IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Yogyawan
2024-03-25 (YYYY-MM-DD)

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Collecting, analyzing, visualizing, querying, make folium map & interactive dashboard, and machine learning of SpaceX data.

Section 1

Methodology

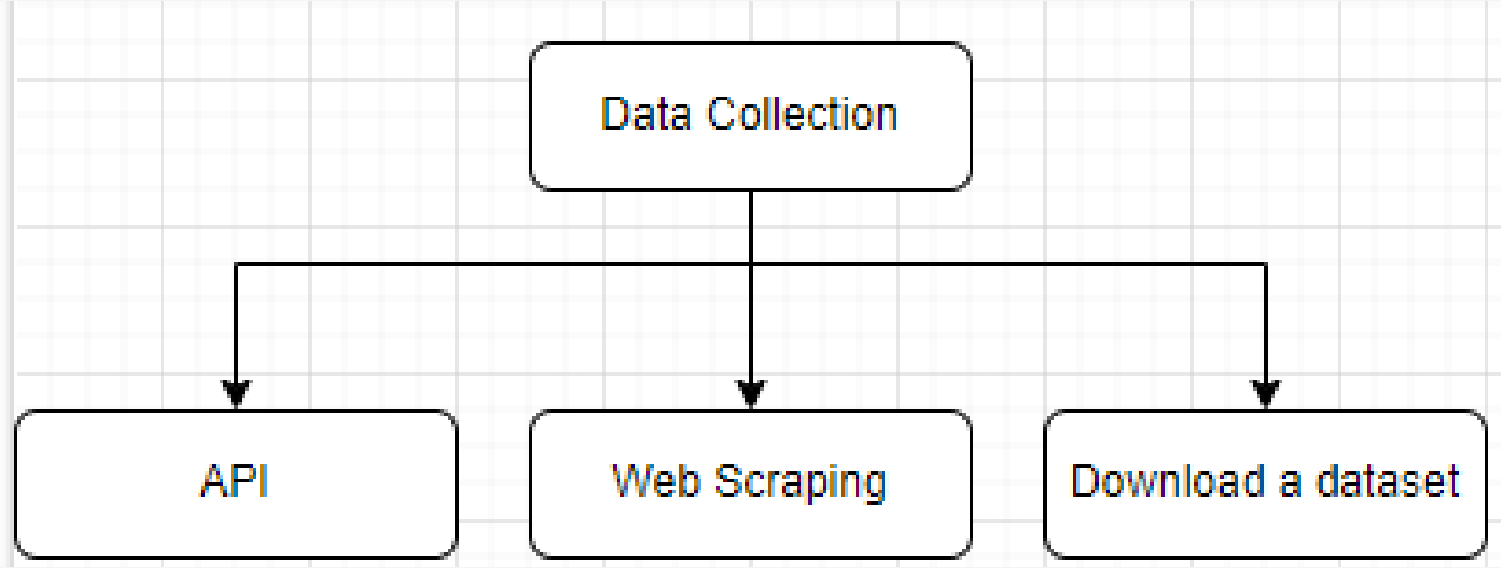
Methodology

Executive Summary

- Data collection methodology:
  - The data collected through API, web scraping, and download a dataset.
- Perform data wrangling
  - Dealing with missing values 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 models
  - Perform Logistics Regression Method, Support Vector Method, Decision Tree Method, and K-Nearest Neighbors Method.

Data Collection

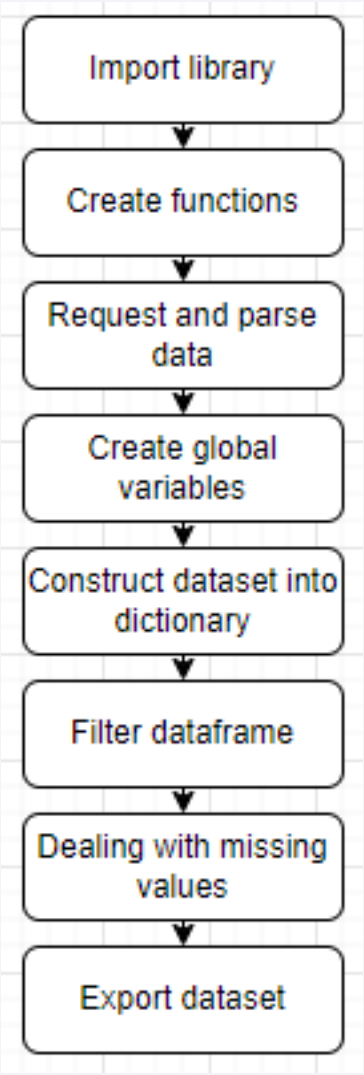
- The data collected through JSON results, web scraping, and download a dataset.



Data Collection – SpaceX API

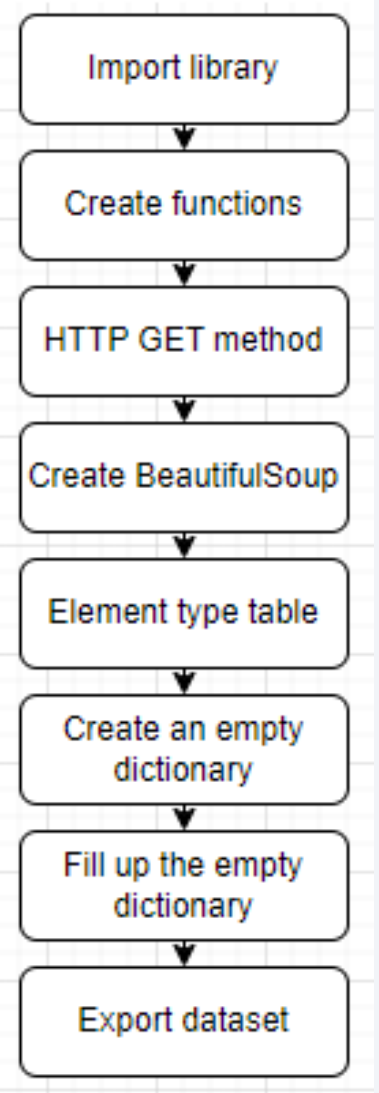
- **Import Libraries and Define Auxiliary Functions**
- **Request and parse the SpaceX launch data using the GET request**
- **Filter the dataframe to only include Falcon9 launches**
- **Data wrangling**
- **Dealing with missing values**

- https://github.com/Twentsix/ForCourseraForLearning/blob/cc699aed01d898c19206a90fbbc46b11743d4499/FinalProject/10-1-2x%20jupyter-labs-spacex-data-collection-api_Executed.ipynb



Data Collection - Scraping

- **Request the Falcon9 Launch Wiki page from its URL**
- **Extract all column/variable names from the HTML table header**
- **Create a data frame by parsing the launch HTML tables**
- https://github.com/Twentsix/ForCourseraForLearning/blob/cc699aed01d898c19206a90fbbc46b11743d4499/FinalProject/10-1-4x%20jupyter-labs-webscraping_Executed.ipynb



Data Wrangling

- **Import libraries, dealing missing values, calculate, and export**
- **Calculate the number of launches on each site**
- **Calculate the number and occurrence of each orbit**
- **Calculate the number and occurence of mission outcome of the orbits**
- **Create a landing outcome label from Outcome column**
- https://github.com/Twentsix/ForCourseraForLearning/blob/cc699aed01d898c19206a90fbbc46b11743d4499/FinalProject/10-1-6x%20labs-jupyter-spacex-Data%20wrangling_Executed.ipynb

EDA with Data Visualization

- Using catplot from seaborn library to see a correlation between variables across dataset, and also using bar chart to see a relationship between variables, and using line chart to see a trends.
- https://github.com/Twentsix/ForCourseraForLearning/blob/cc699aed01d898c19206a90fbbc46b11743d4499/FinalProject/10-2-4x%20jupyter-labs-eda-dataviz.ipynb.jupyterlite_Executed.ipynb

EDA with SQL

- To display unique launch sites in the space mission
- To display the total payload mass, average payload mass, list of date, name of boosters, total number of mission outcomes, and count of landing outcomes carried by boosters launched by NASA
- https://github.com/Twentsix/ForCourseraForLearning/blob/cc699aed01d898c19206a90fbbc46b11743d4499/FinalProject/10-2-2x%20jupyter-labs-eda-sql-coursera_sqllite_Executed.ipynb

Build an Interactive Map with Folium

- Adding marker and label to each points of latitudes and longitudes of launch sites of SpaceX and also distances on given site map
- To add clarity and also knowledge of where the information of launch sites.
- https://github.com/Twentsix/ForCourseraForLearning/blob/cc699aed01d898c19206a90fbbc46b11743d4499/FinalProject/10-3-2x%20lab_jupyter_launch_site_location.jupyterlite_Executed.ipynb

Build a Dashboard with Plotly Dash

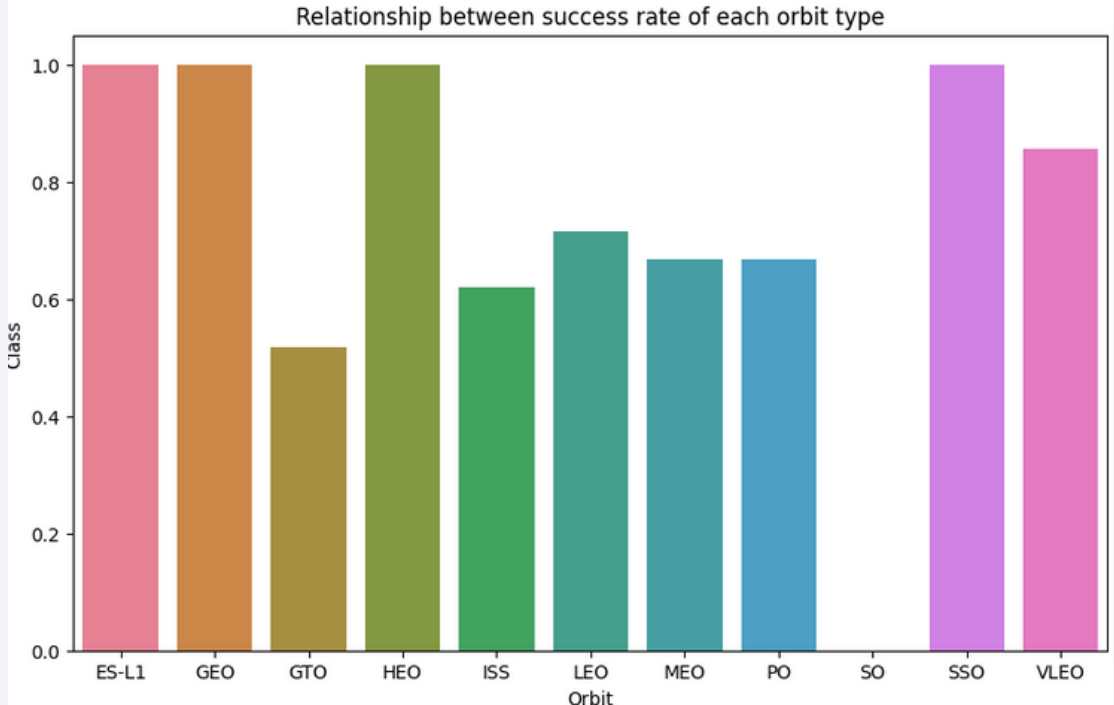
- Adding pie chart and scatter plot to see a percentage and relationships between payload mass (kg) and each classes.
- To see each different classes to its own group.
- https://github.com/Twentsix/ForCourseraForLearning/blob/cc699aed01d898c19206a90fbbc46b11743d4499/FinalProject/spacex_dash_app.py

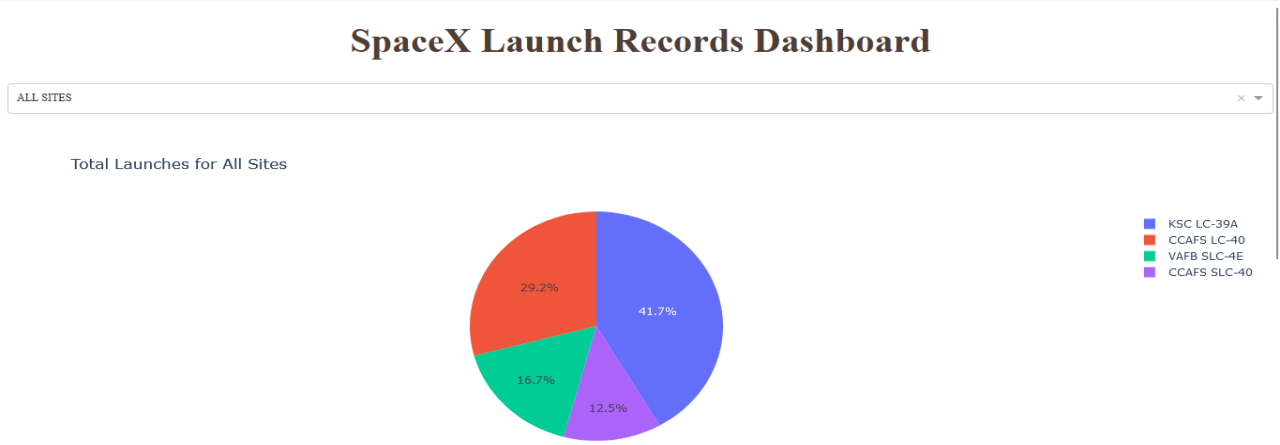
Predictive Analysis (Classification)

- Logistics Regression Method, Support Vector Method, Decision Tree Method, K-Nearest Neighbors Method
- To_numpy(), X_train, X_test, Y_train, Y_test, score(), predict(), fit(), DecisionTreeClassifier()
- https://github.com/Twentsix/ForCourseraForLearning/blob/cc699aed01d898c19206a90fbbc46b11743d4499/FinalProject/10-4-2x%20SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite_Executed.ipynb

Results

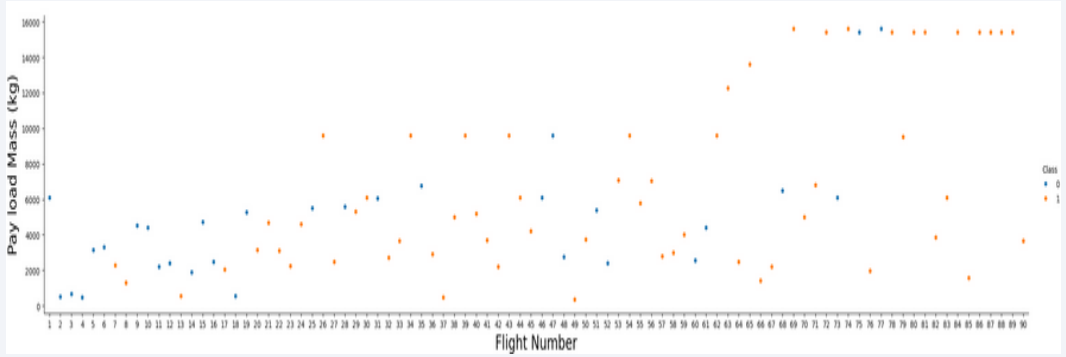
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN

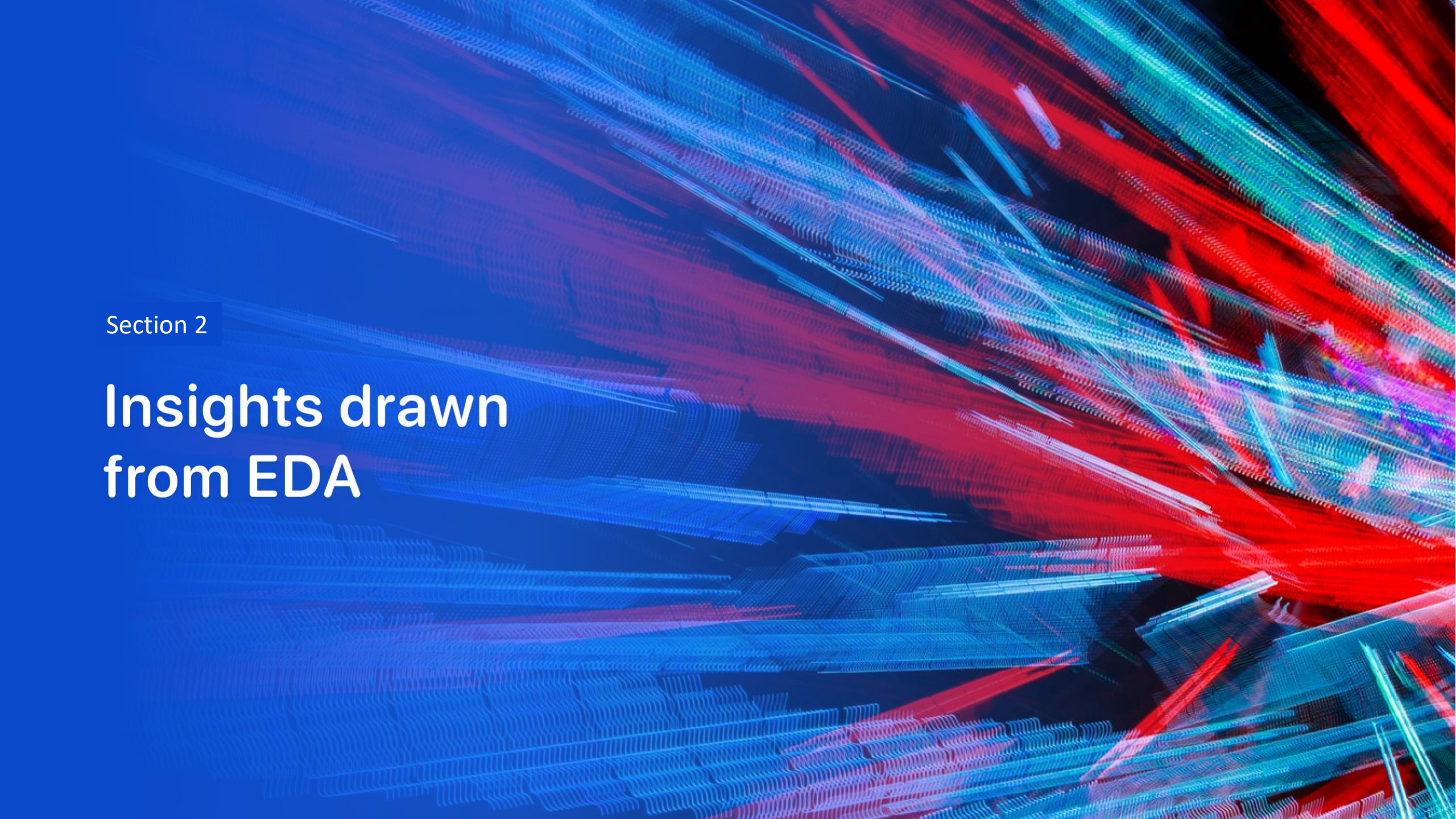





```
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print('Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearsdt neighbors method:', knn_cv.score(X_test, Y_test))
```

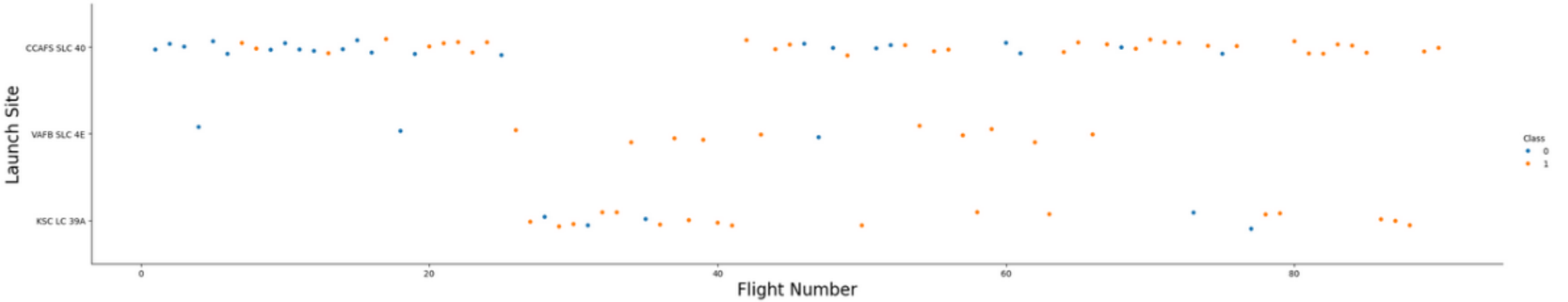
```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.7777777777777778
Accuracy for K nearsdt neighbors method: 0.8333333333333334
```




Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

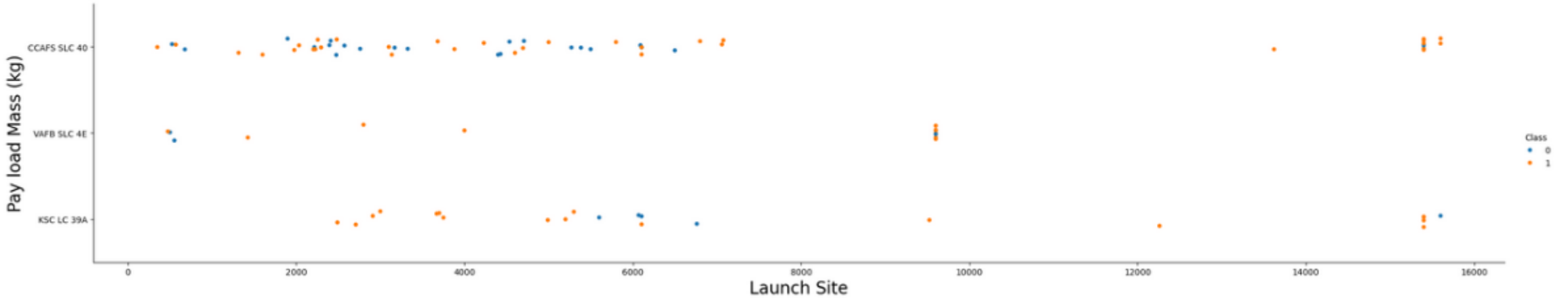
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Relationship between Flight Number vs. Launch Site.

Payload vs. Launch Site

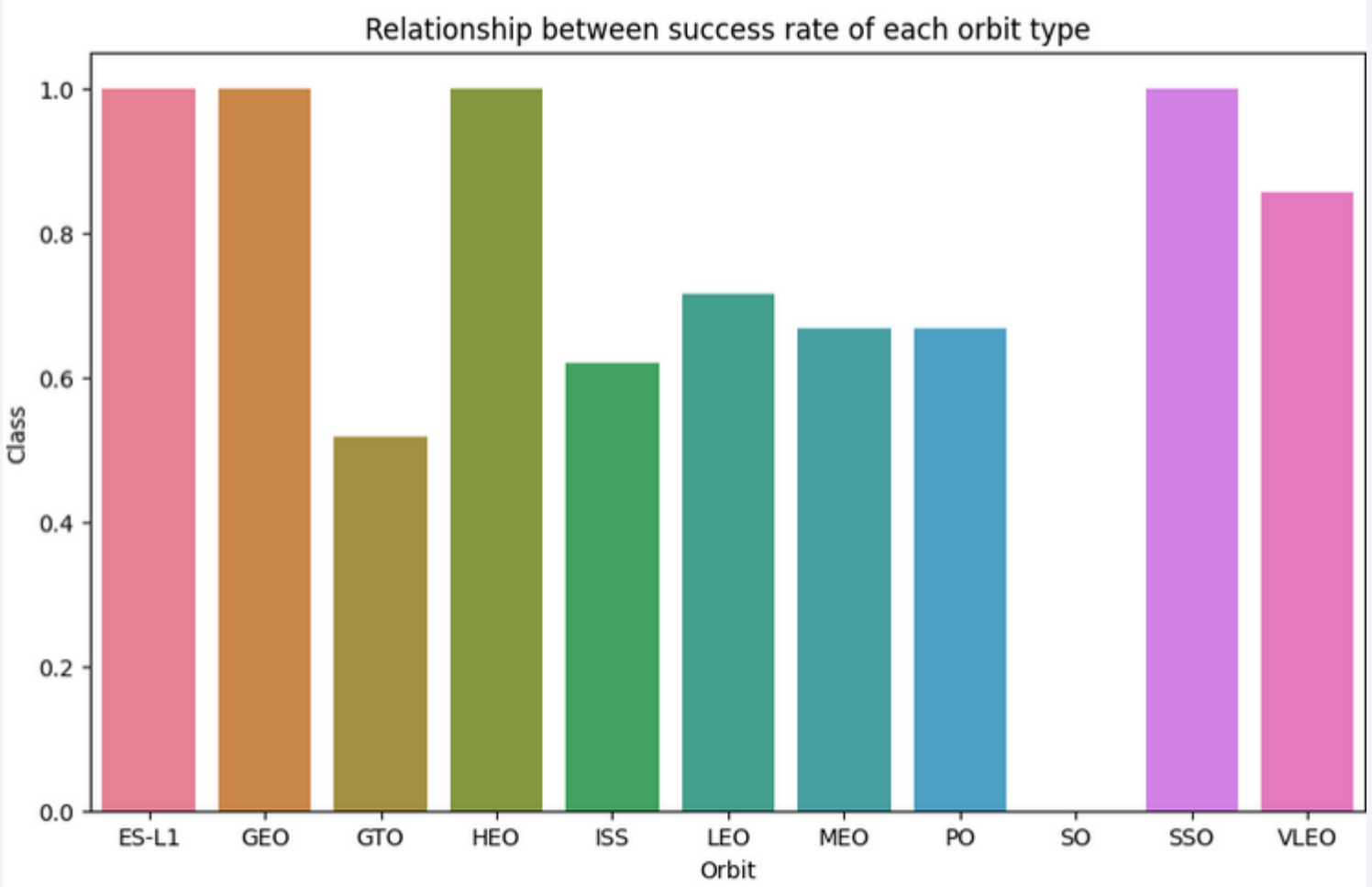
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Launch Site",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```



Relationship between Payload vs. Launch Site.

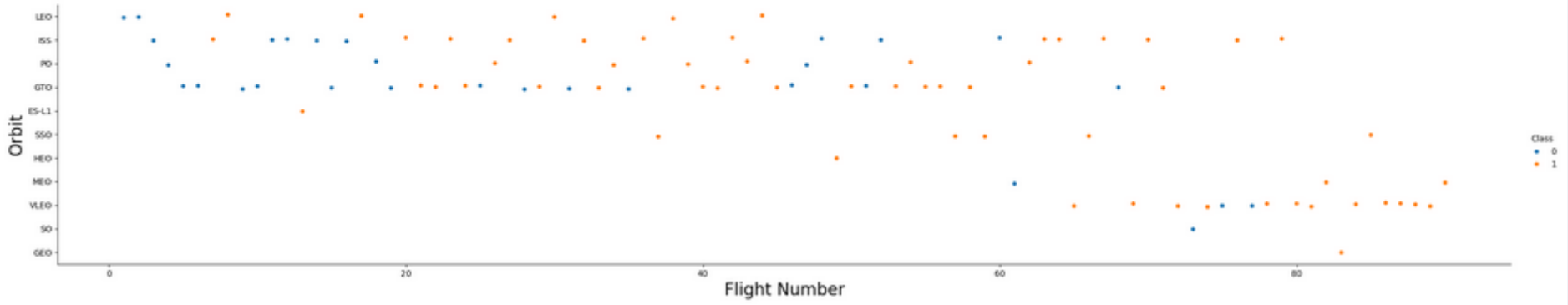
Success Rate vs. Orbit Type

ES-L1, GEO, HEO, and SSO have a higher success.



Flight Number vs. Orbit Type

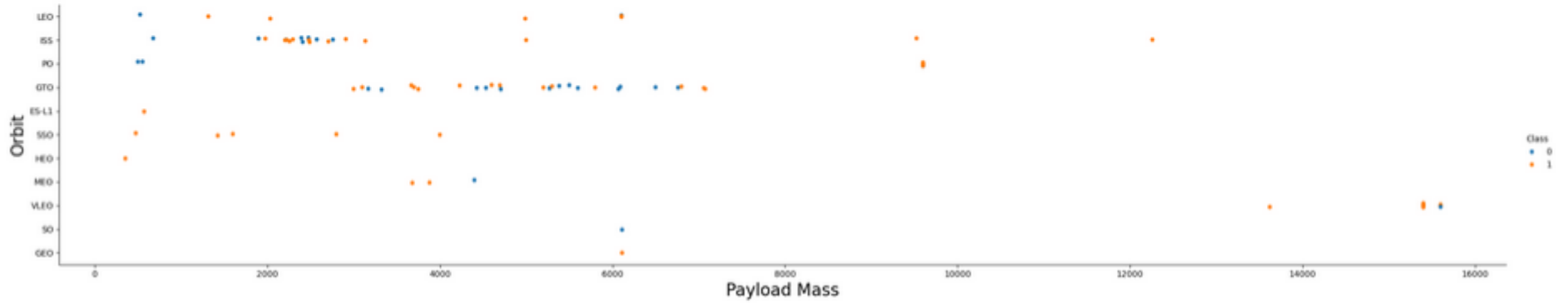
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Relationship between Flight Number vs. Orbit Type.

Payload vs. Orbit Type

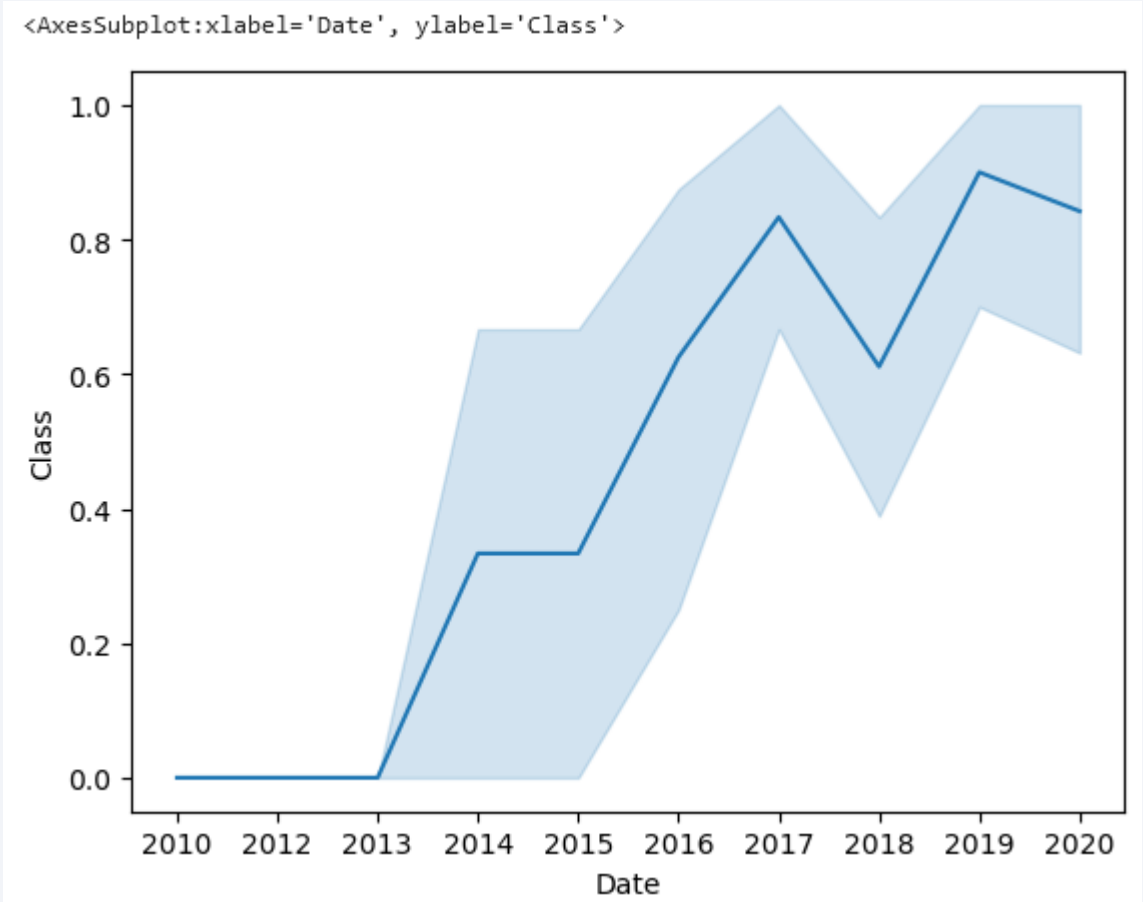
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload Mass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Relationship between Payload vs. Orbit Type.

Launch Success Yearly Trend

The success rate since 2013 kept increasing till 2020.



All Launch Site Names

There are 4 unique name of each launch sites which is:

CCAFS LC-40

VAFB SLC-4E

KSC LC-39A

CCAFS SLC-40

Task 1 ¶

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

```
%sql select distinct(Launch_Site) from SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

Task 2

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

```
%sql select * from SPACEXTABLE 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
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

Display top 5 rows of launch sites with the name begins with ‘CCA’

Total Payload Mass

Total payload mass (Kg) launched by NASA (CRS) with 48213 Kg

Task 3

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

```
#Yang hanya benar-benar NASA (CRS)
#%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer = 'NASA (CRS)'

#Include NASA (CRS)
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer like 'NASA (CRS)%'
```

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

sum(PAYLOAD_MASS__KG_)
48213

Average Payload Mass by F9 v1.1

Task 4

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

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

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

avg(PAYLOAD_MASS__KG_)
2928.4

Average payload mass (Kg) carried by booster version F9 v1.1 with 2928.4 Kg

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

```
#%sql select  from SPACEXTABLE where Mission_Outcome = 'Success' order by Date limit 5
%sql select min(Date) from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)'
```

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

min(Date)
2015-12-22

It was on Dec, 22th 2015 the first successful landing outcome in ground pad achieved.

Successful Drone Ship Landing with Payload between 4000 and 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 from SPACEXTABLE where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000
```

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

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

There are 4 booster version which have success in drone ship and have payload mass greater than 4000 Kg but less than 6000 Kg, which are: F9 FT B1022, F9 FT B1026, F9 FT B1021.2, F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql select distinct(Mission_Outcome) "Mission Outcome", count(*) "Count" from SPACEXTABLE group by Mission_Outcome
```

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

Mission Outcome	Count
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

There are 101 success and 1 failure from the data.

Boosters Carried Maximum Payload

There are 12 booster version which have carried the maximum payload mass (Kg)

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

```
%sql select 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

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

```
%sql select substr(Date,6,2) "Month", \
Landing_Outcome "Landing Outcome", \
Booster_Version "Booster Version", \
Launch_Site "Launch Site" \
from SPACEXTABLE \
where substr(Date,0,5) = '2015' \
and Landing_Outcome = 'Failure (drone ship)'
```

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

Done.
```

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

There are 2 failure landing outcomes in drone ship in the year 2015.

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2015-01-10	9:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)
2016-01-17	18:42:00	F9 v1.1 B1017	VAFB SLC-4E	Jason-3	553	LEO	NASA (LSP) NOAA CNES	Success	Failure (drone ship)
2016-03-04	23:35:00	F9 FT B1020	CCAFS LC-40	SES-9	5271	GTO	SES	Success	Failure (drone ship)
2016-06-15	14:29:00	F9 FT B1024	CCAFS LC-40	ABS-2A Eutelsat 117 West B	3600	GTO	ABS Eutelsat	Success	Failure (drone ship)
2016-07-18	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)

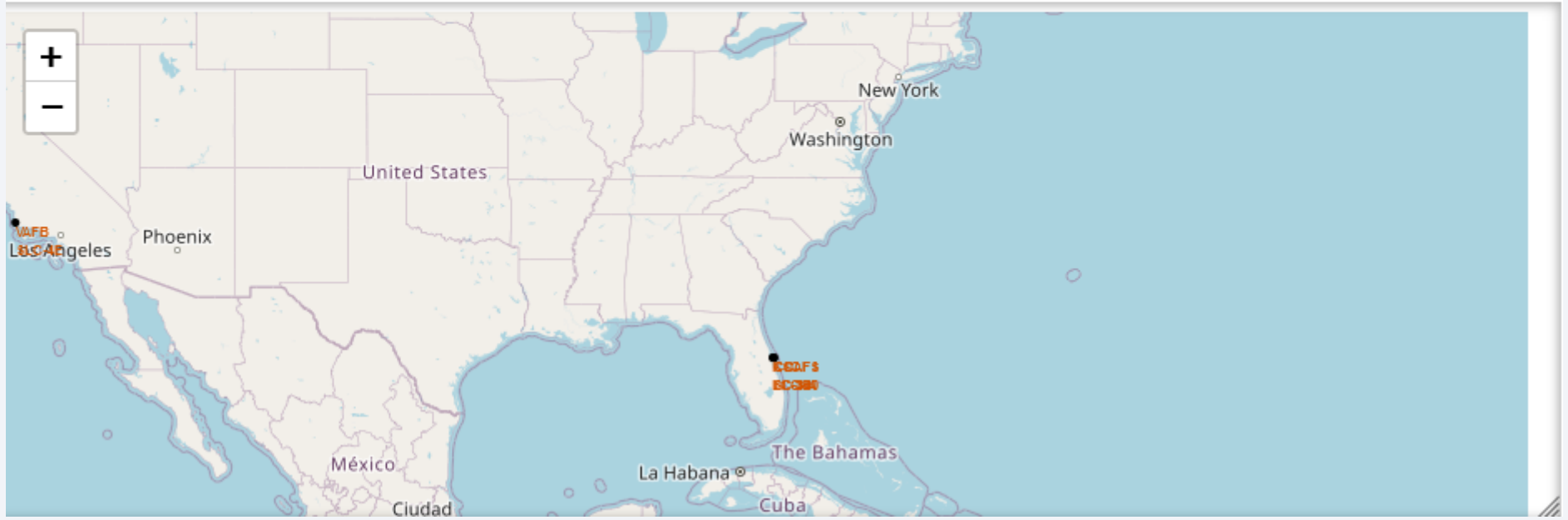
There are 8 records in total, which is landing outcomes such as success and failure between the date June, 4th 2010 until March, 20th 2017.

Section 3

Launch Sites Proximities Analysis

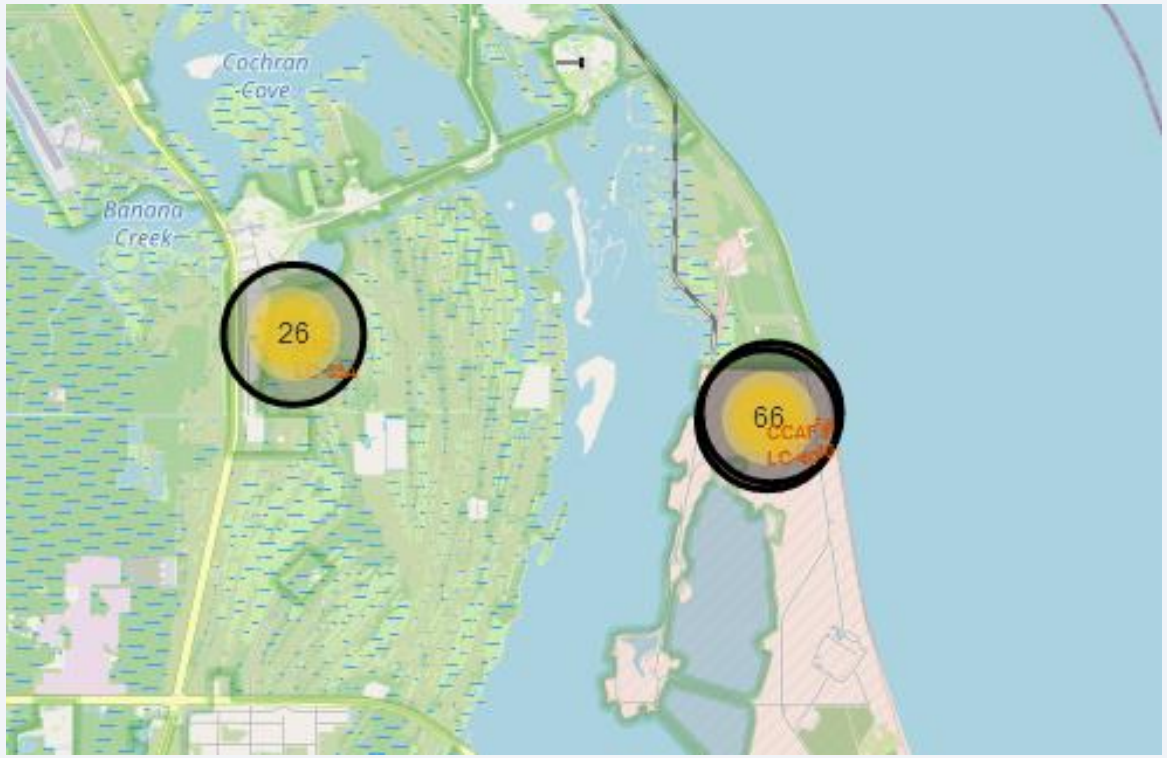
4 Launch Sites

```
site_map.add_child(marker2)
site_map.add_child(circle3)
site_map.add_child(marker3)
"""
for index, row in launch_sites_df.iterrows():
    coordinate = [row['Lat'], row['Long']]
    folium.Circle(coordinate, radius=1000, color='#000000', fill=True).add_child(folium.Popup(row['Launch Site'])).add_to(site_map)
    folium.map.Marker(coordinate, icon=DivIcon(icon_size=(20,20),icon_anchor=(0,0), html='<div style="font-size: 12; color:#d35400;"
site_map
```



Folium map with 4 different sites and markers for each launch sites.

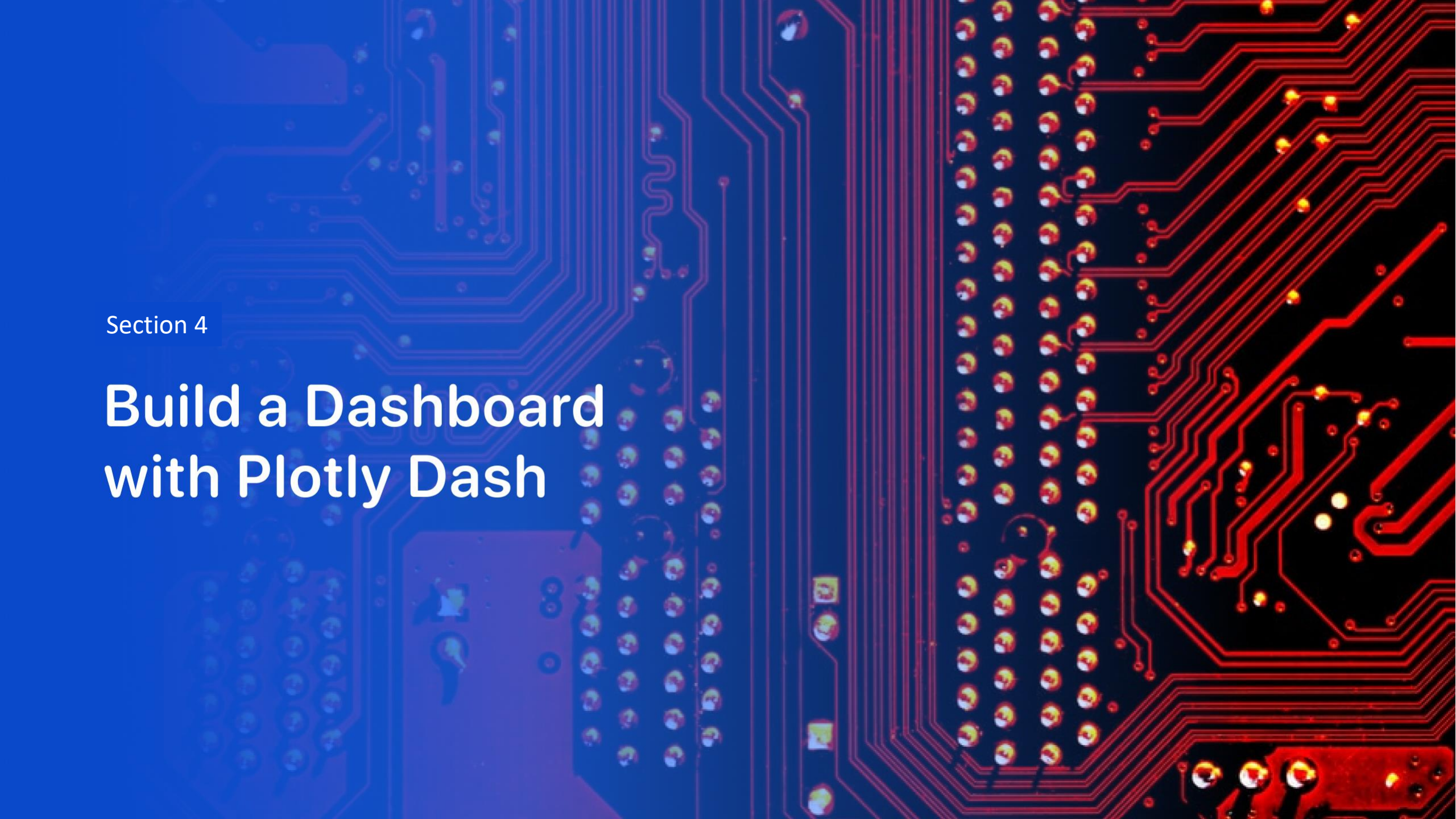
Markers for Success and Failure



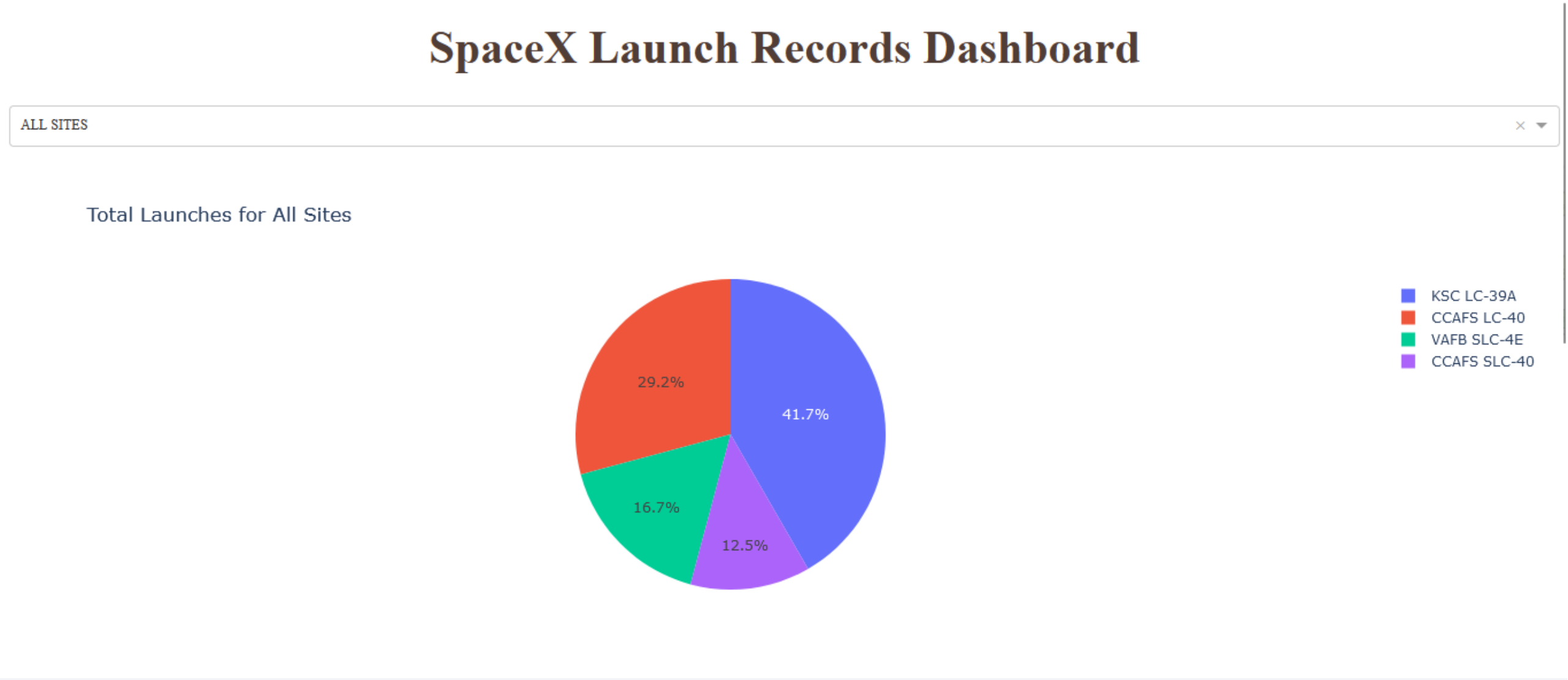
Updated version from the last folium map, which added color for each markers for each launch sites.

Distance between launch site and coastline point

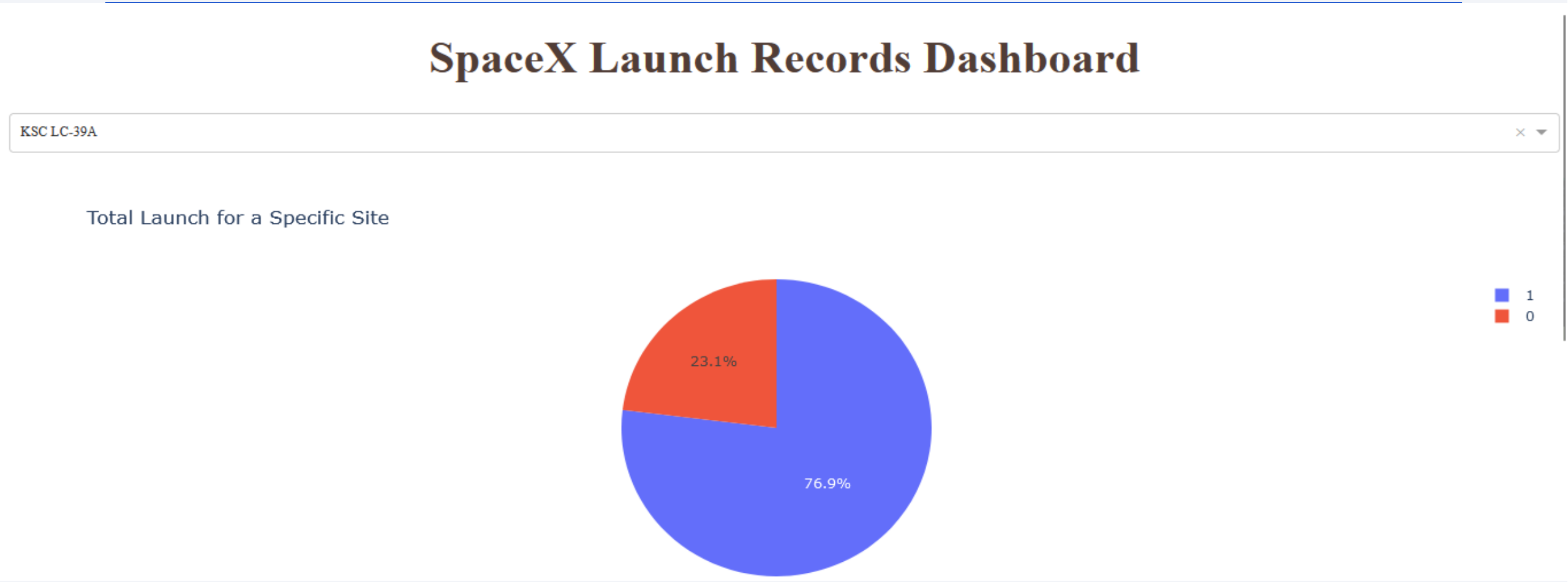
Added Polyline between launch site to the selected coastline point.


Section 4
Build a Dashboard with Plotly Dash

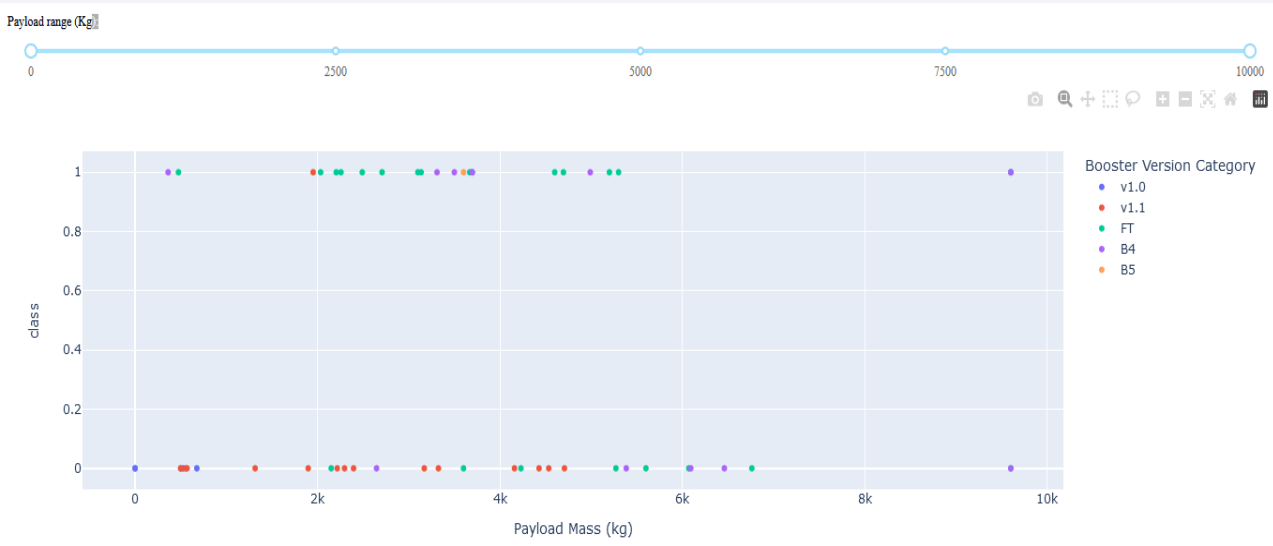
SpaceX Launch Records Dashboard

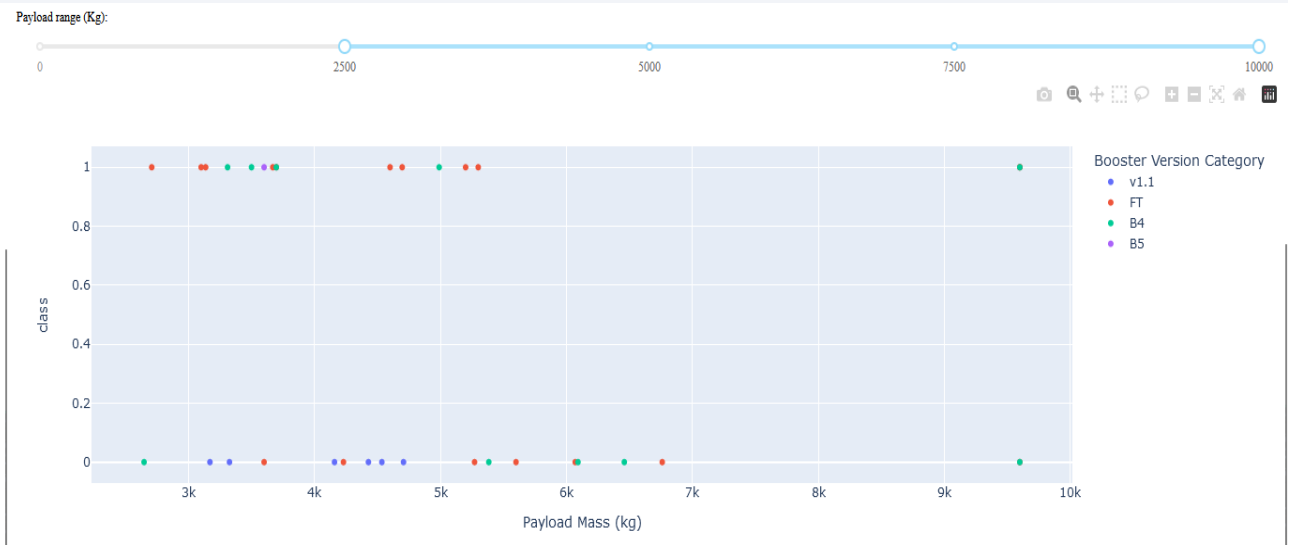


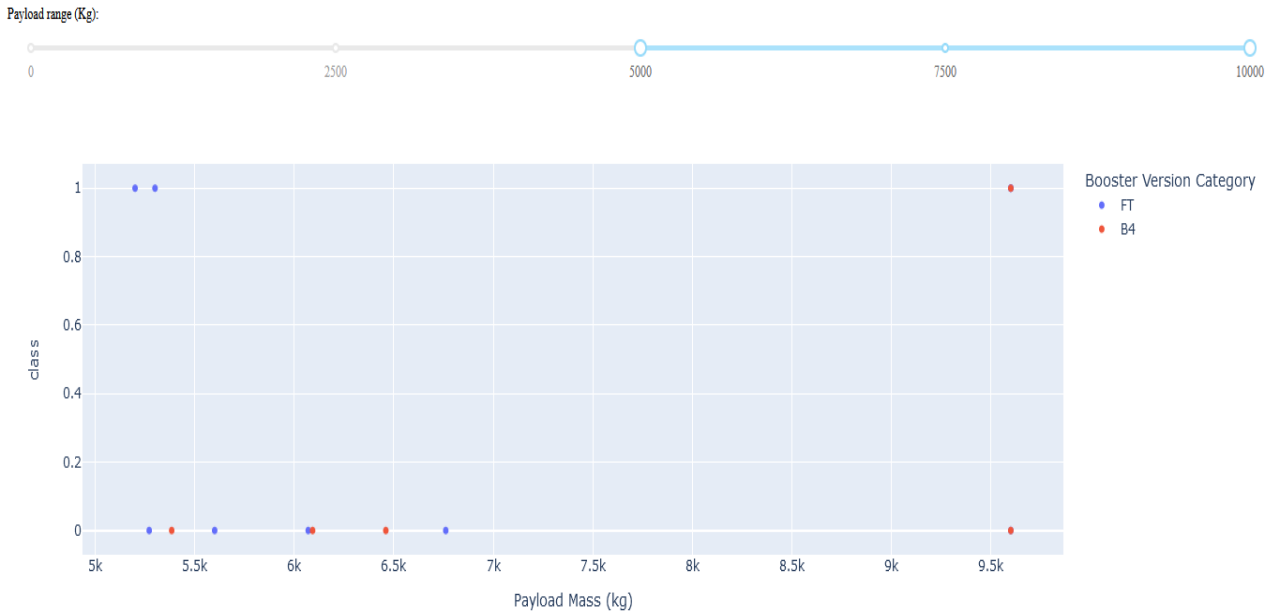
SpaceX Launch Records Dashboard with Highest Launch Success Ratio

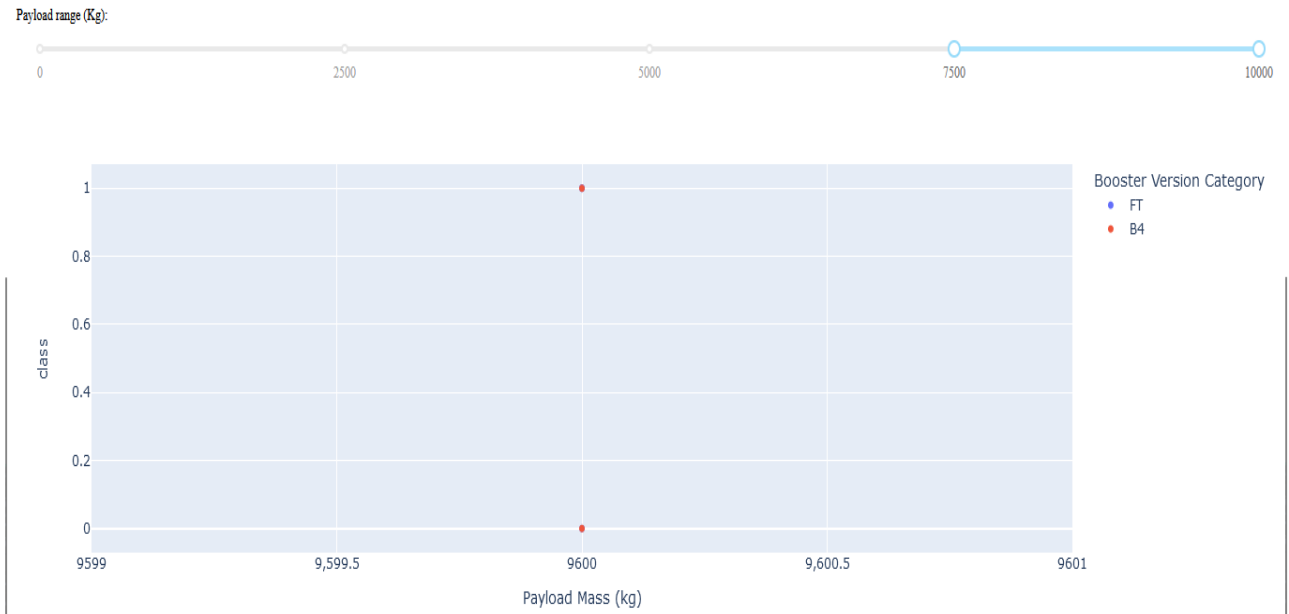


SpaceX Launch Records Dashboard Scatter









Section 5

Predictive Analysis (Classification)

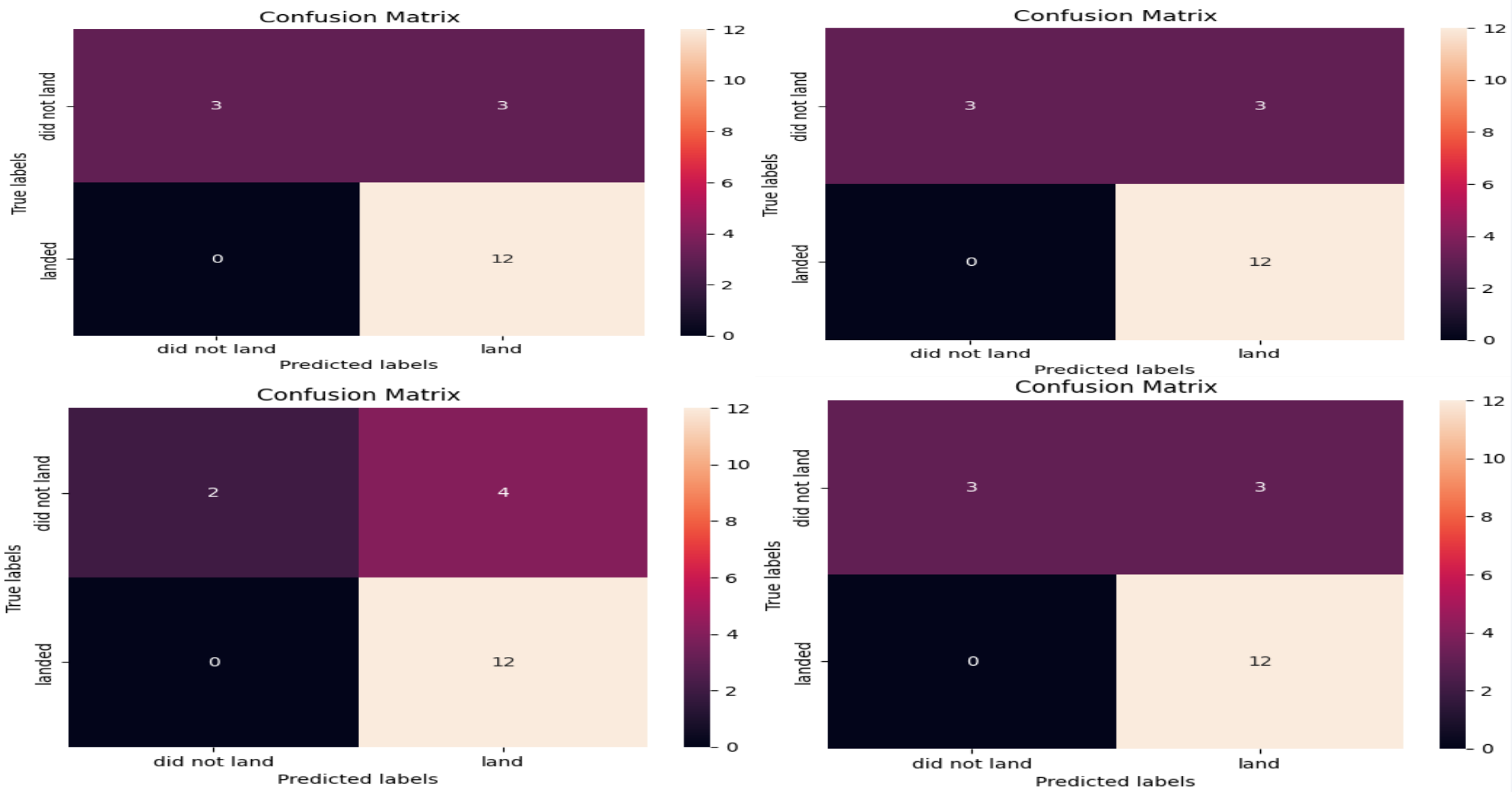
Classification Accuracy

Find the method performs best:

```
# Bagian ane
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print('Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearsdt neighbors method:', knn_cv.score(X_test, Y_test))
```

```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.7777777777777778
Accuracy for K nearsdt neighbors method: 0.8333333333333334
```

Confusion Matrix



Conclusions

- The highest classification accuracy are Logistics Regression method, Support Vector Machine method, and K-Nearest Neighbors method.

Appendix

- https://github.com/Twentsix/ForCourseraForLearning/tree/cc699aed01d898c19206a90fbbc46b11743d4499/FinalProject

Insights

- Perhaps by providing interactive dashboard and folium like map to public might help others interested in data science, rather than dataset in CSV format or others that need to be analyzed first.

Thank you!