IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- Data was collected using a REST API and Data Scraping
- Data was analysed using geographical maps and other visualization tools to find relationship between data and mission success
- Data was prepared for machine learning methods(normalizing, one hot encoding, splitting in train and test data etc.)
- different machine learning objects were trained and tested for accuracy
- the final model reached an accuracy of about 83.3%

Introduction

- SpaceX requires 62 Million dollars compared to 165 million dollar for rocket mission
- Possible due to savings from reusing the first rocket stage
- The reusing of the first stage is not always a success

- Tasks:
- determine the price of Falcon 9 launch
- Determine whether Space X missions will reuse the first stage
  - Done by a machine learning model
  - Use of public information regarding the launches of Space X

-

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - REST API, Data Scraping
- 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

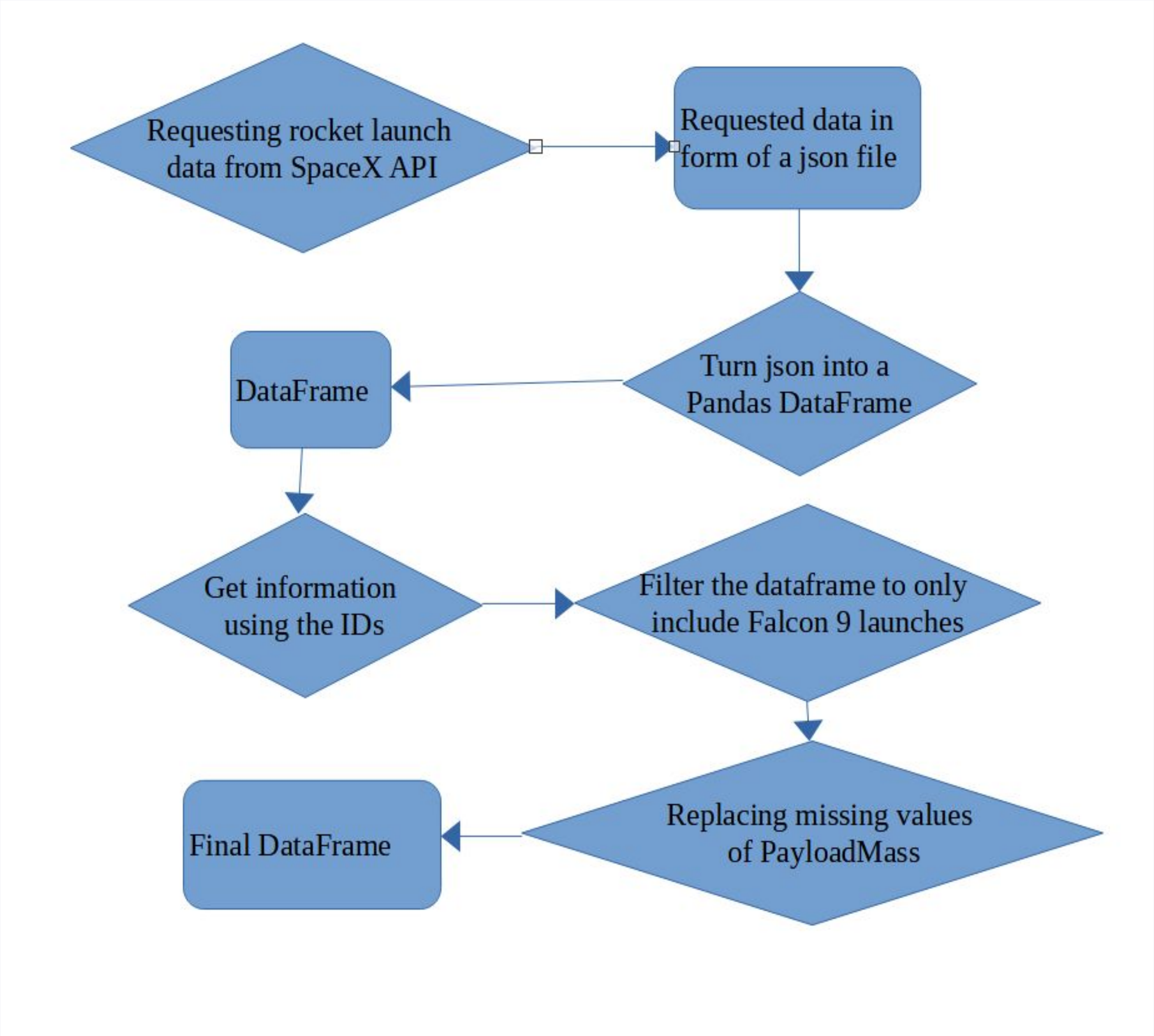
Data Collection

- Data collection from SpaceX REST API:
- Contained information: rocket used, payload delivered, landing outcome...
- Goal: predict attempt to land rocket or not
- Web scraping Falcon 9 Launch records:
- From html tables
- Get specific information of id`s

Data Collection – SpaceX API

- Request and parse the SpaceX launch data using tth GET request
- Json file => Pandas DataFrame
- Get information with the IDs
- Data wrangling

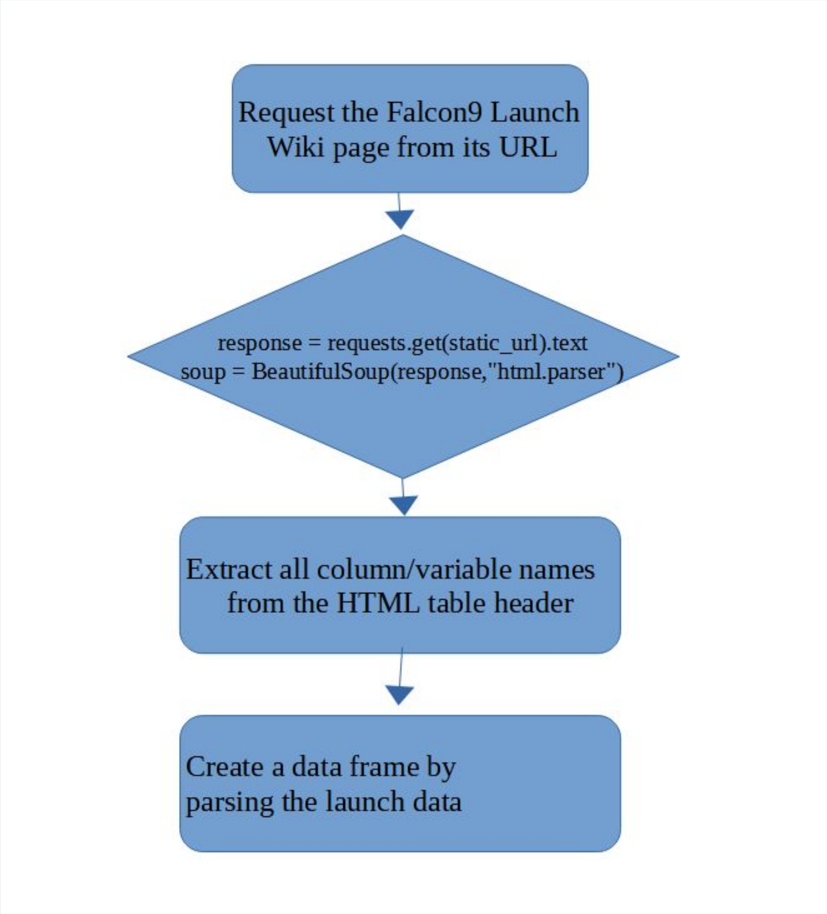
- SpaceX API calls notebook: https://github.com/JannisEbling/Coursera/blob/main/FinalPresentation/notebooks_Python_files/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Request the Falcon9 Launch Wiki
- Extract data from HTML table
- Create a dataframe by parsing the launch data

- Web scraping notebook: https://github.com/JannisEbling/Coursera/blob/main/FinalPresentation/notebooks_Python_files/jupyter-labs-webscraping.ipynb



Data Wrangling

- Number of Launches on each side:

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

- Number and occurrence of each orbit
- number and occurrence of mission outcome per orbit type
- Creation of an landing outcome label
- Success rate of recover stage 1: 67%
- data wrangling notebook: https://github.com/JannisEbling/Coursera/blob/main/FinalPresentation/notebooks_Python_files/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

EDA with Data Visualization

- Kinds of Data Visualization:
  - geographical maps with launch locations
  - bar plot
  - scatter plots
  - line plots
- Data Visualization to find relationships in data to mission success

- EDA with data visualization notebook: https://github.com/JannisEbling/Coursera/blob/main/FinalPresentation/notebooks_Python_files/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Names of the unique launch sites in the space missions
- Records where launch sites begin with the string “CCA”
- Total payload mass carried by boosters launched by NASA (CRS)
- Average payload mass carried by booster version F9 v1.1
- Date when the first successful landing outcome in ground pad was achieved
- names of booster which had success in drone ship and a certain payload
- number of successful and failure mission outcomes
- Names of the booster_versions which have carried the maximum payload
- rank the count of successful landing_outcomes in a date range

- EDA with SQL notebook: https://github.com/JannisEbling/Coursera/blob/main/FinalPresentation/notebooks_Python_files/jupyter-labs-eda-sql-coursera_sqllite.ipynb

-

Build an Interactive Map with Folium

- Markers and circles for launch sites
  - To analyze at which sites, missions where successfull
  - markers were color-coded regarding their success
- Distance measure to highways and coastline to analyze their influence
- interactive map with Folium map: https://github.com/JannisEbling/Coursera/blob/main/FinalPresentation/notebooks_Python_files/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Drop down menu for selection of the site
- Pie chart to show the total launches at each site and their successrate at each site
- slider for the selection of Payload range
- scatter diagram to visualize the success of launches for each site in relation to the payload
- GitHub URL of the Plotly Dash https://github.com/JannisEbling/Coursera/blob/main/FinalPresentation/notebooks_Python_files/dashboard.py

Predictive Analysis (Classification)

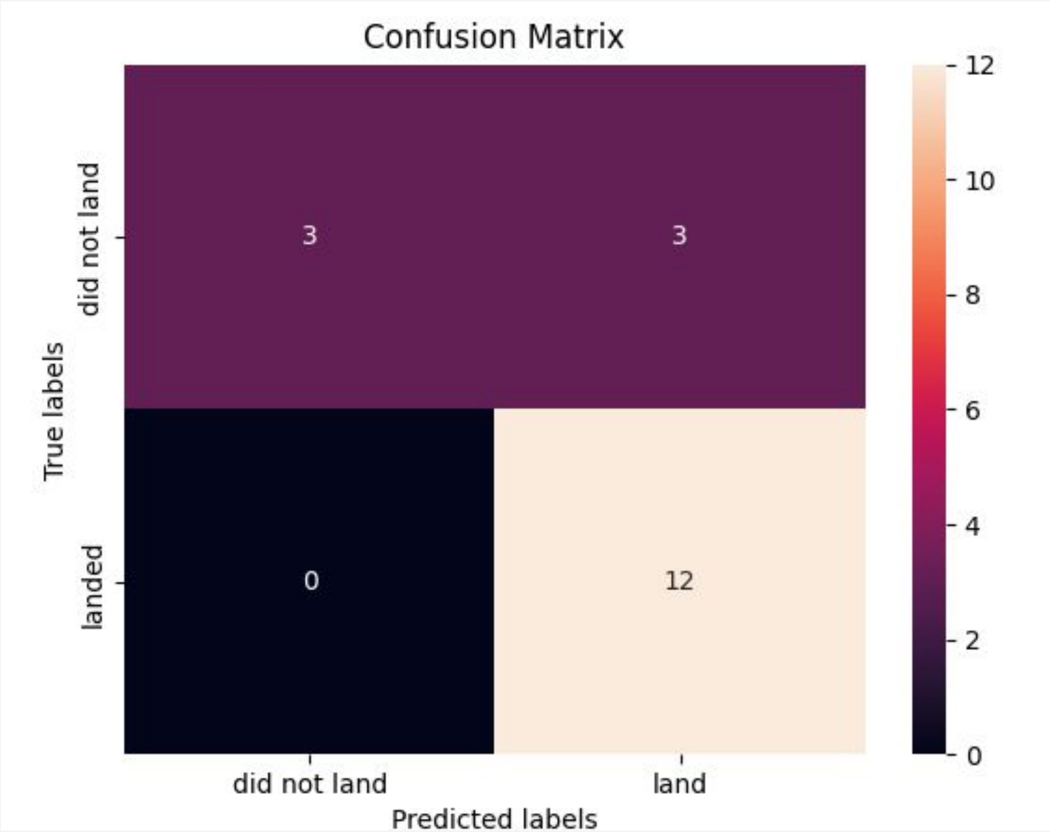
- Data was prepared for training (Standarize, subdivision into trainings and test set)
- Four models where tested:
    - logistic regression
    - Support Vector Machine
    - Decision Tree
    - K Nearest Neighbor

best parameter were found by using Grid search

- GitHub URL of predictive analysis: https://github.com/JannisEbling/Coursera/blob/main/FinalPresentation/notebooks_Python_files/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

- All models had an accuracy of 83.3% after best parameters were selected
- If the same 3 cases were predicted wrong by each model, these might be outliers



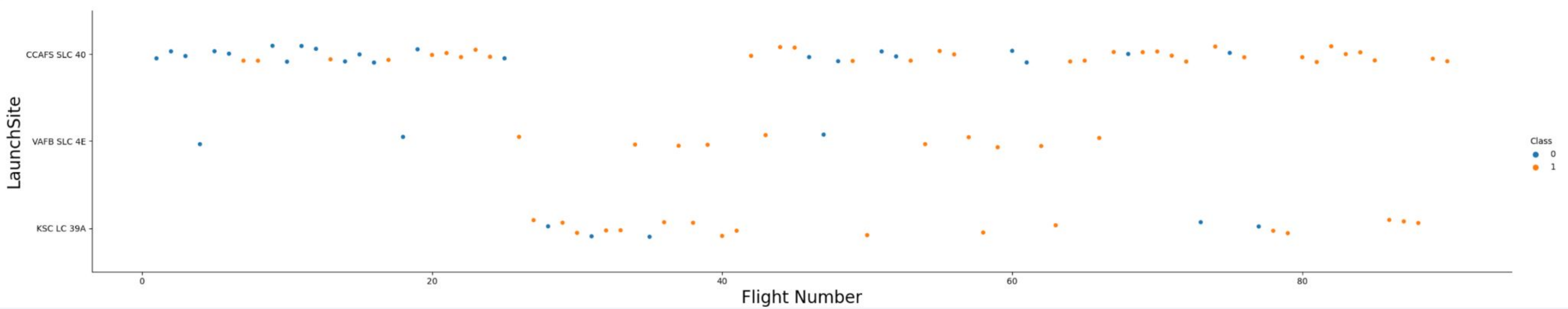
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Shows a scatter plot of Flight Number vs. Launch Site to show their relationship to mission success

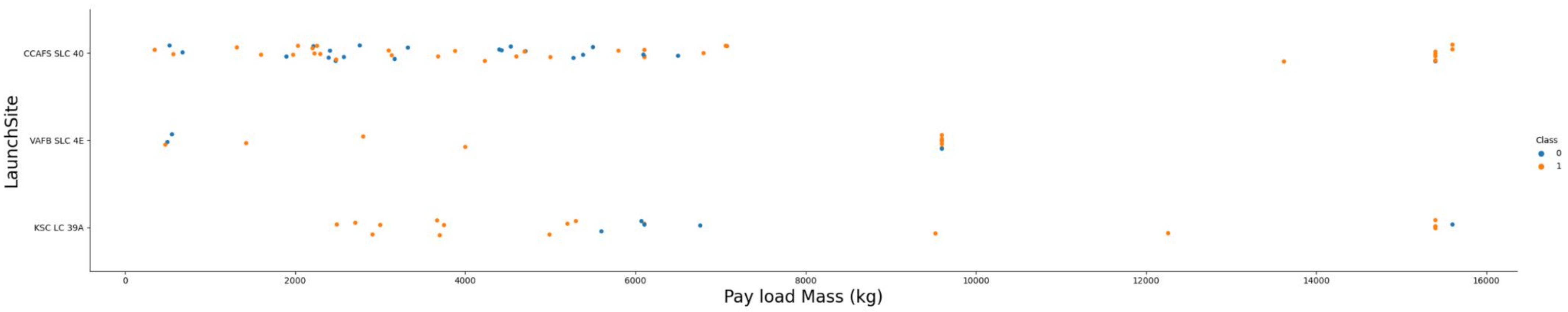
```
### TASK 1: Visualize the relationship between Flight Number and Launch Site
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



Payload vs. Launch Site

- Shows a scatter plot of Payload vs. Launch Site to show their relationship to mission success

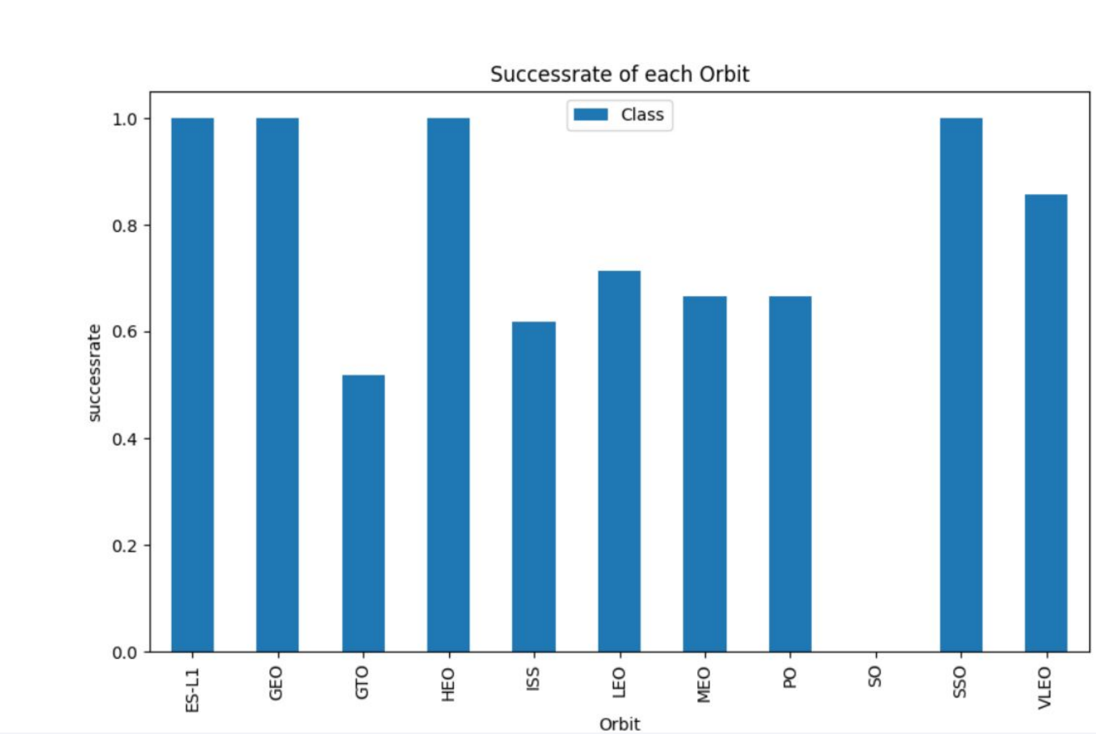
```
### TASK 2: Visualize the relationship between Payload and Launch Site
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



Success Rate vs. Orbit Type

- Shows a bar chart for the success rate of each orbit type to showcase whether orbit type influences success rate

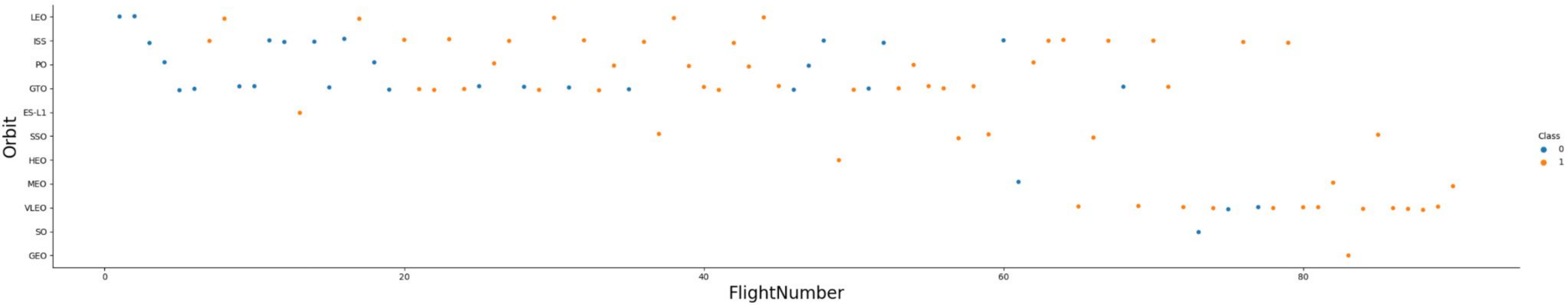
```
### TASK  3: Visualize the relationship between success rate of each orbit type
# step 1: get the data
df_s = df.groupby(['Orbit']).mean()
df_s = df_s[[ "Class"]]

# step 2: plot data
df_s.plot(kind='bar', figsize=(10, 6))
plt.xlabel('Orbit') # add to x-label to the plot
plt.ylabel('successrate') # add y-label to the plot
plt.title('Successrate of each Orbit') # add title to the plot
plt.show()
```



Flight Number vs. Orbit Type

- Shows a scatter point of Flight number vs. Orbit type to show their relationship to success

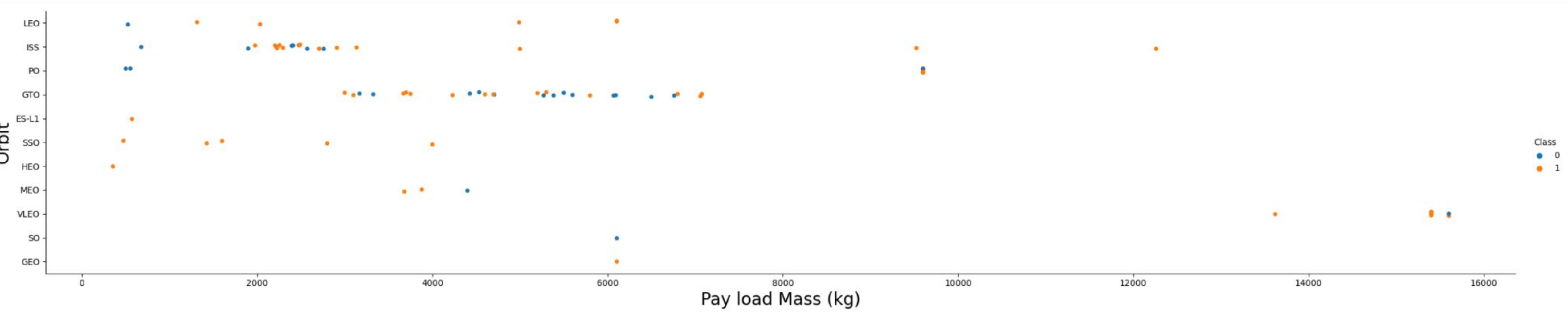
```
### TASK  4: Visualize the relationship between FlightNumber and Orbit type
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Payload vs. Orbit Type

- Shows a scatter point of payload vs. orbit type to show their relationship to success

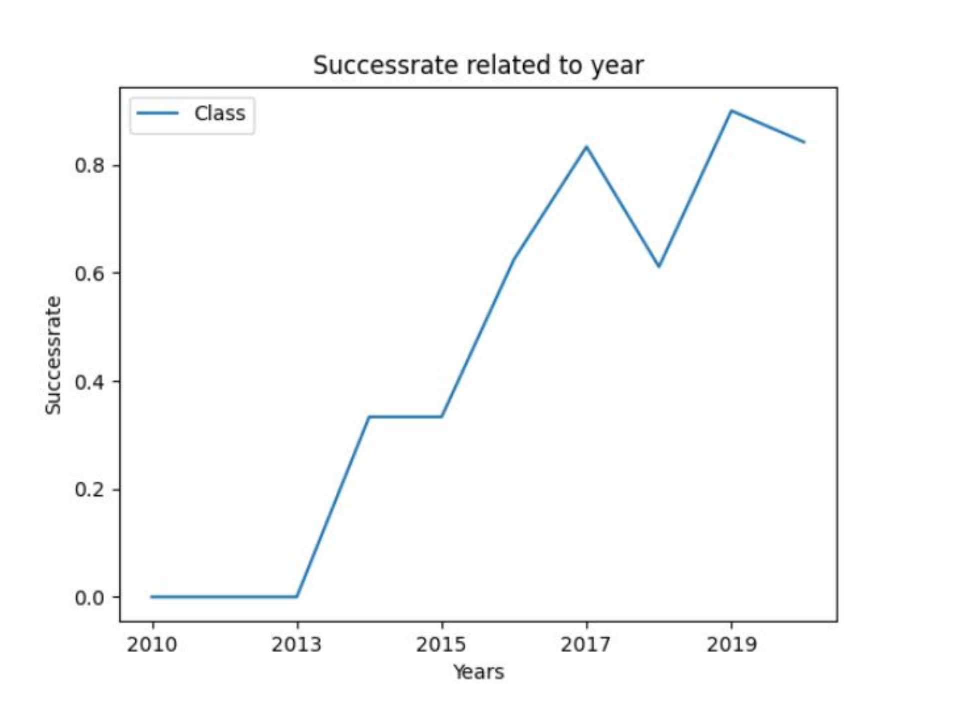
```
## TASK  5: Visualize the relationship between Payload and Orbit type

ns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
lt.xlabel("Pay load Mass (kg)",fontsize=20)
lt.ylabel("Orbit",fontsize=20)
lt.show()
```



Launch Success Yearly Trend

- Shows a line chart of yearly average success rate to show their relationship

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
df_2 = df.groupby(['Date']).mean()
df_2 = df_2[[ "Class"]]
df_2.plot(kind='line')

plt.title('Successrate related to year')
plt.ylabel('Successrate')
plt.xlabel('Years')

plt.show() # need this line to show the updates made to the figure
```



All Launch Site Names

- In this table, all distinct launch_sites can be seen

```
%sql select distinct launch_site from SPACEXTBL
```

```
 * 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'

- In this table, 5 missions can be seen where the launch site started with "CAA"

Task 2

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

```
%sql select * from SPACEXTBL 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
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- This picture shows the total payload carried by boosters from NASA

Task 3

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

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where customer = "NASA (CRS)"
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- This picture shows the average payload mass carried by booster version F9 v1.1

Task 4

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

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where booster_Version = "F9 v1.1"
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- This picture shows the date of the first successful landing date

Task 5

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

Hint:Use min function

```
%sql select min(date) from SPACEXTBL  where "Landing _Outcome" = "Success (ground pad)"
```

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

min(date)
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

- In this table are the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 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 SPACEXTBL  where "Landing _Outcome" = "Success (drone ship)" and Payload_Mass__KG_ >4000 and Payload_Mass__KG_ <6000
```

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

- This picture shows total number of successful and failure mission outcomes

Task 7

List the total number of successful and failure mission outcomes ¶

```
%sql select Mission_Outcome, count(Date) from SPACEXTBL group by Mission_Outcome like "%Success%"
```

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

Mission_Outcome	count(Date)
Failure (in flight)	1
Success	100

Boosters Carried Maximum Payload

- This tables lists the names of the maximum payload mass each booster has carried

Task 8

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

```
%sql select booster_version, max(PAYLOAD_MASS__KG_) from SPACEXTBL group by booster_version
```

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

Booster_Version	max(PAYLOAD_MASS__KG_)
F9 B4 B1039.2	2647
F9 B4 B1040.2	5384
F9 B4 B1041.2	9600
F9 B4 B1043.2	6460
F9 B4 B1039.1	3310
F9 B4 B1040.1	4990
F9 B4 B1041.1	9600
F9 B4 B1042.1	3500
F9 B4 B1043.1	5000
F9 B4 B1044	6092
F9 B4 B1045.1	362
F9 B4 B1045.2	2697
F9 B5 B1046.1	3600
F9 B5 B1046.2	5800
F9 B5 B1046.3	4000
F9 B5 B1046.4	12050
F9 B5 B1047.2	5300
F9 B5 B1047.3	6500
F9 B5 B1048.2	3000
F9 B5 B1048.3	4850
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.2	9600
F9 B5 B1049.3	13620
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.6	15440
F9 B5 B1049.7	15600

2015 Launch Records

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

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, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%sql select "Landing _Outcome", booster_version, launch_site, date from SPACEXTBL where date like ("%2015") and "Landing _Outcome" = "Failure (drone ship)"
```

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

Landing _Outcome	Booster_Version	Launch_Site	Date
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	10-01-2015
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	14-04-2015

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

- This ranks 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
- date selection failed for some reason, i am not sure if date was integrated into the sql table as type date

Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%sql select count("Landing _Outcome"), "Landing _Outcome" from SPACEXTBL group by "Landing _Outcome" order by "date" desc where "date" between "04-06-2010" and "20-03-2017"
```

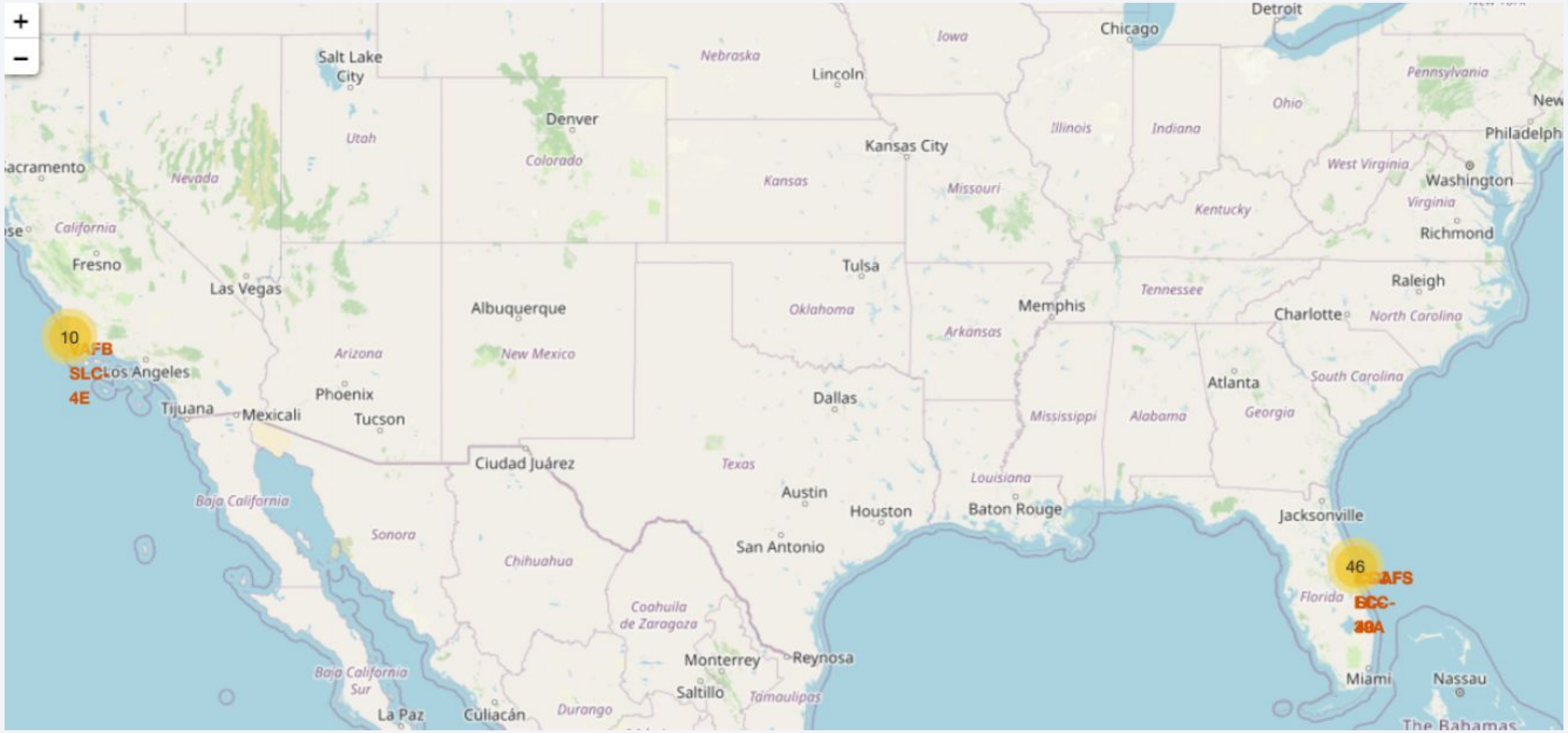
```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) near "where": syntax error
[SQL: select count("Landing _Outcome"), "Landing _Outcome" from SPACEXTBL group by "Landing _Outcome" order by "date" desc where "date" between "04-06-2010" and "20-03-2017"]
(Background on this error at: http://sqlalche.me/e/e3q8)
```

Section 3

Launch Sites Proximities Analysis

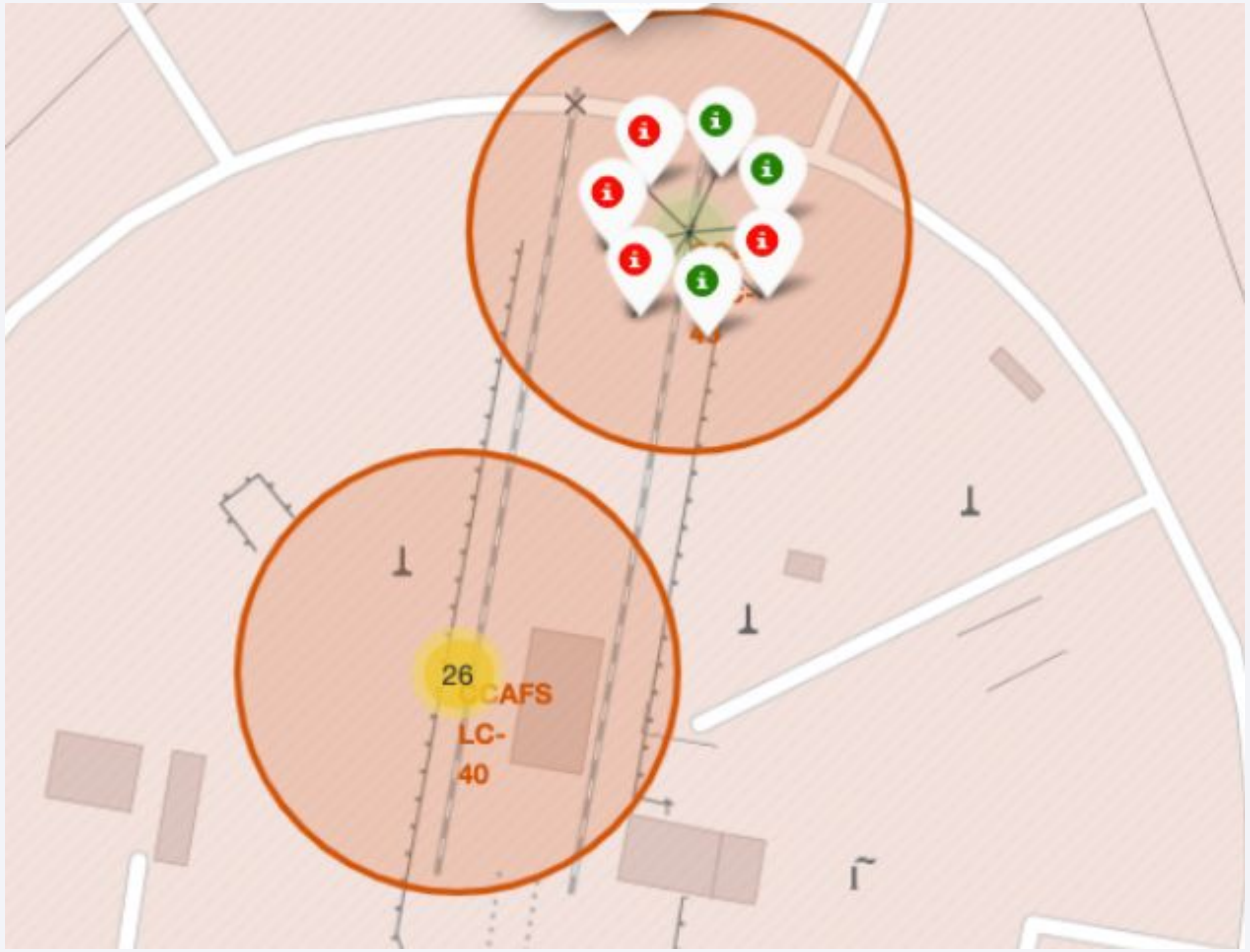
Geographical Map of Launch sites

- This geographical Map shows where the launch sites are and put their location in perspective



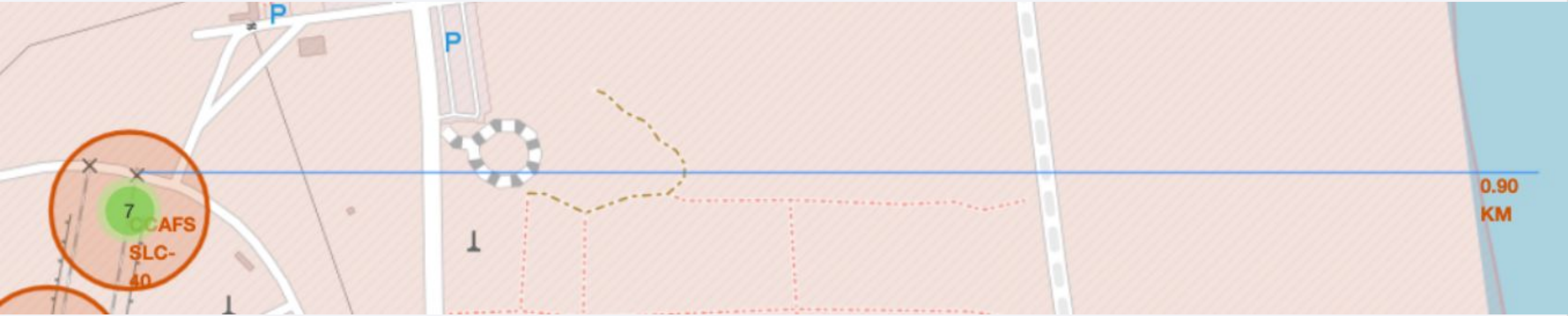
labeled launches to show success

- This map shows where launches were successful and were not through color-coding



Proximity of launch sites to landmarks

- Shoows the proximity of launch sites to landmarks such as coastline and highways to examine their influence on mission success

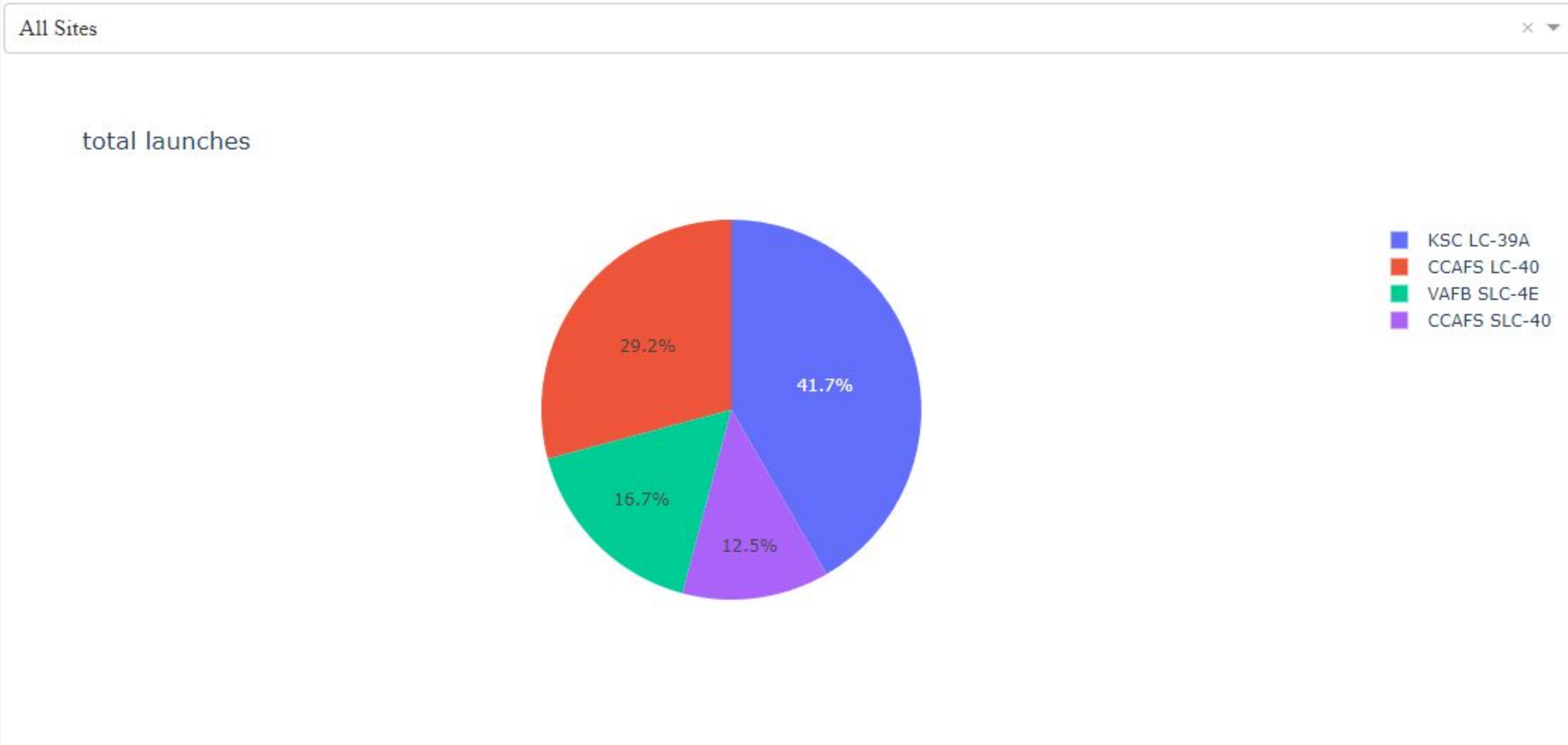


Section 4

Build a Dashboard with Plotly Dash

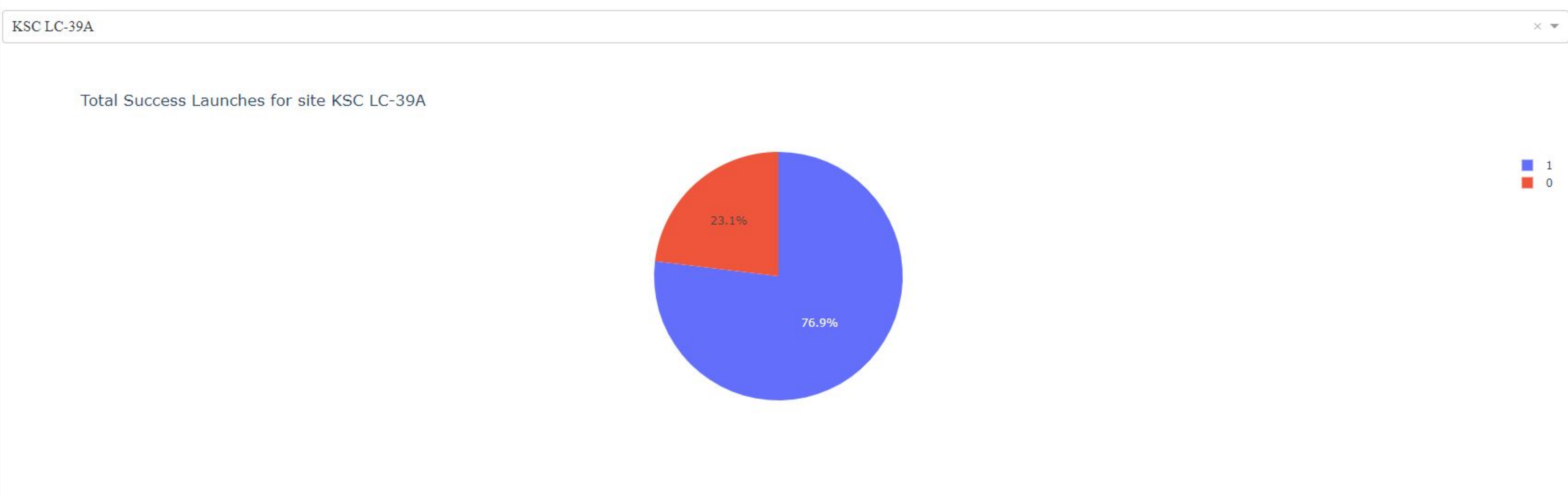
Pie Chart of number of launches at each site

- Shows the number of launches at each launch site



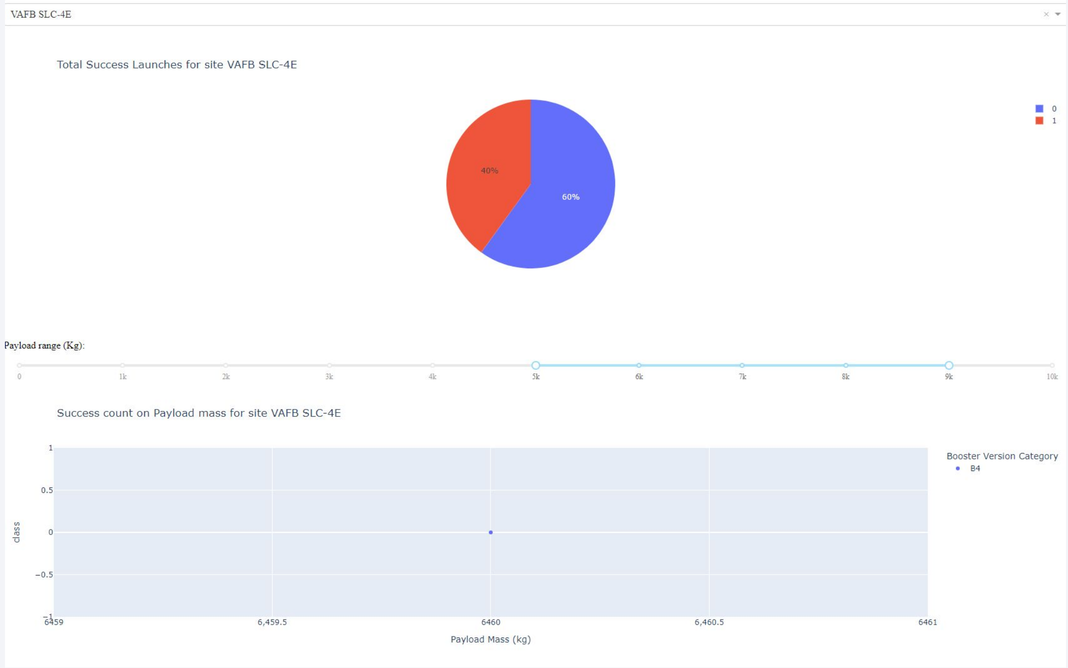
Pie Chart of mission success at selected site

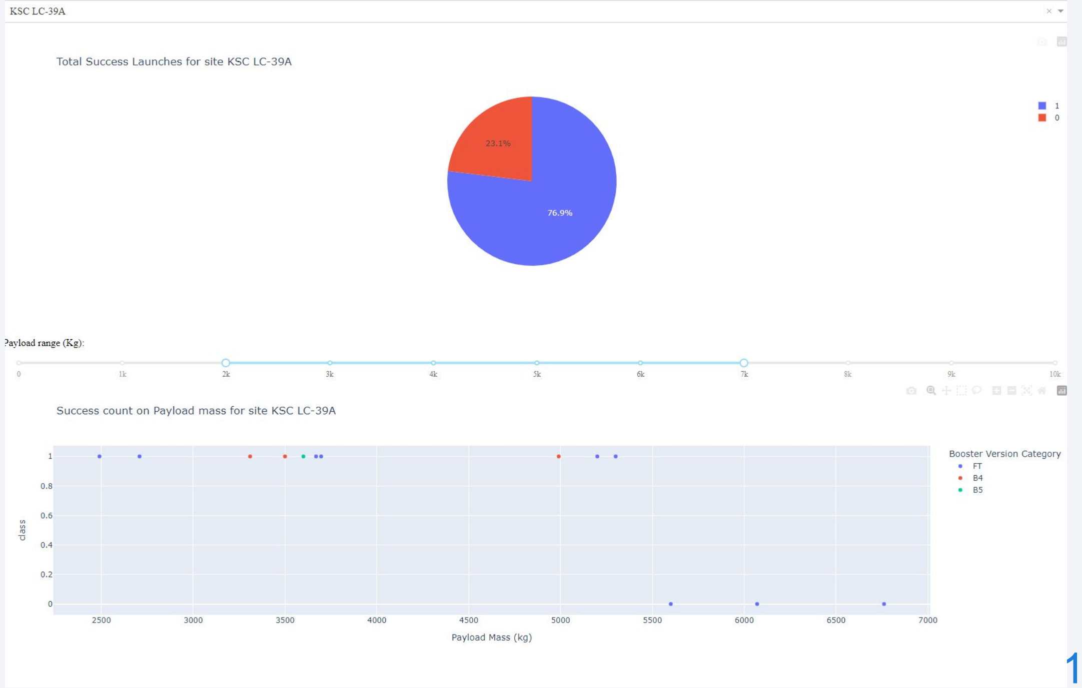
- Shows the Successrate of launches at each site, makes it possible to analyse where successful mission are probably



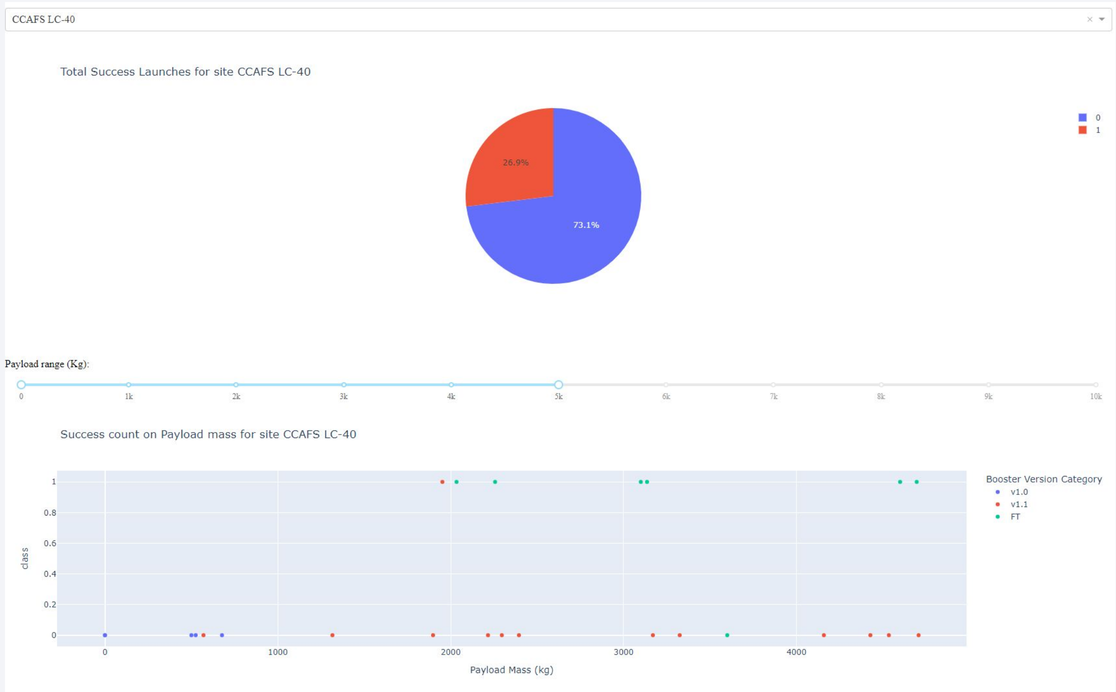
Scatter plot of success at sites and payload ranges

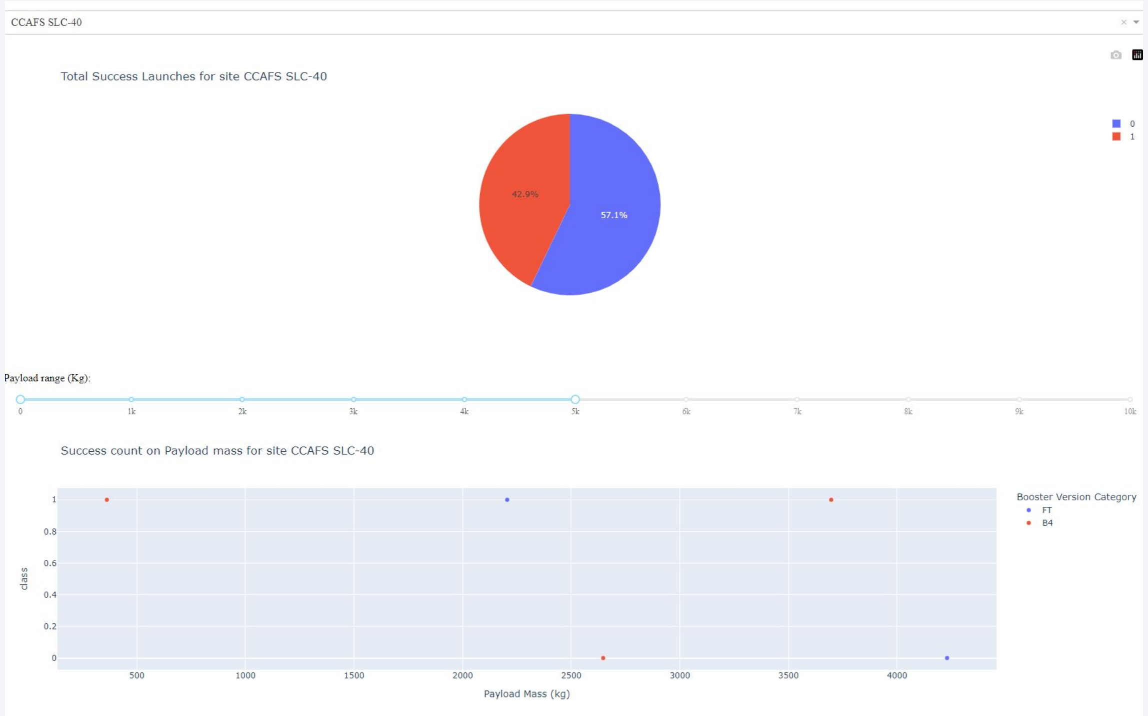
- Shows at which sites and at which payload range missions were successful or not
- Helps to analyse the influence of payload and launch site on success





Scatter plot of success at sites and payload ranges



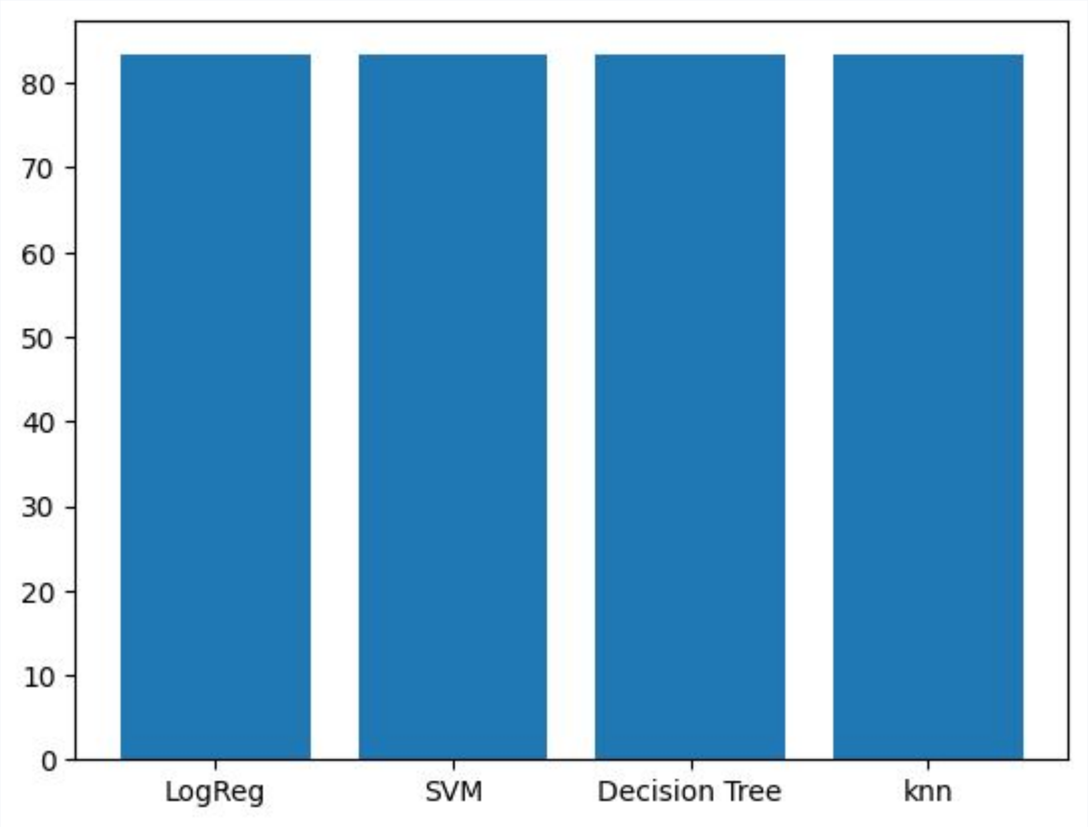


Section 5

Predictive Analysis (Classification)

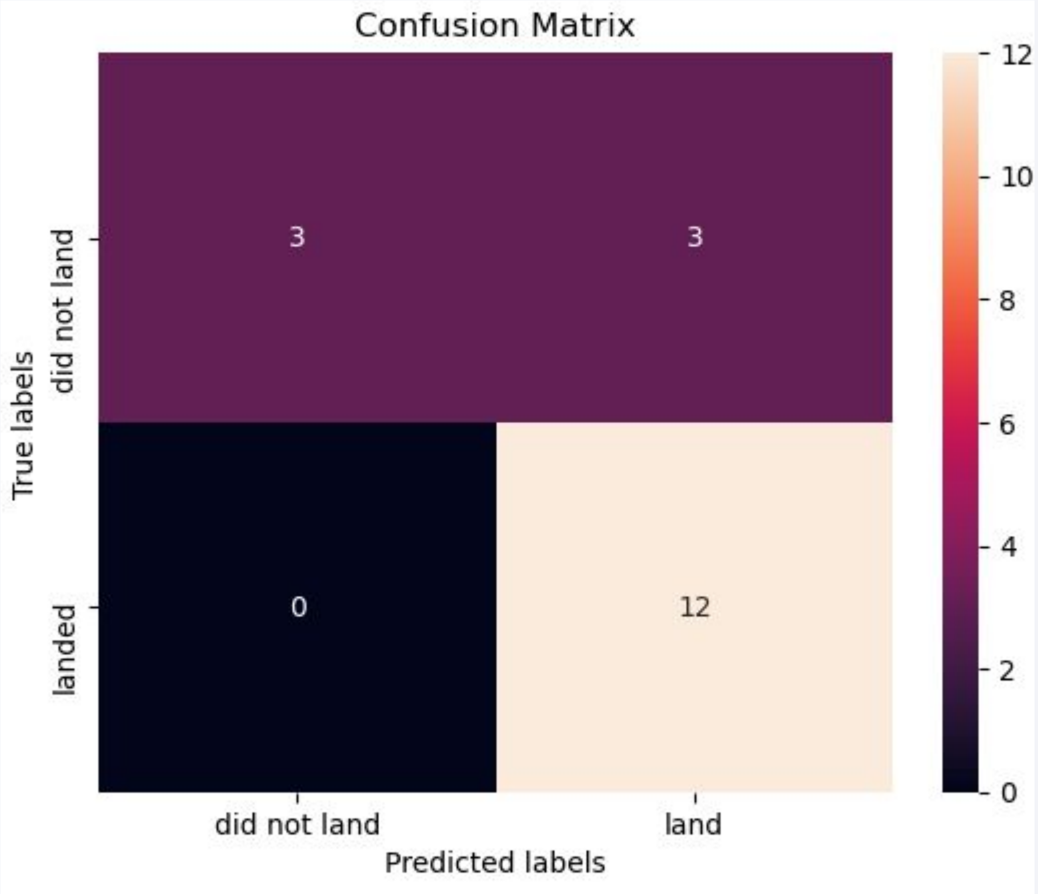
Classification Accuracy

- Each model has the same classification accuracy of 83.33%



Confusion Matrix

- This model predicted 3 cases correctly were the first rocket section did not land
- It also predicted 12 cases correctly were it did land
- 3 cases were predicted wrong and did not land, even though the model predicted, that the rocket will land



Conclusions

- Launch success is primarily influenced by
  - Launchsite
  - Payload
  - Orbit-type
  - Year/ Flight number
- All models performs the same
- Probably more data is needed to improve results
- Accuracy of 83% can be achieved

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!