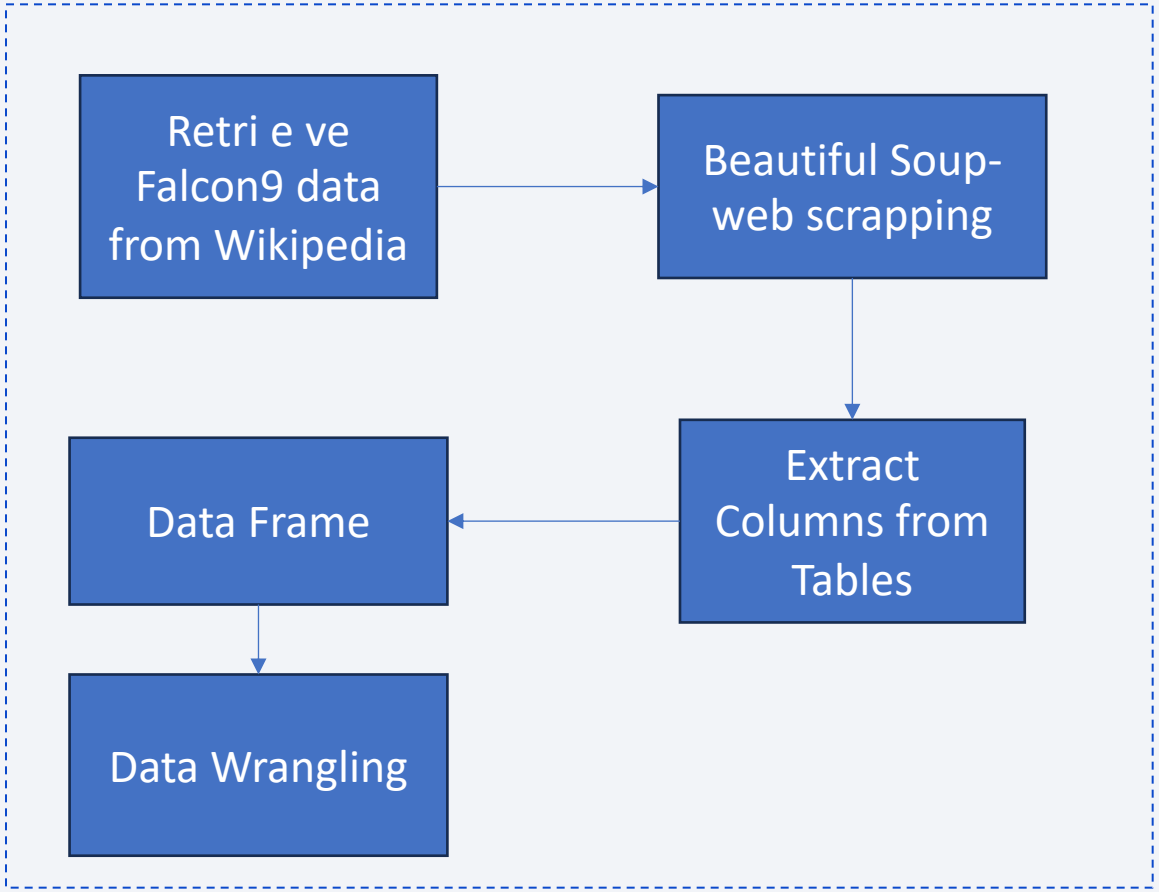
- Used the request to get the SpaceX API to collect data and did data wrangling and formatting

- Git Hub repository:github_link



Data Collection - Scraping

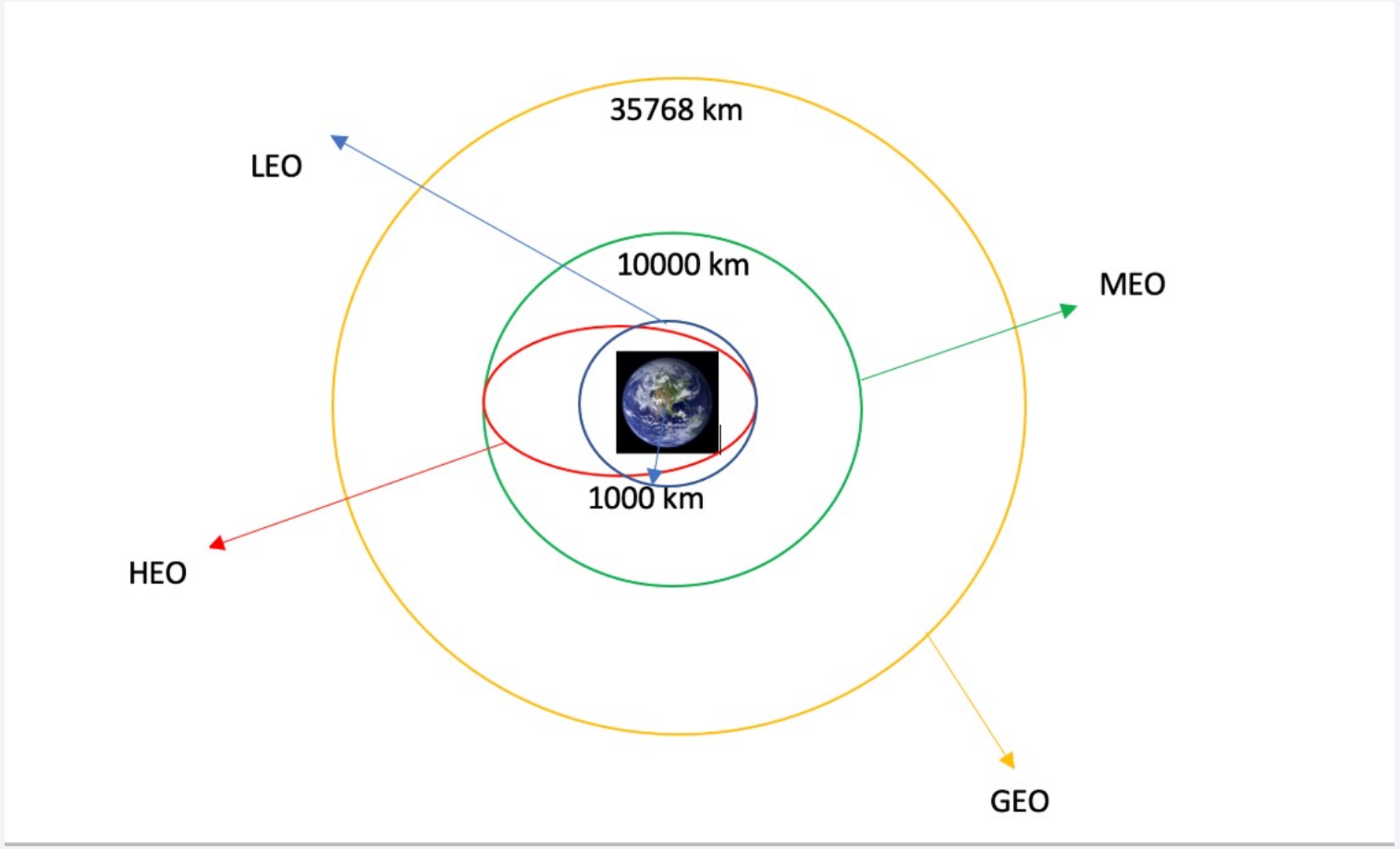
- Retrieved Falcon9 data from Wikipedia
- Used Beautiful Soup library to read columns and extract data
- Create data frame out of the table columns
- Data wrangling
- GitHub: Git Hub Link



Data Wrangling

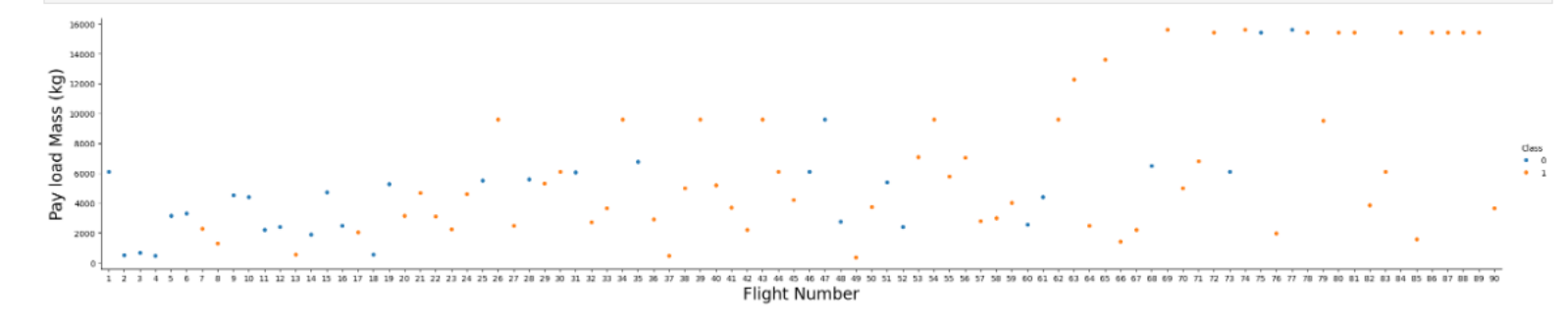
- Performed exploratory Data Analysis and determined Training Labels
- Calculated the number of launches on each site
- Calculated the number and occurrence of each orbit
- Calculate the number and occurrence of mission outcome of the orbits
- Created a landing outcome label from the Outcome column

Data Wrangling



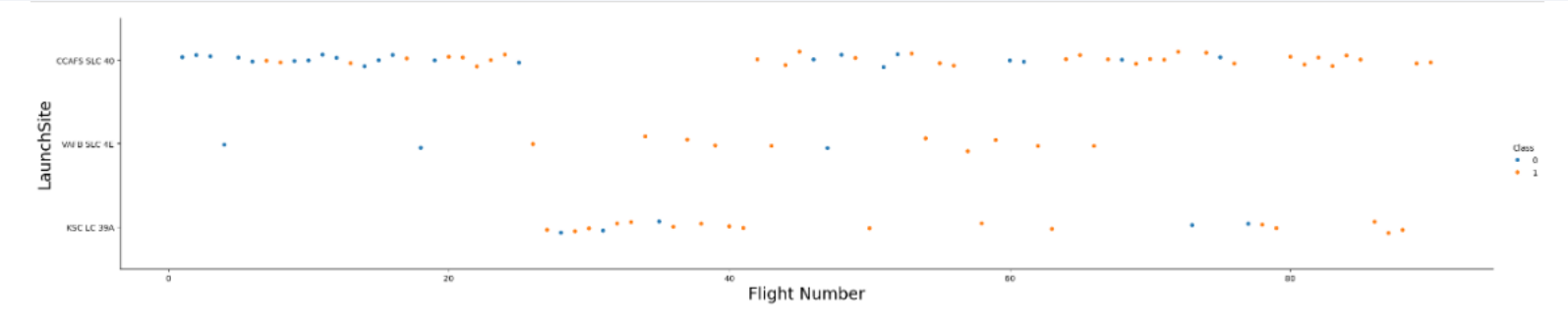
EDA with Data Visualization

- Performed exploratory Data Analysis and Feature Engineering using Pandas and Matplotlib
- Imported Libraries and Define Auxiliary Functions
- Exploratory Data Analysis
- Plotted FlightNumber vs. PayloadMassand overlay the outcome of the launch



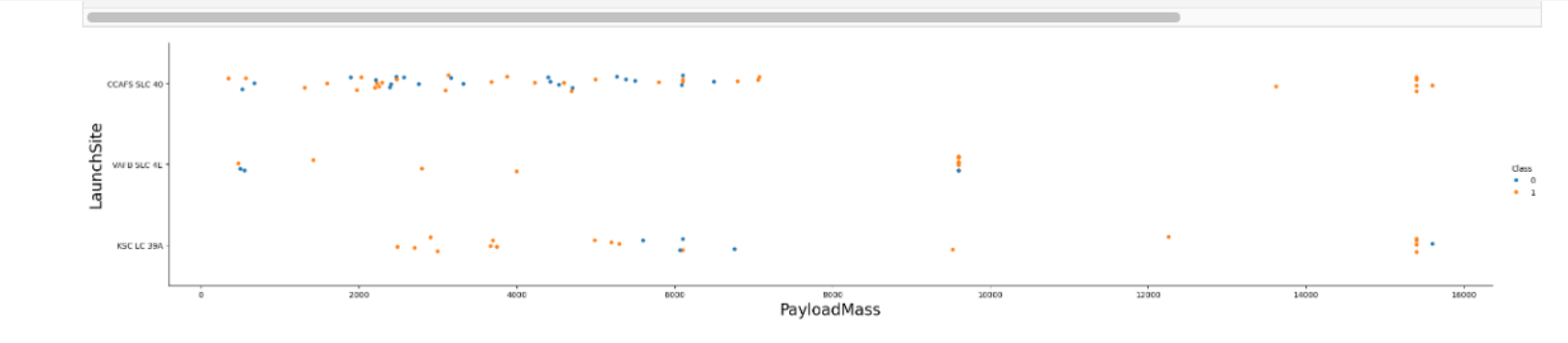
EDA with Data Visualization

- Visualized the relationship between Flight Number and Launch Site



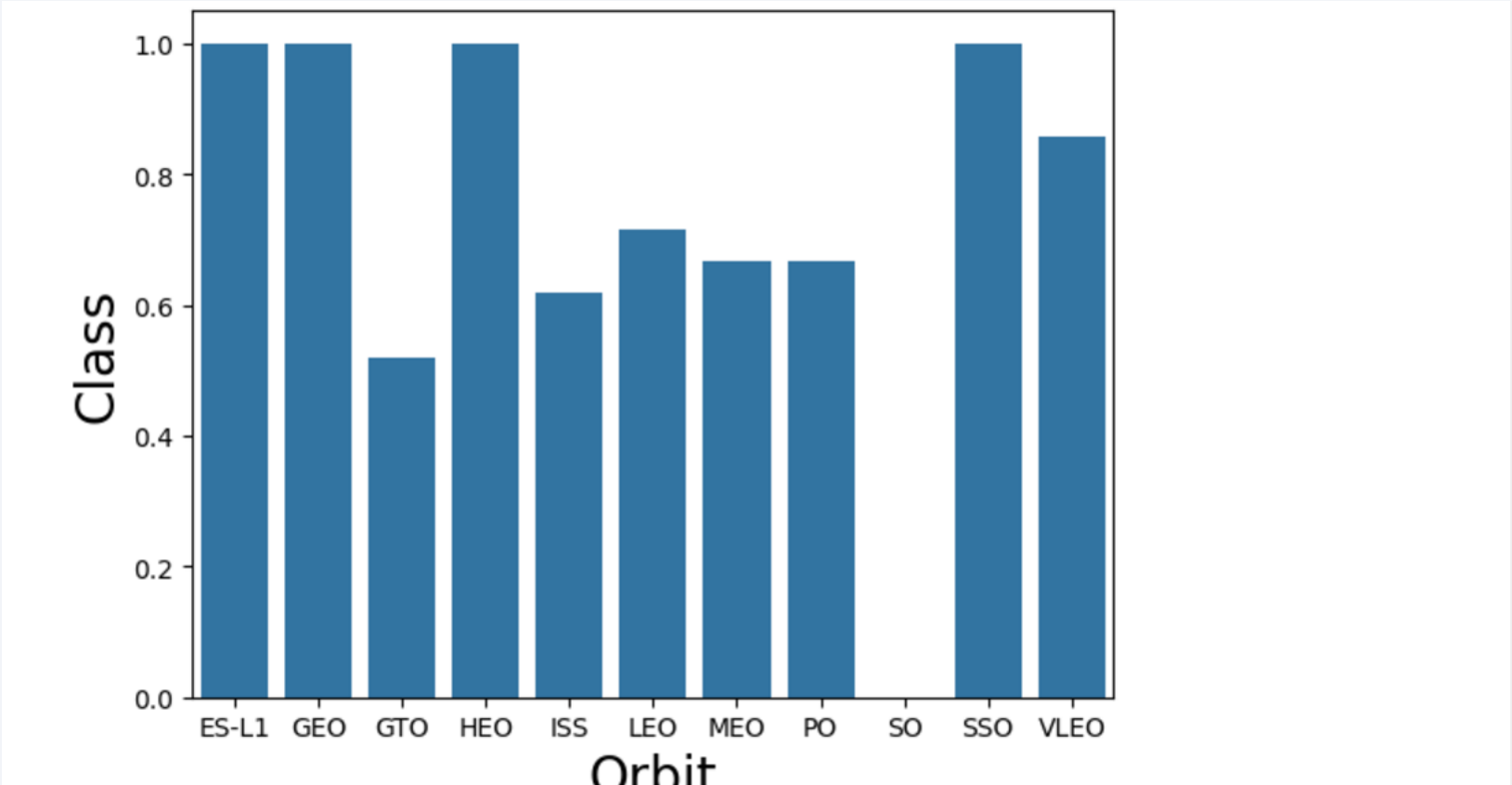
EDA with Data Visualization

- Visualized the relationship between Payload Mass and Launch Site



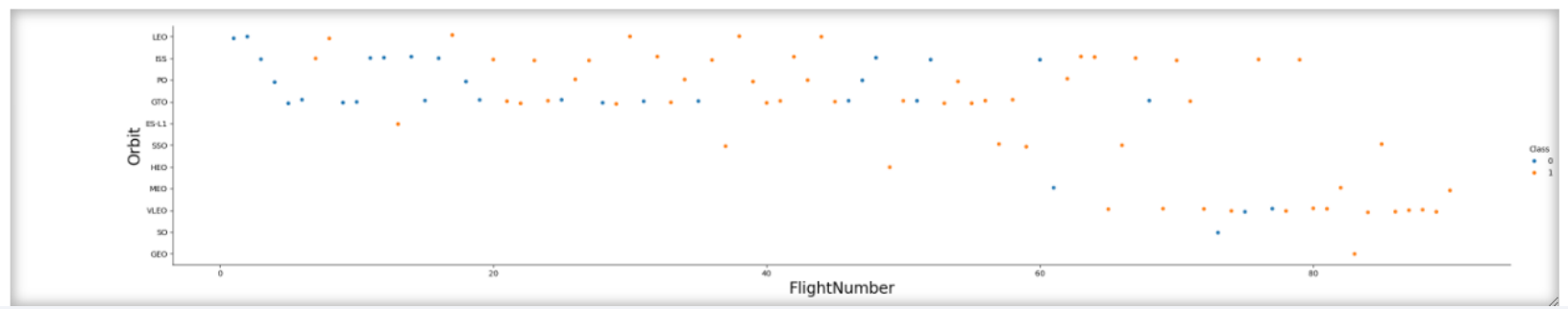
EDA with Data Visualization

- Visualized the relationship between success rate of each orbit type



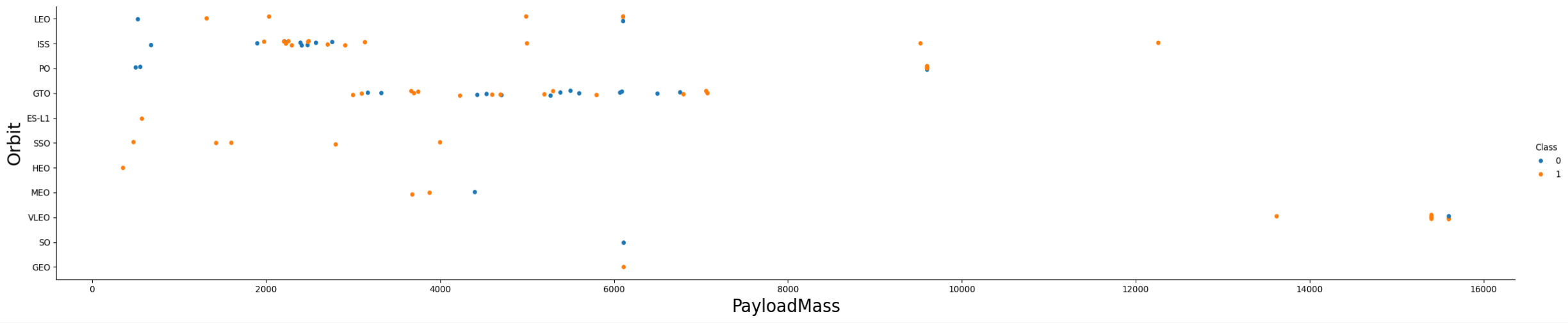
EDA with Data Visualization

- Visualized the relationship between FlightNumber and Orbit type



EDA with Data Visualization

Visualized the relationship between Payload Mass and Orbit type



EDA with Data Visualization

- GitHub URL: GitHub

EDA with SQL

- Downloaded the datasets
- Connected to the database
- Displayed the names of the unique launch sites in the space mission
- %sql select distinct(Launch_Site) from SPACEXTABLE
- Displayed 5 records where launch sites begin with the string 'CCA'
- %sql select * from SPACEXTABLE where "Launch_Site" like 'CCA%' limit 5
- Displayed the total payload mass carried by boosters launched by NASA (CRS)
- %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Customer" = 'NASA (CRS)'

EDA with SQL

- **Displayed average payload mass carried by booster version F9 v1.**

- %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Booster_Version" = 'F9 v1.1'

- **Listed the date when the first succesful landing outcome in ground pad was acheived**

- %sql select min(Date) from SPACEXTABLE where "Landing_Outcome" = 'Success (ground pad)'

- **Listed the names of the boosters which have success in drone ship s and have payload mass greater than 4000 but less than 6000**

- **%**sql select distinct(Booster_Version) from SPACEXTABLE where "Landing_Outcome" = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000

EDA with SQL

- **Listed the total number of successful and failure mission outcomes**
- %sql select "Mission_Outcome", count(*) from SPACEXTABLE group by "Mission_Outcome”
- **Listed the names of the booster_versions which have carried the maximum payload mass. Use a subquery**
- %sql select distinct(Booster_Version) from SPACEXTABLE where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
- **Listed the records which will display the month names, failure outcomes in drone ship , booster versions, and launch_site for the months in year 2015.**
- %sql select substr(Date,6,2) as month ,substr(Date,0,5) as year,Booster_Version,Launch_Site,Landing_Outcome from SPACEXTABLE where Landing_Outcome='Failure (drone ship)'

EDA with SQL

- **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.**

- %sql select Landing_Outcome, count(1) as count from SPACEXTABLE where date between "2010-06-04" and "2017-03-20" group by Landing_Outcome order by count desc

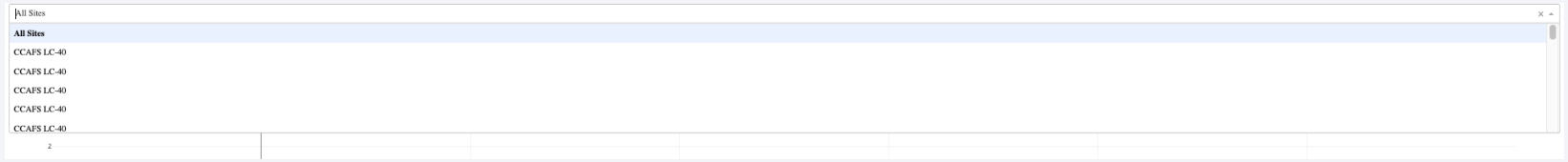
- Git Hub Ref: https://github.com/rajumca/spacex_analysis/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Marked all launch sites on a map
- Marked the success/failed launches for each site on the map using `folium.Marker`
- Calculated the distances between a launch site to its proximities
- Used **folium.Circle** to add a highlighted circle area with a text label on a specific coordinate.
- Added MousePosition on the map to get coordinate s for a mouse over a point on the map.
- Git Hub Repo:
- https://github.com/rajumca/spacex_analysis/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

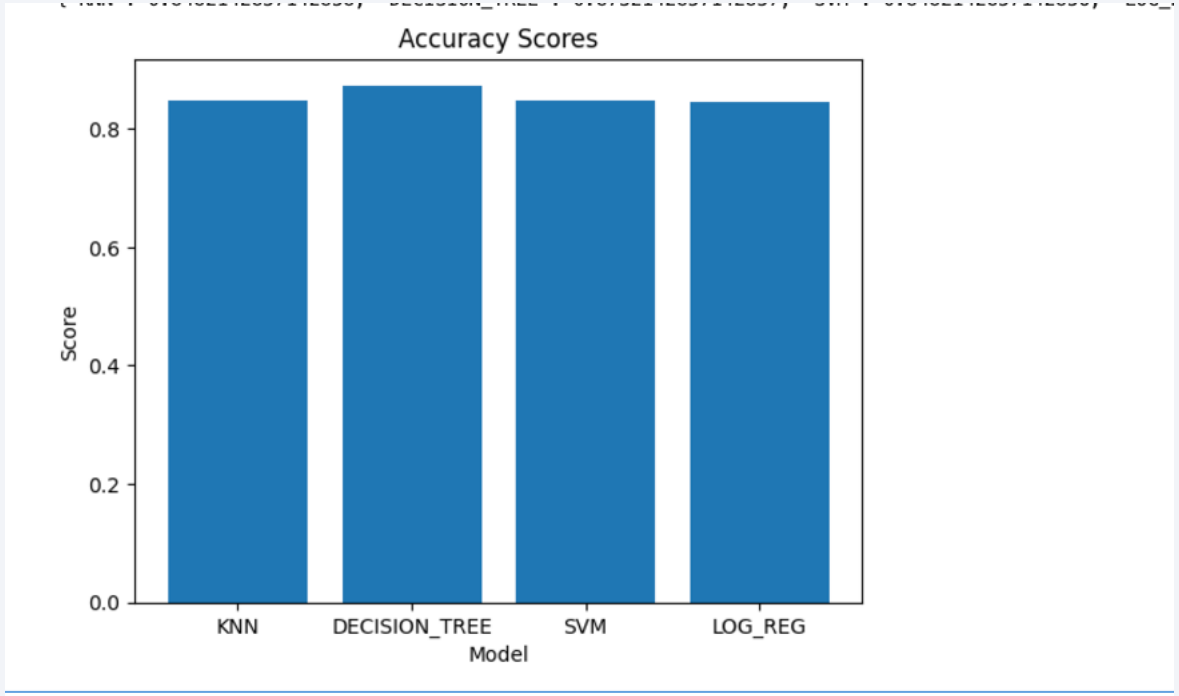
- Iterated Launch sites and created a dropdown



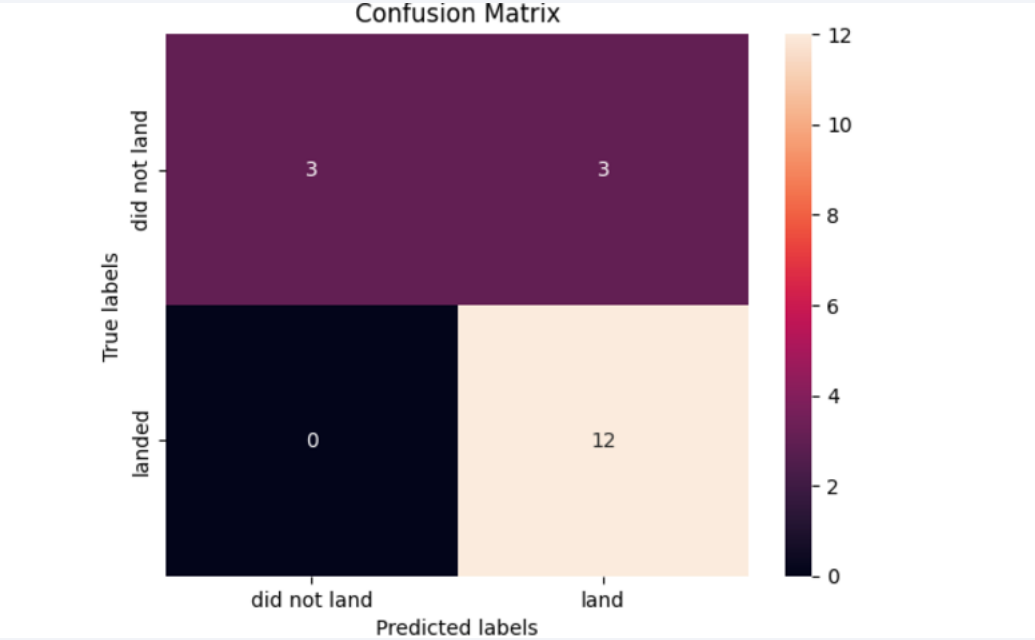
- Generated Pie chart for success rate for each launch site
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

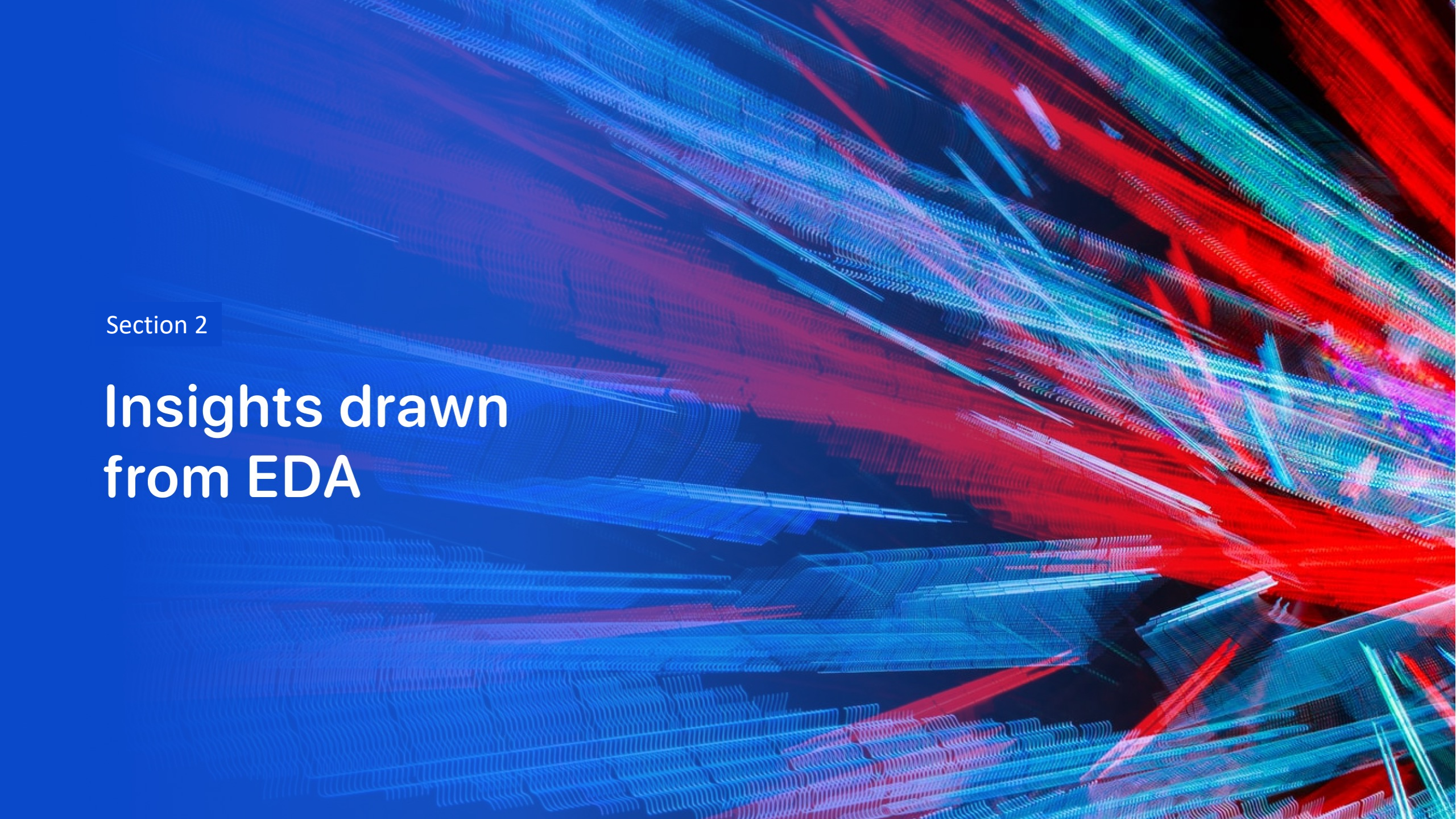
Predictive Analysis (Classification)

- Decision Tree classifier has the best accuracy compared to other models



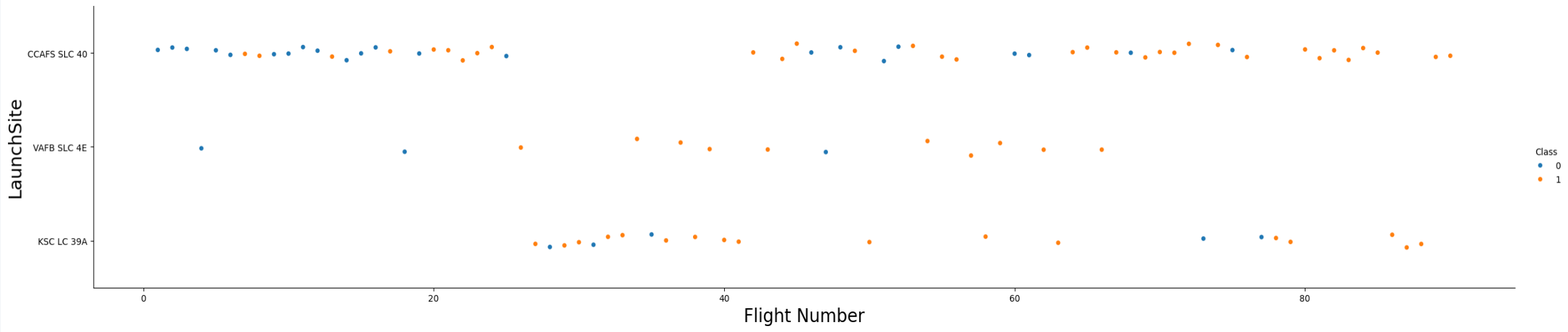
Results


Confusion Matrix
True labels
did not land
landed
3
3
0
12
did not land
land
Predicted labels
12
10
8
6
4
2
0


Section 2
Insights drawn from EDA

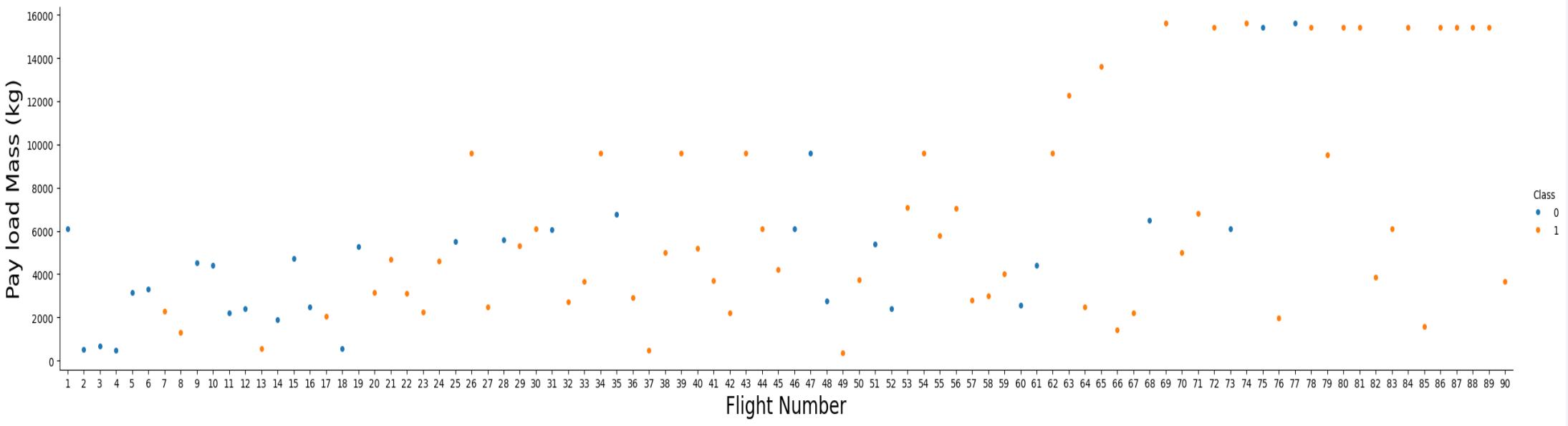
Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site



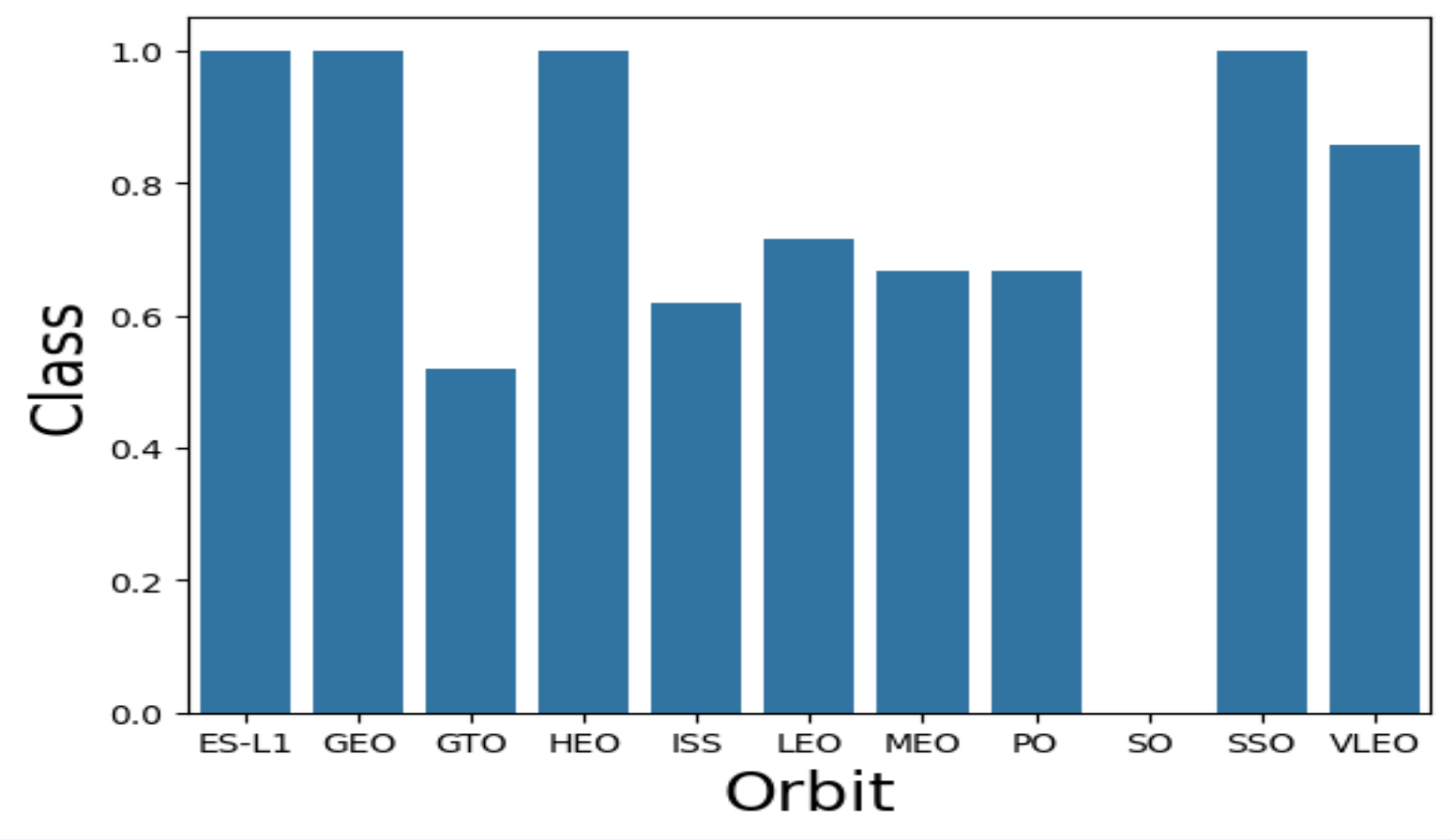
Payload vs. Launch Site

- Scatter plot of Payload vs. Launch Site



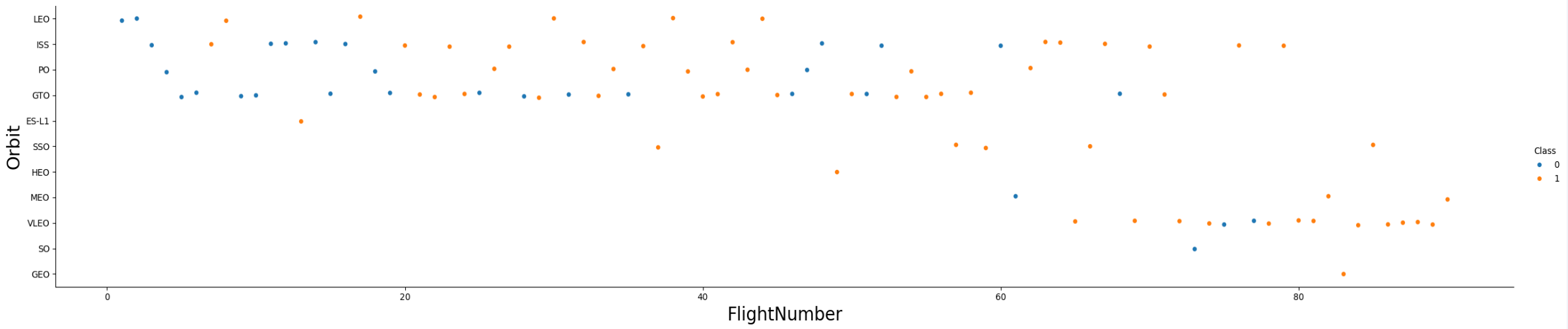
Success Rate vs. Orbit Type

- Bar chart for the success rate of each orbit type



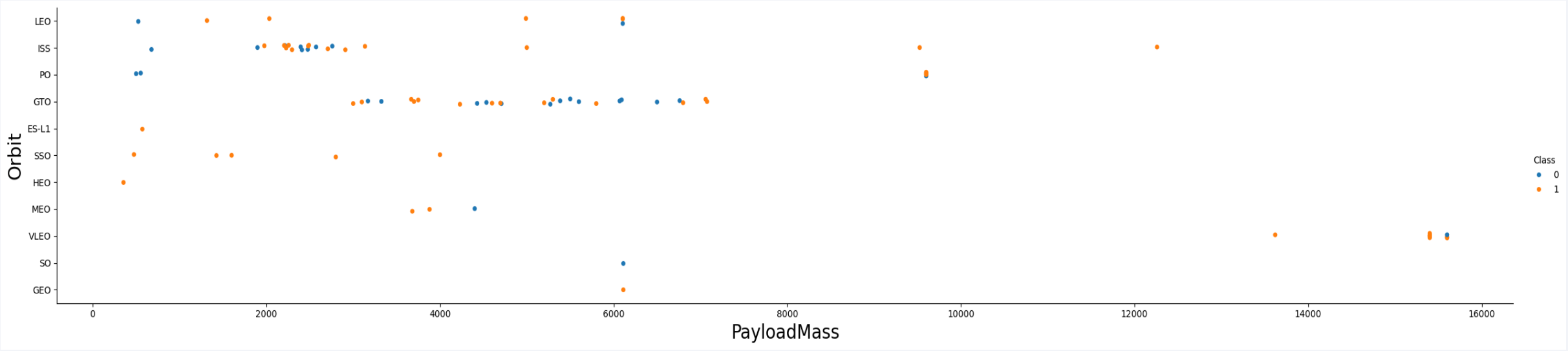
Flight Number vs. Orbit Type

- Scatter point of Flight number vs. Orbit type



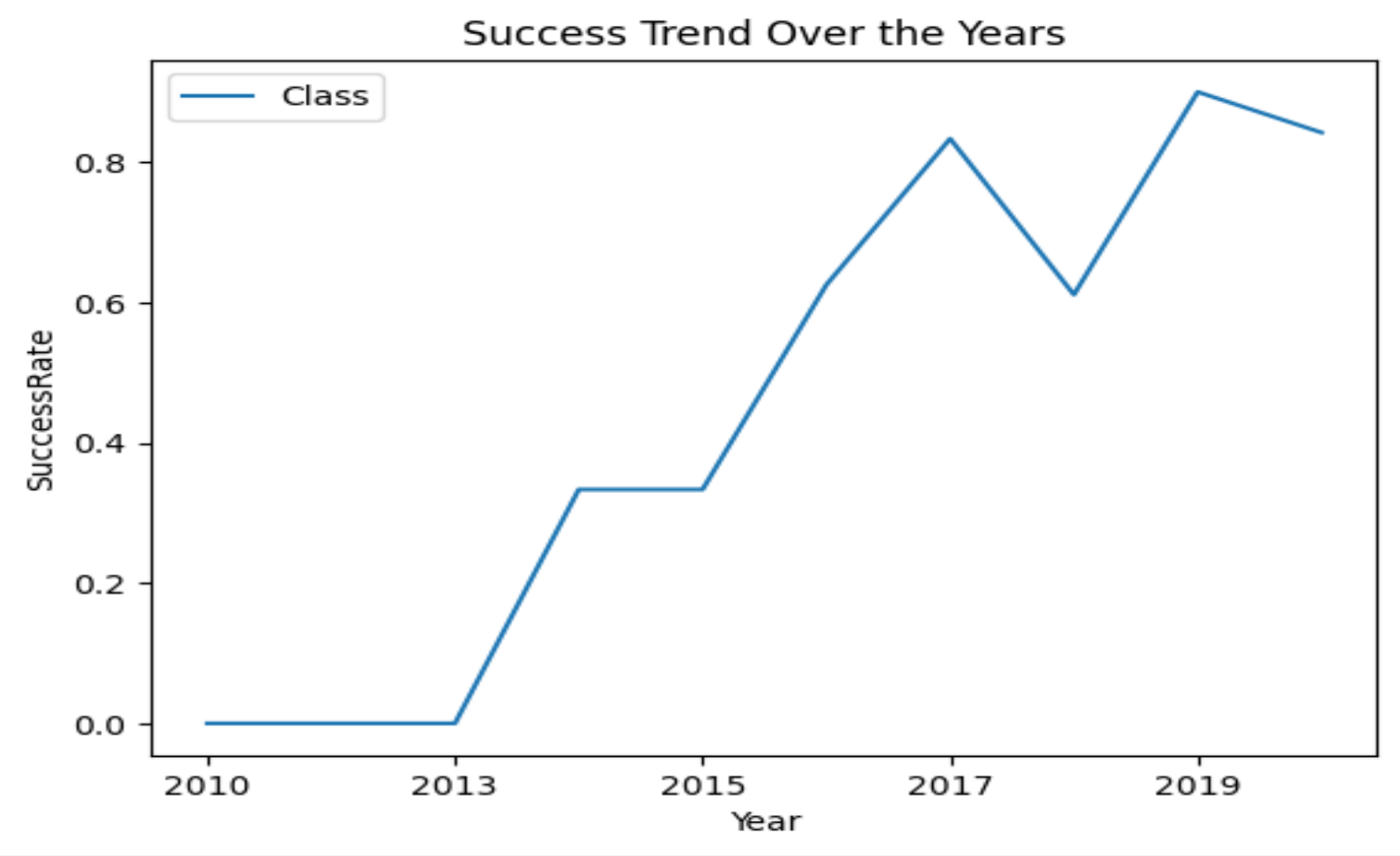
Payload vs. Orbit Type

- Scatter plot for payload vs. orbit type



Launch Success Yearly Trend

- Success rate kept increasing from 2023



All Launch Site Names

- There are four launch sites

```
42]: %sql select distinct(Launch_Site) from SPACEXTABLE
 * sqlite:///my_data1.db
Done.
```

42]:

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

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

[26]:
```
%sql select * from SPACEXTABLE where "Launch_Site" like 'CCA%' limit 5
```

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

[26]:

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

Total Payload Mass

- Total payload carried by boosters from NASA is 45596 KG

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

```
[27]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Customer" = 'NASA (CRS)'
 * sqlite:///my_data1.db
Done.
```

[27]:

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1 is 2928.4

Task 4

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

```
[28]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Booster_Version" = 'F9 v1.1'
 * sqlite:///my_data1.db
Done.
```

[28]:

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on the ground pad
- First successful landing outcome in ground pad was on 2015-12-22

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

Hint:Use min function

```
%sql select min(Date) from SPACEXTABLE where  "Landing_Outcome" = 'Success (ground pad)'
```

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

min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of boosters that have successfully landed on drone ships and had payload mass greater than 4000 but less than 6000

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

```
[31]: %sql select distinct(Booster_Version) from SPACEXTABLE where  "Landing_Outcome" = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000
 * sqlite:///my_data1.db
Done.
```

[31]:

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

[35]:
```
%sql select "Mission_Outcome", count(*) from SPACEXTABLE group by "Mission_Outcome"
```

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

[35]:

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

Boosters Carried Maximum Payload

- List 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 distinct(Booster_Version) from SPACEXTABLE where PAYLOAD_MASS__KG_=( select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

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

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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
13]: %sql select substr(Date,6,2) as month ,substr(Date,0,5) as year,Booster_Version,Launch_Site,Landing_Outcome from SPACEXTABLE where Landing_Outcome='Failure
```

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

13]:

month	year	Booster_Version	Launch_Site	Landing_Outcome
01	2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015	F9 v1.1 B1015	CCAFS LC-40	Failure (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

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.

```
[40]: %sql select Landing_Outcome, count(1) as count from SPACEXTABLE where date between "2010-06-04" and "2017-03-20" group by Landing_Outcome order by count desc
```

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

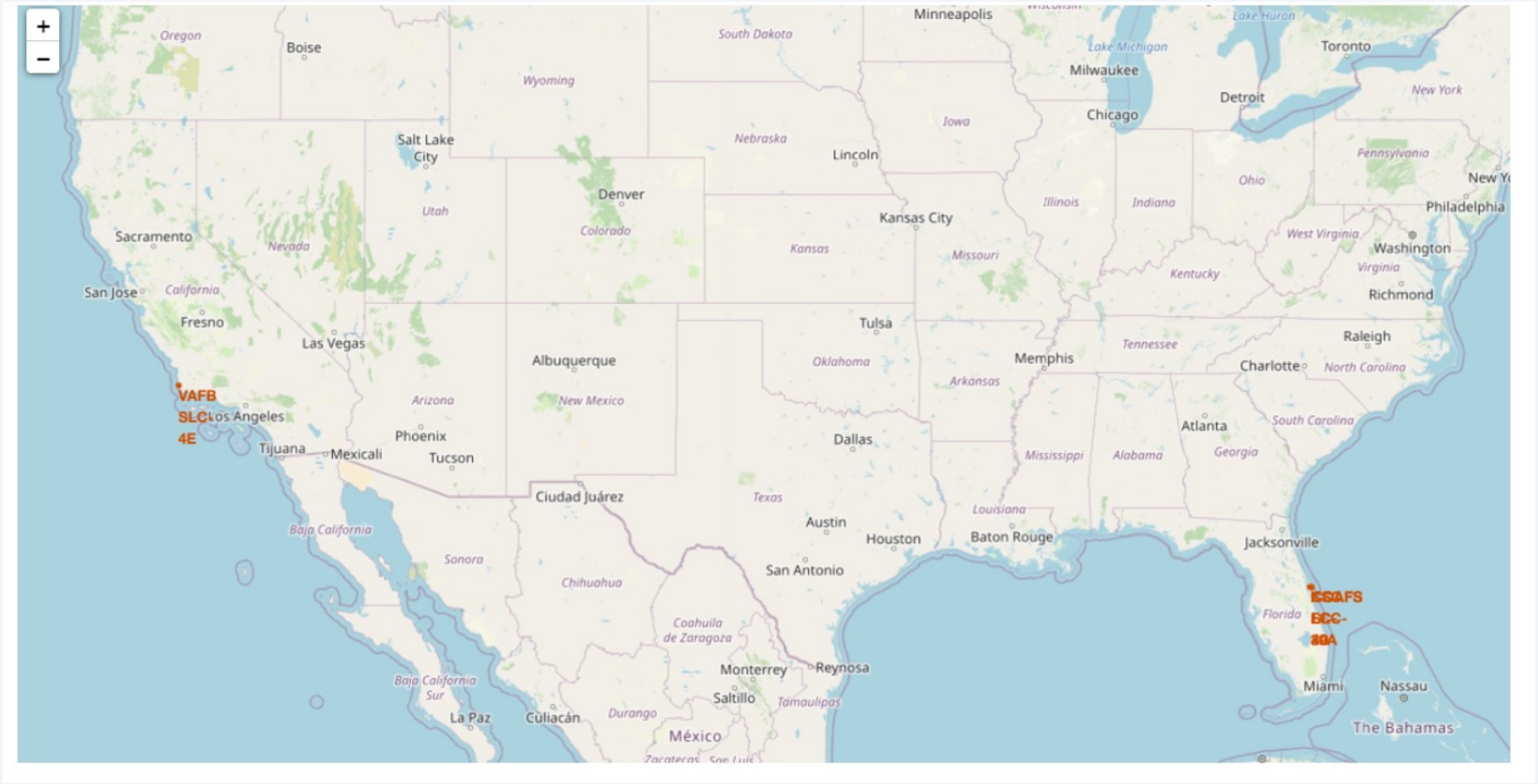
[40]:

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


Section 3
Launch Sites
Proximities Analysis

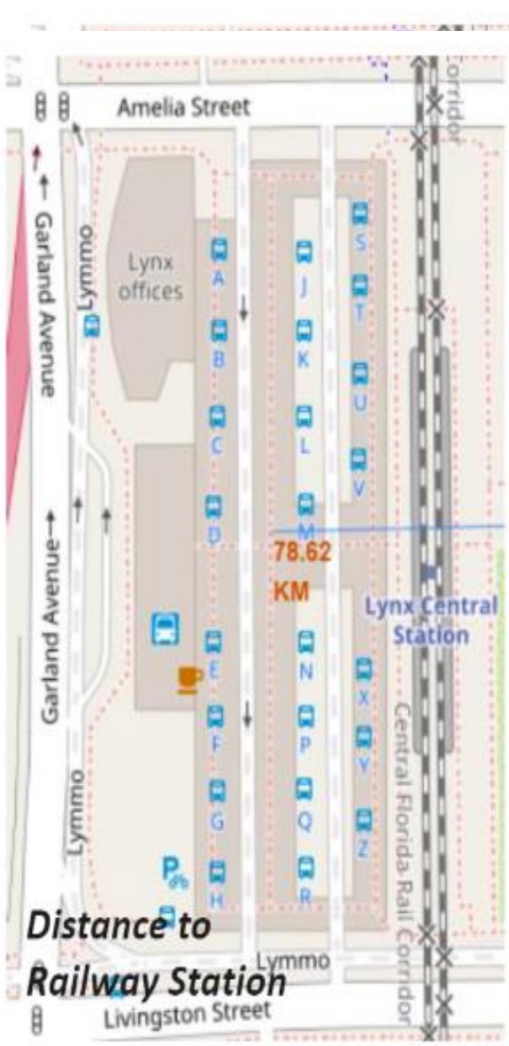
Launch Site Proximity analysis with Folium

- Launch sites locations

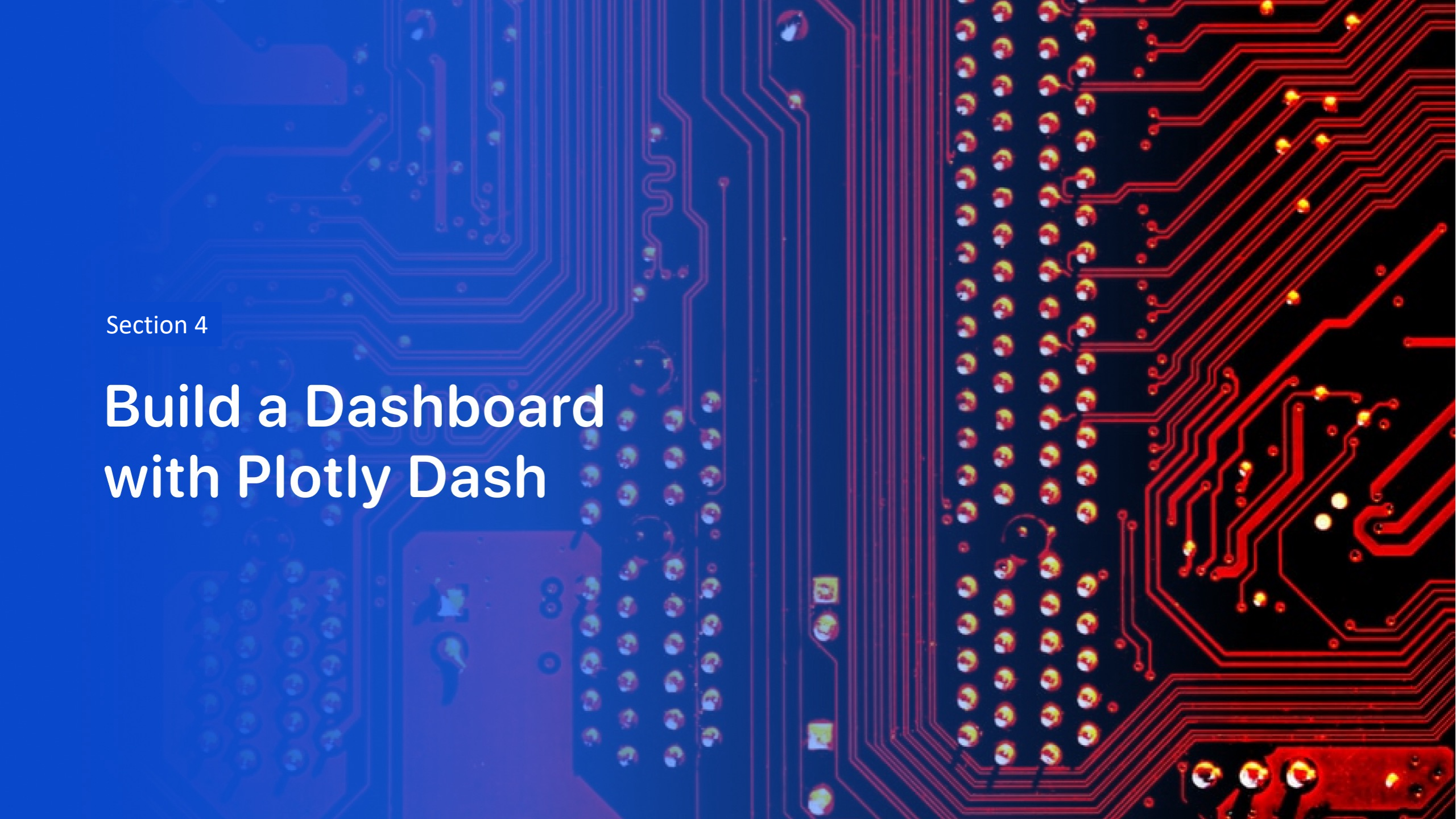


Launch outcome by launch site

Launch Site distance from Highways and Railway Lines

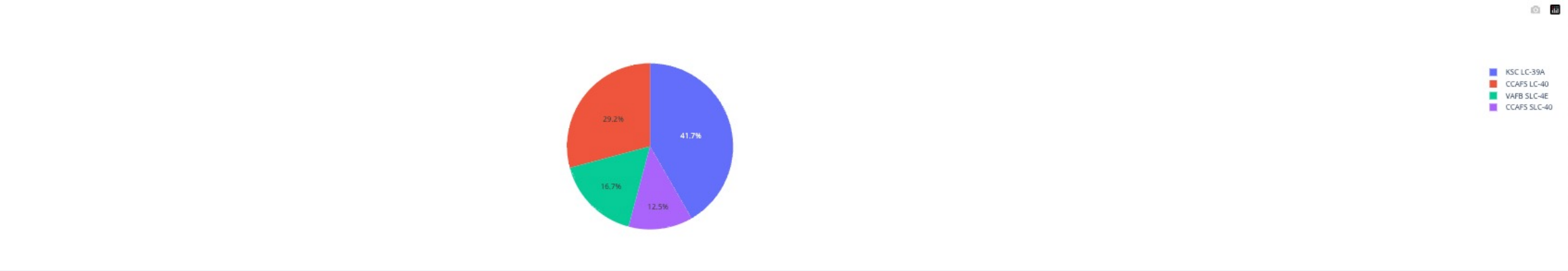


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

Launch success rate for each Launch site

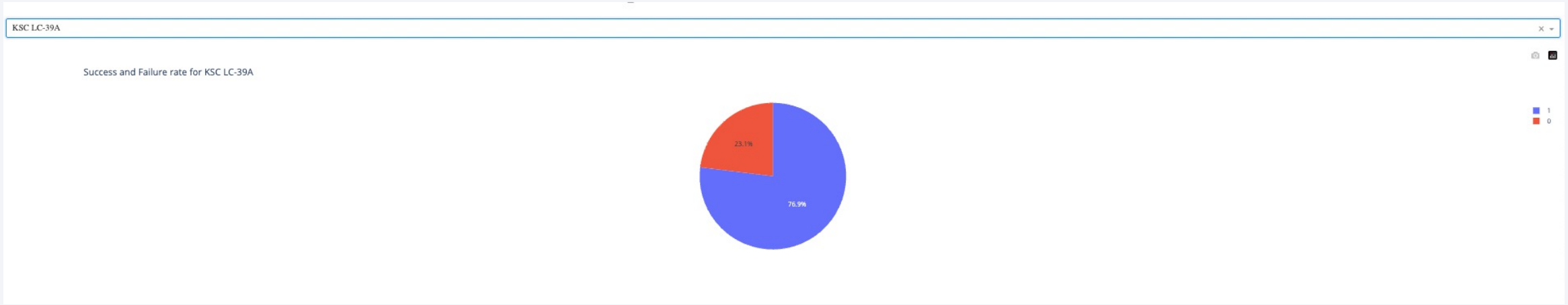
Launch success rate for each Launch site



Most successful launches are from KSC LC-39A

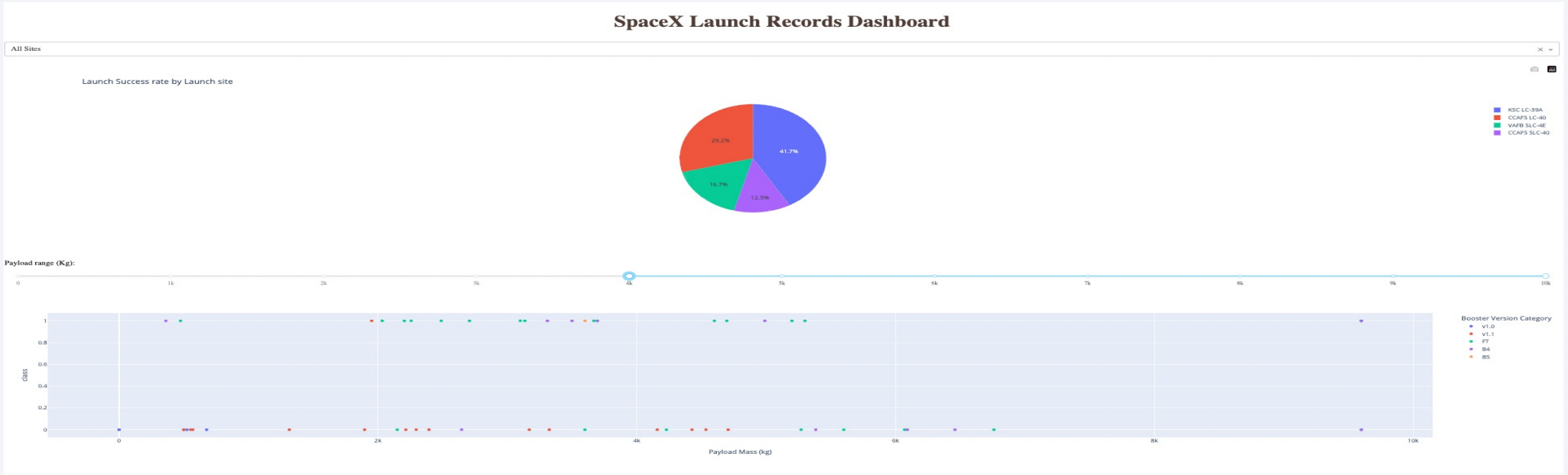
Launch site with most success ratio

- KSC LC-39A has most success rate of 79.6%



Success Rate and impact of Payload mass

KSC LC-39A has most success rate of 79.6%
Most successful launches are from KSC LC-39A
Payload range 2k to 4k has the most successful launches

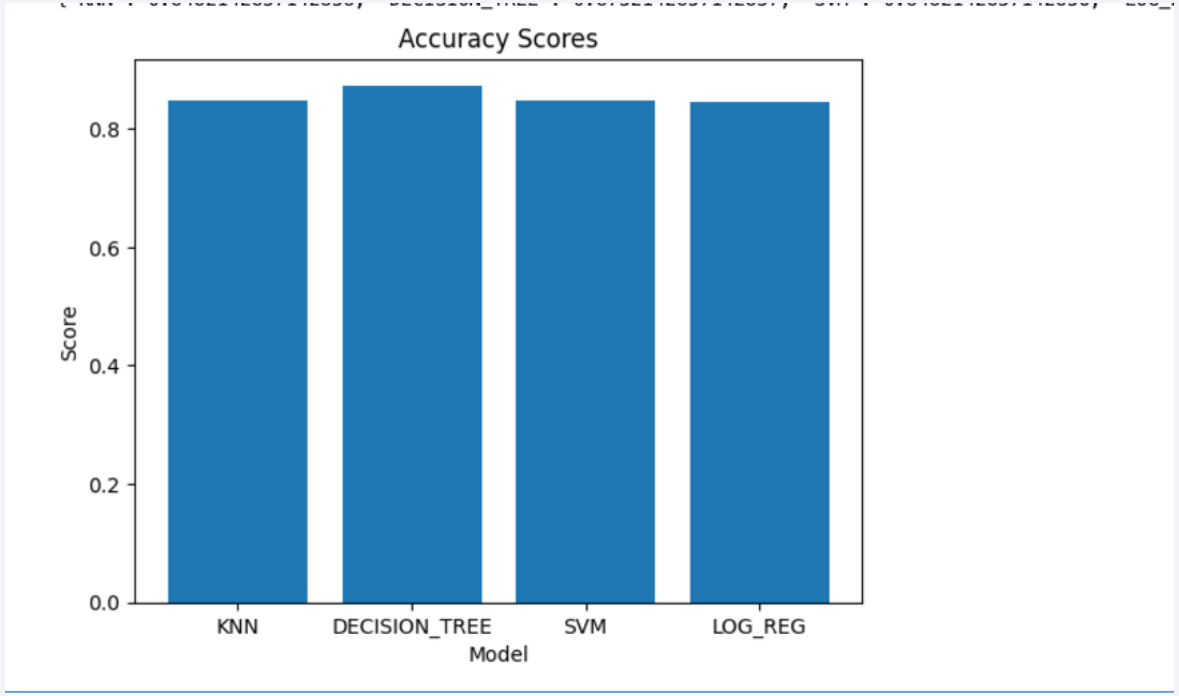


Section 5

Predictive Analysis (Classification)

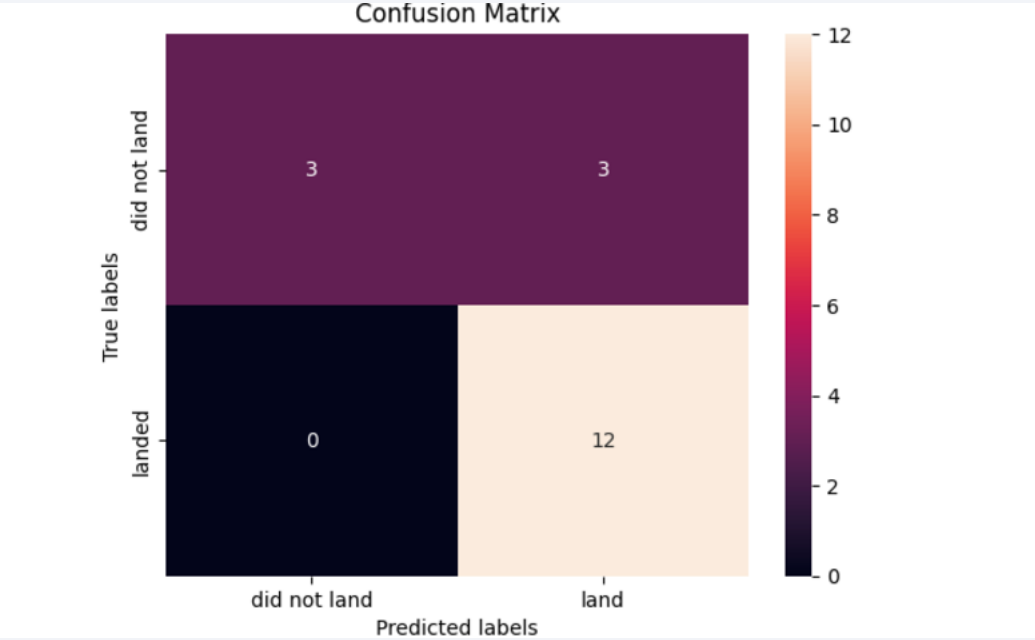
Classification Accuracy

- Decision Trees have the highest accuracy of 0.87



Confusion Matrix

- Decision Tree Classification has the best confusion matrix



Conclusions

- All the models performed equally better, but the Decision Tree Classifier has the best accuracy
- Confusion matrix is same for all the models

Thank you!