Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

Goal of this capstone project is to predict whether the SpaceX Falcon 9 first stage will land successfully. This is crucial because if we can determine if the first stage will land, we can determine the cost of a launch. In order to achieve this following methodologies were used:

- Data Collection through API
- Data Collection with Web Scraping
- Data Wrangling
- Exploratory Data Analysis with SQL
- Exploratory Data Analysis with Data Visualization
- Interactive Visual Analytics with Folium
- Machine Learning Prediction

Results of the analysis show that there are some features of rocket launches that have a correlation with the successfull outcome of the launches.
Final conclusion is that Decision Tree may be the best machine learning algorithm for this problem.

Introduction

- Project background and context

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. This goal of the project is we will predict if the Falcon 9 first stage will land successfully.

- Problems you want to find answers

The launch success rate may depend on many factors such as orbit type, payload mass, etc. Question to be answered by this project is: for a given set of features will the first stage of rocket launch finish successfully?

Section 1

Methodology

Methodology

Executive Summary

- Data collection was performed in two ways: requesting data from the SpaceX API and webscrapinglaunch data from a Wikipedia page
- Data wrangling was performed to transform and clean the data using Python's pandas library
- Exploratory data analysis (EDA) was then performed on clean data using visualization tools such as Python's matplotlib and seaborn libraries, and some questions were answered using SQL queries.
- Visualization of results was performed using Folium for creating maps while Plotly Dash was used to create interactive data visualizations.
- Predictive analysis was performed using four different machine learning classification models (logistic regression, support vector machines, k-nearest neighbour and decision tree classifier). Each model was trained, tuned and evaluated to find the best one.

Data Collection

The data was collected using following methods:

- Data collection was done using get request to the SpaceX API. Then, the response content was decoded as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize(). Data were afterwards cleaned, checked for missing values that were filled where necessary.
- Also, data collection was performed using web scraping from Wikipedia (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches) for Falcon 9 launch records with BeautifulSoup. Launch records were extracted as HTML table, table was parsed and converted to pandas dataframe needed for further analysis.

Data Collection – SpaceX API

- Get request was used to the SpaceX API to collect data, data were normalized, cleaned and some data wrangling and formatting was performed (extracting data only for Falcon 9 and replacing missing values for PayloadMass with a mean value).

- For details check GitHub URL: https://github.com/mbacalja/SpaceX-Capstone-project/blob/main/Data-collection-api.ipynb

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_
```

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

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

```
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, payload_mean)
data_falcon9.isnull().sum()
```

Data Collection - Scraping

- Web scrapping was done with BeautifulSoup and Falcon 9 launch records HTML table from Wikipedia was extracted.
- Table with Falcon 9 launch data was parsed and converted into a pandas dataframe
- For details check GitHub URL: https://github.com/mbacalja/SpaceX-Capstone-project/blob/main/Data-collection-webscraping.ipynb

```
# use requests.get() method with the provided static_url
response = requests.get(static_url)
```

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
BeautifulSoup = BeautifulSoup(response.text, "html5lib")
```

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = BeautifulSoup.find_all("table")
```

Starting from the third table is our target table contains the actual launch records.

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
df.head()
```

Data Wrangling

- Exploratory data analysis was performed and determined the training labels to find some patterns in the data and determine what would be the label for training supervised models.
- Missing values were found and categorical and numerical columns were identified
- Outcomes were converted using one-hot encoding (successful (1) and unsuccessful (0))
- Number of launches on each site, number and occurrence of each orbit number and occurence of mission outcome per orbit type were calculated
- We created landing outcome label from outcome column and exported the results to csv.
- For details check GitHub URL: https://github.com/mbacalja/SpaceX-Capstone-project/blob/main/Data%20wrangling.ipynb

EDA with Data Visualization

Exploratory Data Analysis and Feature Engineering were performed using Pandas and Matplotlib

- Scatter plots were used to represent the relationship between two variables. Differentsets of features were compared such as Flight Number vs. Launch Site, Payload vs. LaunchSite, FlightNumber vs. Orbit Type and Payload vs. Orbit Type.
- Bar charts was used to compare Success Rate for different Orbit Types
- Line charts are useful for showing data trends over time and in this project was used to show Success Rate over years.

- For details check GitHub URL: https://github.com/mbacalja/SpaceX-Capstone-project/blob/main/EDA-with-Data-Visualization.ipynb

EDA with SQL

- Data was also explored using SQL queries. Some of SQL queries performed on dataset are listed below:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- For details check GitHub URL: https://github.com/mbacalja/SpaceX-Capstone-project/blob/main/EDA-with-SQL.ipynb

Build an Interactive Map with Folium

- All launch sites were identified on the map, and map objects such as markers, circles are added to mark the success or failure of launches for each site on the folium map.
- For each launch site launch outcomes feature was displayed (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success using the color-labeled marker clusters
- Distances between a launch site and its proximities was calculated in order to answer following questions:
  - Are launch sites near railways, highways and coastlines?
  - Do launch sites keep certain distance away from cities?
- For details check GitHub URL: https://github.com/mbacalja/SpaceX-Capstone-project/blob/main/Visual-Analytics-with-Folium.ipynb

Build a Dashboard with Plotly Dash

Interactive dashboard created with Plotly dash contains two charts:

- A pie chart that shows the successful launch by each site, it shows distribution of landing outcomes across all launch sites or shows the success rate of launches for a specific site
- A scatter chart that shows the relationship between landing outcomes an the payloadmass of different boosters. The dashboard takes two inputs, namely the site(s) and payload mass and shows how different variables affect the landing outcomes.
- For details check GitHub URL: https://github.com/mbacalja/SpaceX-Capstone-project/blob/main/spacex_dash_app.py

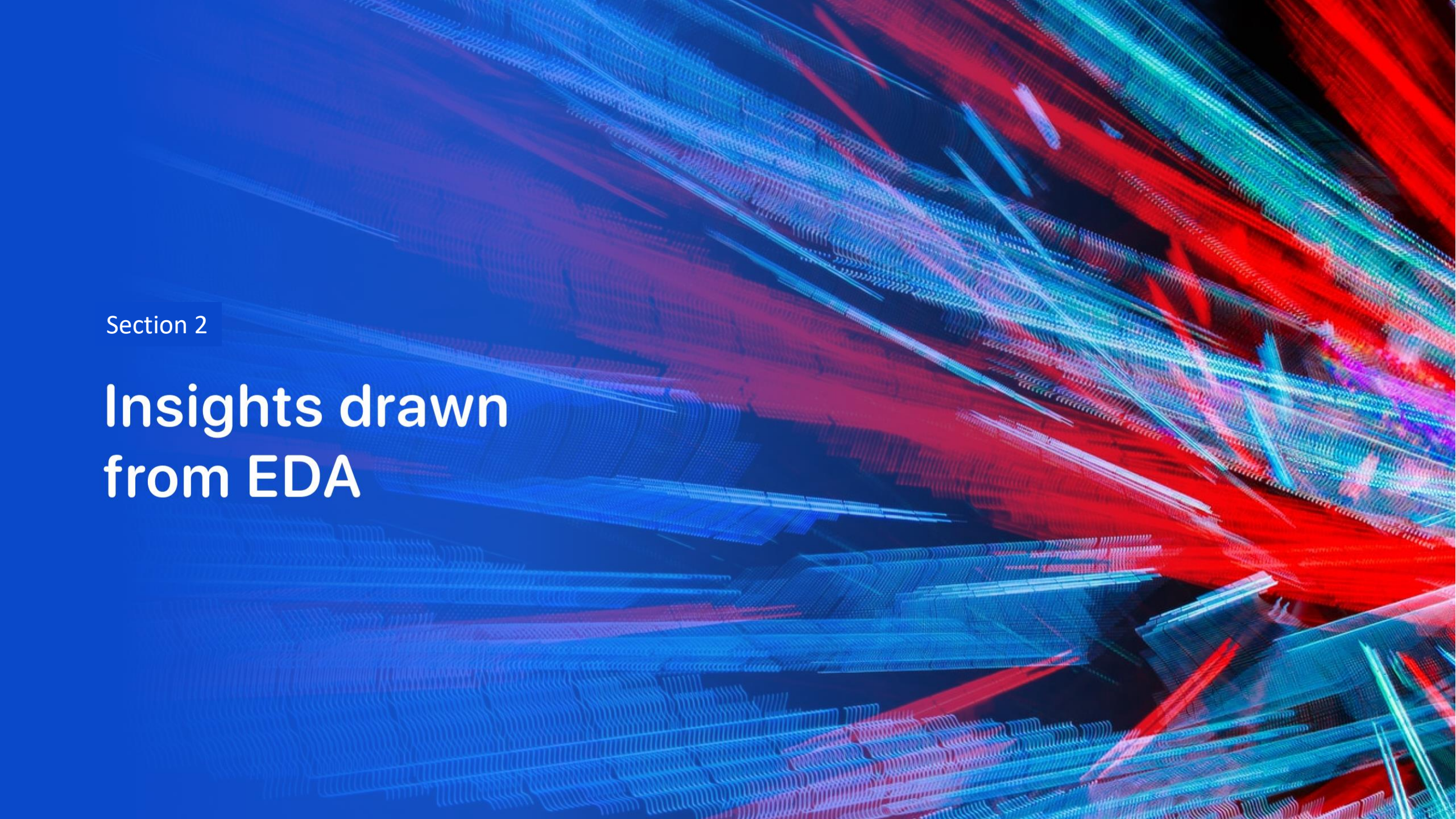
Predictive Analysis (Classification)

- Data was loaded using numpy and pandas and then transformed and split into training and testing data
- . Best Hyperparameter for SVM, Decision Trees, K-Nearest Neighbours and Logistic Regression were found
- Test data was used to evaluate models based on their accuracy scores and confusion matrix
- Best performing classification model was found.

- For details check GitHub URL: https://github.com/mbacalja/SpaceX-Capstone-project/blob/main/SpaceX_Machine%20Learning%20Prediction.ipynb

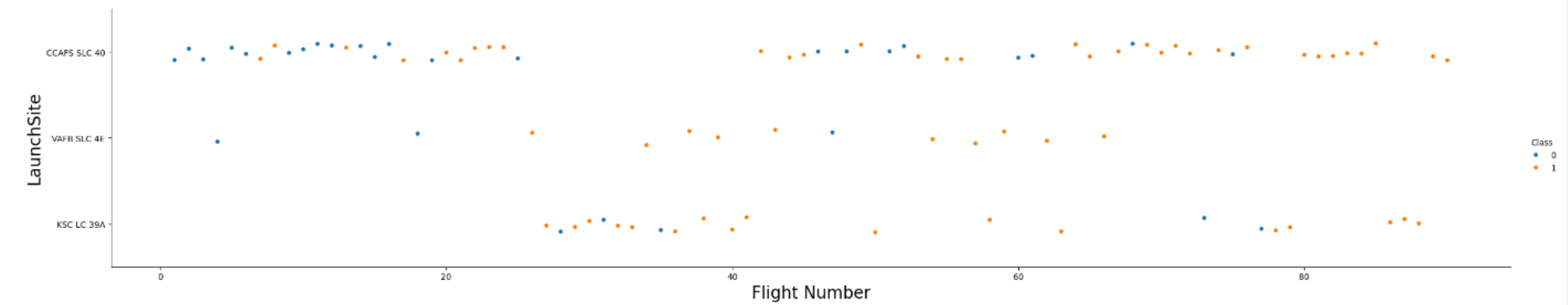
Results

- Exploratory data analysis results:
  - There is a positive correlation between number of flights and success rate
  - Success rate improved over the years
  - The most successful launches were for orbits: SSO, HEO, GEO, and ES-L1
- Interactive analytics demo in screenshots
  - Lighter payloads generally have higher success rate than the heavier payloads.
- • Predictive analysis results:
  - The results of the exploratory data analysis revealed that the success rate of the Falcon9 landings was 66.66%
  - The predictive analysis results showed that the Decision Tree algorithm was the best classification method


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

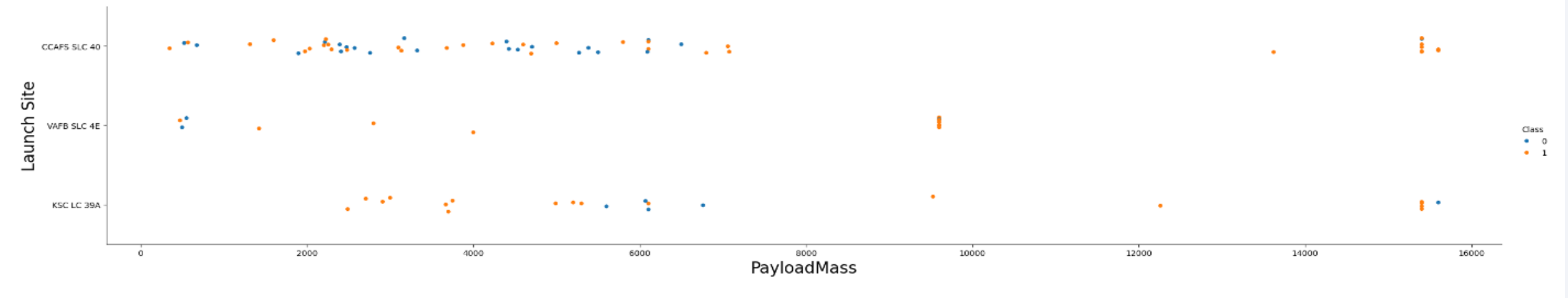
- This figure shows:
  - Success rate increased as the number of flights increased.
  - Increase in successful flights can be seen after the 40th launch



Legend: Blue dot – unsuccessful launch / Red dot – successful launch

Payload vs. Launch Site

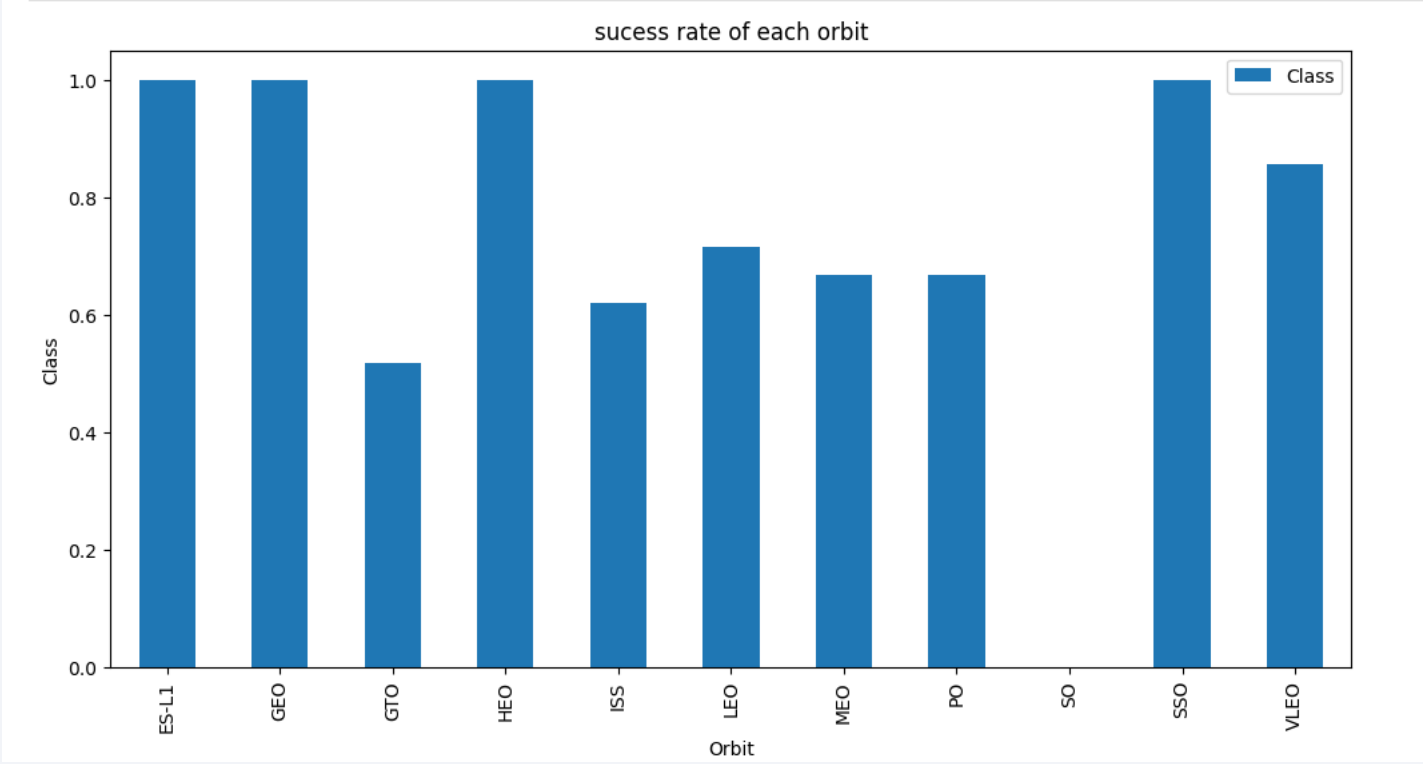
- For the VAFB-SLC launch site there are no rockets launched for heavy Payload Mass
- There seems to be a weak correlation between Payload Mass and Launch Site and therefore this relationship can not be used for decision making.



Legend: Blue dot – unsuccessful launch / Red dot – successful launch

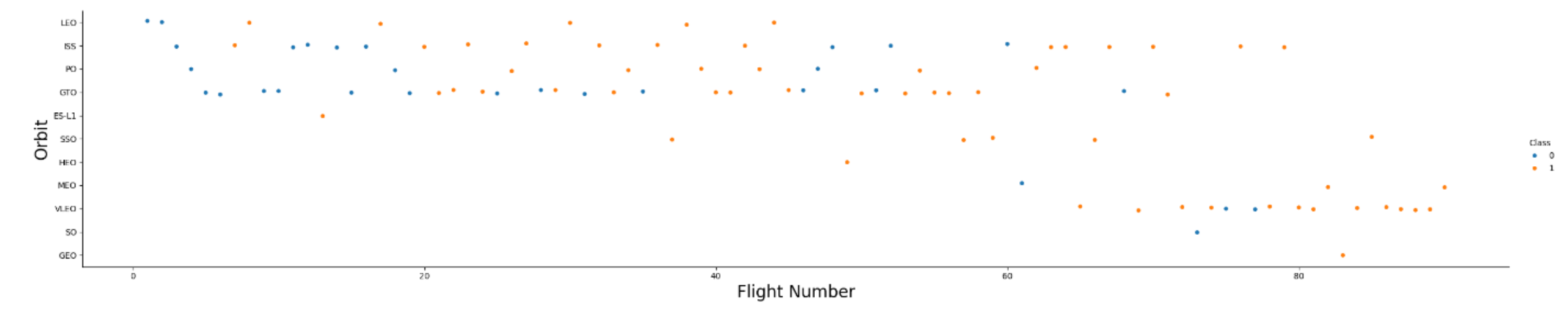
Success Rate vs. Orbit Type

- Figure shows that orbits ES-L1, GEO, HEO, SSO, VLEO had the 100% success rate.
- Orbit SO did not have any successful launches



Flight Number vs. Orbit Type

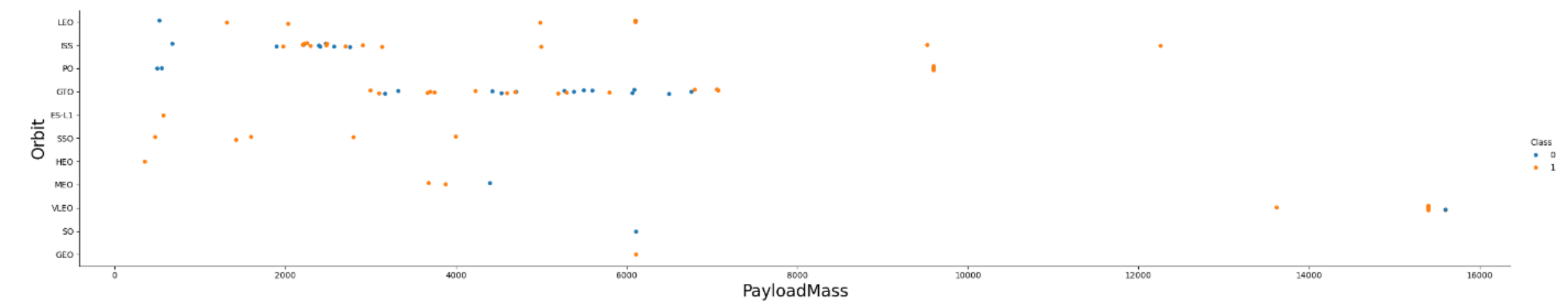
- The figure below that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.
- The SSO orbit has a 100% success rate, although with fewer flights than the other orbits
- Flight numbers greater than 40 have higher success rate than flight numbers below 40.



Legend: Blue dot – unsuccessful launch / Red dot – successful launch

Payload vs. Orbit Type

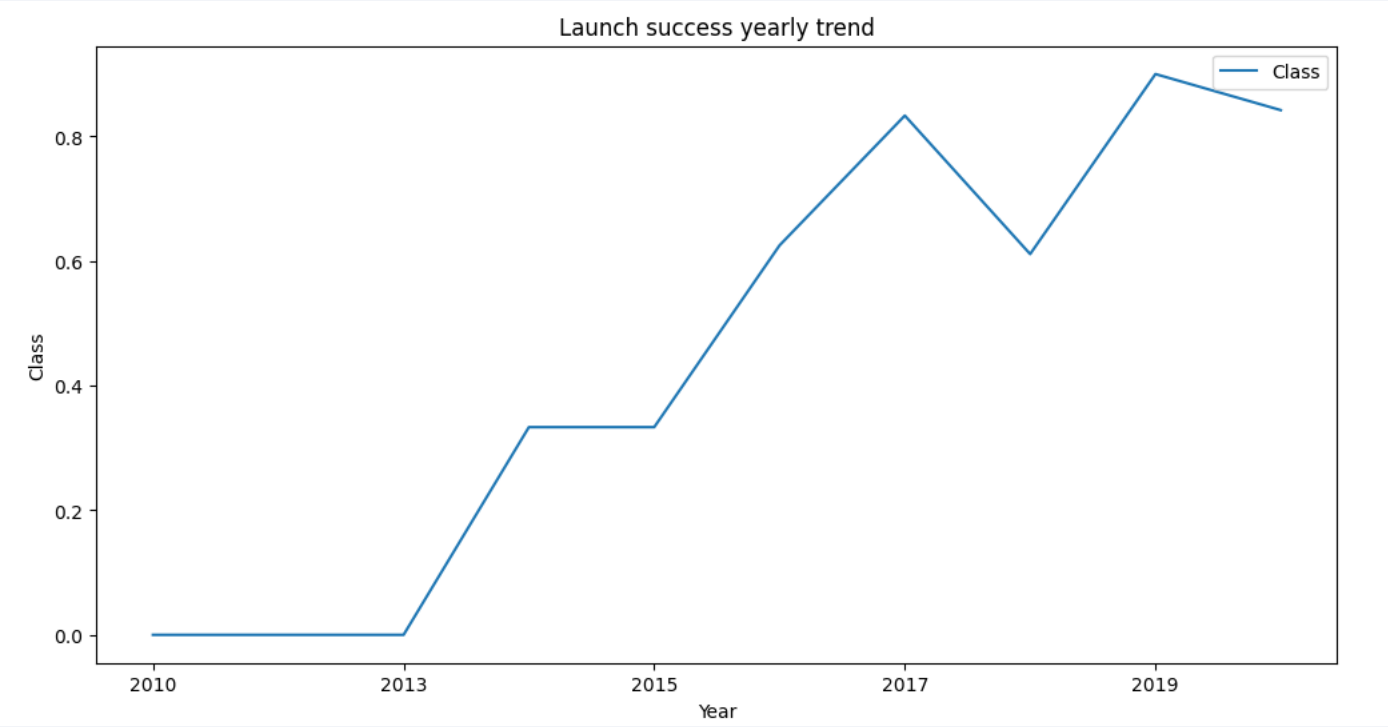
- Figure that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.
- For GTO orbit there seems to be no direct correlation between orbit type and payload mass



Legend: Blue dot – unsuccessful launch / Red dot – successful launch

Launch Success Yearly Trend

- From the figure it can be seen that in general success rate increases over the years, although there is a slight decrease in 2018 and even slighter in 2020



All Launch Site Names

- DISTINCT key word is used to show only unique launch sites from the SpaceXdata
- There are four launch sites, as it will be shown later on the map

```
%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'

- Query used to filer only launch site beginning with CCA used LIKE for selecting those names and LIMIT was used to display only 5 records

```
%sql select "Launch_Site" FROM SPACEXTBL WHERE "Launch_Site" LIKE '%CCA%' LIMIT 5
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- Total Payload Mass was calculated using SUM and filtered for NASA customer with WHERE customer = NASA

```
%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

- AVG function was used the get the average Payload Mass and WHERE was used to get only average Payload Mass for launches where booster version was 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

- Two filters were applied with WHERE, for successful Mission and Landing Outcomes to be “ground pad”, and MIN function was used on date column of that data to get earliest successful landing outcome

```
%sql SELECT MIN("DATE") from SPACEXTBL WHERE "Mission_Outcome"="Success" AND "Landing_Outcome" LIKE "%ground pad%"
```

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

MIN("DATE")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- WHERE clause was used to filter data for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with Payload Mass between 4000 and 6000

```
ECT "Booster_Version" from SPACEXTBL WHERE "Landing_Outcome" LIKE "%Success (drone ship)%" and "PAYLOAD_MASS__KG_" BETWEEN 40
```

```
 * 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

- COUNT for each type of Mission Outcome was used, where data were grouped by Mission Outcome
- Total successful missions were 100, and there was only 1 failed
- LIKE “success” /”failure” could have been used for Mission Outcome to have only for total for success and failure without details of the outcome

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") from SPACEXTBL GROUP BY "Mission_Outcome"
```

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

Mission_Outcome	COUNT("Mission_Outcome")
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Boosters that have carried the maximum payload were identified by using a subquery in the WHERE clause and the MAX() function.

```
%sql select booster_version, payload_mass__kg_ from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) from
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- Combinations of WHERE, LIKE, AND clauses were used for failed landing outcomes in exact month, their booster versions, and launch site names for year 2015

```
%sql select SUBSTR("Date",6,2), "Booster_Version", "Launch_Site" from SPACEXTBL where "landing_outcome" = "Failure (drone sh
```

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

SUBSTR("Date",6,2)	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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

- COUNT() function was used to count the different landing outcomes.
- The WHERE and BETWEEN clauses filtered the results to only include results between in defined timeframe. GROUPBY ensured that the counts were grouped by their outcome. ORDERBY and DESC clauses were used to sort the results by descending order

```
,"landing_outcome" from SPACEXTBL where DATE between '2010-06-04' and '2017-03-20' group by "landing_outcome" order by count(
```

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

count("landing_outcome")	Landing_Outcome
10	No attempt
5	Success (drone ship)
5	Failure (drone ship)
3	Success (ground pad)
3	Controlled (ocean)
2	Uncontrolled (ocean)
2	Failure (parachute)
1	Precluded (drone ship)

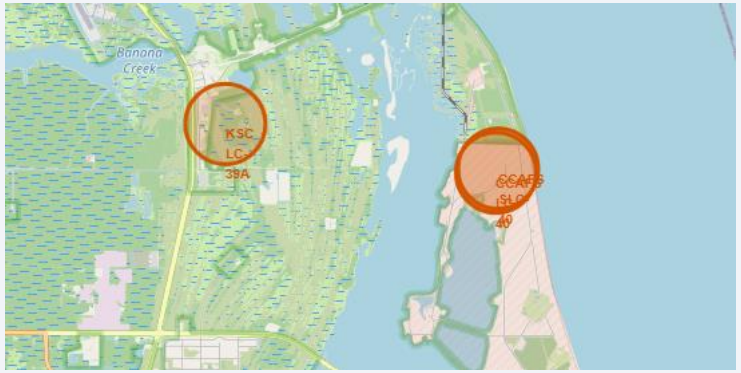
Section 3

Launch Sites Proximities Analysis

SpaceX launch site locations

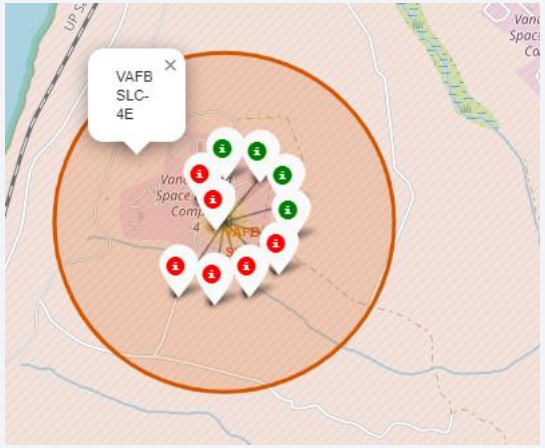
- SpaceX launch sites are in the United States of America, strategically placed near the coast, in Florida and California

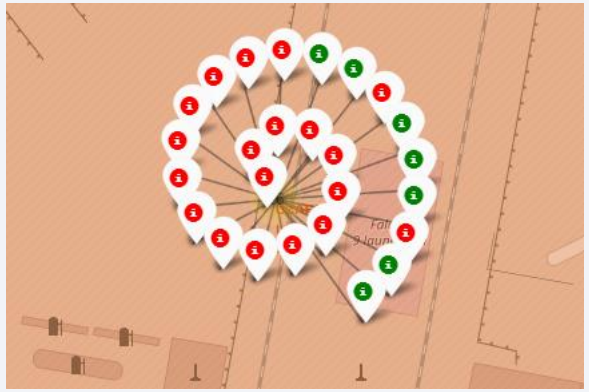
- In Florida there are three launch sites

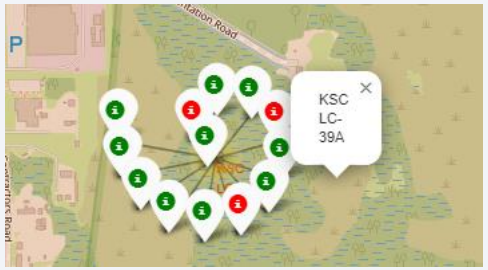


Launch sites (un)successful outcomes

- Green markers are used for success and red for failure for each launch site

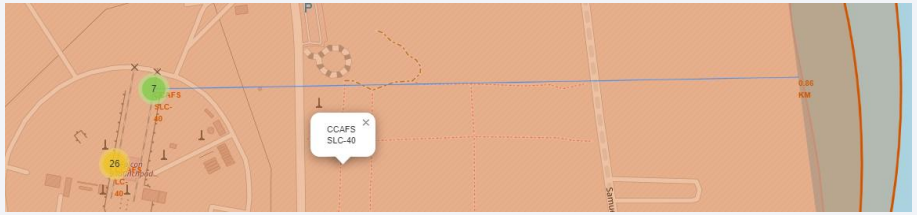




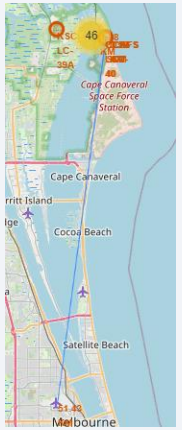


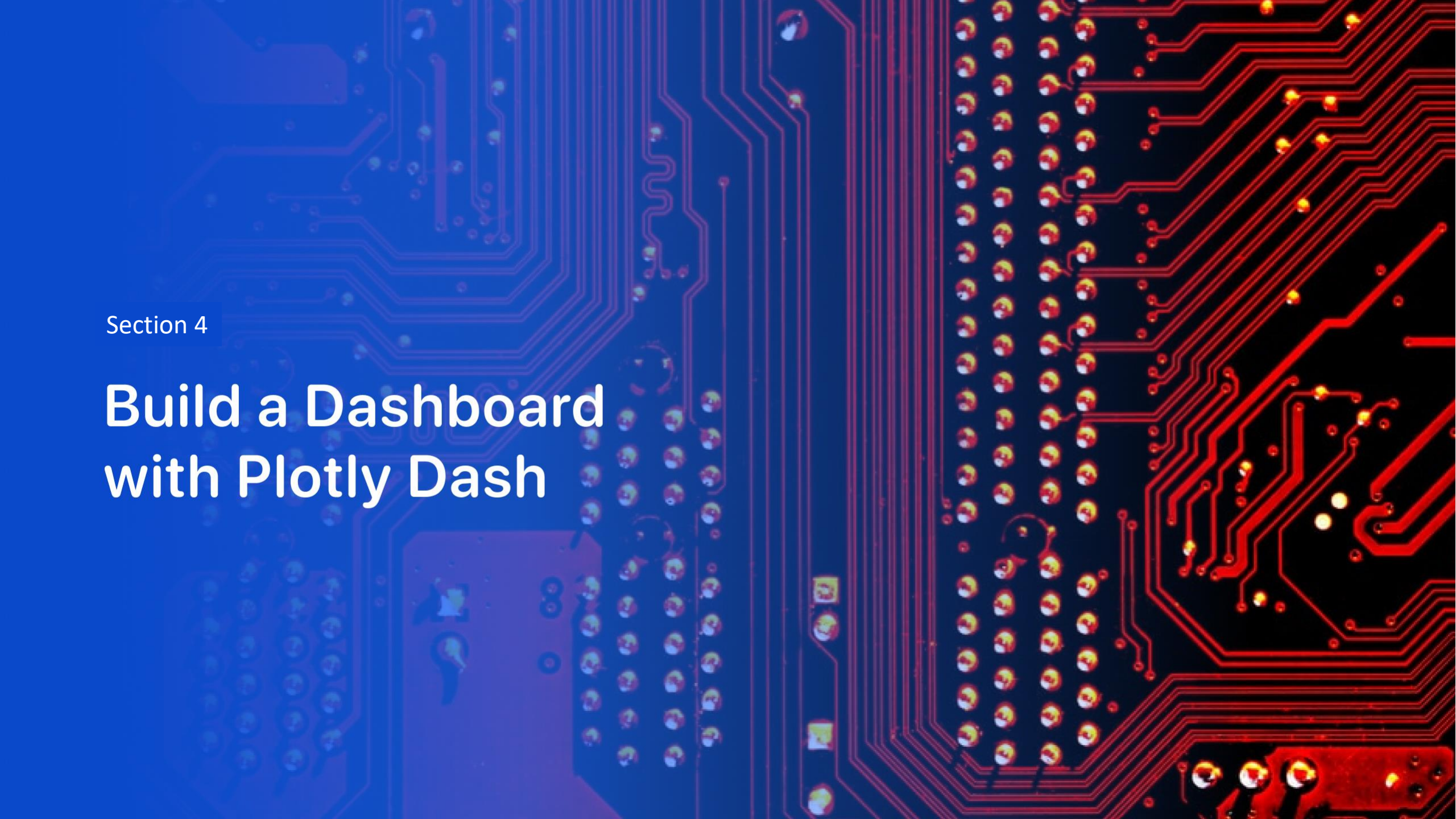
Launch Site distance to landmarks

- Distance of specific launch site to its proximities such as railway, highway, coastline was calculated and displayed, and the distance here is not that big



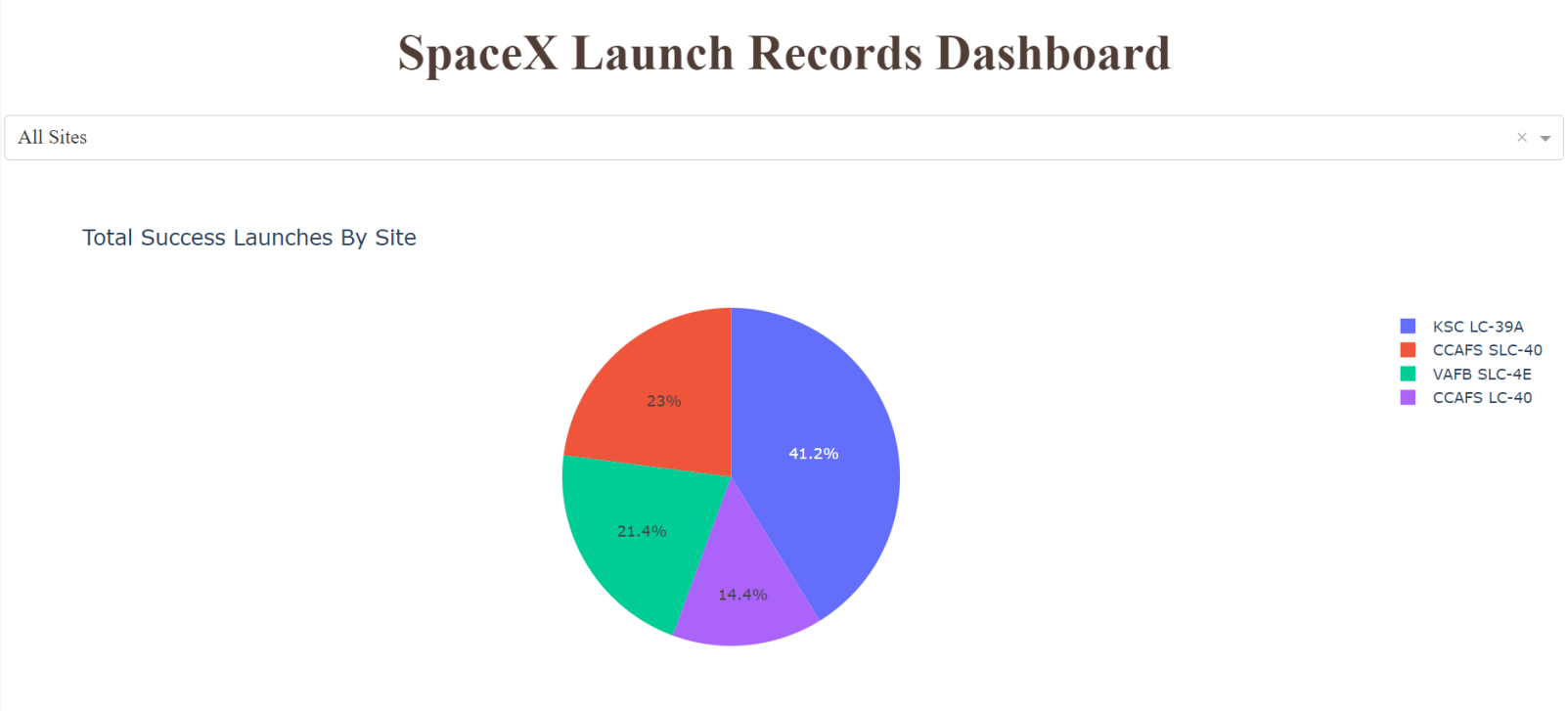
- Distance to the nearest city (Melbourne) is much bigger for safety reasons




Section 4
Build a Dashboard with Plotly Dash

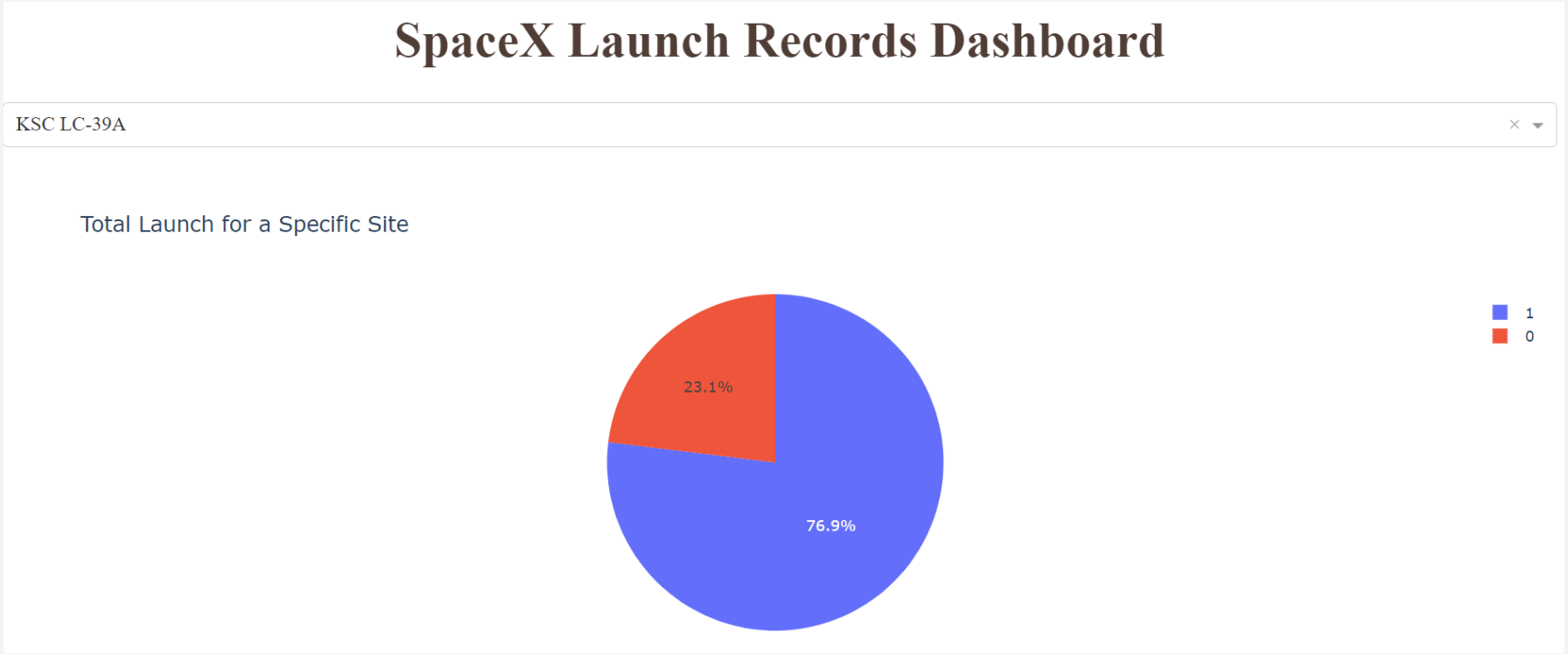
Total successful launches by launch site

- The KSC LC-39A Launch site has the most successful launches out of all launches (in percentage 41,7%)

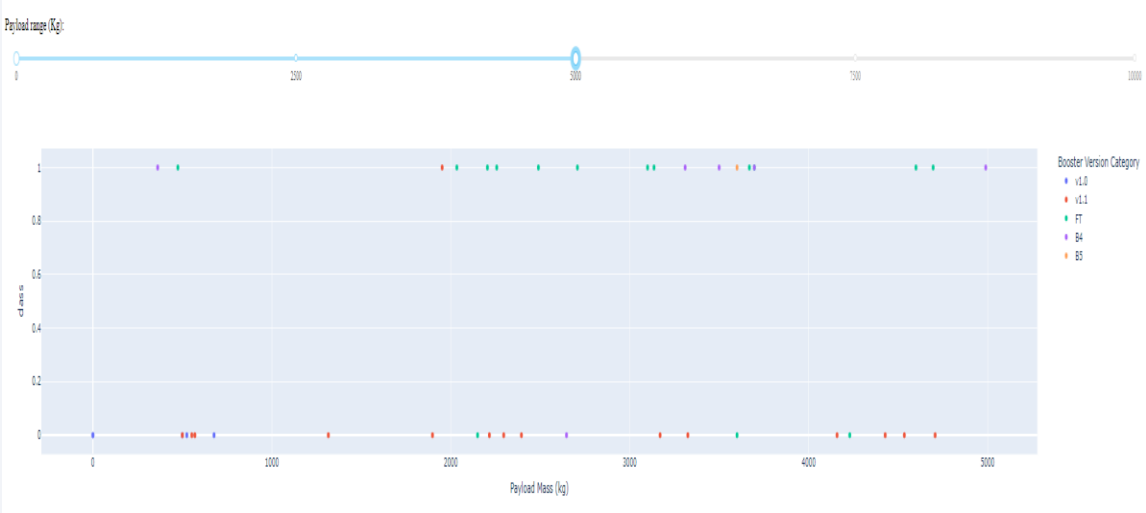


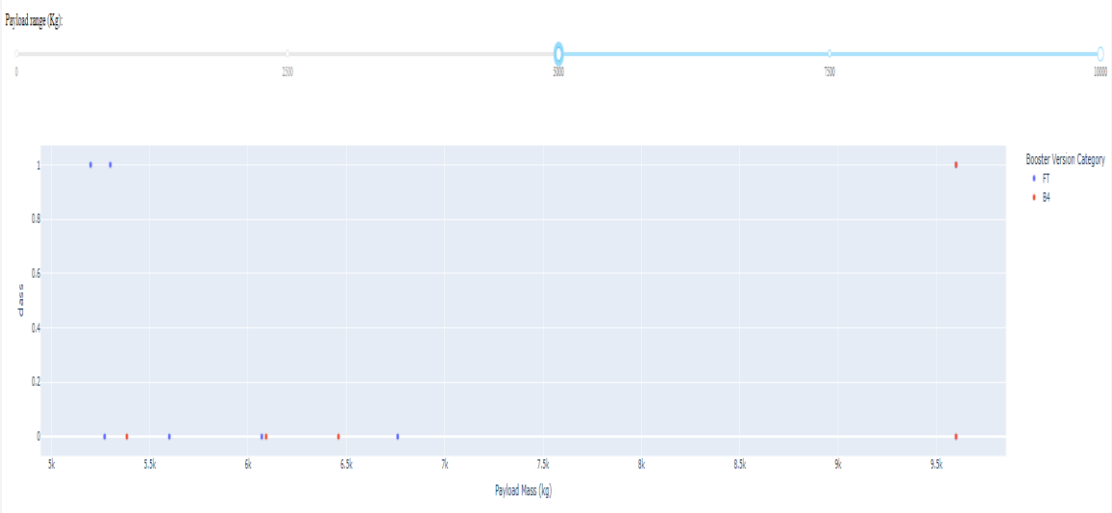
Launch site with the highest launch success ratio

KSC LC-39A has a highest success rate with 76,9% success rate, while getting only 23,1% failure rate



Payload vs Launch Outcome for all sites





- Success rates for low weighted payloads is higher than the heavy weighted payloads (below or above 5000 kg)

Section 5

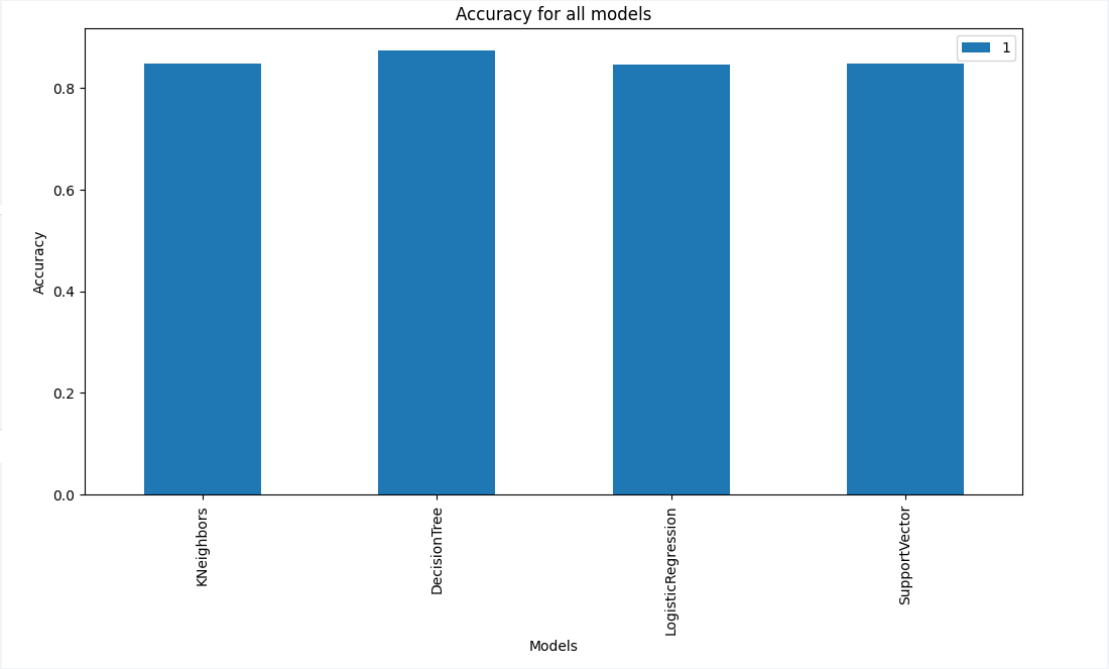
Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy (with a score of 0.873)

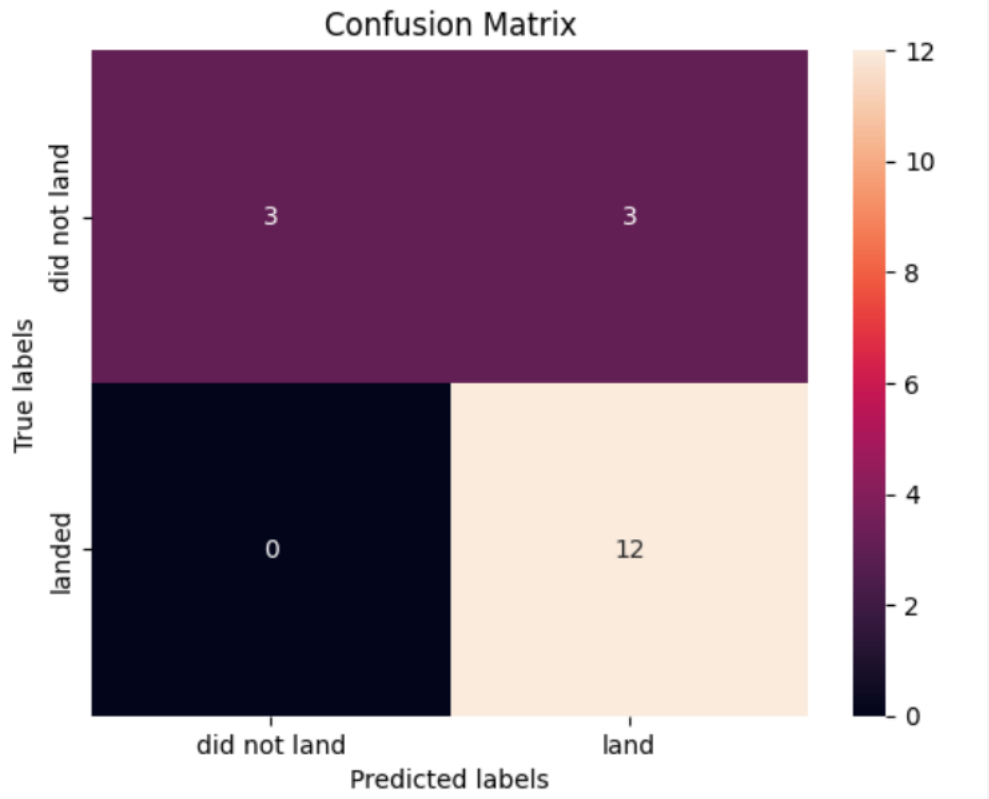
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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

```
Best model is DecisionTree with a score of 0.8732142857142856
```



Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the model predicted 12 successful landings when the True label was successful (TruePositive) and 3 unsuccessful landings where True label was failure (TrueNegative)
- The major problem are 3 unsuccessful landings when the True label was successful (False Positive).
- The model generally predicted successful landings.



Conclusions

- With the increase of the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate in general shows increase over the years.
- Orbits ES-L1, GEO, HEO, SSO had the most success rate.
- The launch sites are nearer to the highways and railways, possibly for transportation, but also far away from cities for safety.
- KSC LC-39A had the most successful launches of any sites.
- Success rate can be linked to payload mass as the lighter payloads generally proved to be more successful than the heavier payloads
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!