IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mikail Memis
09.12.2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Acquistion: leveraged APIs and web scraping techniques for comprehensive data collection
  - Data Preparation: Performed rigorous data cleaning and transformation for analysis readiness
  - Analytical Exploration: Conducted SQL-driven trend analysis and pattern identification
  - Data Visualization: Created impactful visual representations to elucidate findings
  - Interactive Visual Analytics with Folium: Mapped data geographically for spatial analysis
  - Machine Learning Prediction: Developed predictive models with validated accuracy
- Summary of all results
  - Exploratory Data Analysis Result: Uncovered pivotal insights that inform business strategy
  - Interactive Analytics in Screenshots: Provided a visual narrative of data stories through interactive maps
  - Predictive Analytics Result: Achieved high predictive accuracy, enhancing forecast reliability

Introduction

- Project background and context

  Space X lists the Falcon 9 rocket launch service at 62 million dollars, a significant cost reduction compared to other providers who charge upwards of 165 million dollars. This saving largely stems from the reuse of the rocket's first stage. If we can predict the landing outcomes of this stage, we can better estimate launch costs, a valuable insight for companies considering bids against Space X. The project aims to build a machine learning model to forecast the first stage landing success.

- Problems you want to find answers

  - Determining the critical factors that predict a successful first-stage landing.
  - Examining the interplay between different variables affecting landing success.
  - Establishing the operational conditions required for a consistent landing process.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- 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

- API Query Execution:
  - Initiated 'GET' requests to the SpaceX API.
- Data Decoding:
  - Transformed API response content into JSON format.
  - Utilized pandas.json_normalize() to structure JSON data into a tabular form within a pandas DataFrame.
- Data Cleaning:
  - Performed data quality checks for completeness.
  - Imputed missing values to maintain data integrity.
- Web Scraping Enhancement:
  - Applied BeautifulSoup to extract Falcon 9 launch records from Wikipedia.
  - Converted HTML tables into a structured pandas DataFrame.
- Integration and Preparation:
  - Merged API and scraped data into a singular DataFrame for comprehensive analysis.

Data Collection – SpaceX API

- The SpaceX API's GET request was utilized for data acquisition, followed by the cleansing, organization, and formatting of the data into a structured form.

- The link to the notebook:

[jupyter-labs-spacex-data-collection-api.ipynb]

1. Get request for rocket launch data using API

In [6]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [7]:
```
response = requests.get(spacex_url)
```

2. Use json_normalize method to convert json result to dataframe

In [12]:
```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:
```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

In [30]:
```
rows = data_falcon9['PayloadMass'].values.tolist()[0]

df_rows = pd.DataFrame(rows)
df_rows = df_rows.replace(np.nan, PayloadMass)

data_falcon9['PayloadMass'][0] = df_rows.values
data_falcon9
```

Data Collection - Scraping

- Web scraping was applied to extract Falcon 9 launch records using BeautifulSoup, and the resulting table was parsed and transformed into a pandas DataFrame format.
- The link to the notebook:

[jupyter-labs-webscraping.ipynb]

1. Apply HTTP Get method to request the Falcon 9 rocket launch page

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

Out[5]:
```
200
```

2. Create a BeautifulSoup object from the HTML response

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

3. Extract all column names from the HTML table header

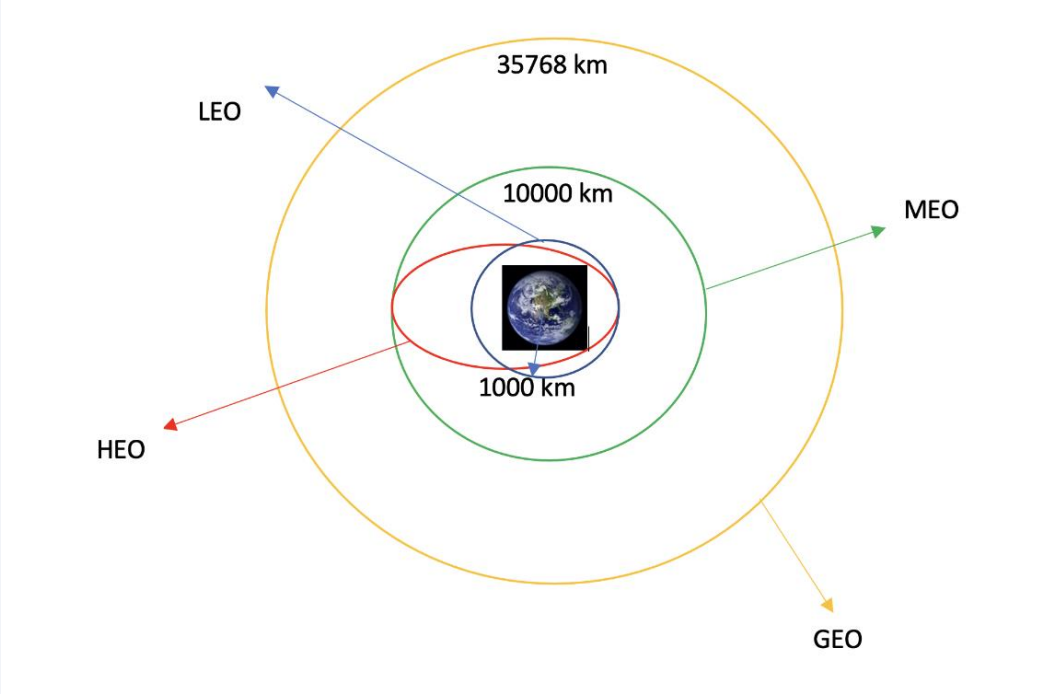
In [10]:
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

Data Wrangling

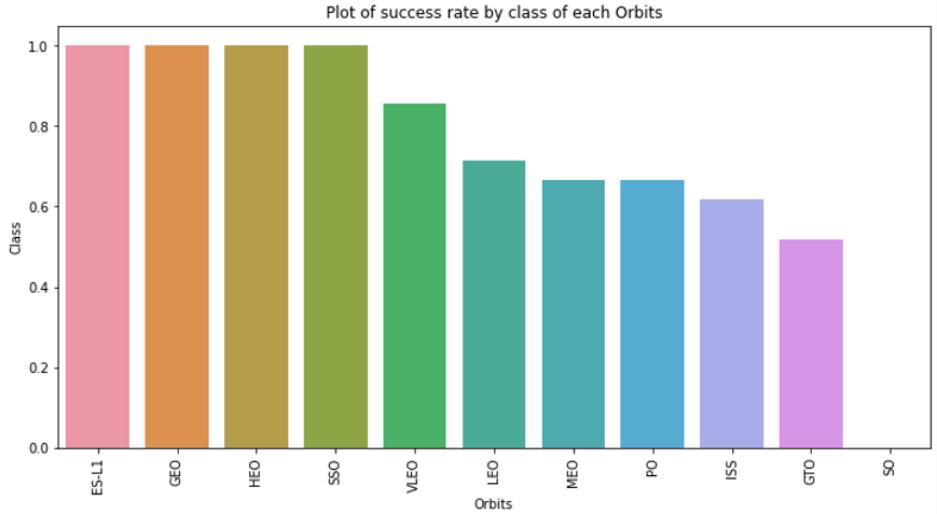
- Exploratory data analysis was conducted, and training labels were determined.
- The number of launches at each site, as well as the number and frequency of each orbit, were calculated.
- Landing outcome labels were generated from the outcome column, and the results were exported to a CSV file.
- The link to the notebook:

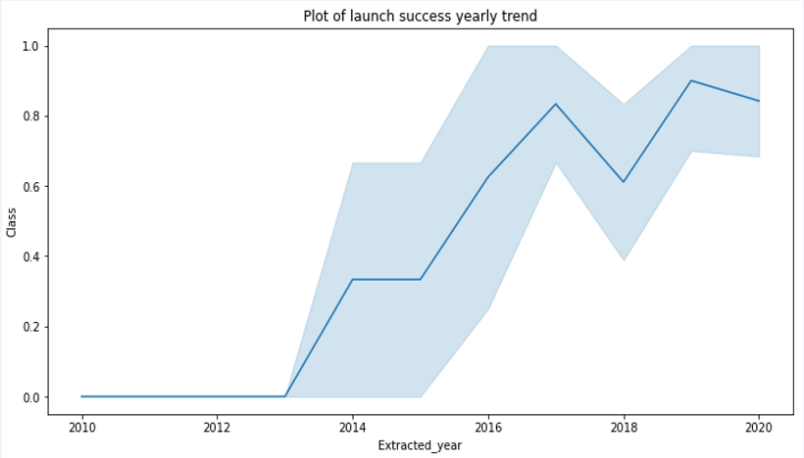
[labs-jupyter-spacex-Data wrangling.ipynb]



EDA with Data Visualization

- Two charts were plotted: The bar chart outlines success rates for various orbital classes, showing variance in success. The line graph traces success over years, highlighting trends and annual success fluctuations.
- The link to the notebook: jupyter-labs-eda-dataviz.ipynb





EDA with SQL

- Successfully integrated the SpaceX dataset into a PostgreSQL environment within a Jupyter notebook for seamless analysis.
- Conducted SQL-driven EDA to derive actionable insights, executing queries that:
  - Identified each unique launch site used in space missions.
  - Calculated the cumulative payload mass delivered by NASA (CRS) missions.
  - Determined the average mass per payload for the Falcon 9 v1.1 booster variant.
  - Aggregated mission outcomes to quantify the success and failure rates.
  - Extracted details pertaining to unsuccessful landings, including drone ship incidents, booster versions, and associated launch sites.

The link to the notebook: jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- All launch sites were marked on the folium map, enhanced with map objects like markers, circles, and lines to depict launch outcomes at each location.
- Launch outcomes were categorized as 0 for failure and 1 for success for analytical clarity.
- Marker clusters, color-coded for ease of visualization, indicated the relative success rates of different sites.
- Distances from launch sites to nearby infrastructures were calculated, providing insights into questions such as the proximity of launch sites to railways, highways, coastlines, and urban areas, and whether a certain distance is maintained from populated centers for safety and logistical reasons.

Build a Dashboard with Plotly Dash

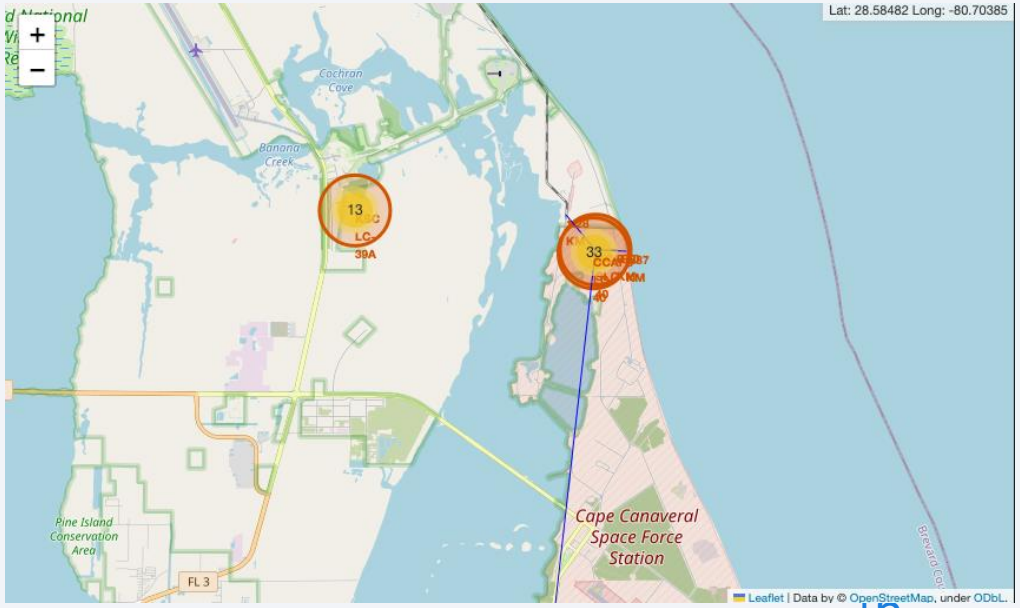
- An interactive dashboard was developed using Plotly Dash, enabling dynamic data exploration.
- Pie charts were added to display the distribution of total launches from various sites, offering a visual summary of launch activity per location.
- Scatter plots were created to analyze the correlation between launch outcomes and payload mass across different booster versions, providing insights into performance trends.
- These visual elements facilitate an immediate understanding of key metrics and their impacts on launch success.
- The link to the notebook: spacex_dash_app.py

Predictive Analysis (Classification)

- The dataset was loaded and processed with numpy and pandas, then divided into training and test sets.
- A suite of machine learning models was constructed, with hyperparameters finely tuned via GridSearchCV.
- Model performance was assessed using accuracy as the key metric, with enhancements made through feature engineering and meticulous algorithm tuning.
- The optimal classification model was identified through comparative analysis.
- The link to the notebook: SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

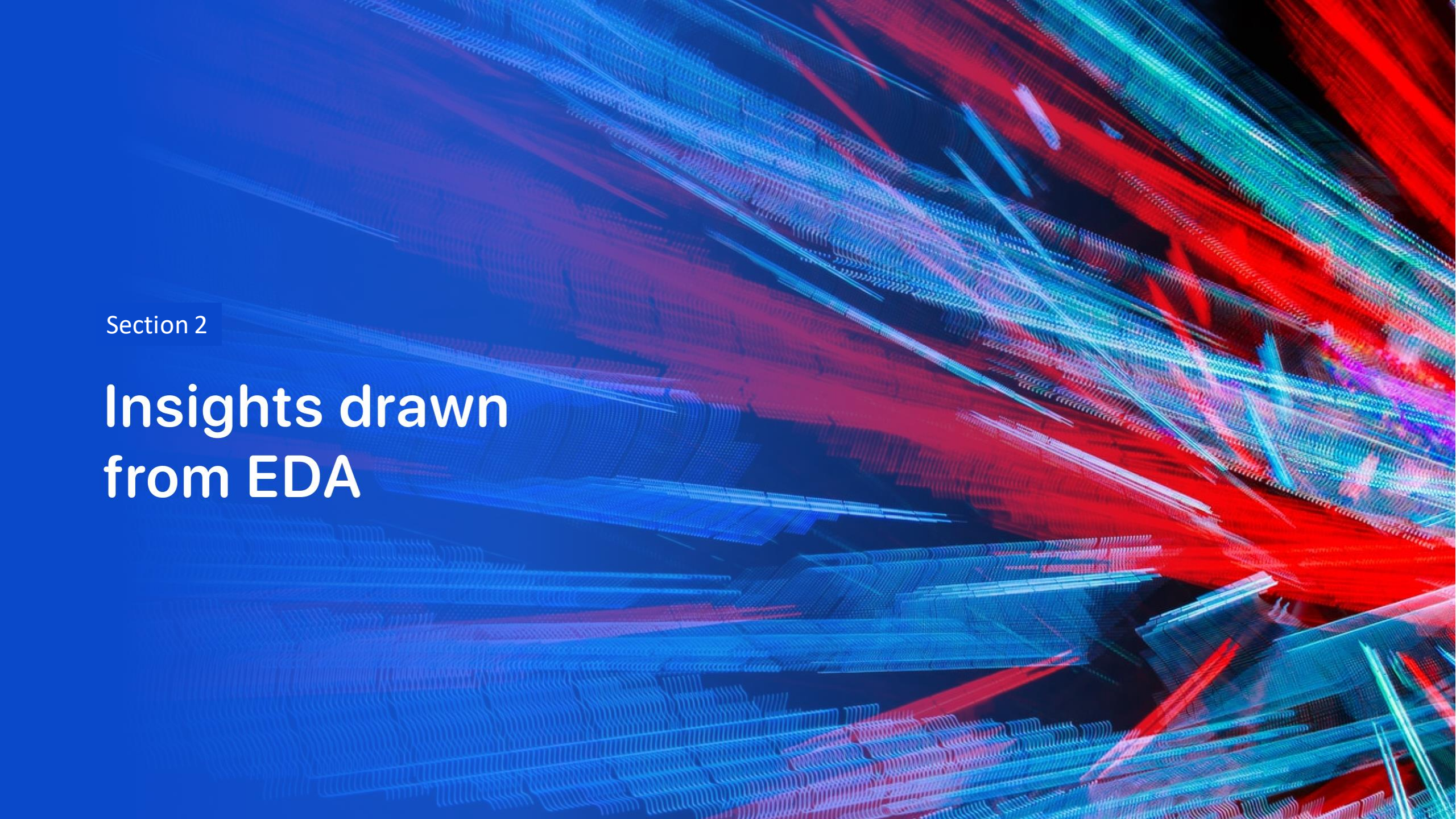
Results

- Exploratory data analysis
  - Over time, launch success rates have improved.
  - KSC LC-39A boasts the highest success rate among launch pads.
  - Certain orbits like ES-L1, GEO, HEO, and SSO achieved a 100% success rate.
- Interactive analytics
  - Launch sites are predominantly positioned near the equator and coastal lines.
  - Strategic placement of launch sites balances safety from populated areas and logistical efficiency for support operations.
- Predictive analysis results
  - The Decision Tree model emerged as the top performer for predicting launch outcomes.



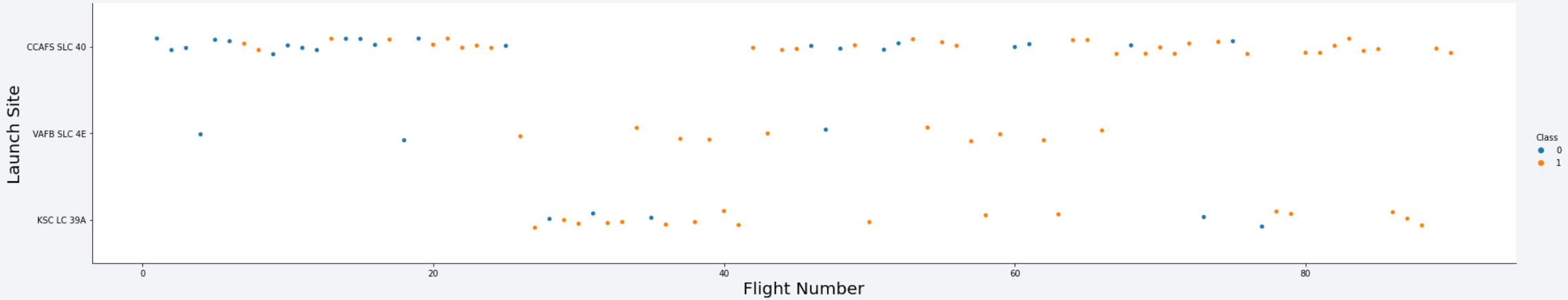
Section 2

Insights drawn from EDA



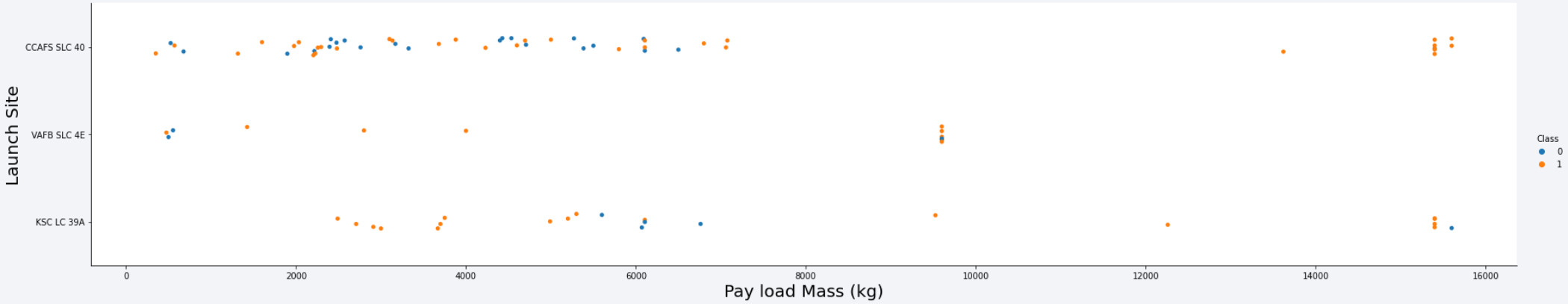
Flight Number vs. Launch Site

- This scatter plot illustrates the relationship between the number of flights conducted and the success rate at each launch site. It suggests that sites with more flights tend to have higher success rates, potentially due to the refinement of processes and increase in experience over time.



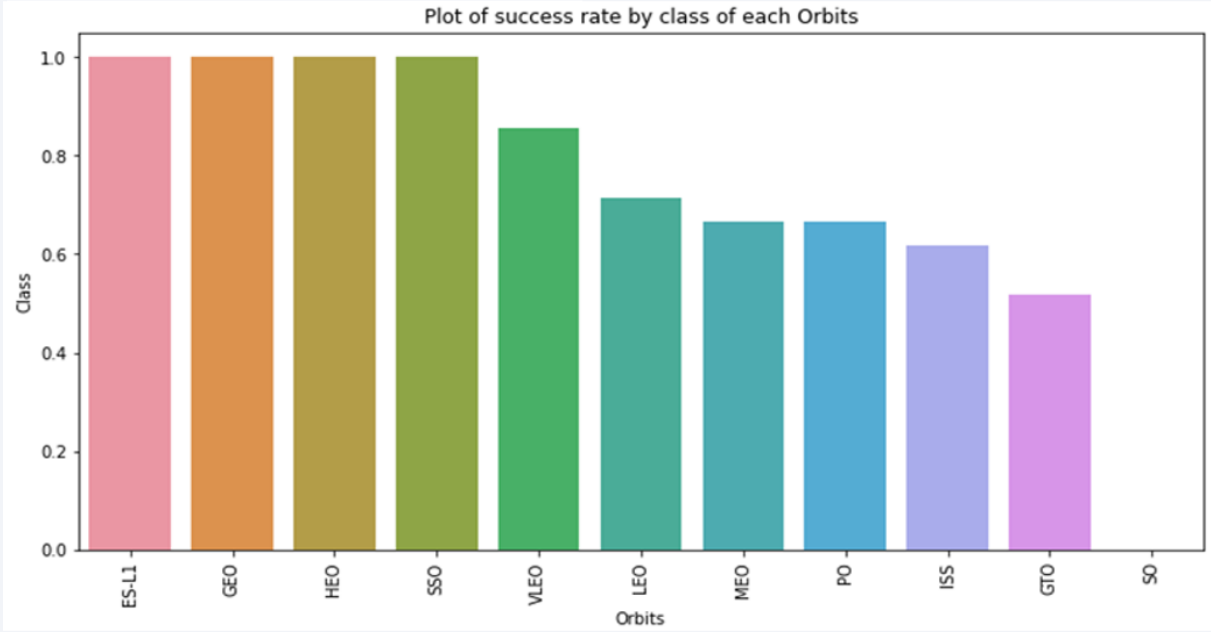
Payload vs. Launch Site

- This scatter plot compares the payload mass to the launch site, particularly focusing on CCAFS SLC 40. The analysis indicates that as the payload mass for launches from this site increases, so does the success rate of the rockets. This trend could suggest that heavier payloads are associated with more successful missions, possibly due to the use of more advanced technology or greater investment in those particular launches.



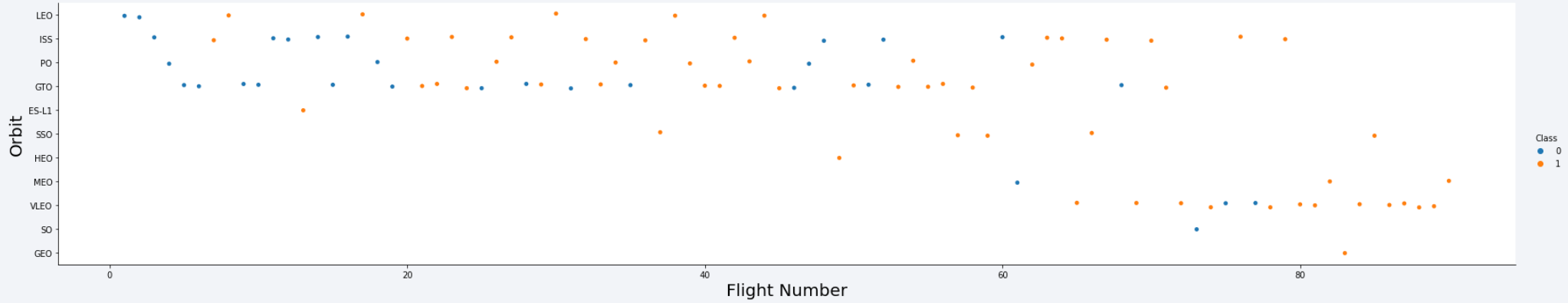
Success Rate vs. Orbit Type

- This bar chart illustrates the success rates across various orbit types. It shows that ES-L1, GEO, HEO, SSO, and VLEO orbits have the highest success rates. Each bar represents a different orbit type, with the height of the bar indicating the proportion of successful missions.



Flight Number vs. Orbit Type

- The scatter plot visualizes the relationship between flight numbers and the types of orbits achieved. It highlights a pattern where success in Low Earth Orbit (LEO) missions correlates with a higher flight frequency. However, this trend does not hold for Geostationary Transfer Orbit (GTO) missions, indicating that the success of flights in GTO is not dependent on the number of flights conducted.

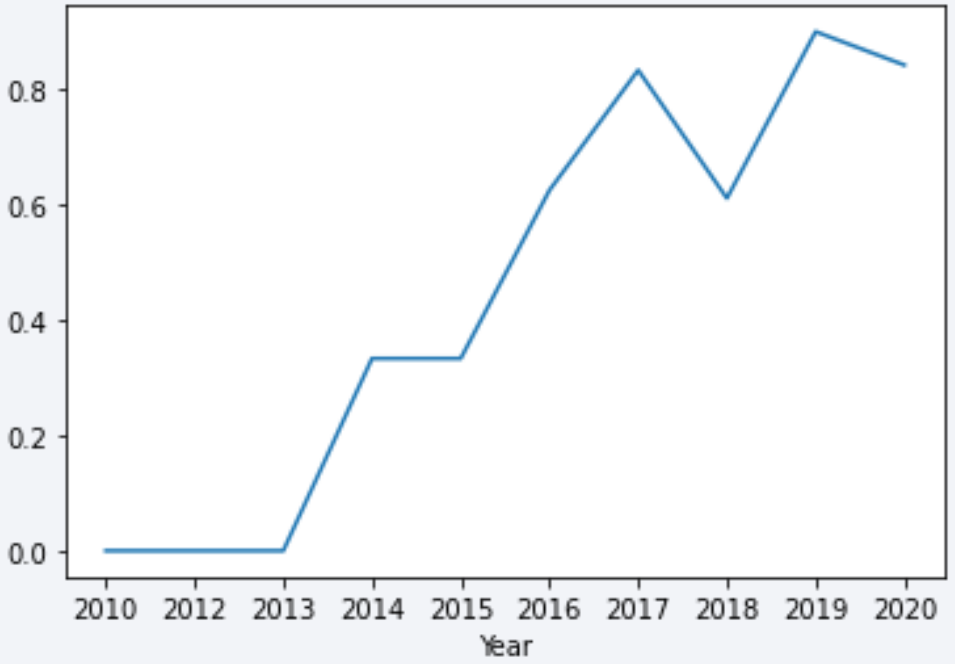


Payload vs. Orbit Type

- This scatter plot illustrates the relationship between the mass of payloads and the types of orbits they are sent to. It indicates that heavier payloads are more frequently associated with successful landings in Polar and Low Earth Orbits, as well as missions to the International Space Station. Each dot represents a specific mission, with the horizontal axis measuring the payload mass and the vertical axis indicating the orbit type.

Launch Success Yearly Trend

- This line chart displays the trend of launch success rates over the years. Starting from 2013, there is a visible increase in the rate of successful launches, continuing through to 2020. The shaded area around the line may represent the confidence interval or variability in the data, emphasizing the general trend of improvement in launch success. This upward trend could reflect advancements in technology, improvements in operational processes, or a combination of factors contributing to space mission success.



All Launch Site Names

- The keyword DISTINCT was employed to extract unique launch site names from the SpaceX dataset.
- A DataFrame displayed the results, showcasing each site's name once.

In [8]:
```
%sql select distinct "Launch_Site" from SPACEXTABLE;
```

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

Out[8]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

- A SQL query with the LIKE operator was utilized to retrieve records starting with 'CCA' from launch site names.
- Five records were displayed, showcasing the initial match criterion.

```
In [9]: %sql select * from SPACEXTABLE where "Launch_Site" like 'CCA%' limit 5;
 * sqlite:///my_data1.db
Done.
```

Out[9]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Lan
6/4/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Fail
12/8/2010	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	Fail
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	
10/8/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	
3/1/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	

Total Payload Mass

- The total mass of payloads launched by NASA boosters was computed as 45,596 kg through an SQL aggregation query.

```
In [10]: %%sql select sum("PAYLOAD_MASS__KG_") as total_payload_mass_launched_by_nasa
         from SPACEXTABLE where "Customer" = "NASA (CRS)";
 * sqlite:///my_data1.db
Done.
Out[10]: total_payload_mass_launched_by_nasa
                                       45596
```

Average Payload Mass by F9 v1.1

- An SQL query was utilized to determine the average payload mass for the Falcon 9 version 1.1 booster, calculated to be 2928.4 kg.

In [11]:
```
%%sql select avg("PAYLOAD_MASS__KG_") as avg_payload_mass_carried_by_F9v11
from SPACEXTABLE where Booster_Version = "F9 v1.1";
```

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

Out[11]:

avg_payload_mass_carried_by_F9v11
2928.4

First Successful Ground Landing Date

- The first landing outcome on ground pad was 1/8/2018.

```
In [12]: %%sql select min("Date") as first_successfull_landing_on_ground_pad from SPACEXTABLE
         where "Landing_Outcome" = "Success (ground pad)";

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

Out[12]:

first_successfull_landing_on_ground_pad
1/8/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

- The WHERE clause was implemented to select boosters that achieved a successful drone ship landing carrying payloads between 4000 and 6000 kg.

In [13]:
```
%%sql select "Booster_Version" from SPACEXTABLE
where "Landing_Outcome" = "Success (drone ship)" and "PAYLOAD_MASS__KG_"<6000
and "PAYLOAD_MASS__KG_">4000;
```

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

Out[13]:

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

Total Number of Successful and Failure Mission Outcomes

- SQL's LIKE wildcard was utilized to count successful and failed missions, revealing 100 successes and 1 failure.

```
In [14]: %sql select count(*) as success from SPACEXTABLE where "Mission_Outcome" like '%Success%';
 * sqlite:///my_data1.db
Done.
Out[14]: success
         100

In [15]: %sql select count(*) as failure from SPACEXTABLE where "Mission_Outcome" like '%Failure%';
 * sqlite:///my_data1.db
Done.
Out[15]: failure
         1
```

Boosters Carried Maximum Payload

- The maximum payload each booster has carried was identified using a subquery with the MAX() function within the WHERE clause.

```
In [16]: %%sql select "Booster_Version" from SPACEXTABLE
         where "PAYLOAD_MASS__KG_" = (select max("PAYLOAD_MASS__KG_") from SPACEXTABLE);
```

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

Out[16]:

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

- SQL queries with WHERE, LIKE, AND, and BETWEEN were applied to extract failed drone ship landing data for 2015, including month, booster versions and launch sites.

In [19]:
```
%%sql select substr(Date, 4, 2) as Month, Booster_Version, Launch_Site, Landing_Outcome
from SPACEXTABLE where substr(Date, 7, 4) = '2015'
and Landing_Outcome = 'Failure (drone ship)'
```

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

Out[19]:

Month	Booster_Version	Launch_Site	Landing_Outcome
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- Landing outcomes were counted and ranked using SQL clauses WHERE, GROUP BY, and ORDER BY for the specified date range.

```
%%sql select "Landing_Outcome", count("Landing_Outcome") as "Outcome_Count"
from SPACEXTABLE where STR_TO_DATE(Date, '%d/%m/%Y') between '2010-06-04'
and '2017-03-20' group by "Landing_Outcome" order by "Outcome_Count" DESC;
```

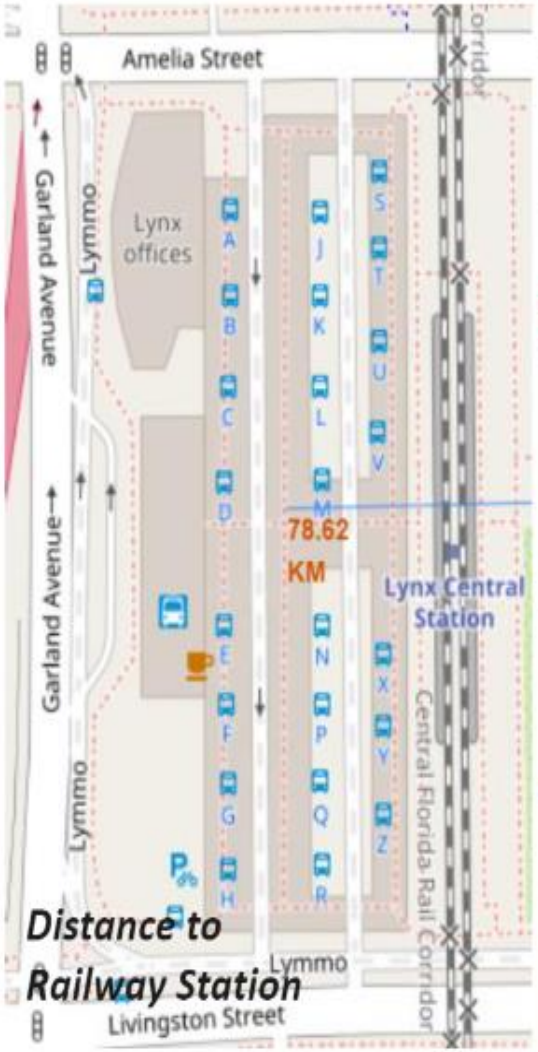
landingoutcome	count
No attempt	10
Success (drone ship)	6
Failure (drone ship)	5
Success (ground pad)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1


Section 3
Launch Sites
Proximities Analysis

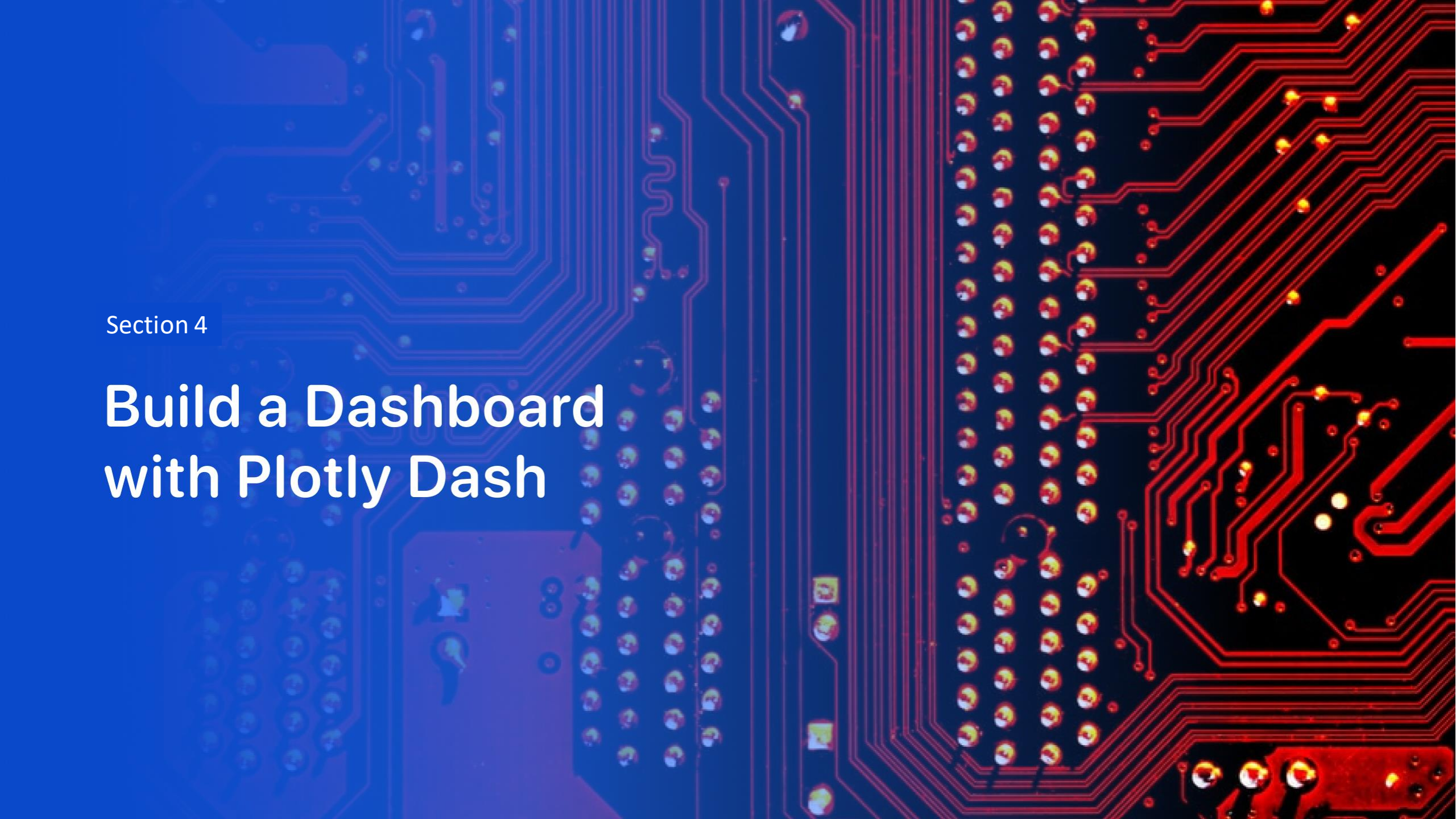
All launch sites global map markers

Markers showing launch sites with color labels

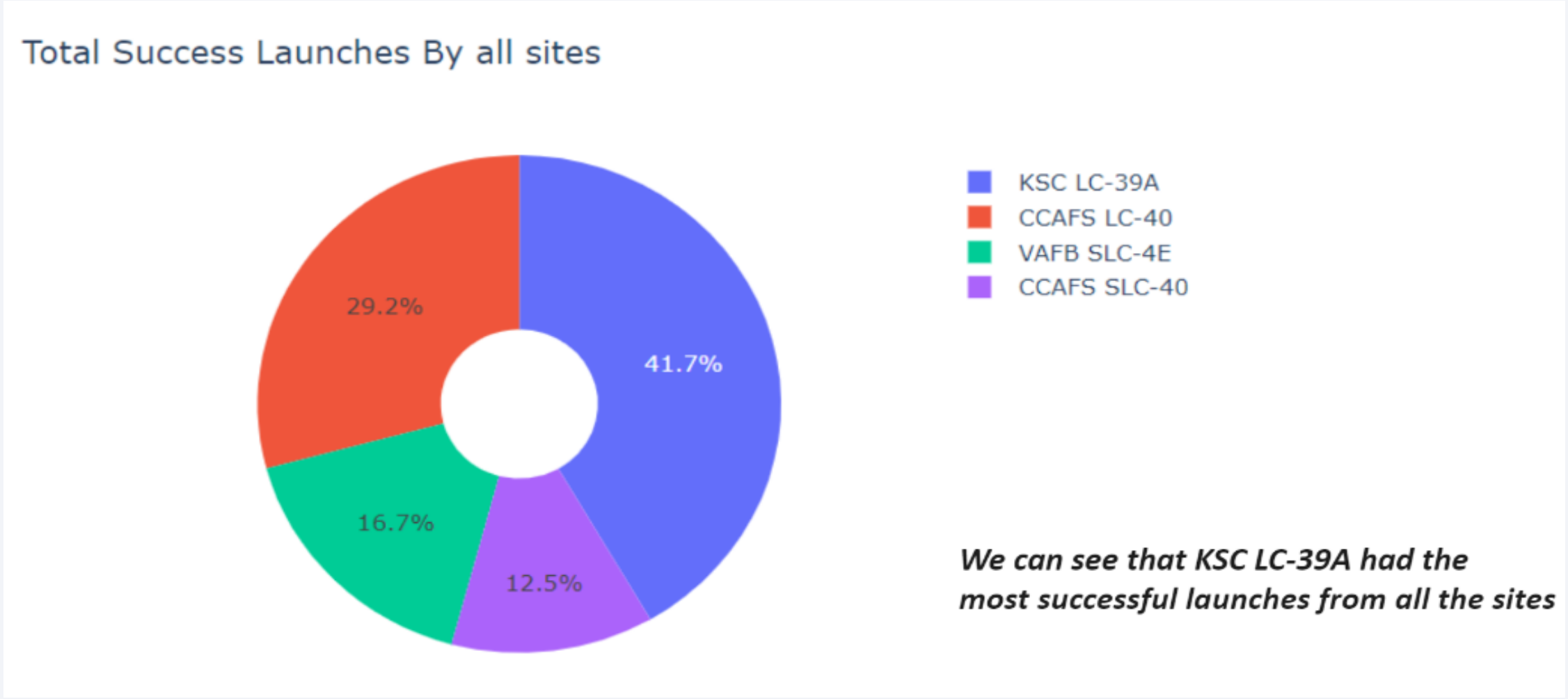
Launch Site distance to landmarks



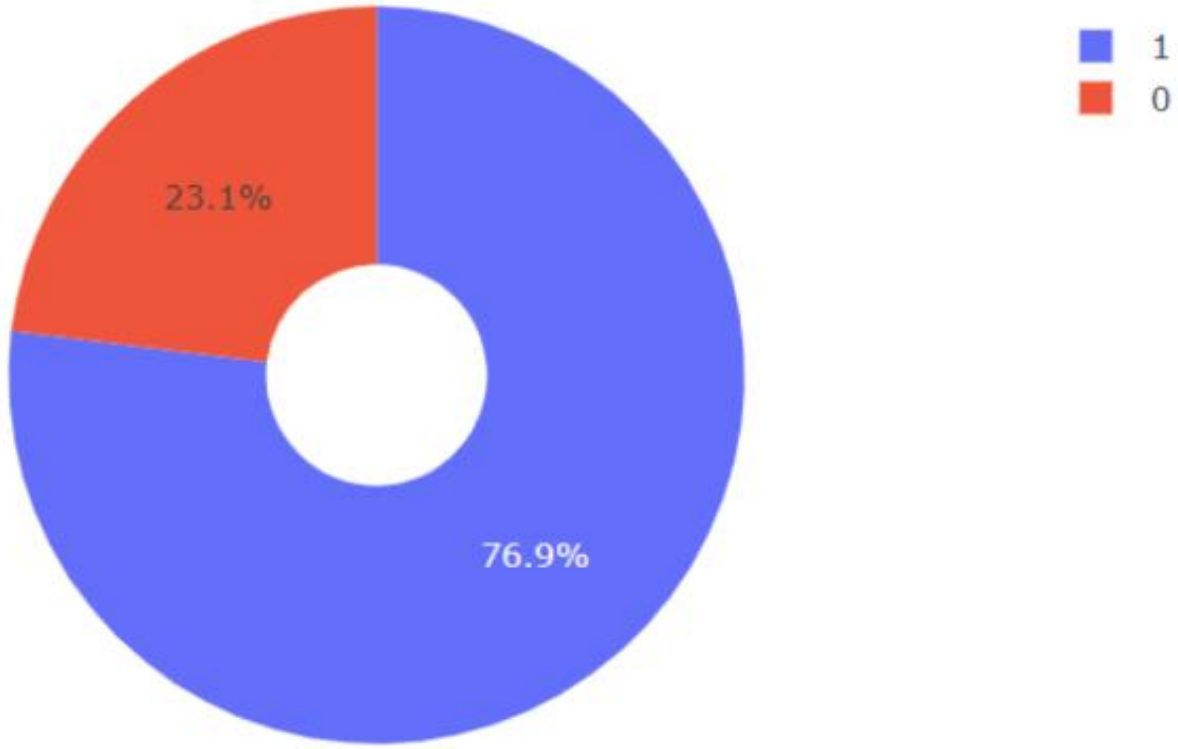
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site

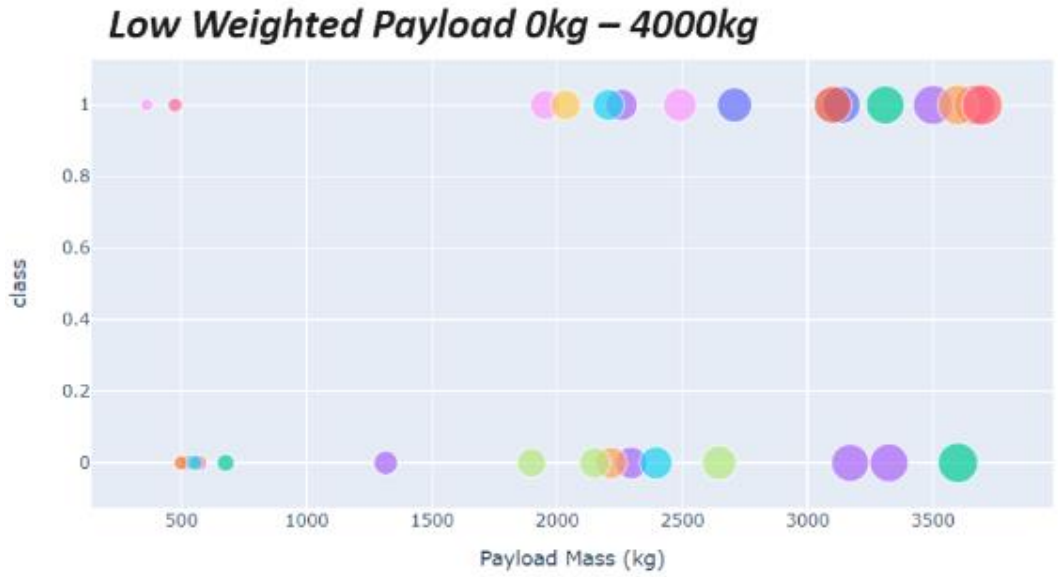


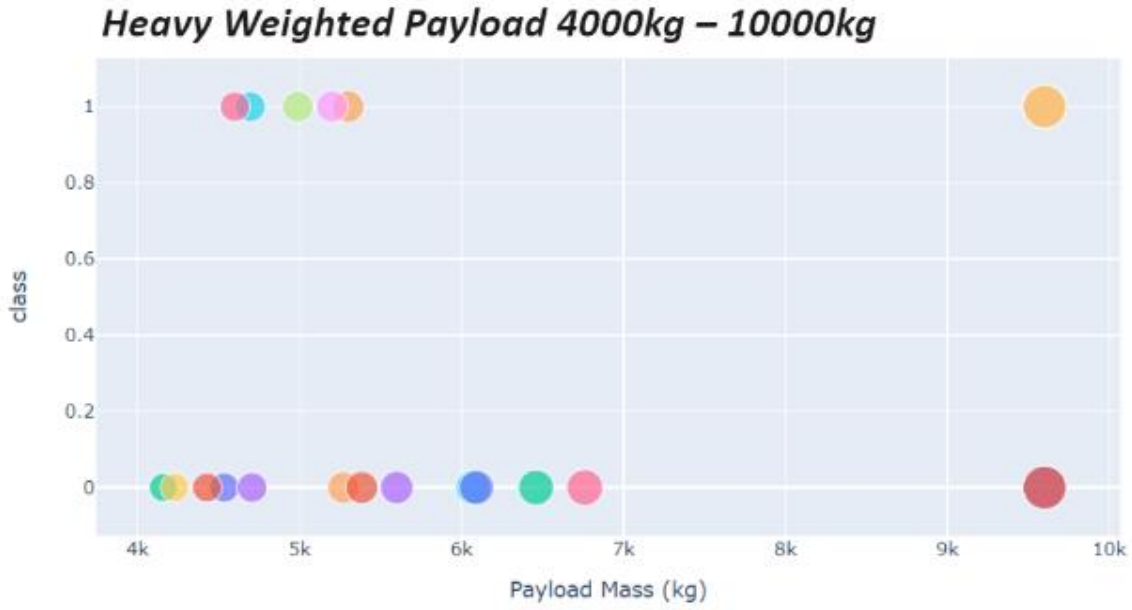
Pie chart showing the Launch site with the highest launch success ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider





We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

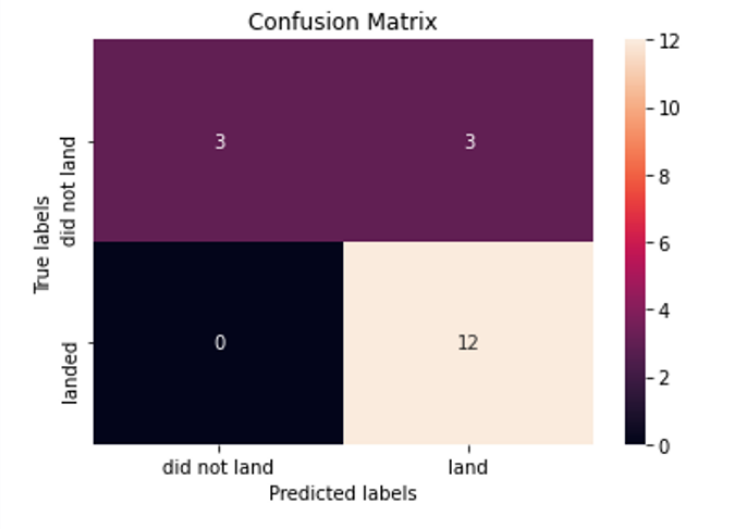
•The decision tree classifier is the model with the highest classification accuracy

```
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])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

- The classifier's confusion matrix reveals its capability to differentiate between classes, with the primary issue being false positives, where unsuccessful landings were incorrectly labeled as successful.



Conclusions

- A positive correlation exists between the number of flights at a launch site and its success rates, suggesting operational efficiency improves with experience.
- Launch success rates have shown an upward trend from 2013 to 2020, possibly reflecting advancements in spaceflight technology and processes.
- Certain orbits such as ES-L1, GEO, HEO, SSO, and VLEO have achieved the highest success rates, indicating their favorable conditions or the effectiveness of missions planned to these orbits.
- KSC LC-39A has demonstrated superior performance with the most successful launches, highlighting it as a significant site for space missions.
- Analysis reveals that heavier payloads are more often associated with successful missions, especially in Polar and Low Earth Orbits, which may be indicative of the capabilities of the Falcon 9 booster.
- The Decision Tree classifier has proven to be the best machine learning algorithm for predicting launch success in this analysis, although it is important to consider the impact of false positives on the overall effectiveness of the model.
- Throughout the various analyses, data-driven insights have been crucial in understanding the factors that contribute to the success of space missions.

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!