IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Muhittin Nasuh Kara
5 March 2023

Outline

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

Executive Summary

- Summary of methodologies
- Data Collection with API and Web Scraping
- Data Wrangling
- Exploratory Data Analysis (EDA) with Data Visualization and SQL
- Interactive Map with Folium
- Machine Learning Prediction
- Predictive Analysis
- Summary of all results
- Exploratory Data Analysis result
- Interactive maps and dashboard –
- Predictive Analytics results

Introduction

- ## Project background and context

The aim is to develop a machine learning pipeline that predicts the success of the first stage landing in SpaceX rocket launches, which determines the cost of the launch as SpaceX can reuse the first stage and offers lower prices compared to other providers. This information can be useful for other companies bidding against SpaceX for rocket launches.

- ## Problems you want to find answers

- What factors determine if the rocket will land successfully?
- How do different features interact to determine the success rate of a rocket landing?
- What operational conditions are necessary to ensure a successful rocket landing program?

Section 1

Methodology

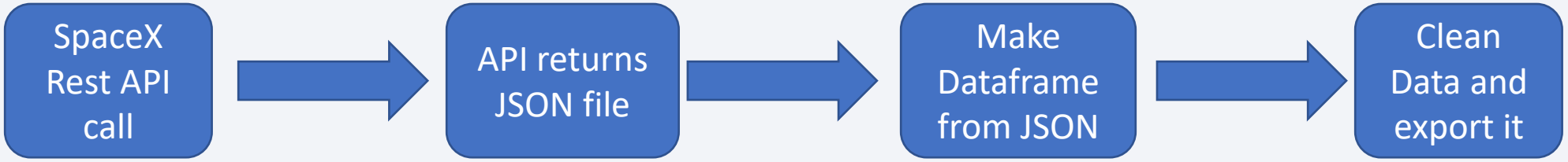
Methodology

Executive Summary

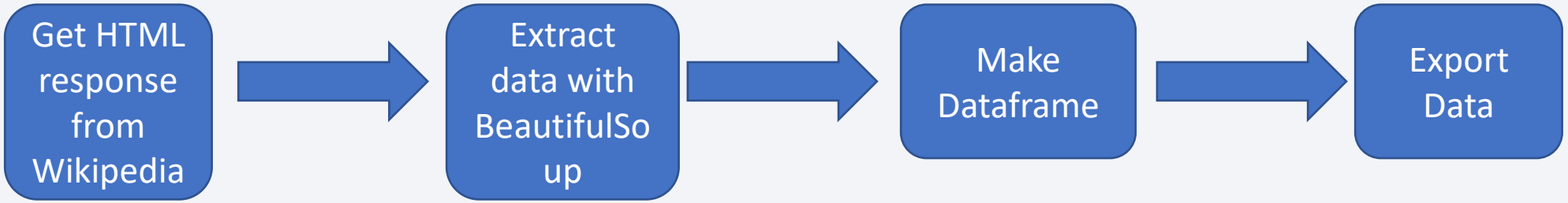
- Data collection methodology:
  - SpaceX API and web scraping from WikIpedia and SpaceX
- Perform data wrangling
  - Data was summarized and analyzed with Python
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The collected data was normalized and split into training and test sets. Four classification models were evaluated using different parameter combinations to determine their accuracy.

Data Collection

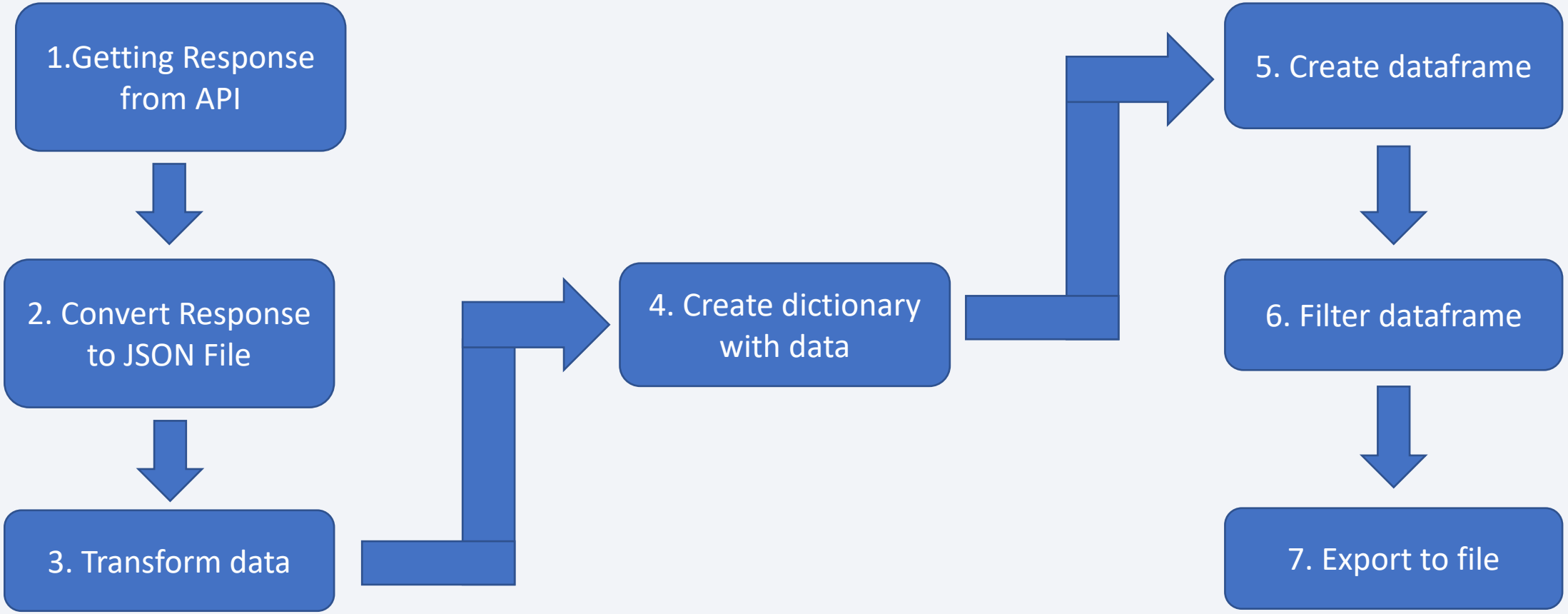
- Datasets are collected from Rest SpaceX API and webscrapping Wikipedia
- Flowchart



- The information obtained by the webscrapping of Wikipedia are launches, landing, payload information.

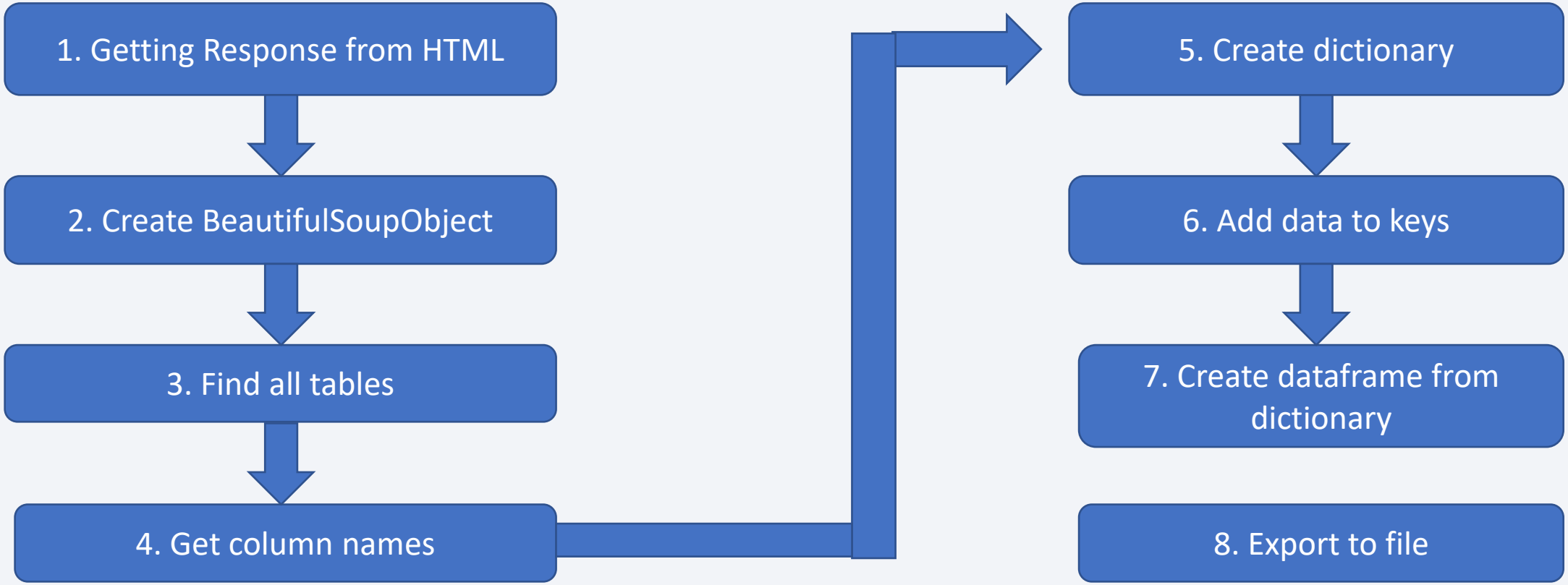


Data Collection – SpaceX API



https://github.com/nasuhkara/Applied_Data_Science_Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

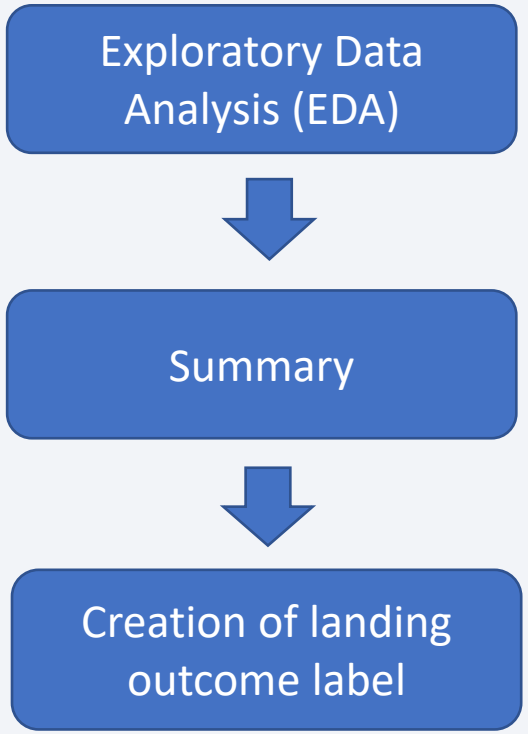
Data Collection - Scraping



https://github.com/nasuhkara/Applied_Data_Science_Capstone/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- Conducted Exploratory Data Analysis (EDA) on the dataset.
- Calculated the summary of launches per site, occurrences of each orbit, and occurrences of mission outcome per orbit type.
- Created a landing outcome label from the Outcome column.



https://github.com/nasuhkara/Applied_Data_Science_Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- **To visualize the relationship between Flight Number and Launch Site:**

Scatter plot with Flight Number on the x-axis and Launch Site on the y-axis, with each point representing a launch.

- **To visualize the relationship between Payload and Launch Site:**

Histogram with Launch Site on the x-axis and Payload on the y-axis, with each bar representing the average Payload for launches from that site.

- **To visualize the relationship between success rate of each orbit type:**

Stacked bar chart with Orbit Type on the x-axis and Success/Failure counts on the y-axis, with each bar representing the total number of launches for that Orbit Type and the sections of the bar representing the number of successful and failed launches.

- **To visualize the relationship between Flight Number and Orbit type**

Scatter plot with Flight Number on the x-axis and Orbit Type on the y-axis, with each point representing a launch.

- **To visualize the relationship between Payload and Orbit type:**

Bar chart with Orbit Type on the x-axis and Payload on the y-axis, with each bar representing the average Payload for launches with that Orbit Type.

- **To visualize the launch success yearly trend:**

Line chart with Year on the x-axis and Launch Success Rate on the y-axis, with each point representing the average success rate of launches for that year.

- https://github.com/nasuhkara/Applied_Data_Science_Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- We performed SQL queries to gather and understand data from dataset:
- Displaying 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.
- 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 successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

https://github.com/nasuhkara/Applied_Data_Science_Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We will use Folium to perform interactive visual analytics to explore the relationship between success rate and various factors such as payload mass and orbit type.
- We will mark all launch sites on a map using Folium and visualize the distribution of successful and unsuccessful launches across different launch sites.
- We will mark the success and failed launches for each site on the map to gain insights into how the location and proximity of a launch site may affect the success rate.
- We will calculate the distances between each launch site and its proximity to identify potential areas for future launch sites to improve the success rate.
- Overall, using Folium for interactive visual analytics will enable us to gain a deeper understanding of the relationship between success rate and various factors related to space missions.

https://github.com/nasuhkara/Applied_Data_Science_Capstone/blob/main/lab_jupyter_launch_site_location-checkpoint.ipynb

Build a Dashboard with Plotly Dash

- The dashboard includes multiple components for interactive data exploration and analysis.
- The dropdown component allows users to select a specific launch site or view data for all launch sites.
- The pie chart displays the total number of successful and failed launches for the selected launch site.
- The rangeslider allows users to select a specific range of payload masses to explore.
- The scatter plot displays the relationship between success rate and payload mass, providing a more detailed view of the data.

https://github.com/nasuhkara/Applied_Data_Science_Capstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- The process for building and evaluating our machine learning models can be broken down into several steps.
- First, we loaded our dataset and preprocessed it by normalizing the data and splitting it into training and test sets.
- Next, we selected machine learning algorithms and set parameters for each algorithm using GridSearchCV. We trained GridSearchCV models with the training dataset and evaluated the performance of each model.
- To evaluate the models, we obtained the best hyperparameters for each type of model and computed accuracy for each model using the test dataset. We also plotted a Confusion Matrix to visualize the model's performance.
- Finally, we compared the models based on their accuracy and selected the model with the best performance. Further details on our results can be found in our notebook.

https://github.com/nasuhkara/Applied_Data_Science_Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5%20.ipynb

Results

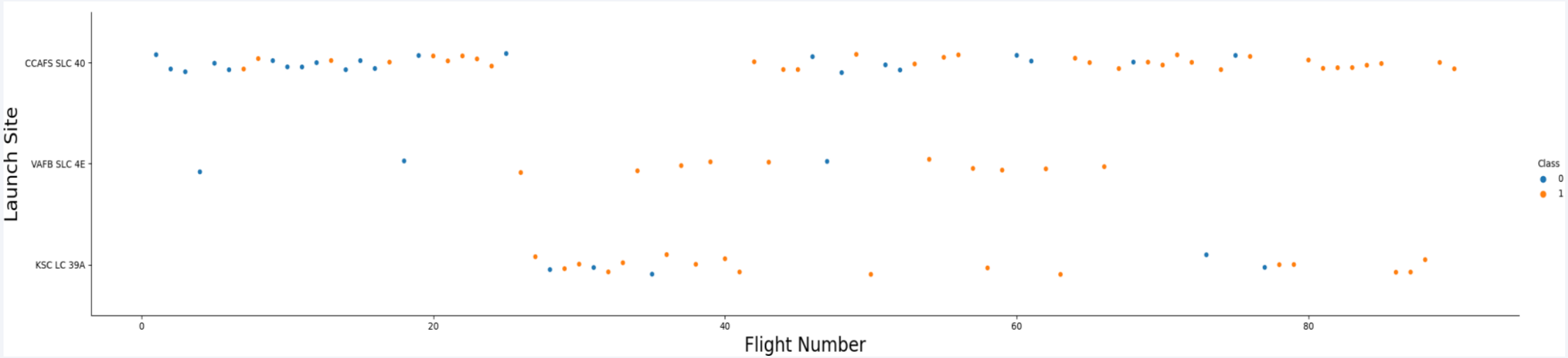
- The success rate of the first stage landing appears to increase with flight number, but decrease with payload mass.
- Different launch sites have different success rates, with CCAFS LC-40 having a success rate of 60%, while KSC LC-39A and VAFB SLC 4E have a success rate of 77%.
- In the LEO orbit, the success rate appears to be related to the number of flights, while in the GTO orbit, there seems to be no relationship between flight number and success rate.
- Heavy payloads have a higher success rate for positive landings in Polar, LEO, and ISS orbits.
- The success rate has been increasing steadily since 2013 until 2020.
- The accuracy of the different machine learning models evaluated are as follows: Logistic Regression Accuracy: 0.8464, SVM Accuracy: 0.8482, Decision Tree Accuracy: 0.889, KNN Accuracy: 0.8482. Based on these results, the decision tree model has the highest accuracy.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

Scatterplot of Flight Number and Launch Site

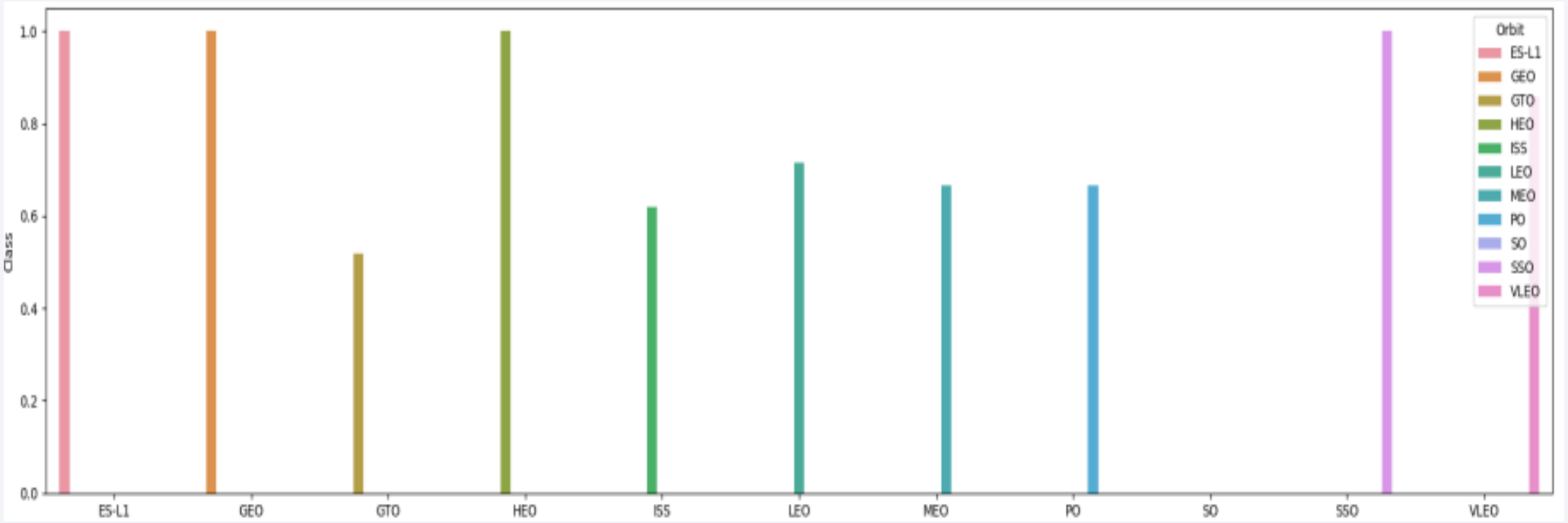


We can conclude that the success rate increases for each site. It can be concluded that the final flight success rate is gradually improving.

Payload vs. Launch Site

Depending on the launch site, a heavier payload may be considered for a successful landing. Too heavy a load will cause the landing to fail.

Success Rate vs. Orbit Type



- Show a bar chart for the success rate of each orbit type
- ES-L1, GEO, HEO, and SSO has 100% success rate, others are better than 50%

Flight Number vs. Orbit Type

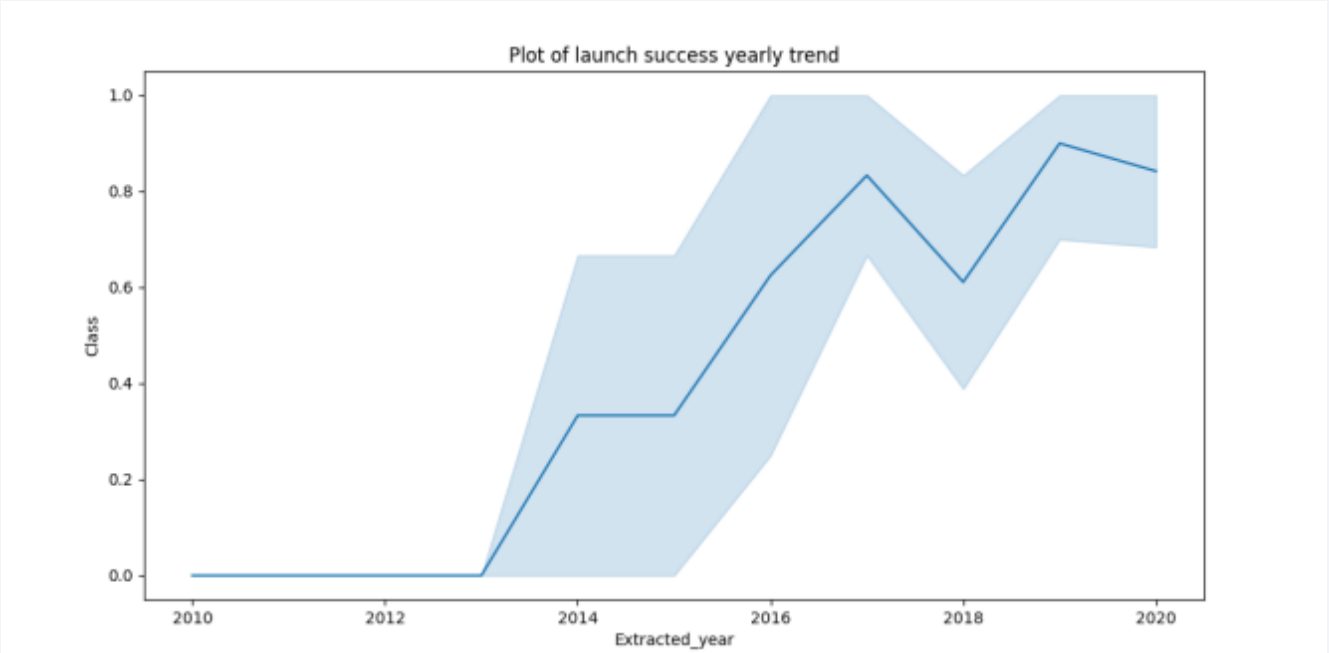
The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.

Payload vs. Orbit Type

Reducing the payload weight for a GTO orbit improves launch success. Reducing the payload weight for a GTO orbit improves launch success.

Launch Success Yearly Trend

- • Show a line chart of yearly average success rate
- • Success rate has steady increased over the years, almost 90% at 2020



All Launch Site Names

- **SQLQuery**
- **Results**
- **Explanation**

The use of DISTINCT in the query allows to remove duplicate LAUNCH_SITE

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

```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- **SQL Query**
- **Results**
- **Explanation**

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

This query returns the sum of all payload masses.

Total Payload Mass

- **SQL Query**

```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

- **Results**

	total_payloadmass
0	45596

- **Explanation**

This query returns the sum of all payload masses where the customer is NASA (CRS).

Average Payload Mass by F9 v1.1

SQL Query

```
SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'
```

Results

AVG("PAYLOAD_MASS__KG_")
2534.666666666665

Explanation

This query returns the average of all payload masses where the booster version contains the substring F9 v1.1.

First Successful Ground Landing Date

SQL Query

```
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

Results

	firstsuccessfull_landing_date
0	2015-12-22

Explanation With this query, we select the oldest successful landing. The WHERE clause filters dataset in order to keep only records where landing was successful. With the MIN function, we select the record with the oldest date.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

Results

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Explanation

With this query, we select the oldest successful landing. The WHEREclause filters dataset in order to keep only records where landing was successful. With the MIN function, we select the record with the oldest date.

Total Number of Successful and Failure Mission Outcomes

SQL Query

Results

Explanation

With the first SELECT, we show the subqueries that return results. The first subquery counts the successful mission. The second subquery counts the unsuccessful mission

```
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

```
The total number of successful mission outcome is:
```

	successoutcome
0	100

```
The total number of failed mission outcome is:
```

	failureoutcome
0	1

Boosters Carried Maximum Payload

SQL Query

```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

Results

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

Explanation

We used a subquery to filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECTDISTINCT) with the heaviest payload mass

2015 Launch Records

SQL Query

```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Results

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Explanation

This query returns month, booster version, launch site where landing was unsuccessful and landing date took place in 2015. Substr function process date in order to take month or year. Substr(DATE, 4, 2) shows month. Substr(DATE,7,4) shows year.

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

SQL Query

```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

Results

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

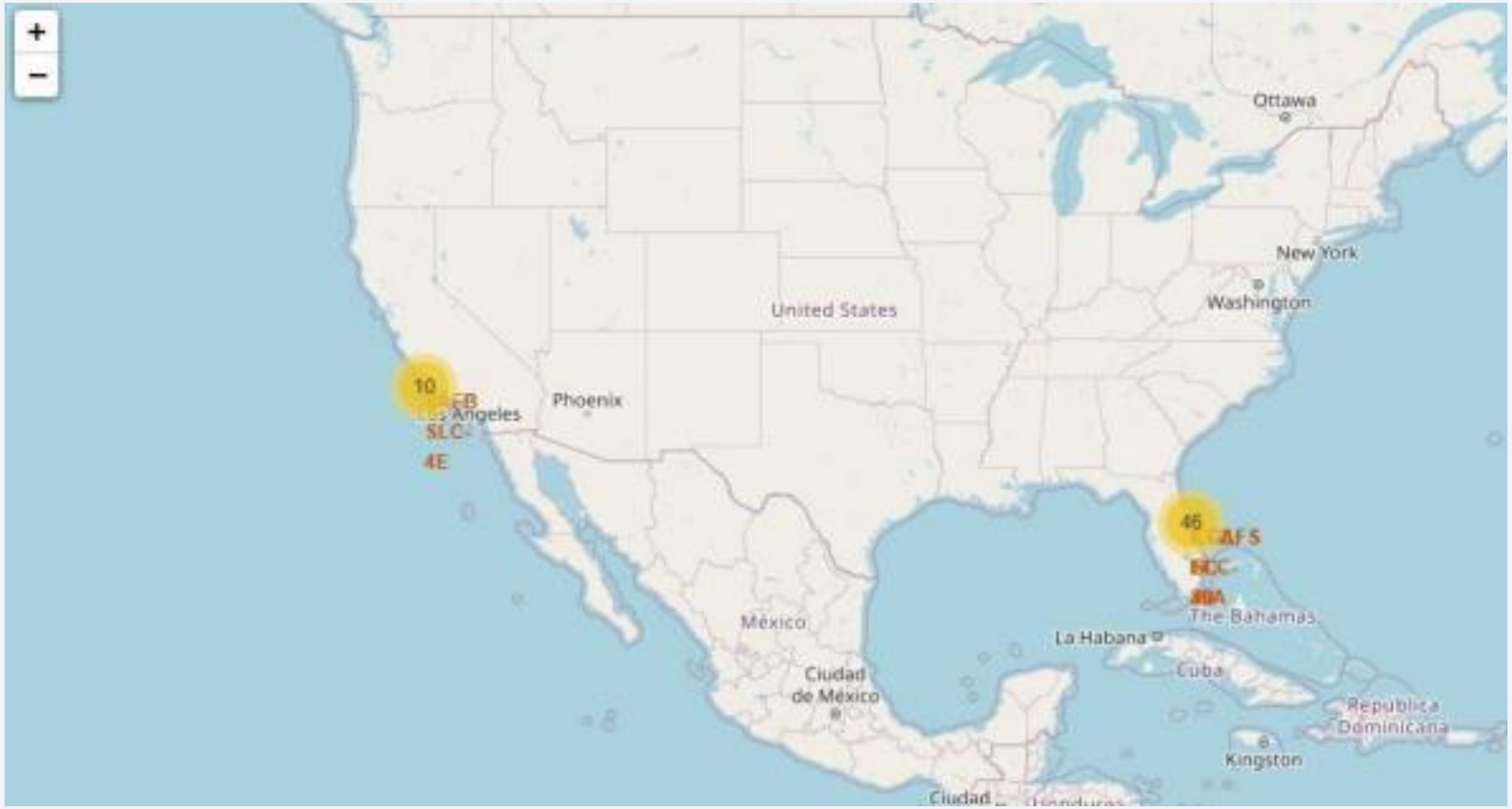
Explanation

This query returns landing outcomes and their count where mission was successful

Section 3

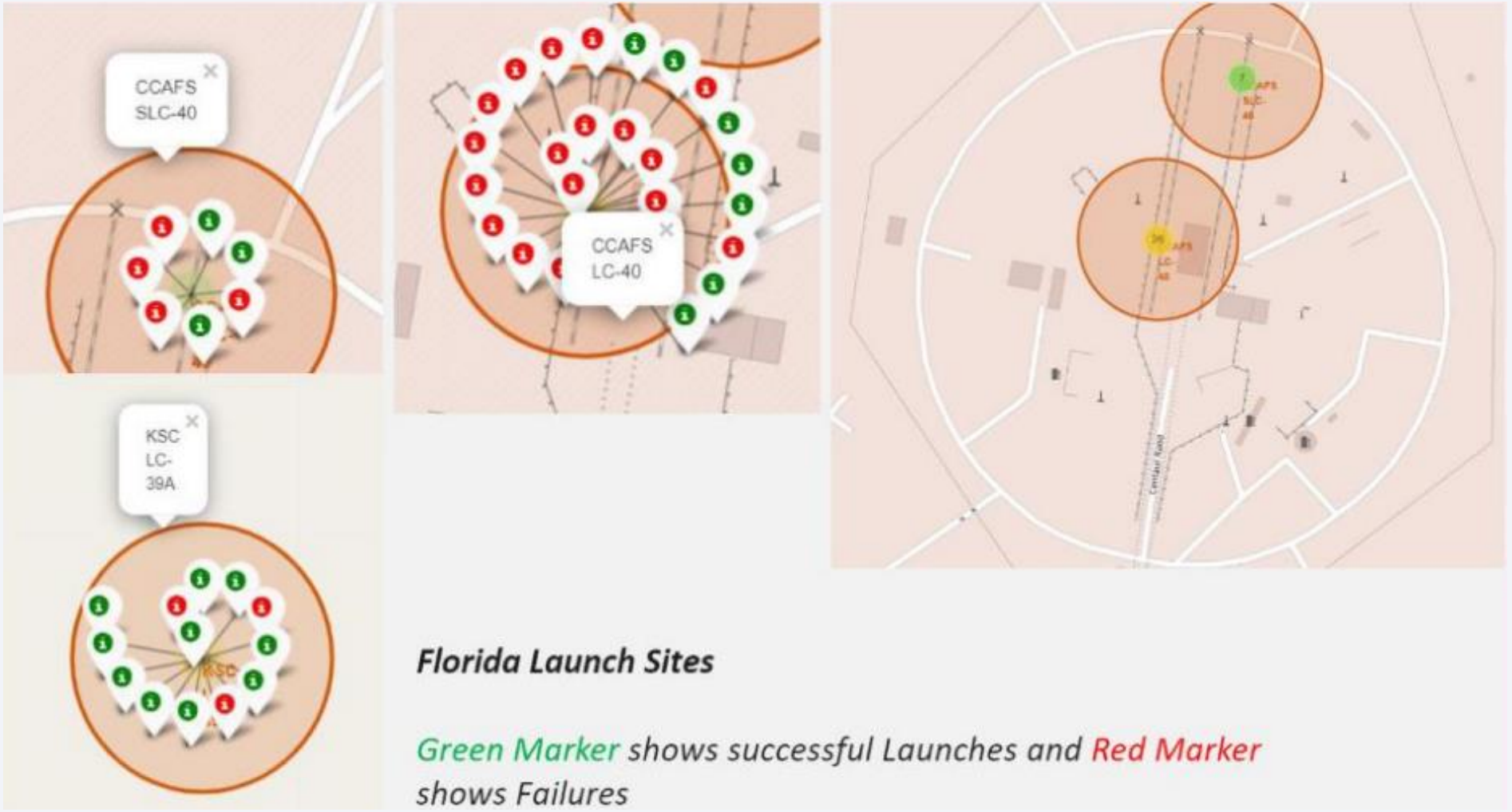
Launch Sites Proximities Analysis

All launch sites global map markers

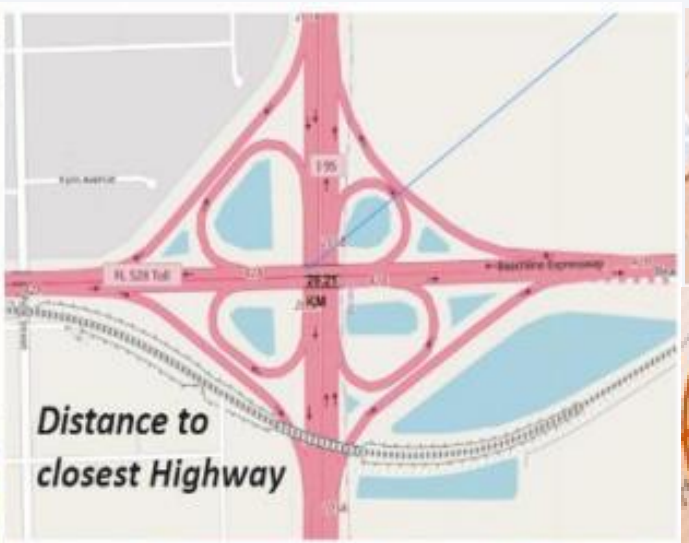


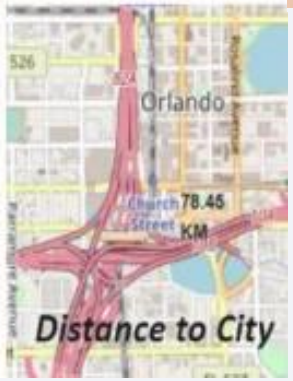
We can see that the SpaceX Launch sites are in the USA. Florida and California

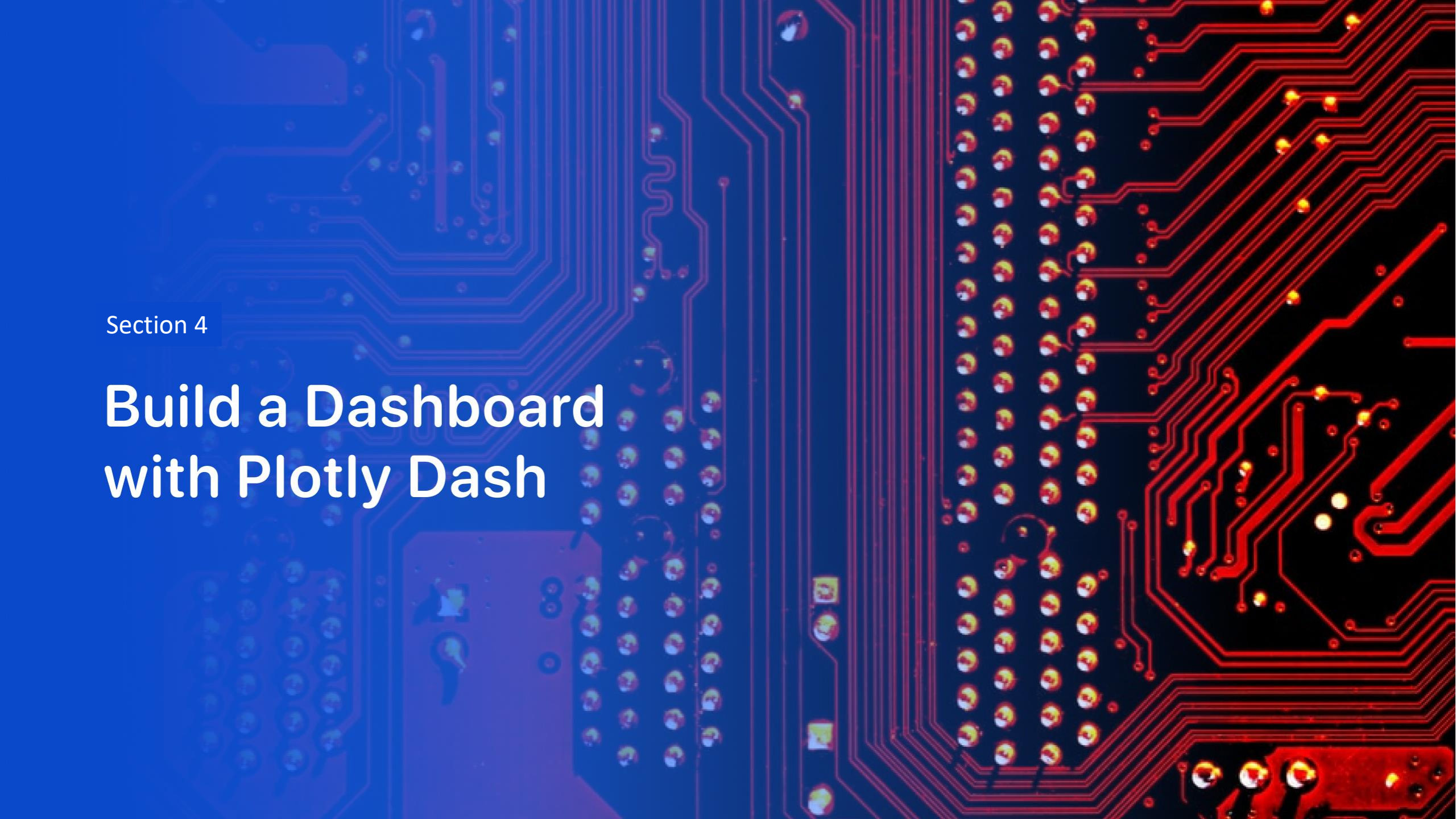
Folium map – Color Labeled Markers



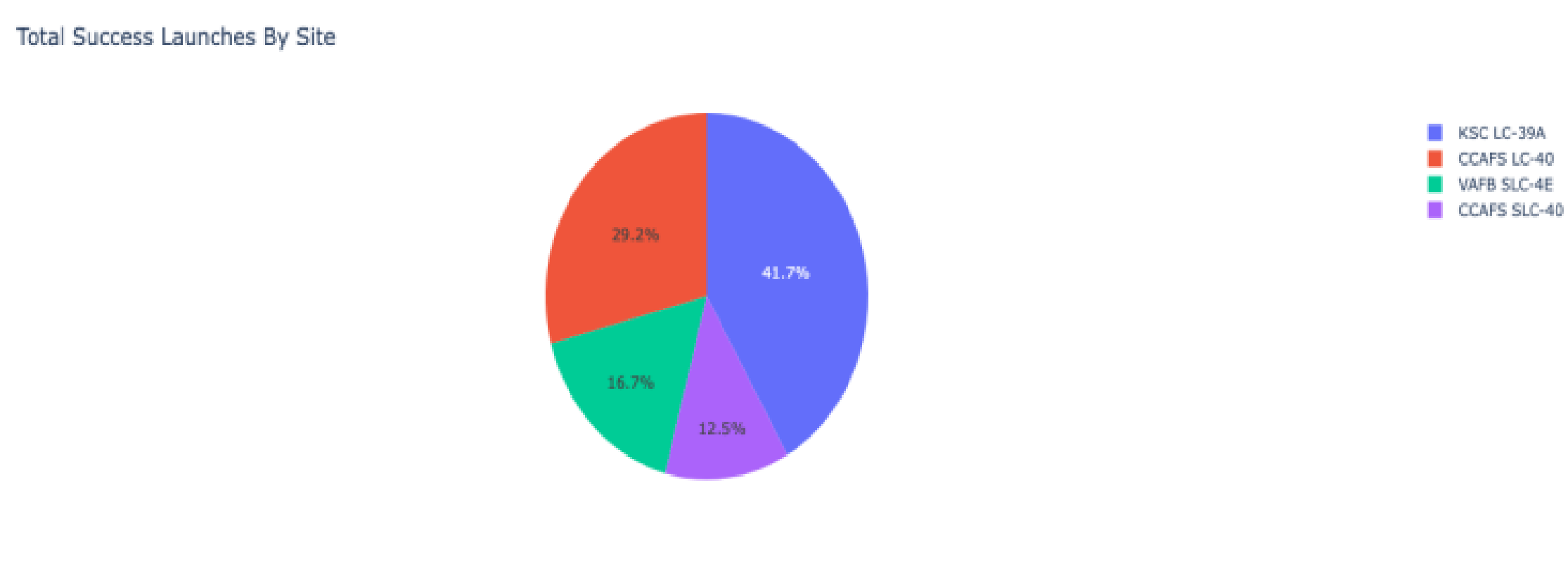
Launch sites to landmarks distance






Section 4
Build a Dashboard with Plotly Dash

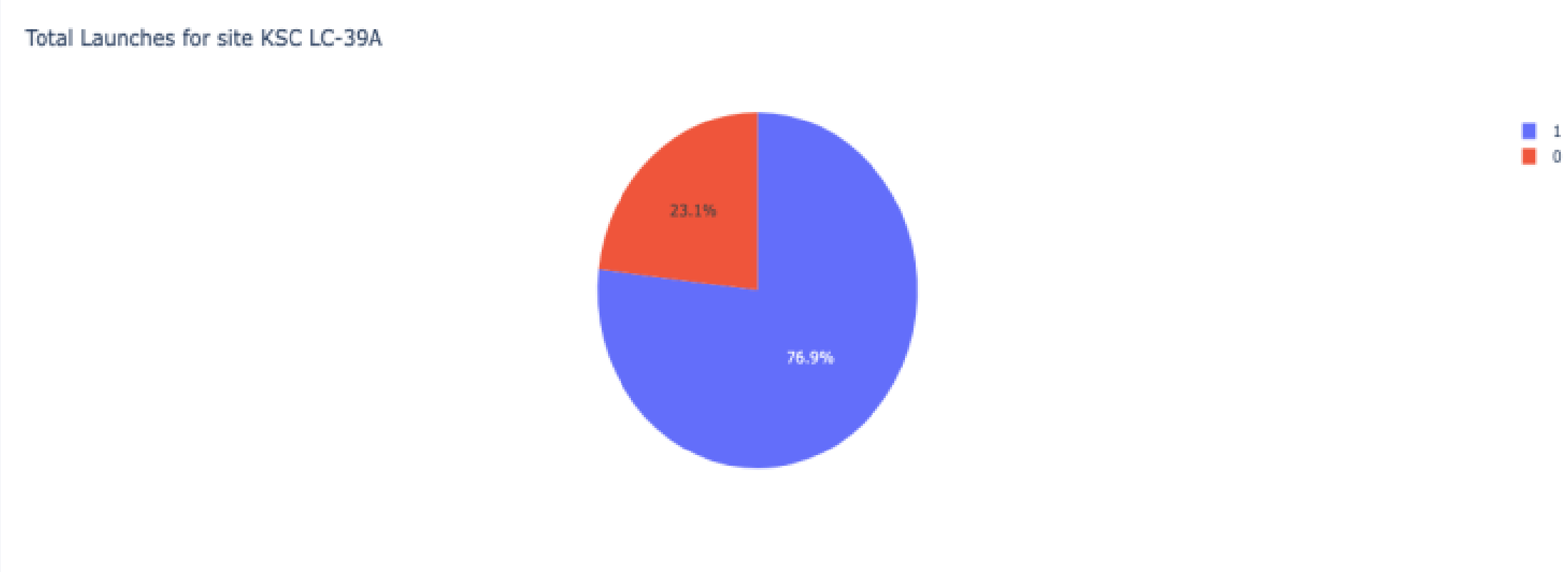
Total Success Launches by Site



KSC LC-39A has the highest success rate
CCAFS SLC-40 has the lowest success rate

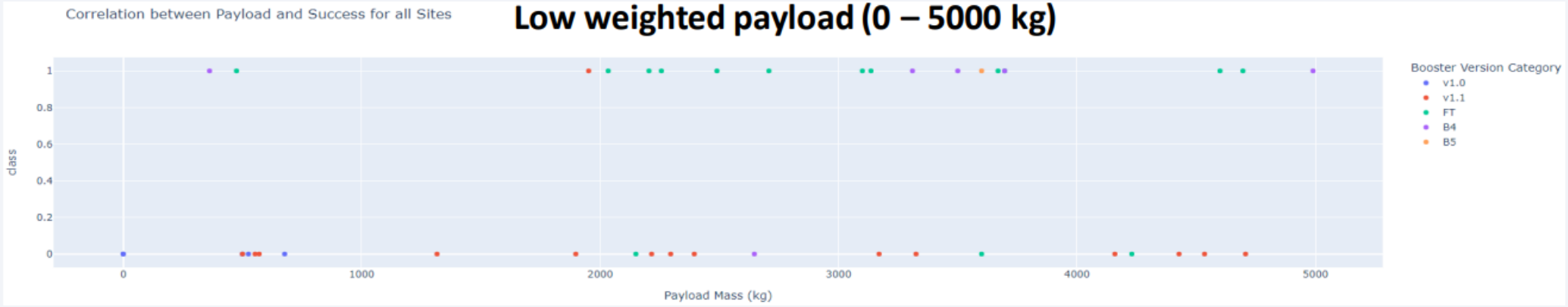
Highest launch site success rate

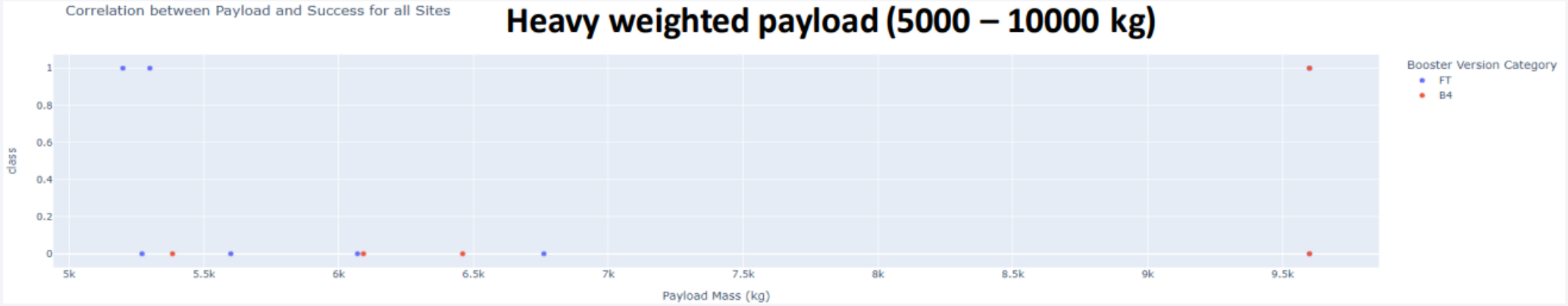
KSC LC-39A has close to 77% success rate



Payload mass vs Outcome for all sites with different payload mass selected

Lightweight loads have a better success rate than heavyweight loads.



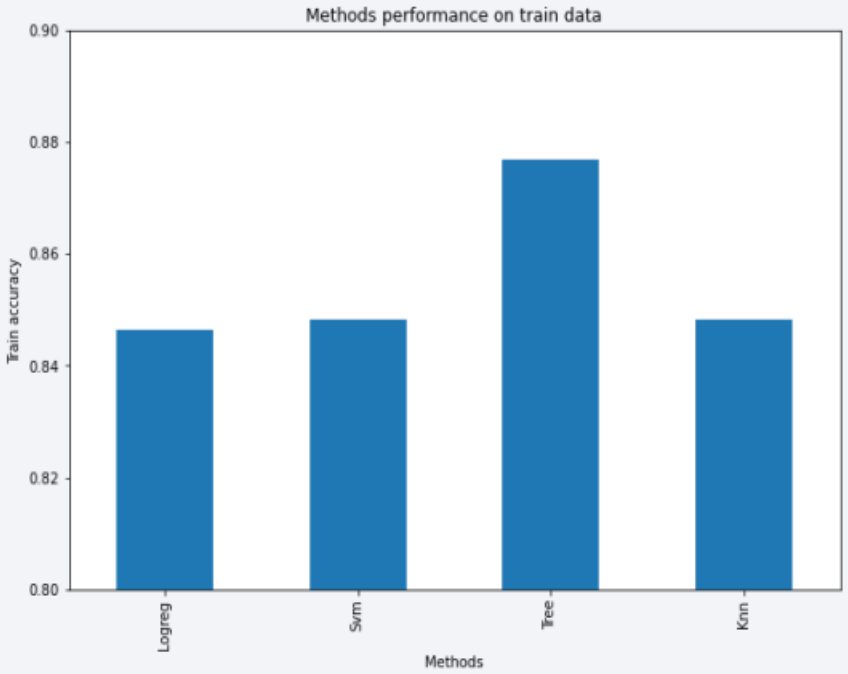


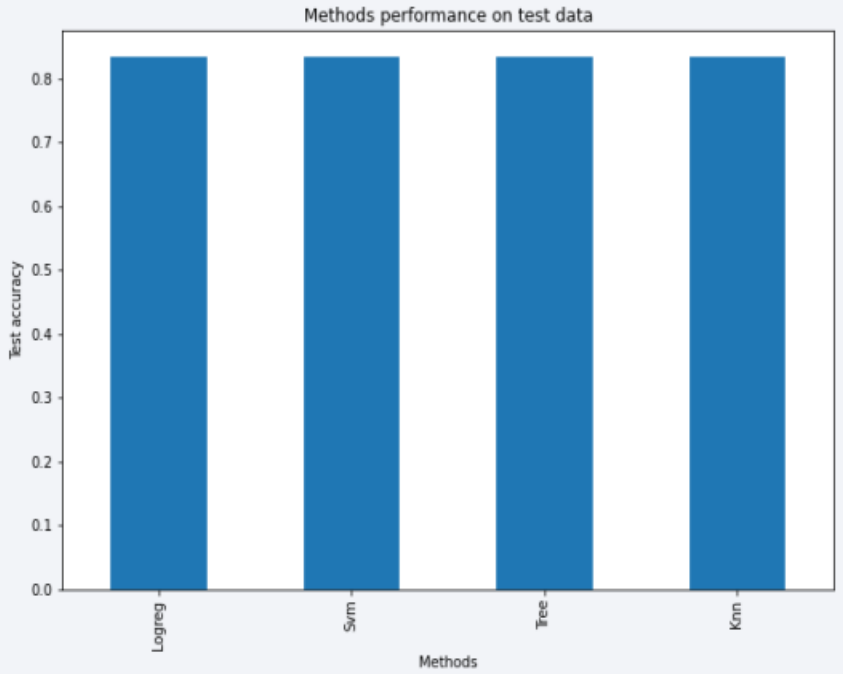
Section 5

Predictive Analysis (Classification)

Classification Accuracy

	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333

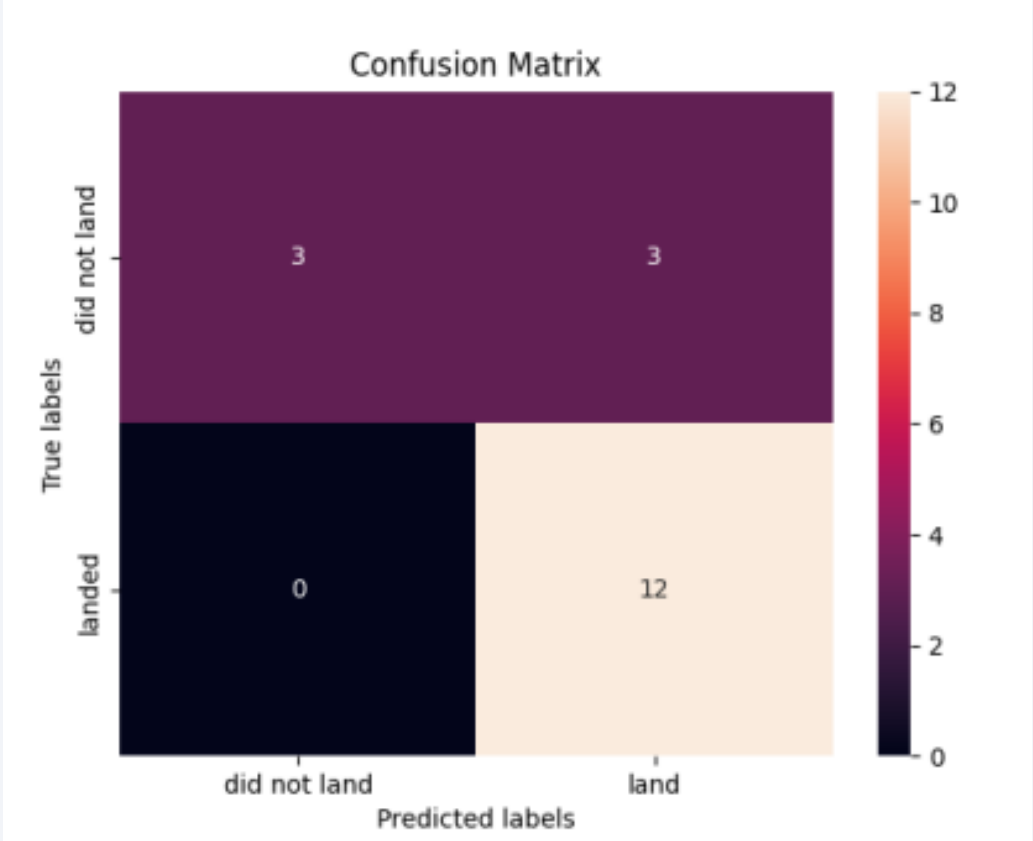




The decision tree classifier is the model with the highest classification accuracy

Confusion Matrix

- Confusion matrix for decision tree
- Classifier shows what classifier can do
- It distinguishes between different classes.
- The biggest problem is false positives.



Conclusions

- Launch success rates depend on factors such as launch site, orbit, and previous launches.
- Success rates have been increasing since 2013.
- The best orbits for success are ES-L1, GEO, HEO, SSO, and VLEO.
- Lighter payloads tend to perform better overall.
- KSC LC-39A had the most successful launches.
- The reason why some sites perform better than others is unclear.
- The Decision Tree Algorithm was the best machine learning algorithm for this dataset.
- Knowledge gained from previous launches has helped improve launch success rates.

Appendix

We utilized Python and SQL to preprocess and analyze the launch success dataset, and created charts and graphs to visualize our findings. Notepad was used to keep track of our progress and store code snippets. The project output includes a final report .

Here my github link

https://github.com/nasuhkara/Applied_Data_Science_Capstone

Thank you!