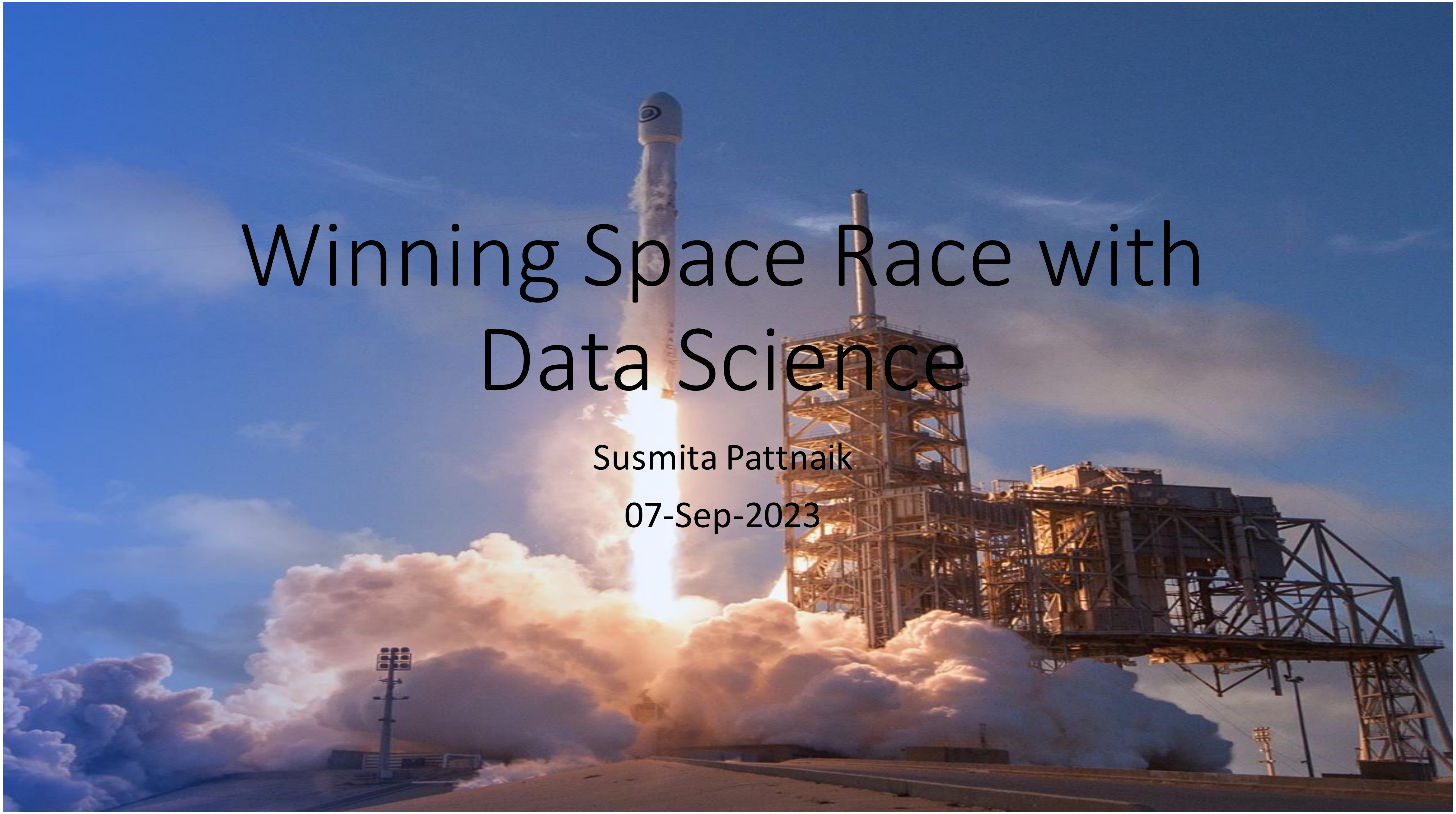


Winning Space Race with Data Science

Susmita Pattnaik

07-Sep-2023

Outline

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

Executive Summary

- Utilize data science methodologies to define and formulate a real-world business problem
- Utilize your data analysis tools to load a dataset, clean it, and find out interesting insights from it

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.

Section 1

Methodology

Methodology

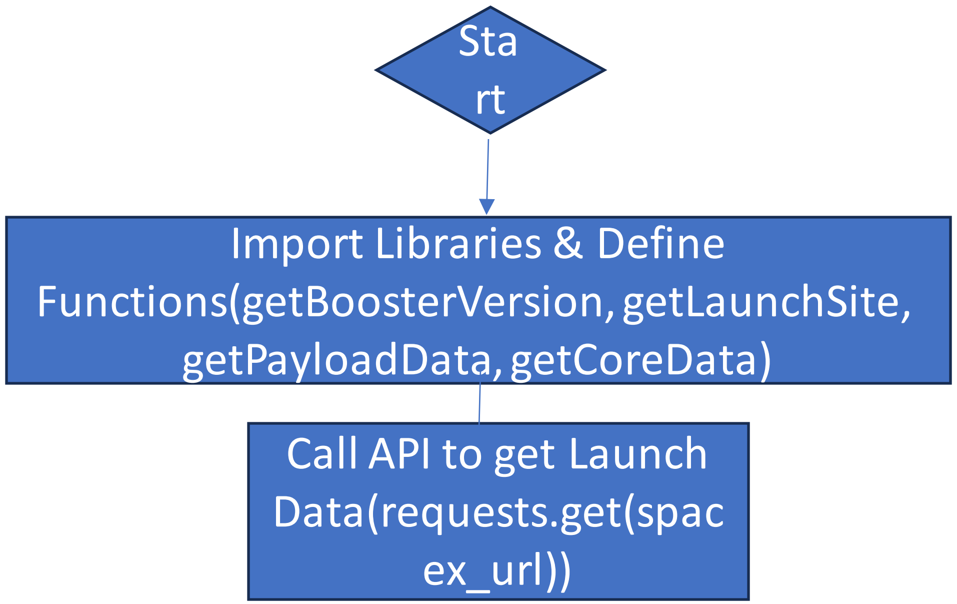
Executive Summary

- Data Collection Methodology
  - Describe how data was collected
- Perform Data Warngling
  - Describe how data was processed
- Perform exploratory data analysis using Visulization and SQL
- Perform interactive visual analytics using Flolium and Plotly Dash
- Perform Predictive Analytics using classification models
  - How to build, tune evaluate classification models

Data Collection

- Describe how Data sets were collected
- Present data collection process use key phrases and flowchart

Data Collection – SpaceX API



- https://github.com/susmitapattnaik/CapStone_Project/blob/main/jupyter-labs-spacex-data-collection-api%20(2).ipynb

Data Collection - Scraping

- Request the Falcon9 Launch Wiki page from its URL
- Extract all column/variable names from the HTML table header. Create Beautifulsoup object from HTML Response
- Extract all column/variable names from the HTML table header
- Create a data frame by parsing the launch HTML tables
- https://github.com/susmitapattnaik/CapStone_Project/blob/main/jupyter-labs-webscraping%20(1).ipynb

Data Wrangling

- https://github.com/susmitapattnaik/CapStone_Project/blob/main/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite%20(1).ipynb
- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurence of mission outcome per orbit type
- Create a landing outcome label from Outcome column

EDA with Data Visualization

EDA with SQL

- %sql select distinct Launch_Site from SPACEXTABLE
- %sql select * from SPACEXTABLE where Launch_Site like 'CCA%' LIMIT 5
- %sql select sum(PAYLOAD_MASS_KG_) as payloadmass from SPACEXTABLE where "Customer" ='NASA (CRS)'
- %sql select AVG("PAYLOAD_MASS_KG_") AS avgpayload from SPACEXTABLE where "Booster_Version"= 'F9 v1.1'
- %sql select min(Date) from SPACEXTABLE where "Landing_Outcome" = 'Success'
- %sql select distinct "Booster_Version" from SPACEXTABLE where "Landing_Outcome" = 'Success (drone ship)'
- %sql select "Mission_Outcome", count(1) from SPACEXTABLE
- %sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
- %sql select substr(Date, 4, 2) , Landing_Outcome, Booster_Version, Launch_Site from SPACEXTBL where substr(date,1,4)='2015'
- %sql SELECT LANDING_OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;

Github Link:

https://github.com/susmitapattnaik/CapStone_Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

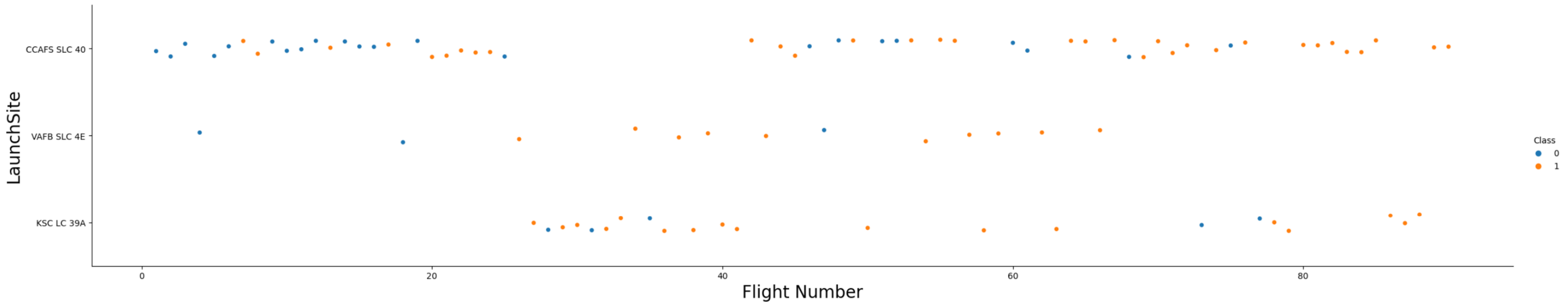
Build and Interactive map with Folium

- Github Link - https://github.com/susmitapattnaik/CapStone_Project/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb
- This Exercise includes using markers, Circles and lines on Folium maps to identify launch sites and study

Section 2

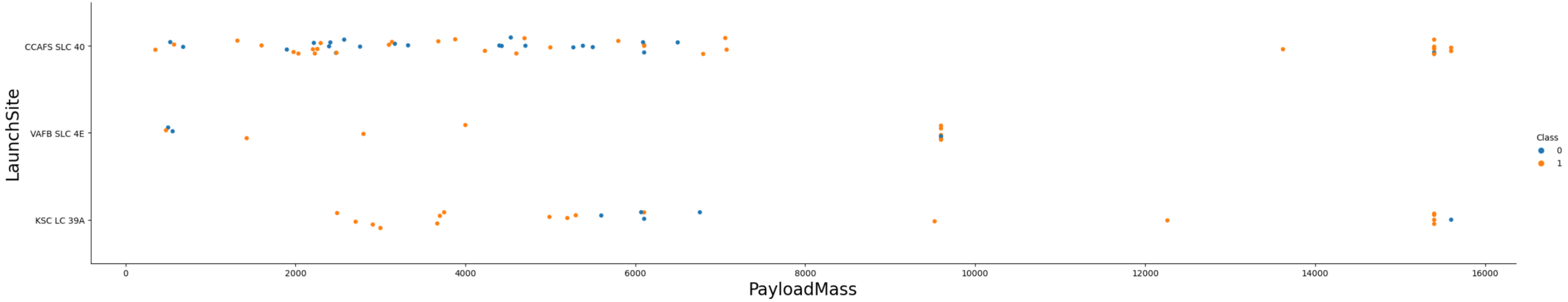
Insights drawn from EDA

Flight Number Vs. Launch Site




```
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```

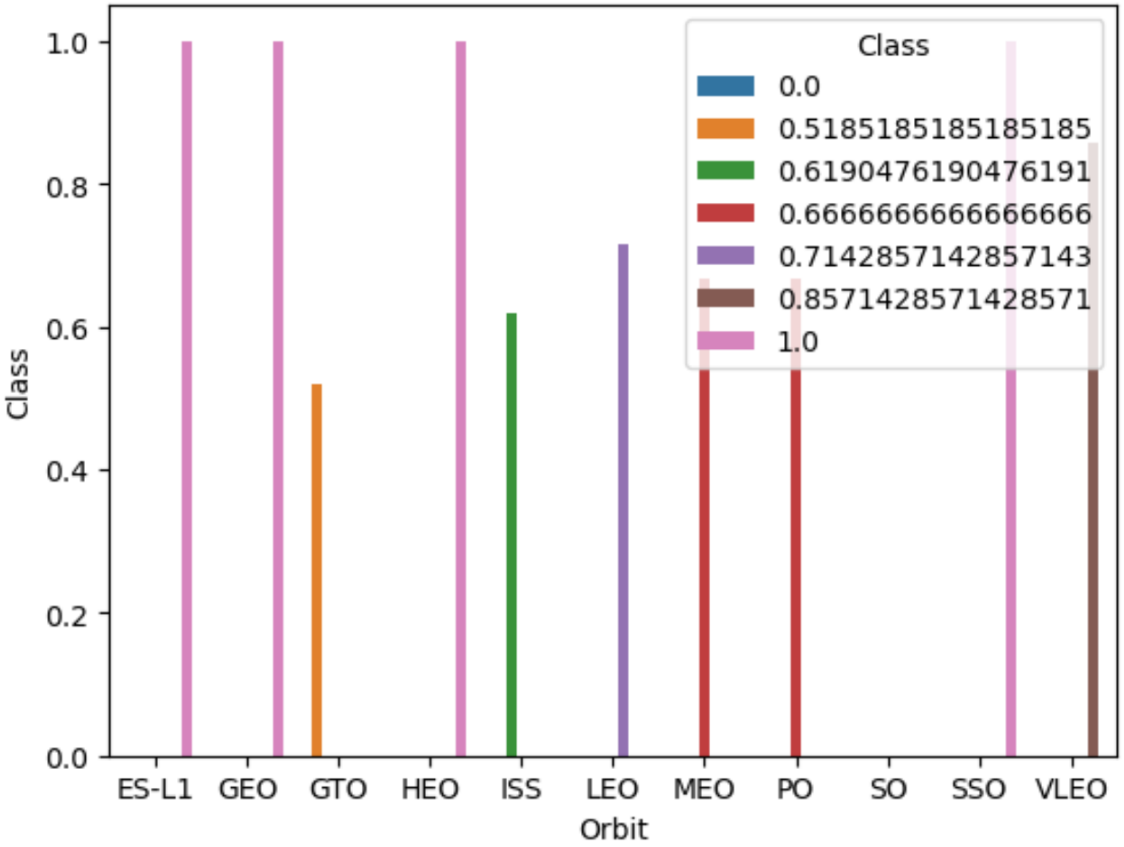
PayLoad Vs. Launch Site




```
### TASK 2: Visualize the relationship between Payload and Launch Site
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```

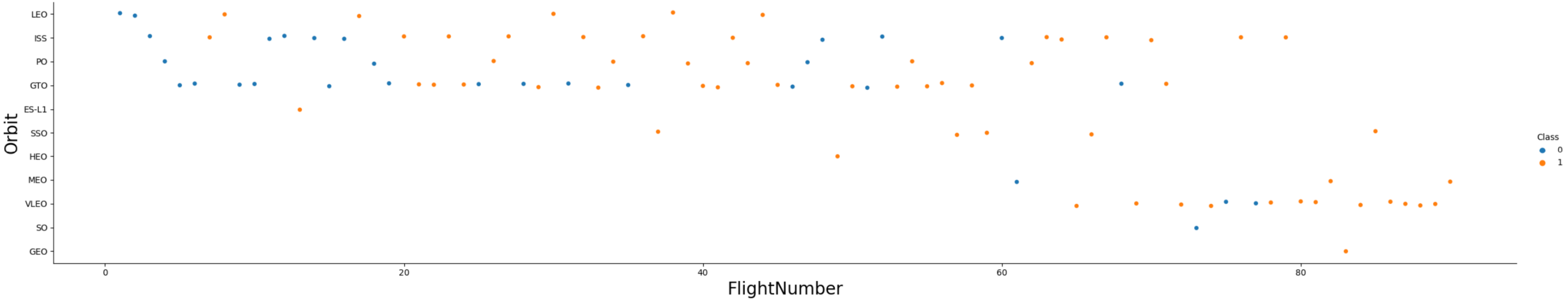
Success Rate Vs. Orbit Type

```
### TASK  3: Visualize the relationship between success rate of each orbit type
orbit_success = df.groupby('Orbit').mean()
orbit_success.reset_index(inplace=True)
sns.barplot(x="Orbit",y="Class",data=orbit_success,hue='Class')
```



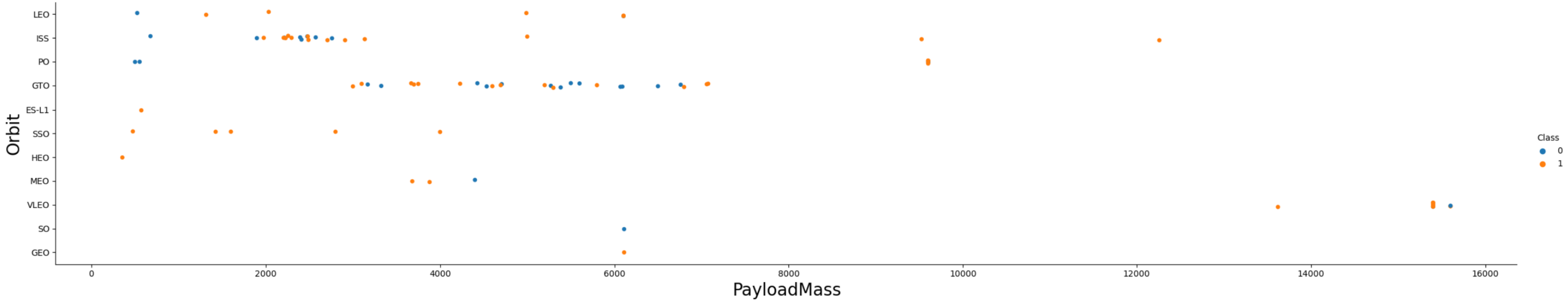
Flight Number Vs. Orbit Type

```python
# 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("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



PayLoad Vs. Orbit Type

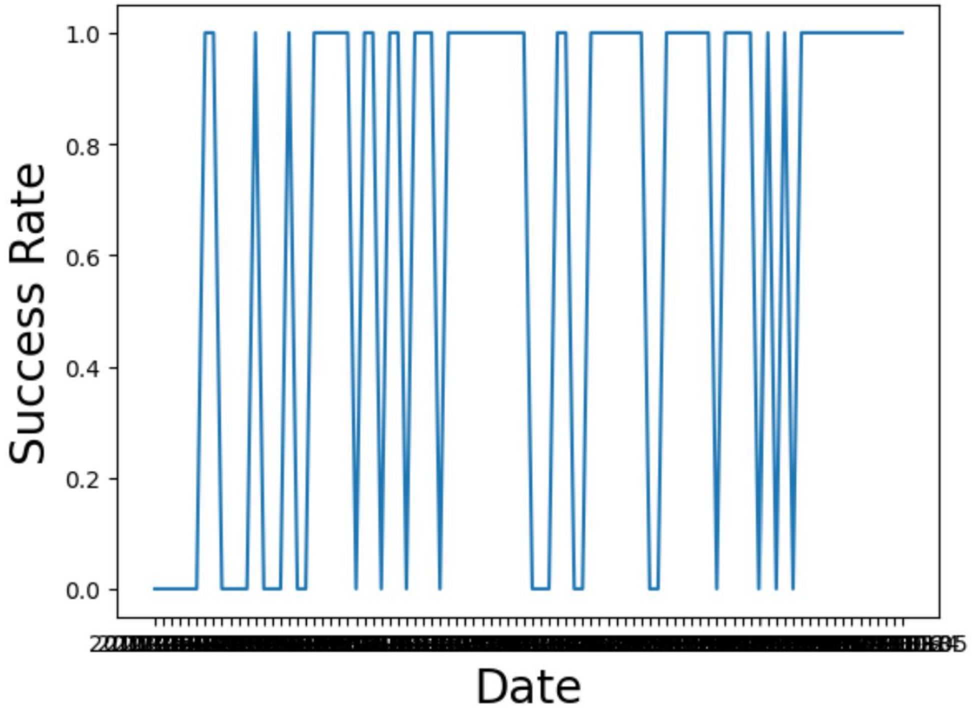
```
[12]: # Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

Launch Success Yearly Trend

```
[6]: # Plot a line chart with x axis to be the extracted year and y axis to be the success rate
sns.lineplot(data=df, x="Date", y="Class")
plt.xlabel("Date",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



All Launch Site Names

```
[10]: %sql select distinct Launch_Site from SPACEXTABLE
```

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

[10]:

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'

```
[2]: %sql select *  from SPACEXTABLE where Launch_Site like 'CCA%' LIMIT 5
```

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

[2]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total PayLoad Mass

Task 3

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

[10]:
```
%sql select sum("PAYLOAD_MASS_KG_") as payloadmass from SPACEXTABLE where  Customer ='NASA (CRS)'
```

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

[10]:

payloadmass
0.0

Average PayLoad Mass by F9 v1.1

Task 4

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

```
[25]: %sql select AVG("PAYLOAD_MASS_KG_") AS avgpayload from SPACEXTABLE where "Booster_Version"= 'F9 v1.1'
```

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

[25]:

avgpayload
0.0

First Successful Ground Landing Date

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

Hint:Use min function

[26]:
```
%sql select min(Date) from SPACEXTABLE where "Landing_Outcome" = 'Success'
```

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

[26]:

min(Date)
2018-03-12

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

[28]:
```
%sql select distinct "Booster_Version" from SPACEXTABLE where "Landing_Outcome" = 'Success (drone ship)'
```

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

[28]:

Booster_Version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

[29]:
```
%sql select "Mission_Outcome", count(1) from SPACEXTABLE
```

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

[29]:

Mission_Outcome	count(1)
Success	101

Boosters Carried Maximum Payload

Task 8

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

[30]:
```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

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

[30]:

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

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.

[32]:
```
%sql select substr(Date, 4, 2) , Landing_Outcome, Booster_Version, Launch_Site from SPACEXTBL where substr(date,1,4)='2015'
```

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

[32]:

substr(Date, 4, 2)	Landing_Outcome	Booster_Version	Launch_Site
5-	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
5-	Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40
5-	No attempt	F9 v1.1 B1014	CCAFS LC-40
5-	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
5-	No attempt	F9 v1.1 B1016	CCAFS LC-40
5-	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40
5-	Success (ground pad)	F9 FT B1019	CCAFS LC-40

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.

[34]:
```
%sql SELECT LANDING_OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;
```

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

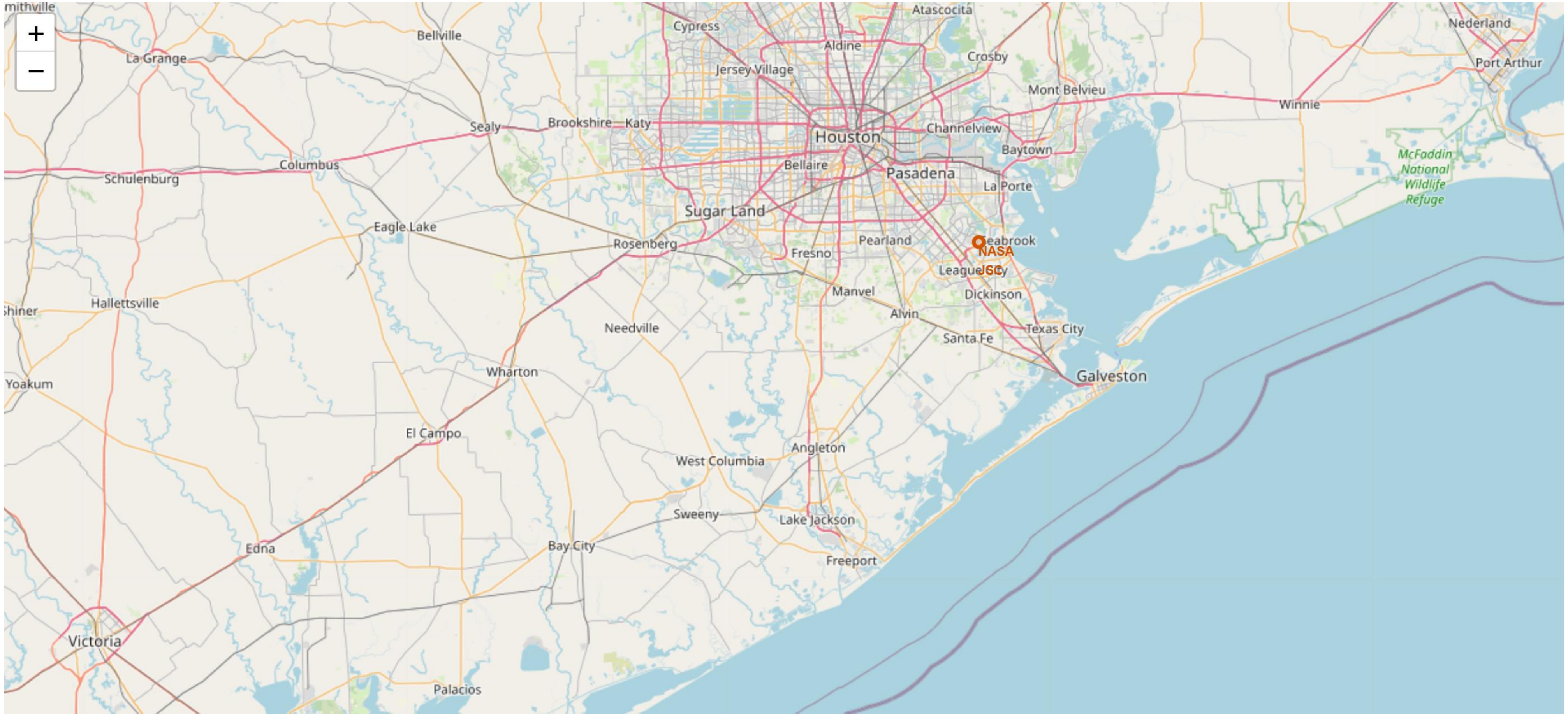
[34]:

Landing_Outcome
No attempt
Success (ground pad)
Success (ground pad)
Success (drone ship)
Success (ground pad)
Success (drone ship)
Success (drone ship)
Success (ground pad)
Failure (drone ship)
Success (drone ship)
Success (drone ship)
Failure (drone ship)
Failure (drone ship)
Success (ground pad)
Controlled (ocean)
Failure (drone ship)
Precluded (drone ship)
No attempt
Failure (drone ship)
No attempt
Uncontrolled (ocean)
Controlled (ocean)

Section 3

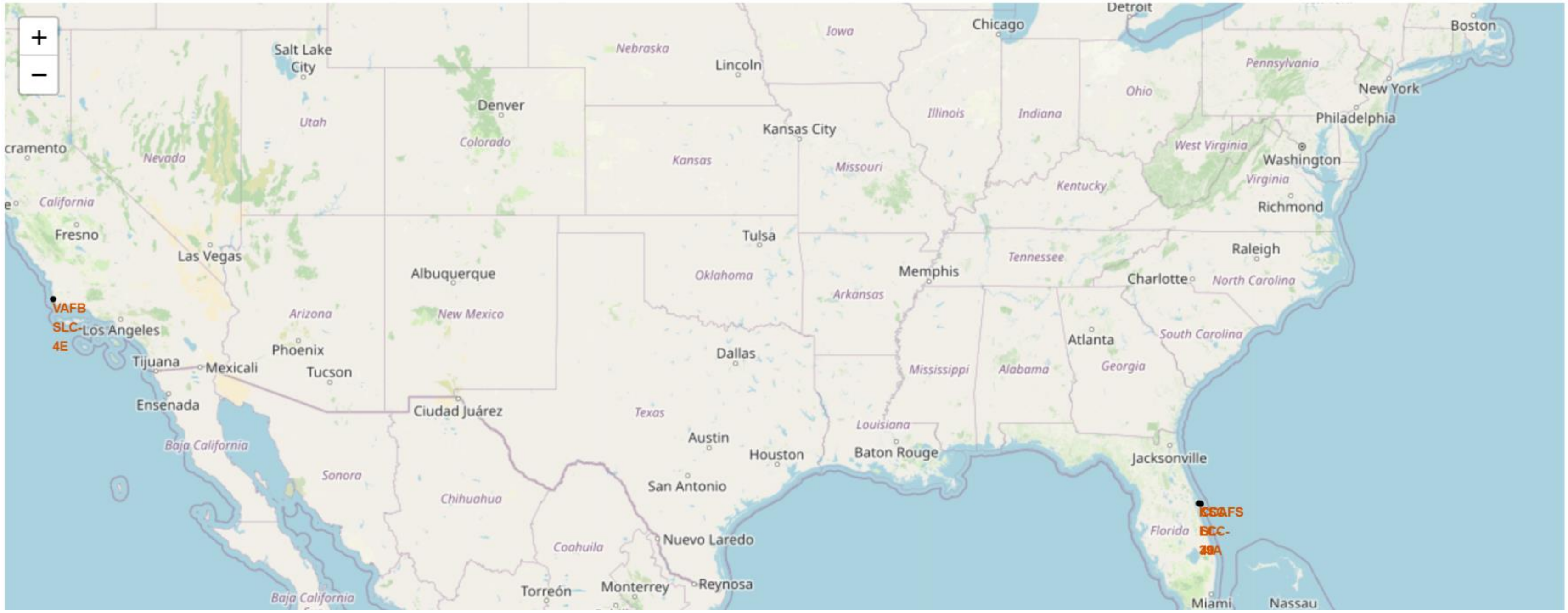
Launch Sites Proximities Analysis

Folium Map Screenshot 1

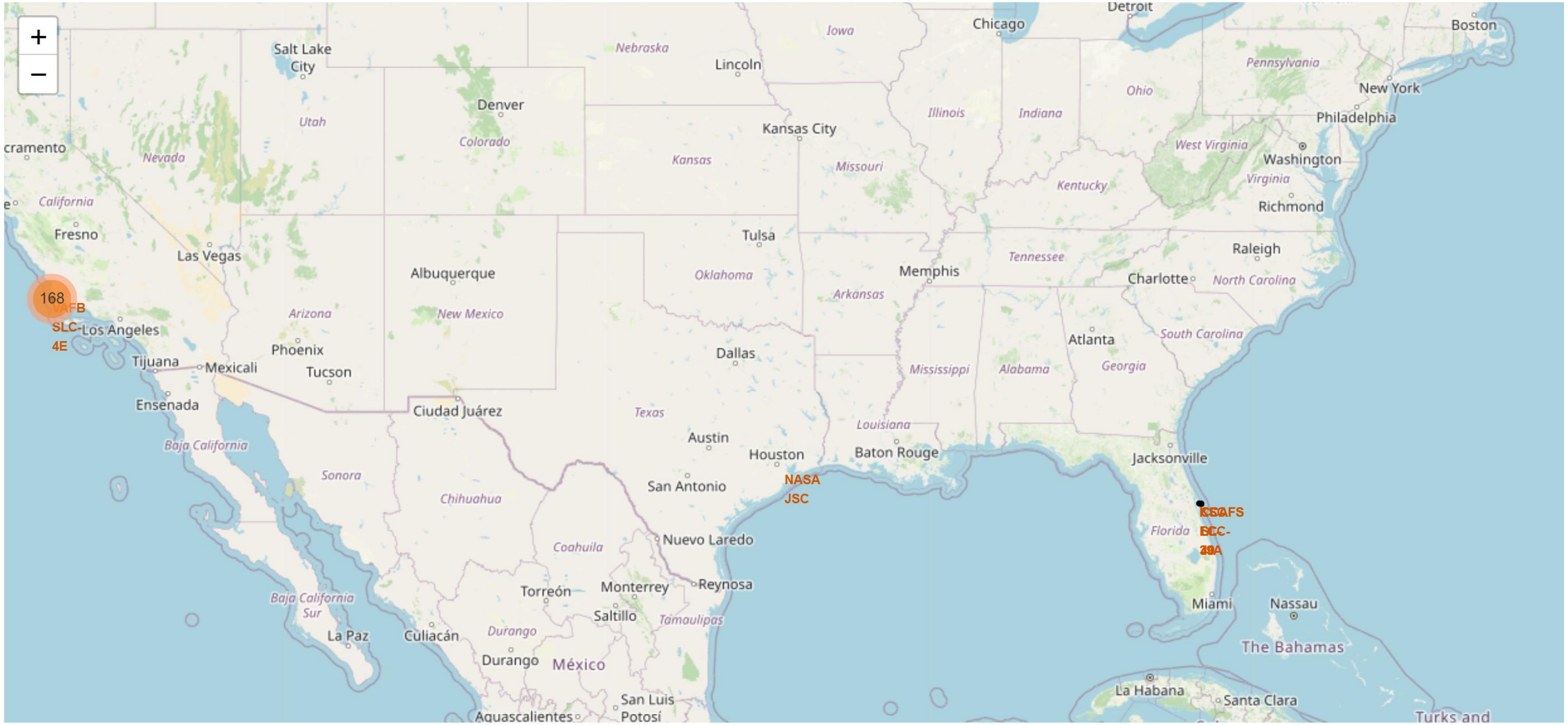


Folium Map Screen Shot 2

```python
# Initial the map
site_map = folium.Map(location=nasa_coordinate, zoom_start=5)
# For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition, add Launch site name as a popup label
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;"><b>%s</b></div>' % row['Launch Site'], )).
site_map
```



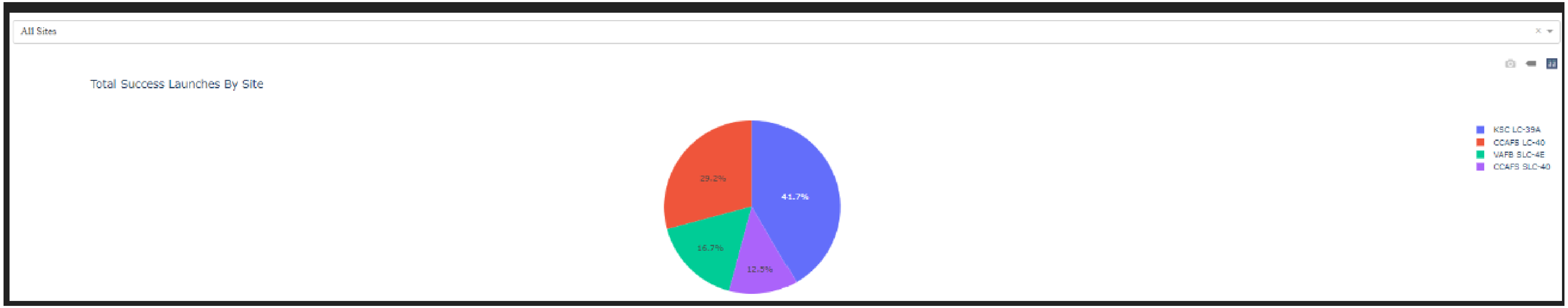
Folium Map Screen Shot 3



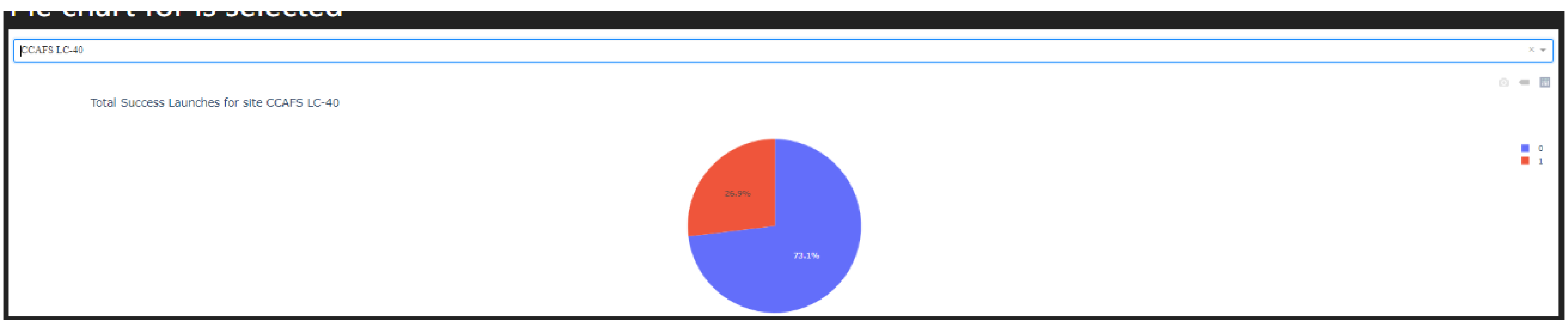
Section 4

Build a Dashboard with Plotly Dash

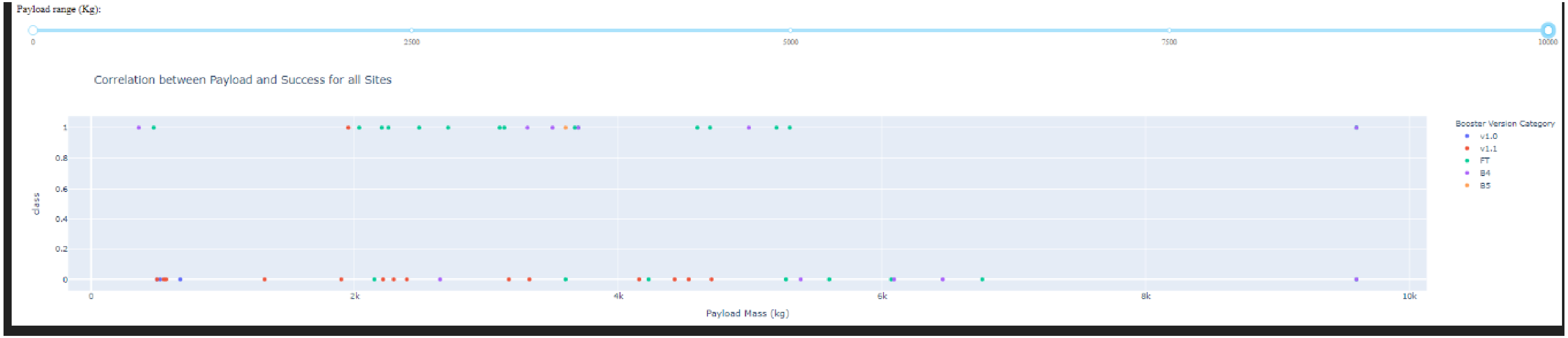
Dashboard Screenshot -1



Dashboard Screenshot -2



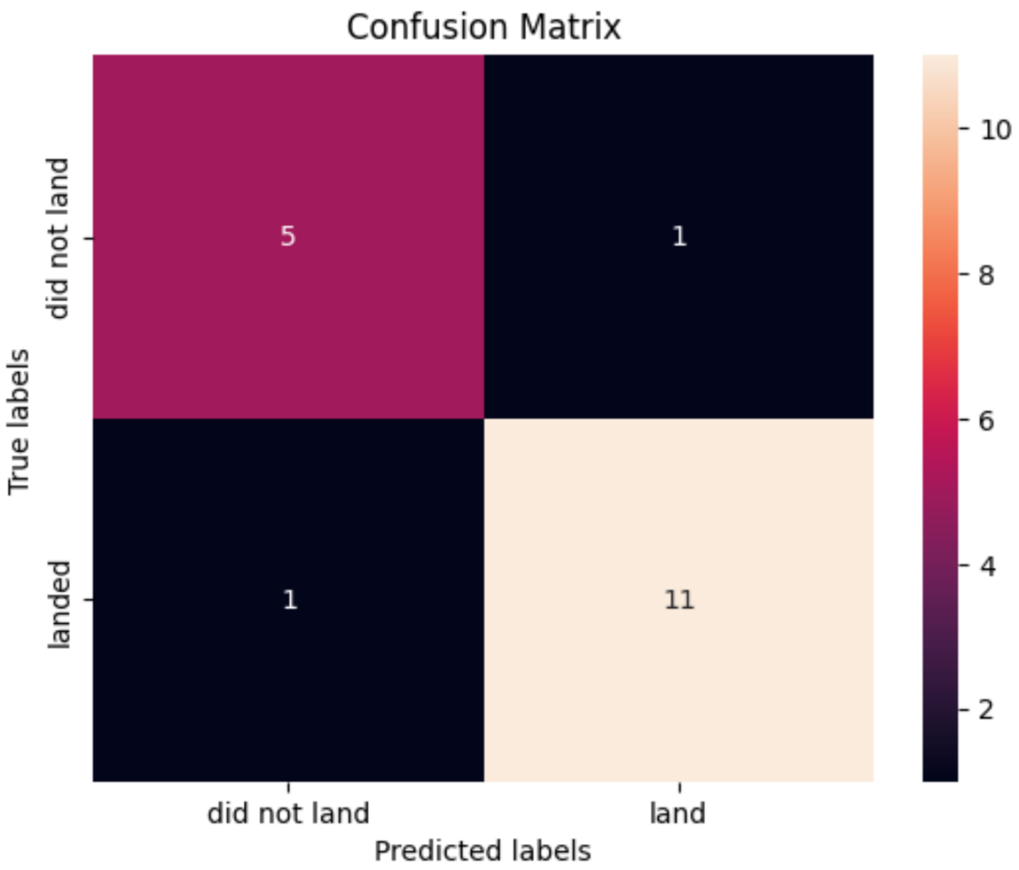
Dashboard Screenshot -3



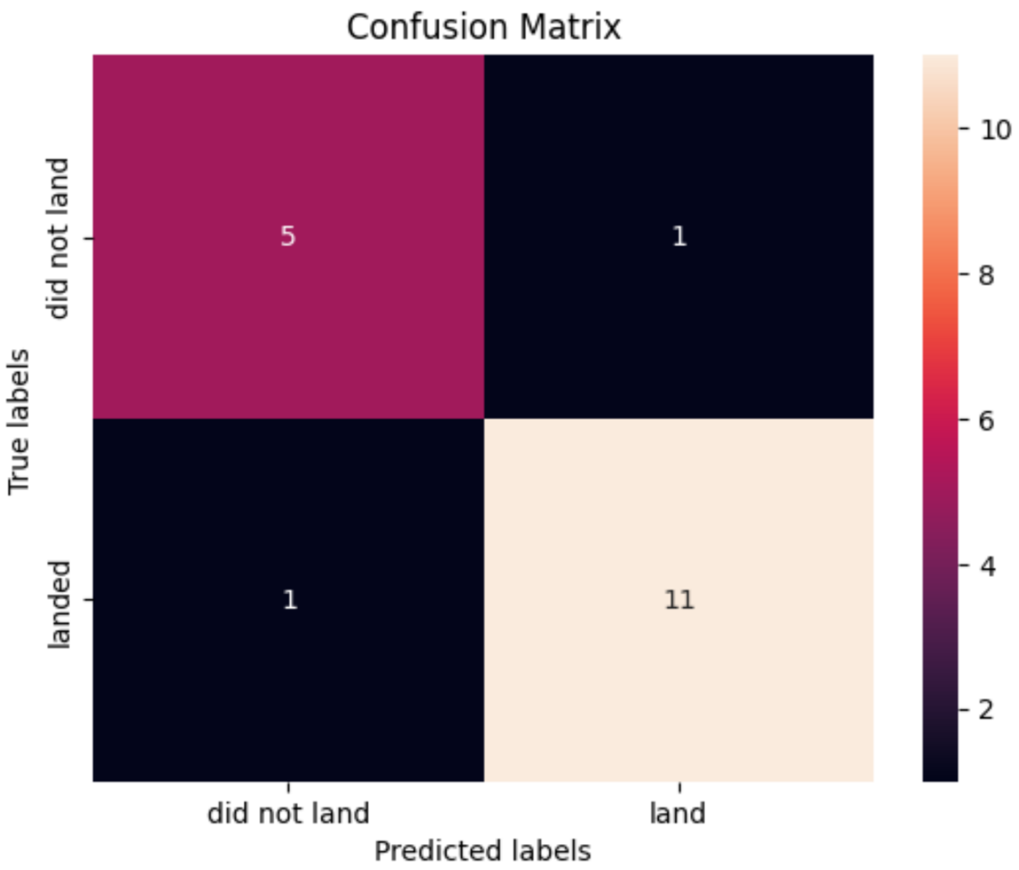
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Conclusions

TASK 12

Find the method performs best:

In [45]:

```
predictors = [knn_cv, svm_cv, logreg_cv, tree_cv]
best_predictor = ""
best_result = 0
for predictor in predictors:

    predictor.score(X_test, Y_test)
```

Out[45]: 0.8333333333333334

Appendix

Thank you!