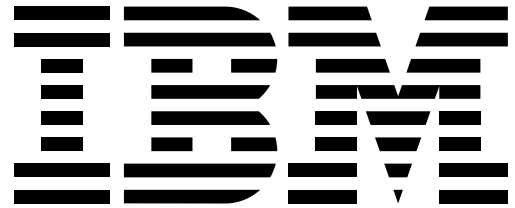
DATA SCIENCE CAPTSONE PROJECT



Javier Pérez Aguirre

Outline

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

Executive Summary

- SpaceX reuses the first stage booster to lower the total cost of launching a payload to orbit.
- The goal is to predict whether if the landing is going to be successful or not using different features such as orbit type and payload mass.
- It was found that as more launches are done, the higher is the success rate of the booster.
- The best algorithm to predict the landing outcome of the rocker booster was Decision Tree.

Introduction

- Falcon 9 is the world's first orbital class reusable rocket. Reusability allows SpaceX to refly the most expensive parts of the rocket, which in turn drives down the cost of space access.
- The rocket has been used to launch a wide variety of payloads, including satellites, the Dragon spacecraft, and the Crew Dragon spacecraft. It has also been used to launch the first commercial crewed spacecraft to the International Space Station.
- They key objective of this project is to analyze and predict whether If the first stage is going to land successfully, using different features such as orbit and payload mass.

METHODOLOGY

Methodology

Data collection methodology:

- SpaceX API
- Web Scraping from Wikipedia

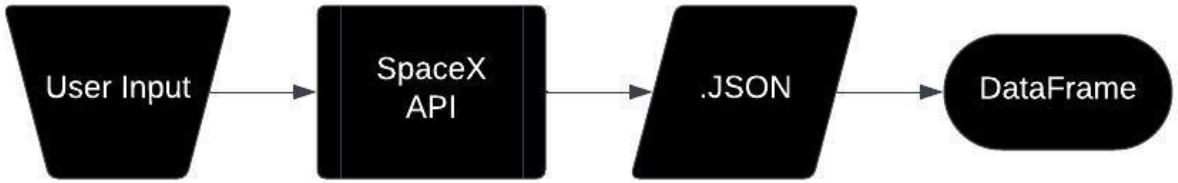
Perform data wrangling

- Dropping unnecessary columns
- One Hot Encoding categorical variables

- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Perform interactive visual analytics using Folium and Plotly Dash**
- **Perform predictive analysis using classification models**

Data Collection

- Data was collected using two methods: SpaceX API and Web scrapping
  - **SpaceX API:**



  - **Web scrapping:**

Data Collection – SpaceX API

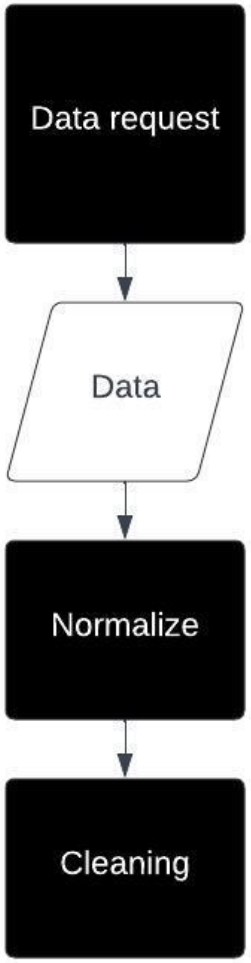
- Data request to SpaceX API:

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

- Normalize data into a data frame:

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

- Data cleaning: Replacing NaN values with means for the Payload Mass column
- Github: https://github.com/crimpydude/Capstone-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Assign response to an object:

```
request = requests.get(static_url)
```

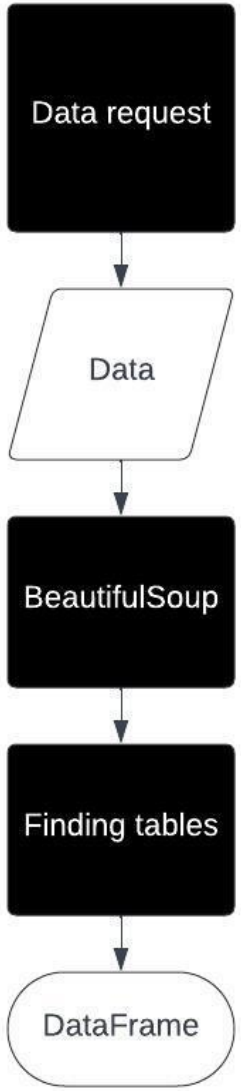
- Create BeautifulSoup object:

```
soup = BeautifulSoup(request.text)
```

- Extract tables:

```
html_tables = soup.find_all('table')
```

- Github: https://github.com/crimpydude/Capstone-Project/blob/main/jupyter-la webscraping.ipynb



Data Wrangling

- The number of launches is calculated for each site:

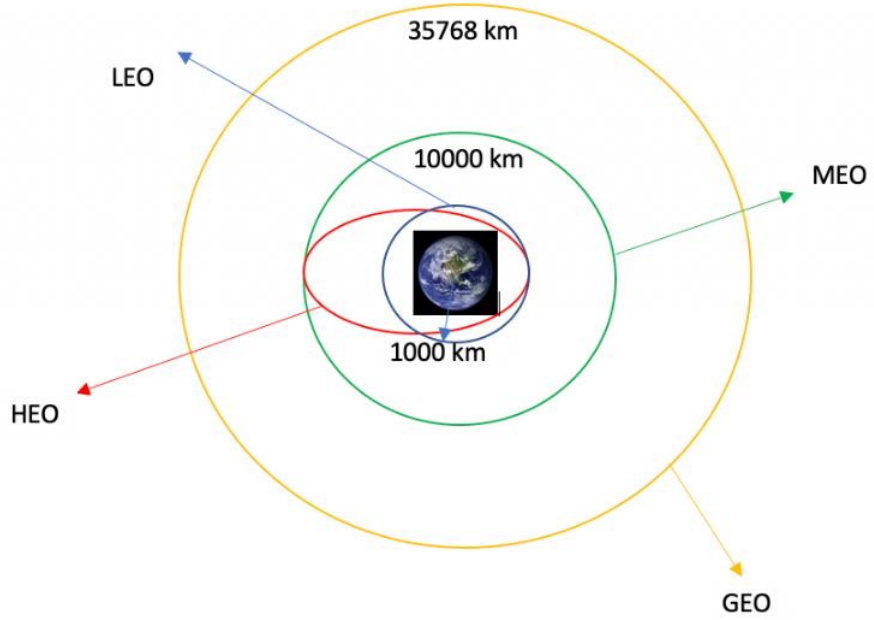
```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

- Number and occurrence for each orbit is calculated:

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

- Number and occurrence of mission outcomes for each orbit:

```
landing_outcomes = df['Outcome'].value_counts()
```



- A new column "Class" is created. Successful landings are labeled as 1 and 0 for unsuccessful landings.
- Github: https://github.com/crimpydude/Capstone-Project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Scatter Plots:

- Payload Mass vs. Flight Number
- Launch Site vs. Flight Number
- Launch Site vs. Launch Site
- Orbit vs. Flight Number
- Orbit vs. Payload Mass

Bar Plots:

- Class mean vs. Orbit

Line Plots:

- Class vs. Year

- Github: https://github.com/crimpydude/Capstone-Project/blob/main/jupyter-labs-eda-dataviz.ipynb

Scatter plots show the relationship or correlation between the two variables.

Bar plots make a visually easy comparison between many variables at the same time.

And line plots are useful to identify trends of a variable over time.

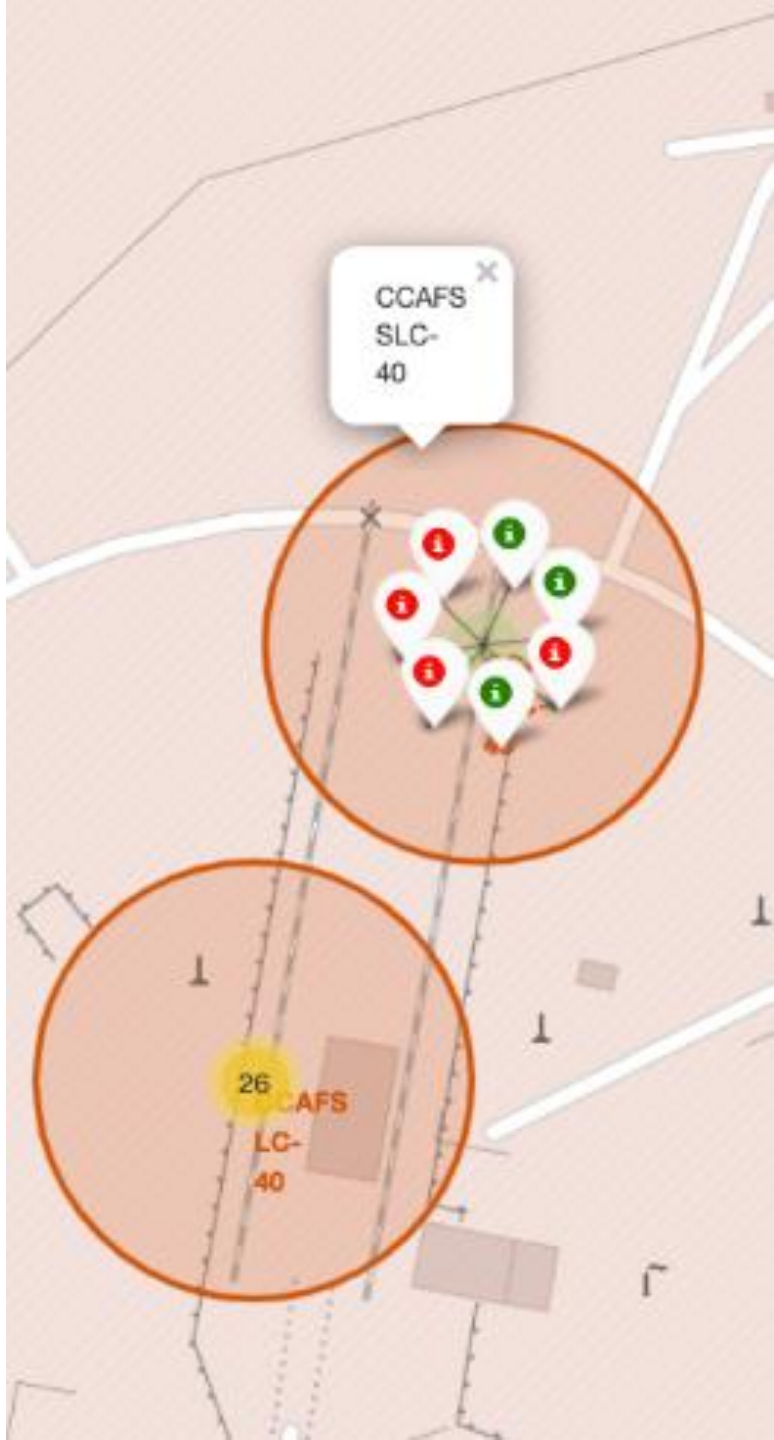
EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- 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 date when the first successful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- 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
- 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

- Github: https://github.com/crimpydude/Capstone-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Launch sites were marked in a map to visually identify their location.
- Green and red markers are used to represent the outcome for missions, where green is successful and red unsuccessful.
- Different distances are calculated from launch sites and landmarks in the surroundings.
- Github: https://github.com/crimpydude/Capstone-Project/blob/main/lab_jupyter_launch_site_location.ipynb



Build a Dashboard with Plotly Dash

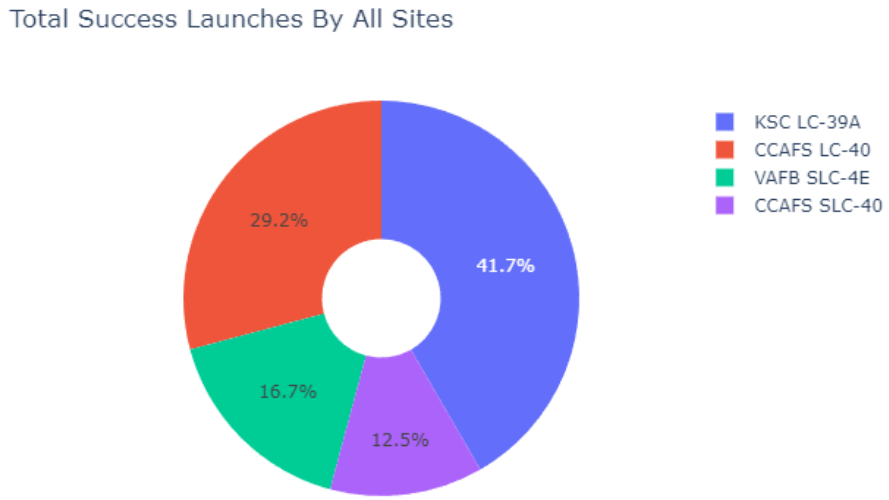
The following plots were added to the dash:

- Pie Chart of total launches by launch site
- Scatter plot of Class vs. Payload Mass for different Booster

Pie Charts allow to visualize the outcome of missions per launch site.

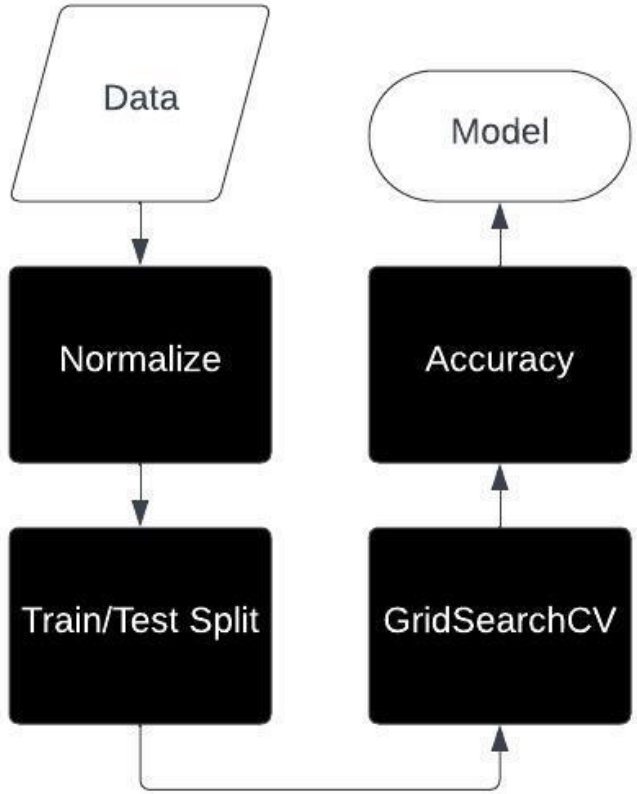
The scatter plot helps to understand the relationship between the two variables.

- Github: https://github.com/crimpydude/Capstone-Project/blob/main/spacex_dash_app.py



Predictive Analysis (Classification)

- Data is normalized to avoid problems of data scale.
- Dataset is divided into train and test set for model training and evaluation.
- Once a model is chosen, a GridSearchCV object is created for parameter tunning.
- Accuracy of the model is evaluated using the test set
- Github: https://github.com/crimpydude/Capstone-Project/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb



Results

EXPLORATORY DATA ANALYSIS RESULTS

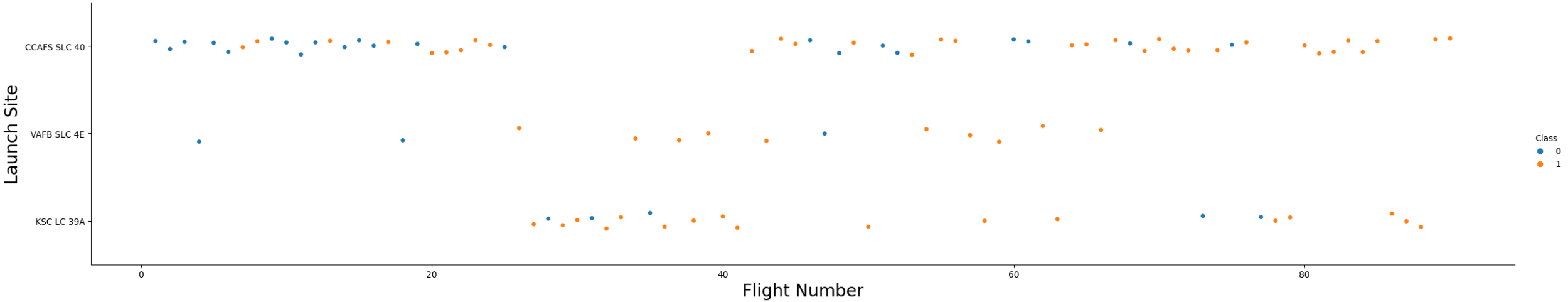
INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS

PREDICTIVE ANALYSIS RESULTS

INSIGHTS DRAWN FROM EDA

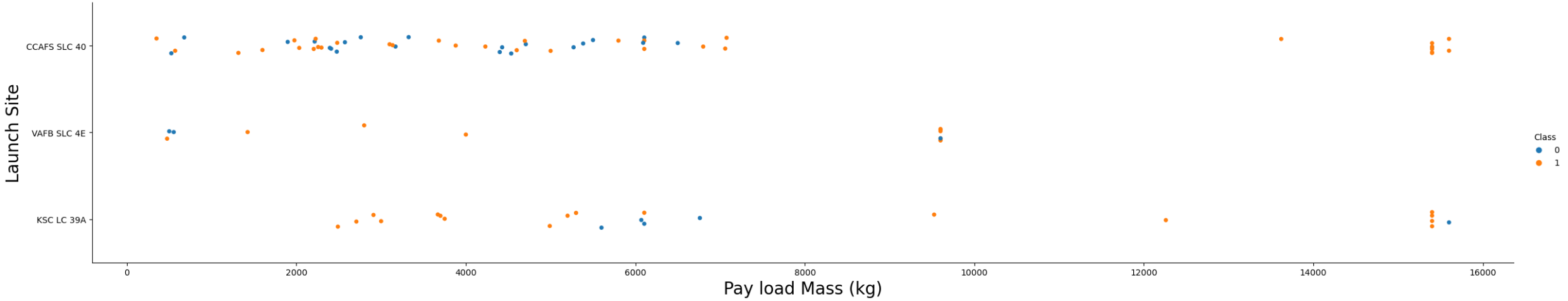
Flight Number vs. Launch Site

- As the flight number increases, the successful landings are more common.



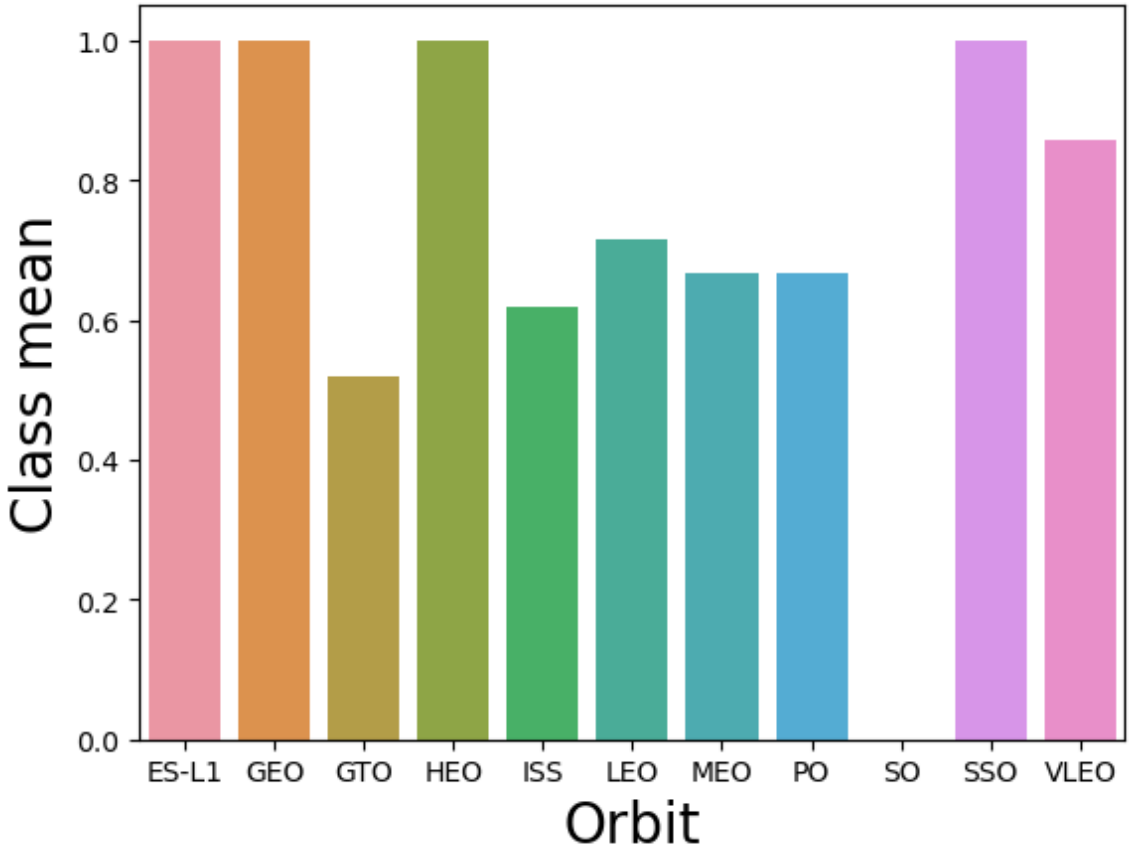
Payload vs. Launch Site

- There is no apparent correlation for lighter Payloads but for heavier Payloads the success rate increases.



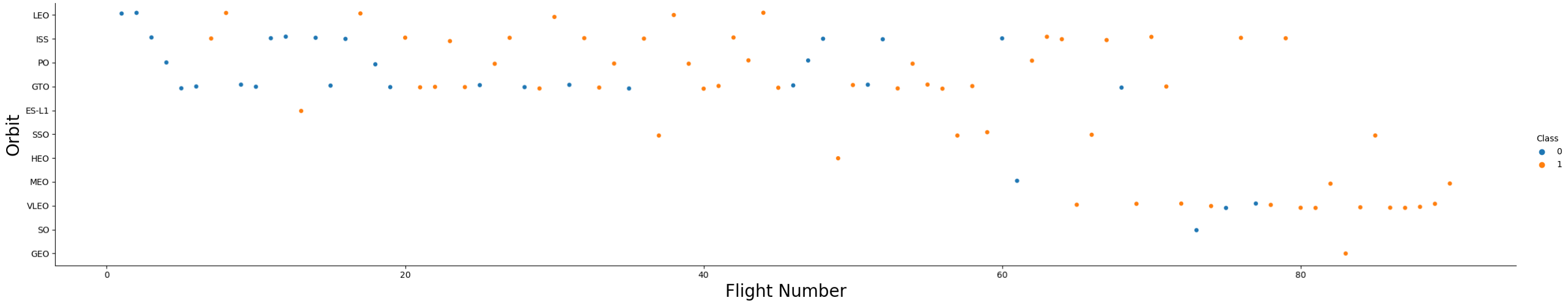
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO have a 100% of success rate. Meanwhile SO has a 0%.



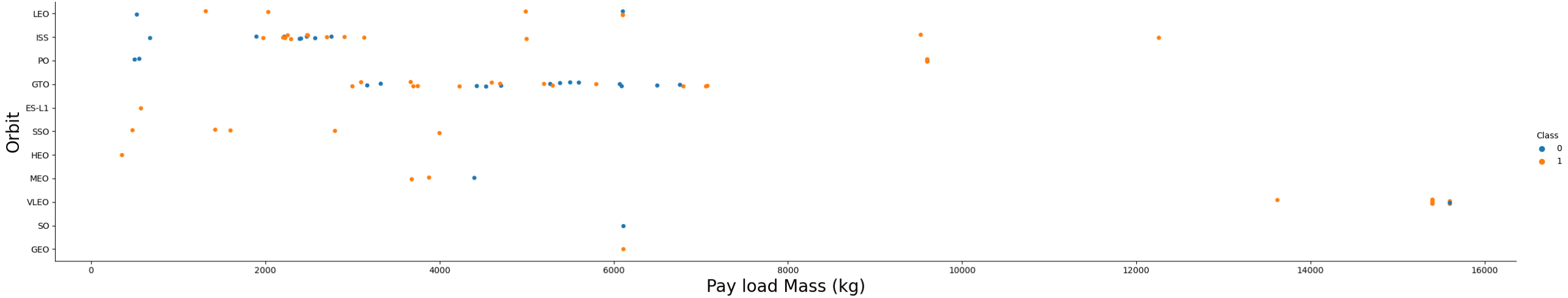
Flight Number vs. Orbit Type

- In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



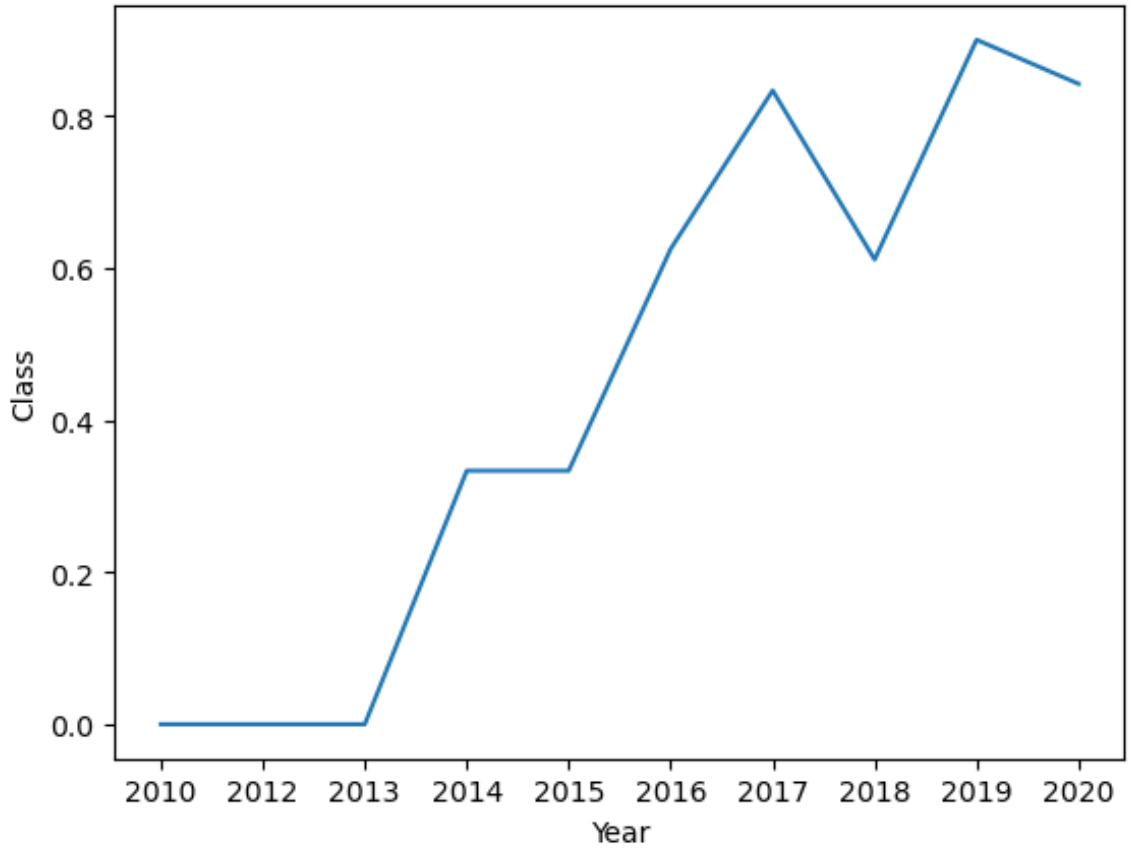
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS. However for GTO cannot be distinguished this well as both positive landing rate and negative landing (unsuccessful mission) are both there here.



Launch Success Yearly Trend

- Success rate increases from 2013 to 2020, having a small decline in 2018.



All Launch Site Names

- Query: `%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL`
- The statement DISTINCT is used to get unique results from the data

Launch Site Names Begin with 'CCA'

- Query: `%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5`
- The statement LIMIT is used to limit the results to 5.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Query: `%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'`

- Data is filtered with a WHERE statement.

	Total Payload Mass
0	45596

Average Payload Mass by F9 v1.1

- Query: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1'

- A WHERE statement is used to filter data for booster F9 v1.1

AVG(PAYLOAD_MASS__KG_)
6138.287128712871

First Successful Ground Landing Date

- Query: `%sql SELECT MIN(DATE) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)' GROUP BY DATE`

- Here data is grouped by date and filtered by landing outcome. The first successful landing corresponds to the date 22/12/2015.

MIN(DATE)
01/08/2018
05/01/2017
06/03/2017
09/07/2017
14/08/2017
15/12/2017
18/07/2016
19/02/2017
22/12/2015

Successful Drone Ship Landing with Payload between 4000 and 6000

- Query: %sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000

- Data is filtered using a BETWEEN statement for PAYLOAD_MASS__KG_

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

Total Number of Successful and Failure Mission Outcomes

- Query: `%sql SELECT Mission_Outcome , COUNT(*) FROM SPACEXTBL GROUP BY Mission_Outcome`

- Here the different Mission Outcomes is displayed, for a total of a 100 successful launches and 1 as a failure.

Mission_Outcome	COUNT(*)
None	898
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Query: `%sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)`

- A subquery is used to select the maximum payload: `SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL`

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

- Query: %sql SELECT Date, Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTBL WHERE Landing_Outcome = 'Failure (drone ship)' AND Date LIKE '%/2015'

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

- Data is filtered where the landing outcome is faliure for Drone Ship within the year 2015.

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

- Query: `%sql SELECT COUNT(*), Landing_Outcome FROM SPACEXTBL GROUP BY Landing_Outcome ORDER BY COUNT(*) DESC`

- Data is ordered by `COUNT(*) DESC` to get a descending list.

COUNT(*)	Landing_Outcome
898	None
38	Success
21	No attempt
14	Success (drone ship)
9	Success (ground pad)
5	Failure (drone ship)
5	Controlled (ocean)
3	Failure
2	Uncontrolled (ocean)
2	Failure (parachute)
1	Precluded (drone ship)
1	No attempt

LAUNCH SITES PROXIMITIES ANALYSIS

World Map and Launch Sites

- Launch sites are located in the United States in both West and East coasts. All sites are located close to the equator.

Colored labeled markers

- Markers where assigned for different outcomes of missions where green is successful and red is failure.

Launch Site distance from Coastline

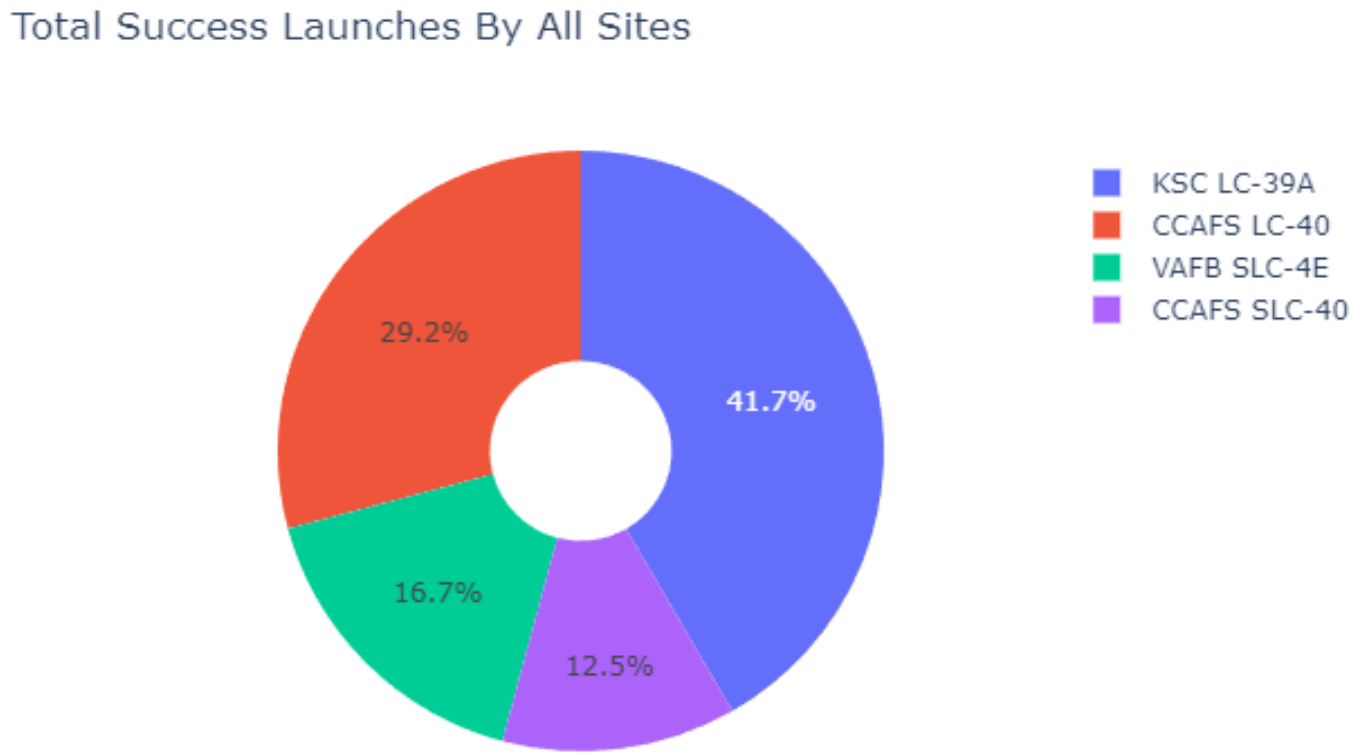
- Distances from the launch site to different landmarks is calculated.
- In this slide, the distance from the coastline is shown.

BUILD A DASHBOARD WITH PLOTLY DASH

Success launches by all Sites

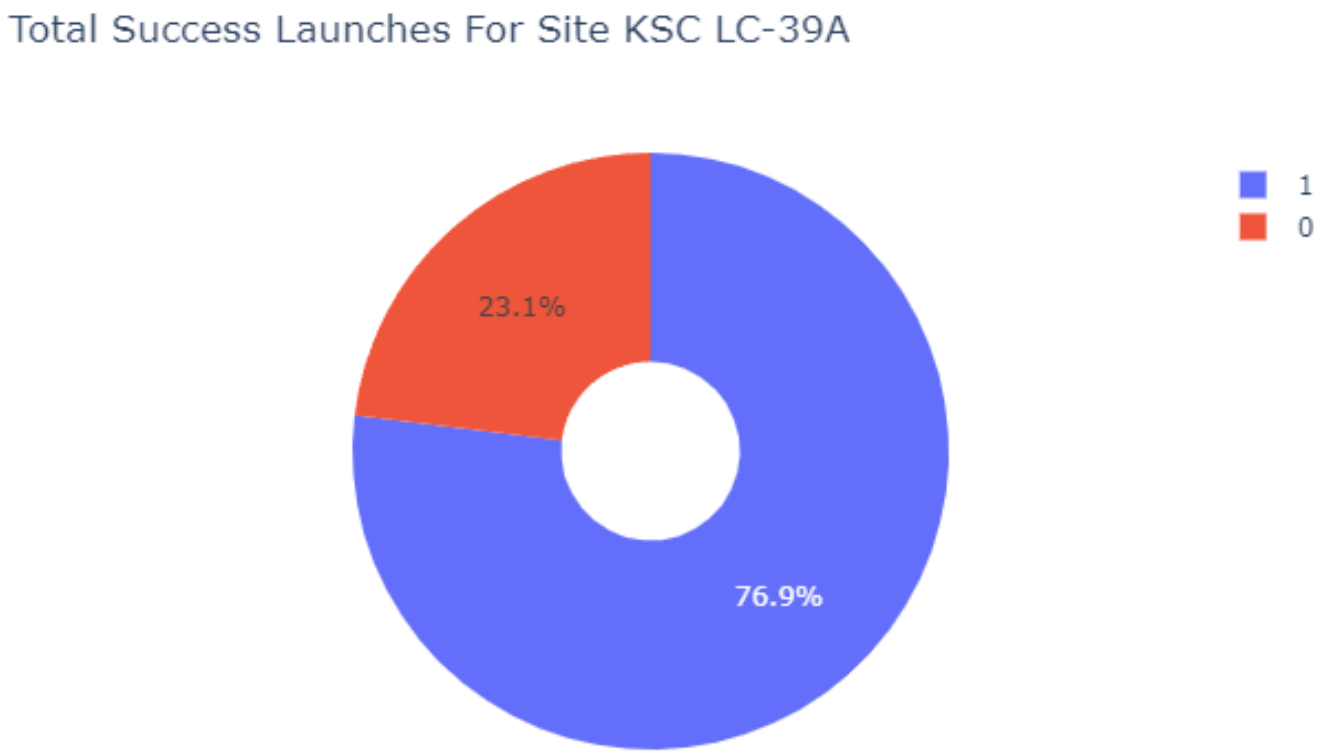
KSC LC-39A is

the most successful site.



Site with highest success ratio

KSC LC 39-A has a 76.9% success rate.



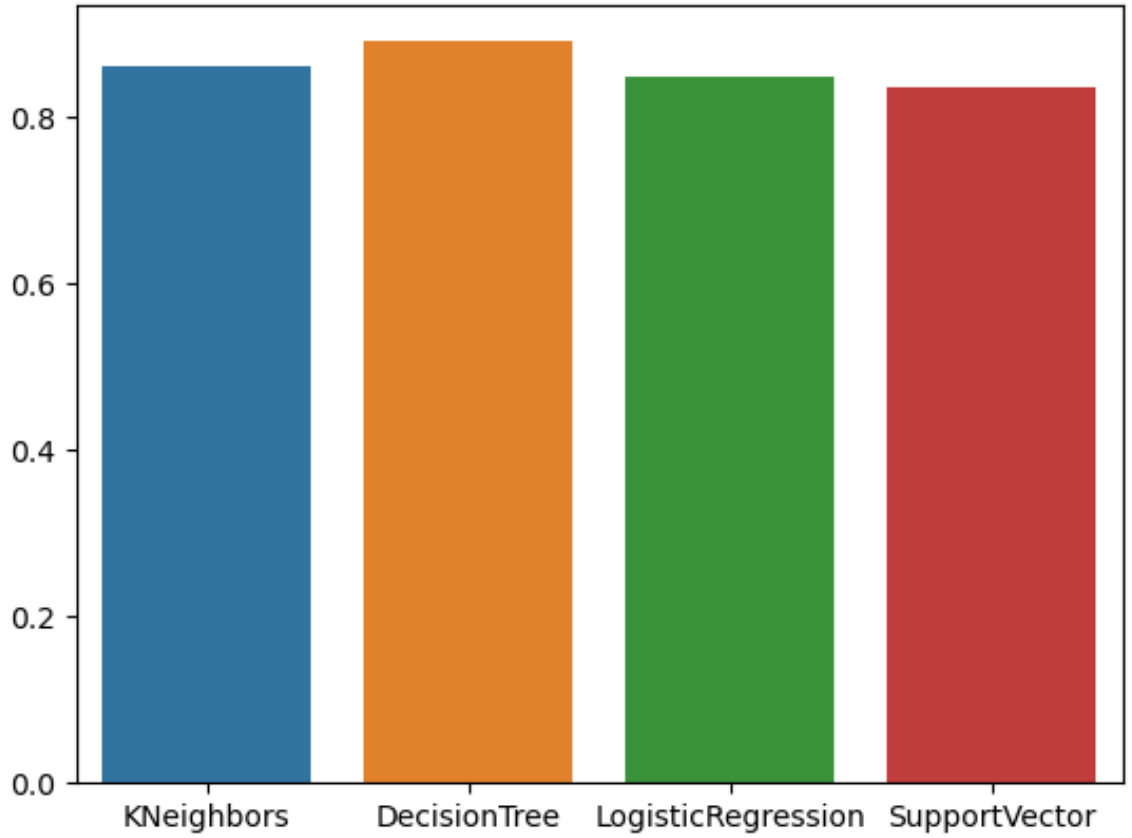
Class vs. Payload Mass

- Payloads between 2000 and 6000 [kg] success rates.

PREDICTIVE ANALYSIS - CLASSIFICATION

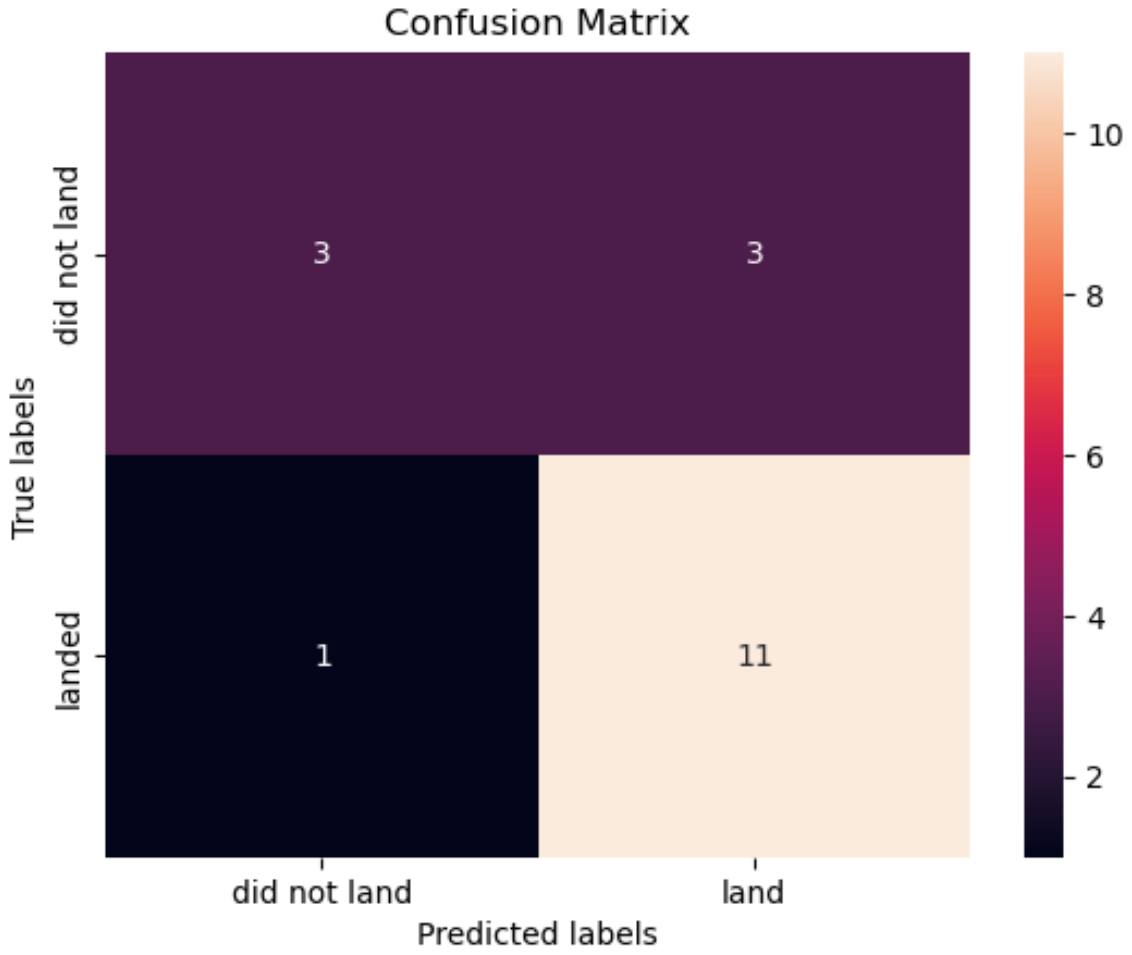
Classification Accuracy

- Decision Tree has the highest accuracy with 89% approximately.



Confusion Matrix for Decision Tree

- For this algorithm 4 entries were labeled wrong: 1 as did not land but landed and 3 that landed and did not land.



Conclusions

- As more launches happen, the success rate increases.
- KSC LC-39A is the most successful site.
- Success rate increases from 2013 to 2020, having a small decline in 2018.
- ES-L1, GEO, HEO and SSO have a 100% of success rate. Meanwhile SO has a 0%.
- Decision Tree was the best algorithm for the dataset.

Appendix

- The bar plot for the accuracy of the models was not part of the tasks. The following code will perform the plot:

```
models = {'KNeighbors':knn_cv.best_score_,
                'DecisionTree':tree_cv.best_score_,
                'LogisticRegression':logreg_cv.best_score_,
                'SupportVector': svm_cv.best_score_}


   algo = max(models, key=models.get)
   print('Best model is', algo,'with a score of', models[algo])


   sns.barplot(pd.DataFrame(models, index = [0]))
```