IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sonam Phuntsho
22/06/2024

Outline

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

Executive Summary

- Summary of methodologies
  - 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
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- ## Project background and context

  SpaceX offers Falcon 9 rocket launches for $62 million, significantly cheaper than the $165 million charged by other providers, largely due to the reusability of the Falcon 9's first stage. Predicting whether the first stage will land successfully can help other companies competitively bid against SpaceX for rocket launches. The project's goal is to develop a machine learning pipeline to predict the successful landing of the Falcon 9's first stage.

- ## Problems you want to find answers

  - What factors determine if the rocket will land successfully?
  - The interaction among various features that influence the success rate of a landing.
  - The operating conditions required to ensure a successful landing.

Section 1

Methodology

Methodology

Executive Summary

Data Collection Methodology

- **SpaceX API**: Retrieved specific data points related to SpaceX launches.
- **Web Scraping**: Extracted additional information from Wikipedia.

Data Wrangling

- Cleaned and pre-processed data.
- Handled missing values and inconsistencies.
- Normalized formats and transformed data types.

Exploratory Data Analysis (EDA)

- Used visualizations and SQL.
- Generated descriptive statistics.
- Visualized distributions, correlations, and identified outliers.

Interactive Visual Analytics

- **Folium**: Created geospatial visualizations for launch sites and trajectories.
- **Plotly Dash**: Developed customizable, interactive plots for in-depth data exploration.

Predictive Analysis Using Classification Models

- **Building Models**: Developed decision trees, random forests, logistic regression models.
- **Tuning Models**: Optimized parameters for enhanced performance.
- **Evaluating Models**: Assessed models using accuracy, precision, recall, F1 score.

Data Collection

Data Collection and Processing

- API Requests: Collected data using GET requests to the SpaceX API.
- Data Conversion: Decoded JSON response with the `.json()` function and transformed it into a pandas dataframe using `.json_normalize()`.
- Data Cleaning: Checked for and filled in missing values to ensure data completeness.
- Web Scraping: In additional used BeautifulSoup to extract Falcon 9 launch records from Wikipedia. Parsed HTML tables and converted them into pandas dataframes for comprehensive analysis.

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting
- Link to the notebook on GitHub:

[-IBM-Data-Science-Capstone-SpaceX/01.Spacex-data-collection-api.ipynb at main · szphunt/-IBM-Data-Science-Capstone-SpaceX (github.com)]

Now let's start requesting rocket launch data from SpaceX API with the following URL:

In [83]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [84]:
```
response = requests.get(spacex_url)
```

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

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

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

```
# Get the head of the dataframe
df.head()
```

Data Collection - Scraping

- We utilized web scraping techniques to gather Falcon 9 launch records using BeautifulSoup.
- We extracted the data from the table and transformed it into a pandas dataframe.
- Link to github notebook:

[-IBM-Data-Science-Capstone-SpaceX/02. SpaceX_Webscraping.ipynb at main · szphunt/-IBM-Data-Science-Capstone-SpaceX (github.com)]

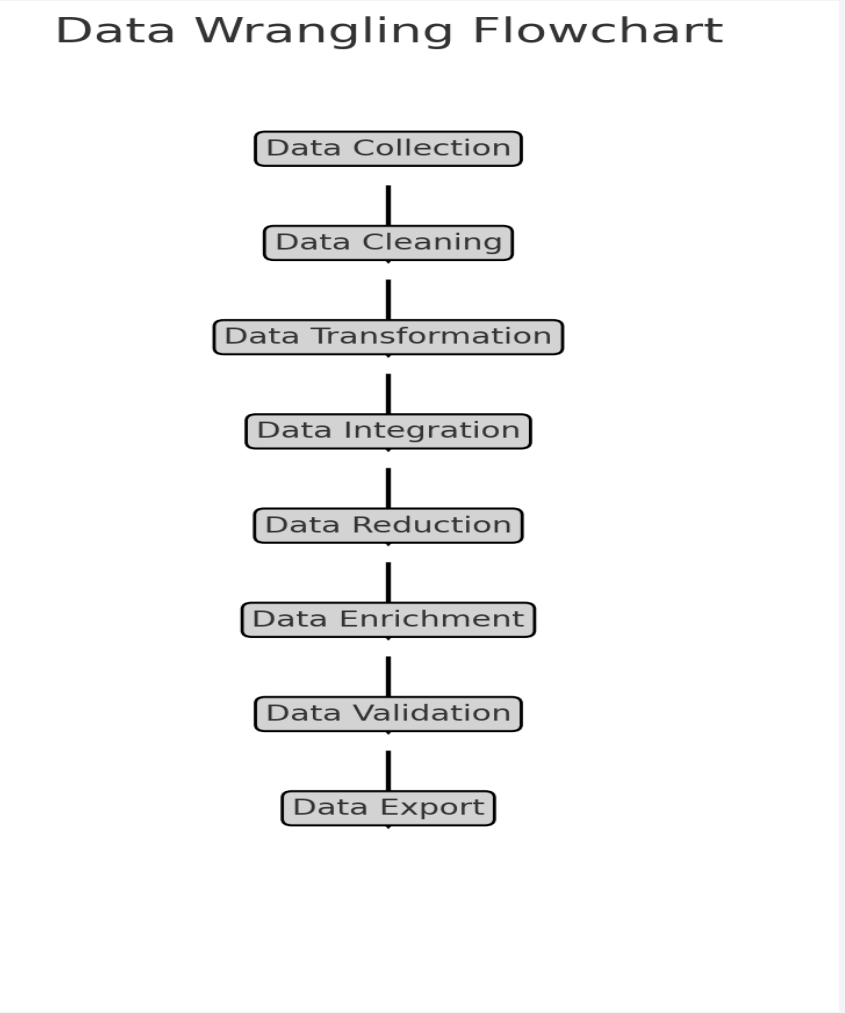
Data collection Flowchart

```
A[Start] --> B[Send HTTP Request to Website]
B --> C[Receive HTML Response]
C --> D[Parse HTML with BeautifulSoup]
D --> E[Locate and Extract Target Data]
E --> F[Convert Data into pandas DataFrame]
F --> G[Store or Process Data]
G --> H[End]
```

Data Wrangling

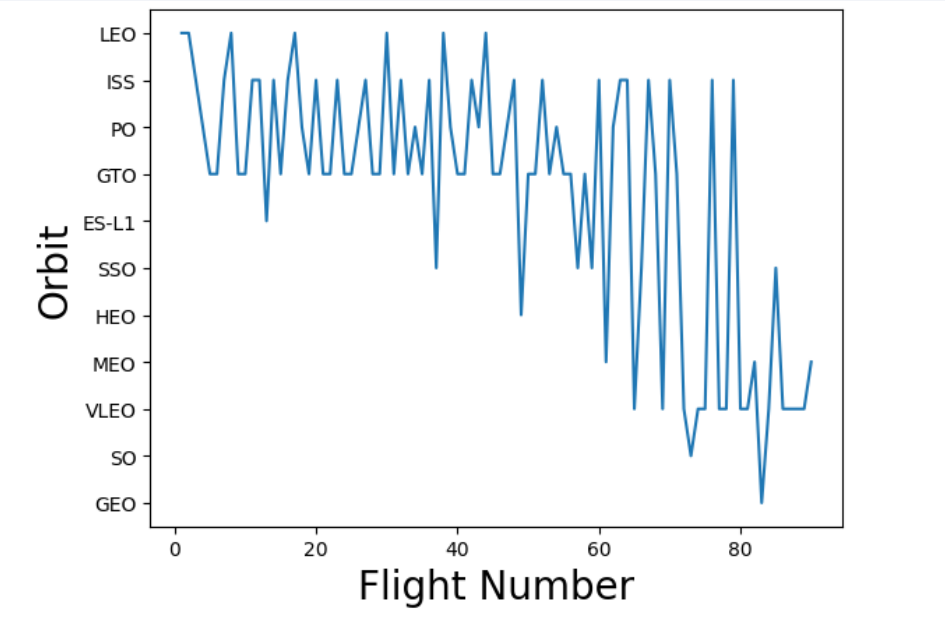
- We conducted exploratory data analysis and identified the training labels.
- We computed the number of launches at each site, as well as the frequency and count of each orbit type.
- We generated the landing outcome label from the outcome column and saved the results to a CSV file.
- Link to GitHub for data wrangling notebook:

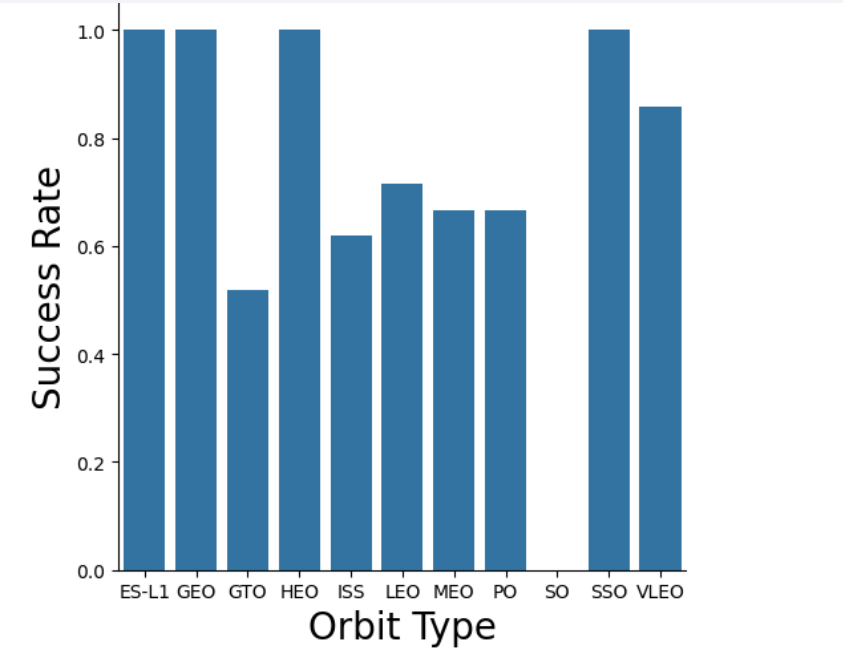
-IBM-Data-Science-Capstone-SpaceX/03. Spacex-Data wrangling.ipynb at main · szphunt/-IBM-Data-Science-Capstone-SpaceX (github.com)



EDA with Data Visualization

- We analyzed the data by visualizing the relationships between various factors: flight number and launch site, payload and launch site, success rate for each orbit type, flight number and orbit type, and the yearly trend in launch success.





- Link to github for EDA with Data Visualization notebook:

[-IBM-Data-Science-Capstone-SpaceX/04. EDA with Data Visualization.ipynb at main · szphunt/-IBM-Data-Science-Capstone-SpaceX (github.com)]

EDA with SQL

- We imported the SpaceX dataset directly into a PostgreSQL database within the Jupyter notebook environment.
- We conducted Exploratory Data Analysis (EDA) using SQL to derive insights from the data. Our SQL queries revealed, for example:
- The names of the unique launch sites used in space missions.
- The total payload mass carried by boosters launched under NASA's CRS program.
- The average payload mass carried by the F9 v1.1 booster version.
- The total count of successful and failed mission outcomes.
- The failed landing outcomes on drone ships, including their booster versions and launch site names.
- Link to github for EDA with SQL Notebook:

  -IBM-Data-Science-Capstone-SpaceX/05. EDA with SQL.ipynb at main · szphunt/-IBM-Data-Science-Capstone-SpaceX (github.com)

Build an Interactive Map with Folium

- We marked all launch sites on a Folium map and added map objects like markers, circles, and lines to indicate the success or failure of launches at each site.
- We categorized launch outcomes as 0 for failure and 1 for success.
- Using color-coded marker clusters, we identified which launch sites had relatively high success rates.
- We calculated the distances from each launch site to nearby features and answered questions such as:
  - Are launch sites near railways, highways, and coastlines?
  - Do launch sites maintain a certain distance from cities?
- We added these objects to the map to visually represent and analyze the data more effectively:
  - Markers, circles, and lines: These indicate the locations of launch sites and the success or failure of launches, providing a clear and immediate visual understanding of where launches occurred and their outcomes.
  - Color-labeled marker clusters: These help quickly identify which launch sites have higher success rates, making it easier to assess performance at a glance.
  - Distance calculations: These allow us to explore the geographical context of launch sites, such as their proximity to railways, highways, coastlines, and cities, which can be crucial for logistical, safety, and strategic planning.
- Link to github Interactive Map with folium Notebook:

  IBM-Data-Science-Capstone-SpaceX/06. Interactive Map with Folium.ipynb at main · szphunt/-IBM-Data-Science-Capstone-SpaceX (github.com)

Build a Dashboard with Plotly Dash

- We developed an interactive dashboard using Plotly Dash.
- We created pie charts displaying the total launches from specific sites.
- We generated scatter plots illustrating the relationship between Outcome and Payload Mass (Kg) across various booster versions.
- This dashboard was created to complete visual analysis and answer the following questions:
  - Which site has the largest successful launches?
  - Which site has the highest launch success rate?
  - Which payload range(s) has the highest launch success rate?
  - Which payload range(s) has the lowest launch success rate?
  - Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highestlaunch success rate

Link to github: szphunt/-IBM-Data-Science-Capstone-SpaceX (github.com)

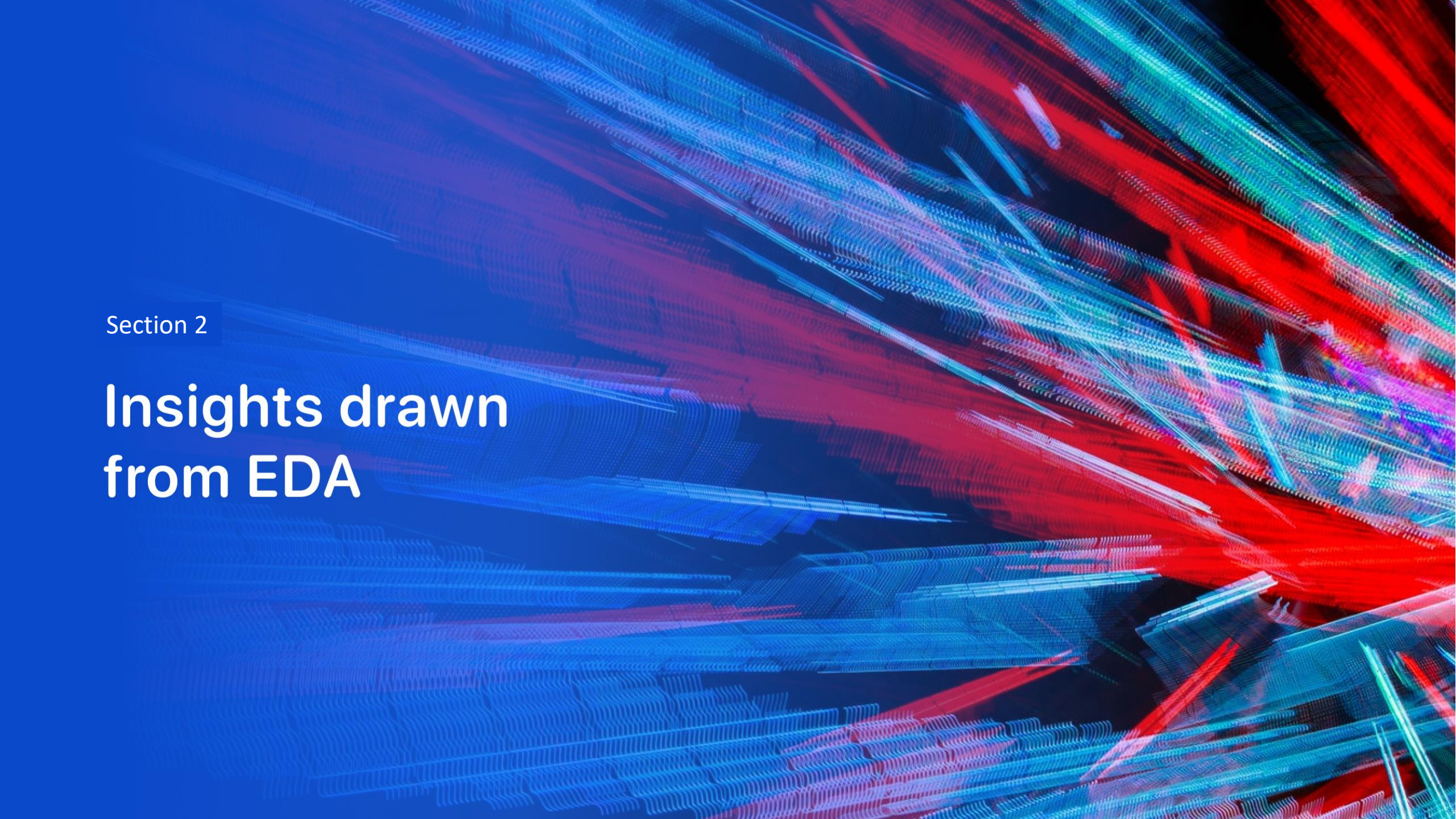
Predictive Analysis (Classification)

- We imported the data using numpy and pandas, processed the data, partitioned our dataset into training and testing subsets.
- We constructed various machine learning models and optimized different hyperparameters using GridSearchCV.
- We evaluated our model's performance using accuracy as the primary metric, enhanced the model through feature engineering and algorithm refinement.
- We identified the top-performing classification model.
- Link to github notebook for predictive analysis:

-IBM-Data-Science-Capstone-SpaceX/07. SpaceX_Machine Learning Prediction.ipynb at main · szphunt/-IBM-Data-Science-Capstone-SpaceX (github.com)

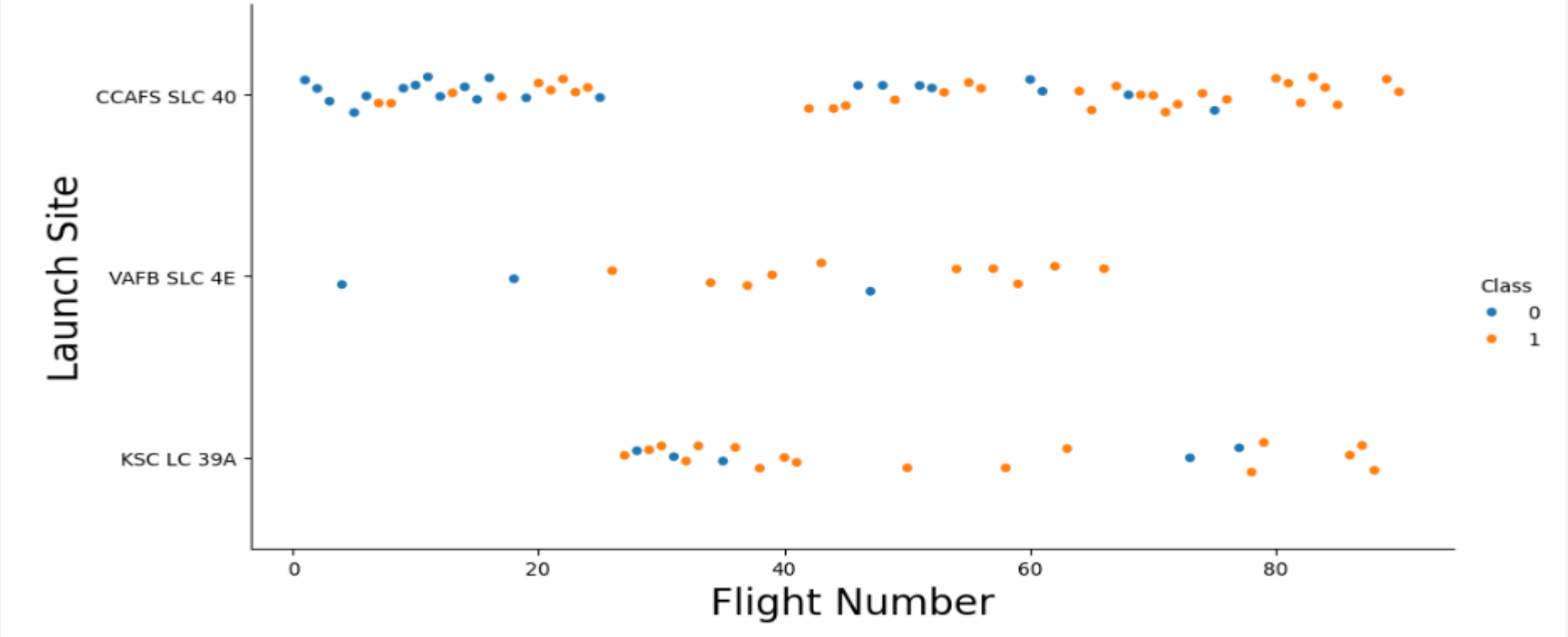
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

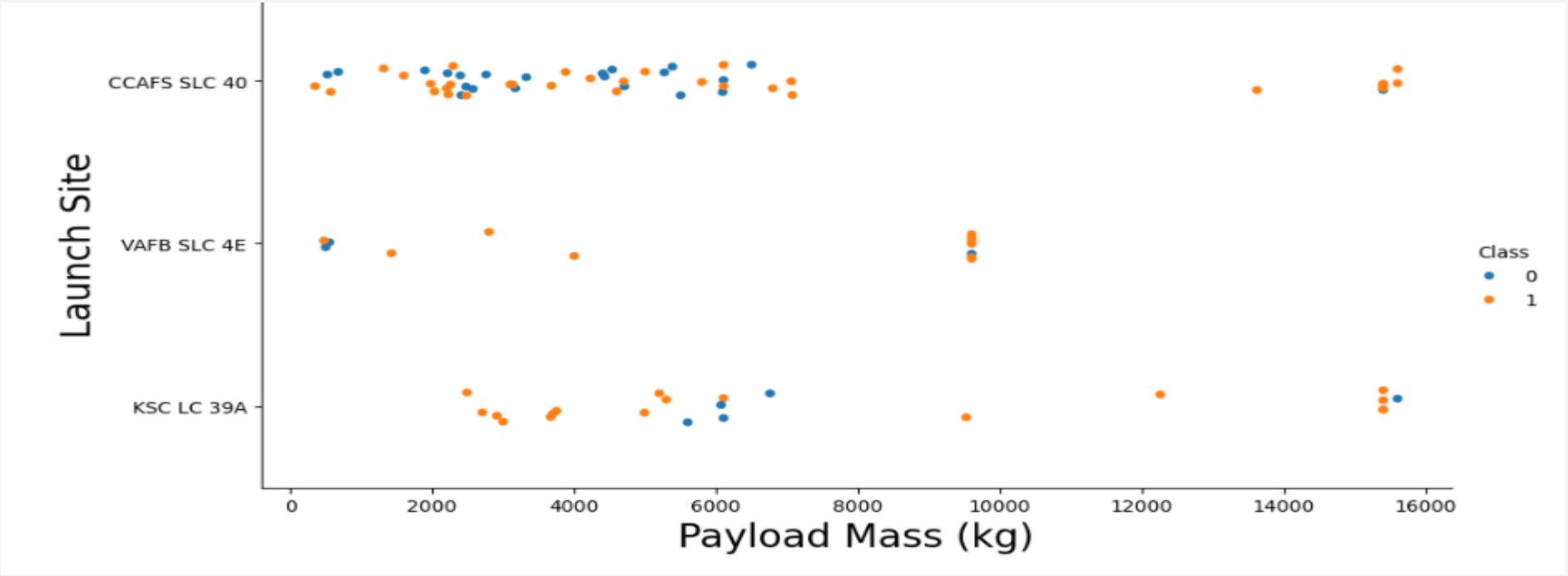
Flight Number vs. Launch Site

- From the plot, it became evident that higher flight volumes at a launch site correlate positively with increased launch success rates at that site.



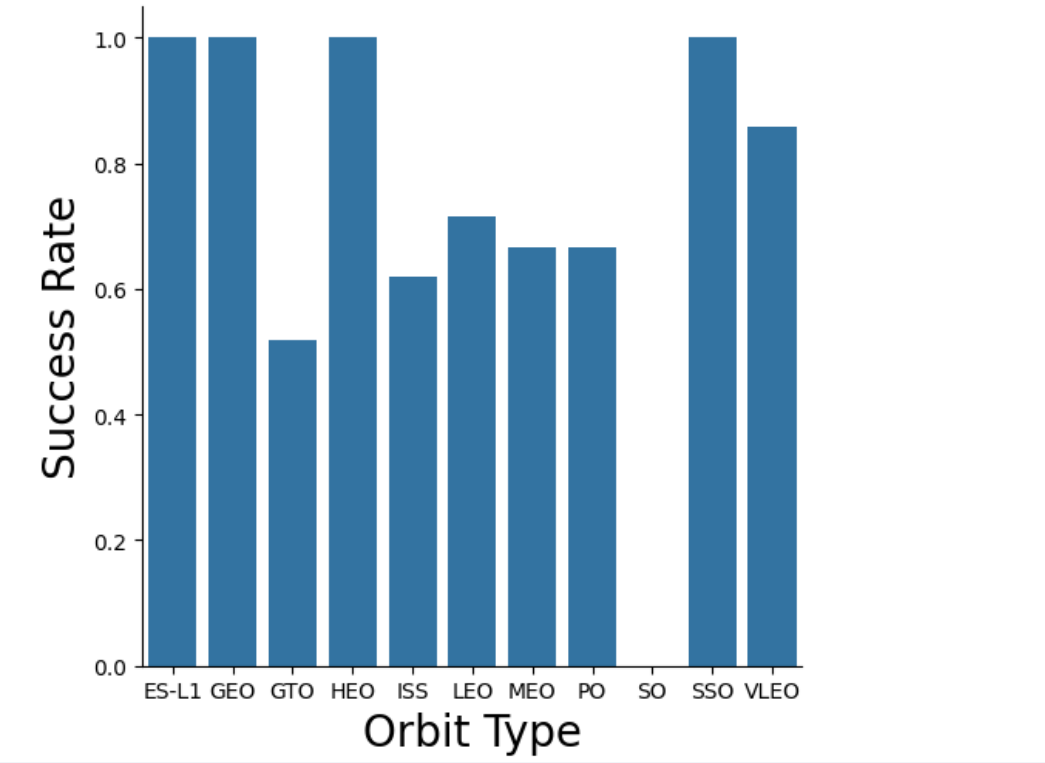
Payload vs. Launch Site

- The higher the payload mass at CCAFS SLC 40, the greater the success rate for the rocket. Additionally, upon examining the scatter plot of Payload versus Launch Site, it is evident that no rockets were launched with heavy payload masses (greater than 10000) from the VAFB-SLC launch site.



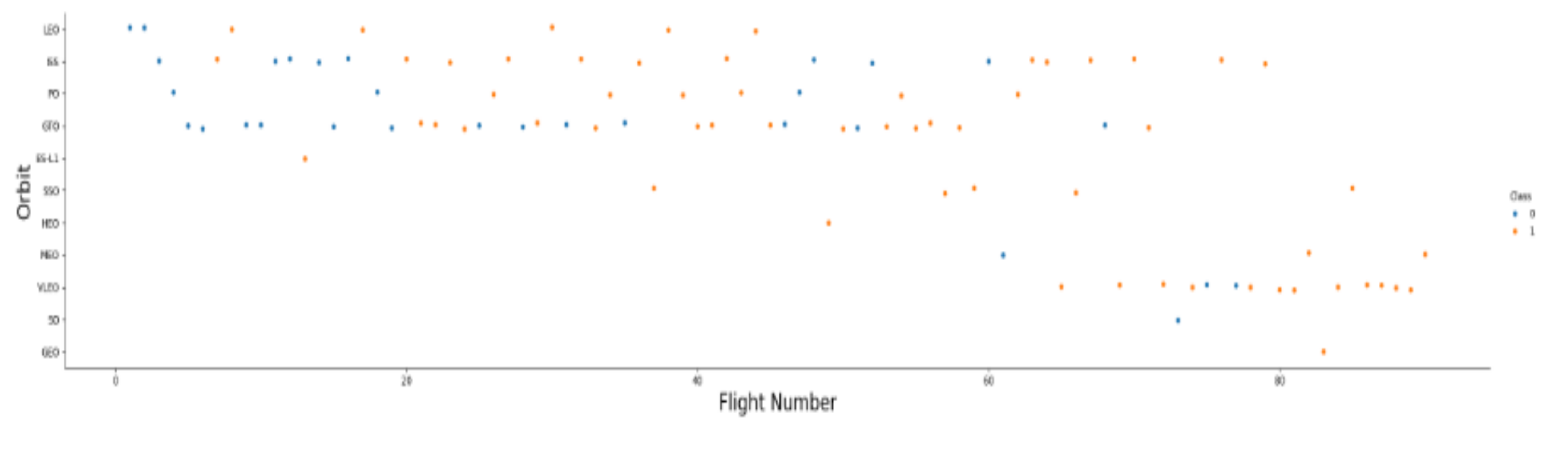
Success Rate vs. Orbit Type

- From the plot, it is clear that missions to Earth-Sun Lagrange Point 1 (ES-L1), Geostationary Earth Orbit (GEO), High Earth Orbit (HEO), Sun-Synchronous Orbit (SSO), and Very Low Earth Orbit (VLEO) achieved the highest success rates compared to other orbital destinations. This suggests that launches targeting these specific orbits experienced consistently successful outcomes, indicating robust reliability and suitability for missions to these orbital destinations.



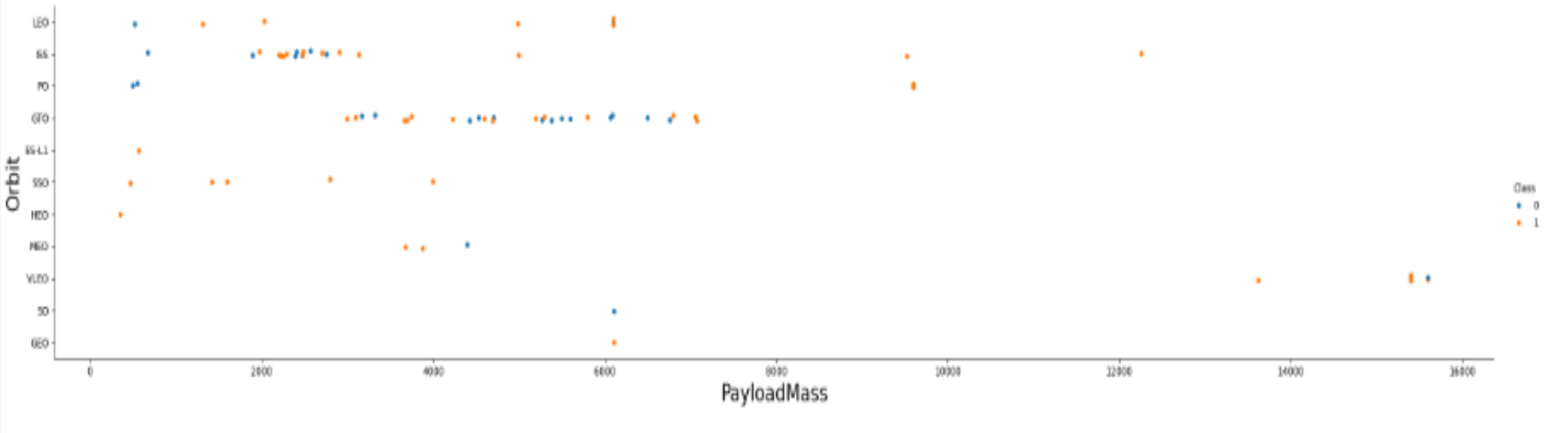
Flight Number vs. Orbit Type

- In the Low Earth Orbit (LEO), the success rate of missions appears to increase with the number of flights observed, suggesting a possible correlation between flight experience and mission success in this orbital regime. In contrast, for missions aiming for Geostationary Transfer Orbit (GTO), the plot indicates no apparent relationship between the number of flights and mission success rates. This disparity suggests that factors other than flight experience might play a more crucial role in determining mission success in GTO missions, such as launch vehicle capabilities or payload specifications.



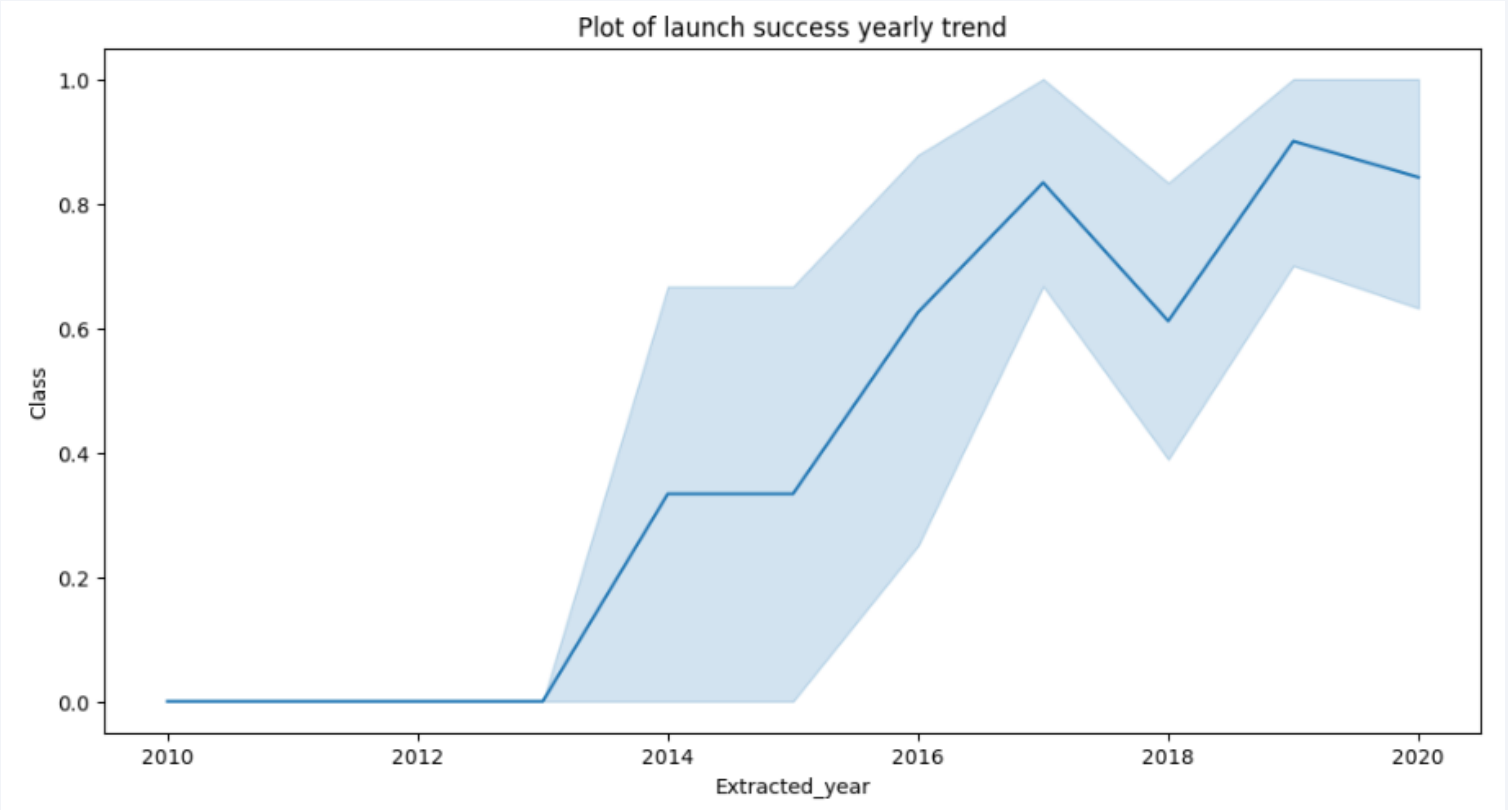
Payload vs. Orbit Type

- With heavy payloads, missions to Polar orbits, Low Earth Orbit (LEO), and the International Space Station (ISS) show higher rates of successful or positive landings. In contrast, distinguishing between successful and unsuccessful landings is less clear for missions to Geostationary Transfer Orbit (GTO), where both outcomes occur frequently.



Launch Success Yearly Trend

- From the plot, we can observe that the success rate of the rockets kept on increasing from 2013 to 2020.



All Launch Site Names

- In order to determine the all the launch site names in the data, we used the DISTINCT keyword to print out the names.

Task 1

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

In [34]:
```
%sql Select Distinct Launch_Site from SPACEXTABLE
```

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

Out[34]:

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

Launch Site Names Begin with 'CCA'

- We used the query given in the image below to extract names of the launch site beginning with ‘CCA’

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

[35]:
```
%sql SELECT * \
    FROM SPACEXTBL \
    WHERE LAUNCH_SITE LIKE'CCA%' LIMIT 5;
```

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

[35]:

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

Total Payload Mass

- This SQL query below was used to calculates the total payload mass in kilograms for missions where the customer is NASA (CRS) from the SPACEXTBL table. The total mass carried by boosters from NASA is found to be 45596.

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) \
    FROM SPACEXTBL \
    WHERE CUSTOMER = 'NASA (CRS)';
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- The SQL query below was used to calculate the average payload mass in kilograms carried by the booster version ‘F9 v1.1’ from SPACEXTBL table. The average mass is found to be 2928.4

```
In [12]: %sql SELECT AVG(PAYLOAD_MASS__KG_) \
             FROM SPACEXTBL \
             WHERE BOOSTER_VERSION = 'F9 v1.1';
          * sqlite:///my_data1.db
         Done.
Out[12]: AVG(PAYLOAD_MASS__KG_)
         2928.4
```

First Successful Ground Landing Date

- From the query it is found that the first successful landing outcome on the ground pad was on 22nd December 2015.

```
[11]: %sql SELECT MIN(DATE) \
      FROM SPACEXTBL \
      WHERE Landing_Outcome = 'Success (ground pad)'

       * sqlite:///my_data1.db
      Done.
[11]: MIN(DATE)

      2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- We employed the WHERE clause to select boosters that have successfully landed on a drone ship and applied the AND condition to identify successful landings with a payload mass between 4000 and 6000 kilograms.

In [54]:
```
%%sql SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_, Landing_Outcome  FROM SPACEXTBL
where 4000 < PAYLOAD_MASS__KG_ and PAYLOAD_MASS__KG_ < 6000 and Landing_Outcome = 'Success (drone ship)'
```

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

Out[54]:

Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- We used the '%' wildcard to filter for records in the WHERE clause where the Mission Outcome indicated either a success or a failure. This wildcard allows for matching any characters that follow 'success' or 'failure' in the Mission Outcome column.

List the total number of successful and failure mission outcomes

In [56]:
```
%%sql SELECT Mission_Outcome, count(Mission_Outcome) as "Total" FROM SPACEXTBL
Group by Mission_Outcome
```

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

Out[56]:

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

Boosters Carried Maximum Payload

- We identified the booster that carried the highest payload by using a subquery within the WHERE clause along with the MAX() function.

In [59]:
```
%%sql SELECT Distinct Booster_version, PAYLOAD_MASS__KG_ FROM SPACEXTBL
where PAYLOAD_MASS__KG_ = (Select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

Out[59]:

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

- We utilized a combination of the WHERE clause, LIKE operator, AND operator, and BETWEEN operator to filter for unsuccessful landings on drone ships, including their corresponding booster versions and launch site names specifically for the year 2015.

```
[12]: %%sql Select Landing_Outcome, Booster_Version, Launch_Site, DATE from SPACEXTBL
      where Landing_Outcome LIKE 'Failure (drone ship)' AND Date between '2015-01-01' AND '2015-12-31'

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

[12]:

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

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

- We retrieved landing results and their respective counts from the dataset, applying a WHERE clause to specify results between June 4, 2010, and March 20, 2017. We grouped these landing outcomes using the GROUP BY clause and sorted them in descending order with the ORDER BY clause.

In [68]:
```
%%sql select landing_outcome, count(landing_outcome) as "Total" from SpacexTBL
where DATE between '2010-06-04' and '2017-03-20'
group by landing_outcome
order by "Total" desc
```

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

Out[68]:

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

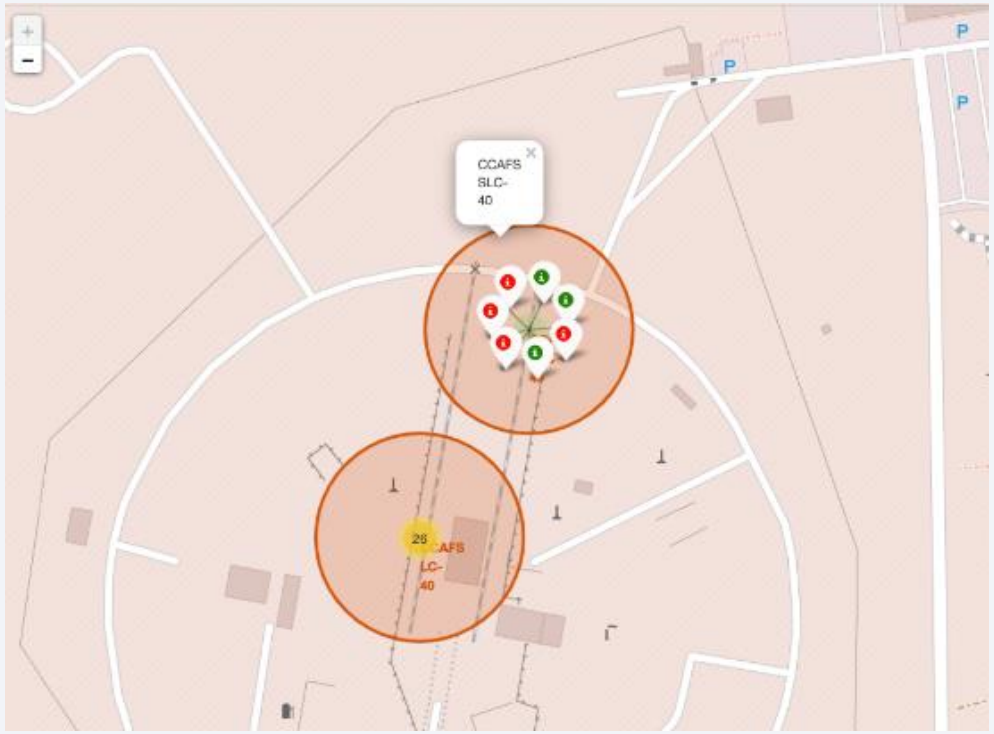
Section 3
Launch Sites
Proximities Analysis

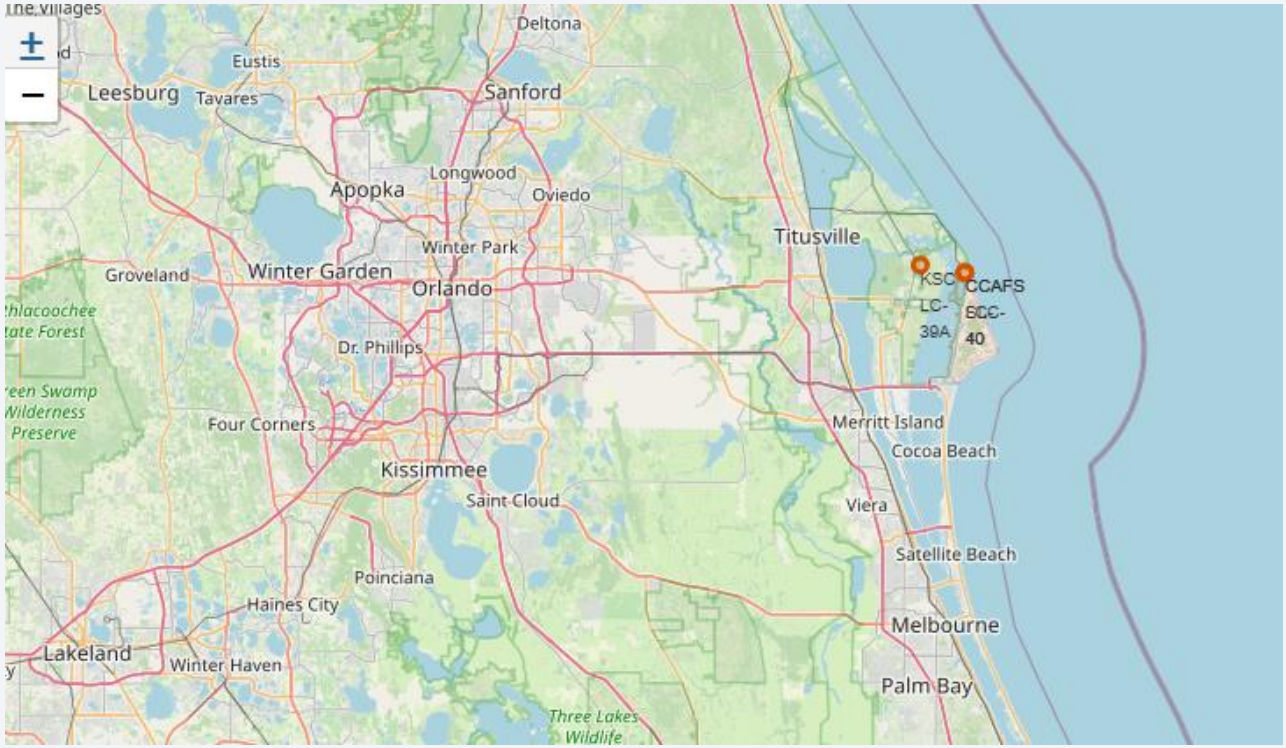
All Launch Site Global Map Markers

- We can see from the map below that all SpaceX launch sites are located at the coast of Florida and California Coastlines in The United States of America.

Launch site with markers depicting success or failure

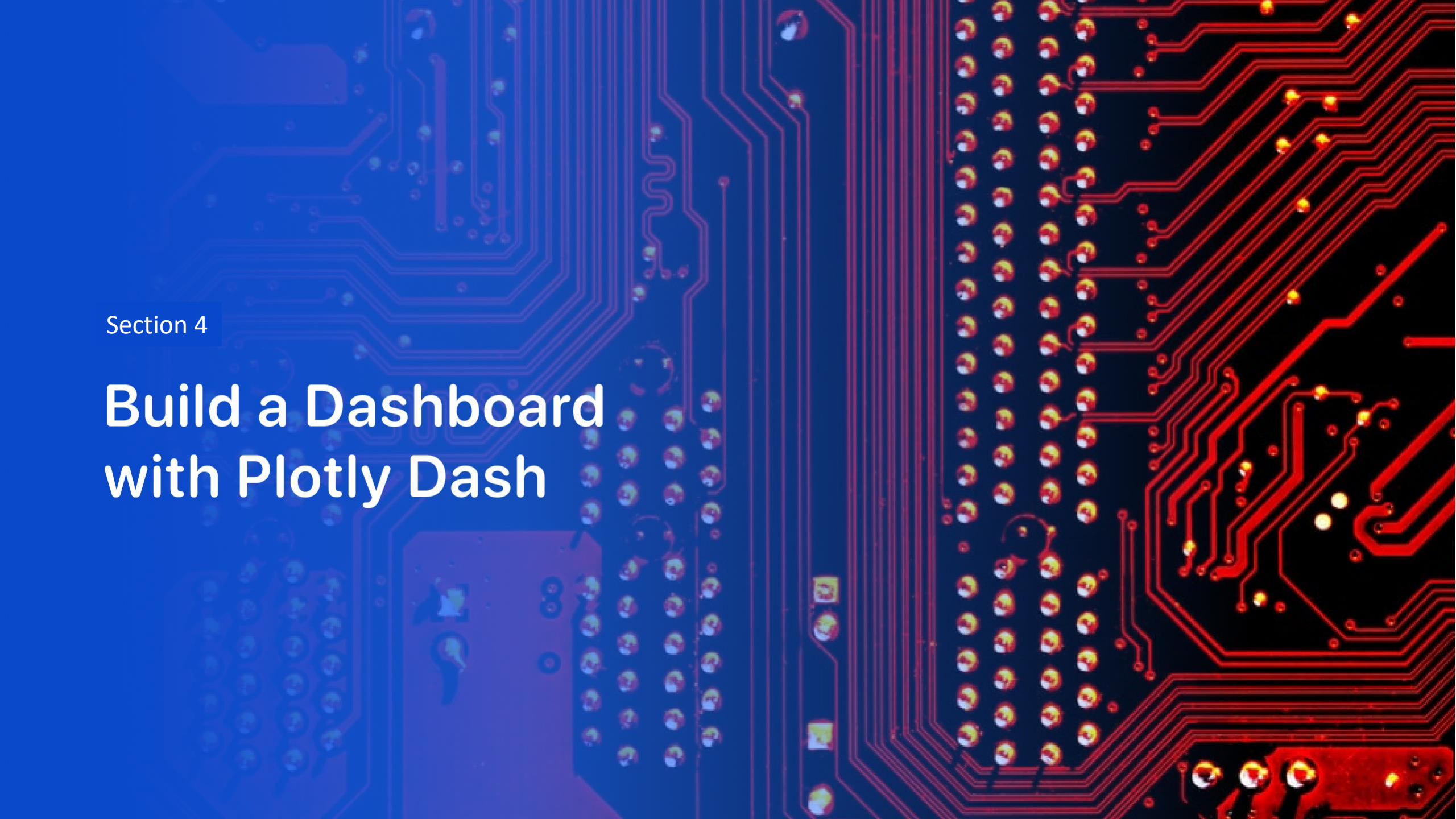
- Green marker shown on the map indicate successful launches and red indicate failure.





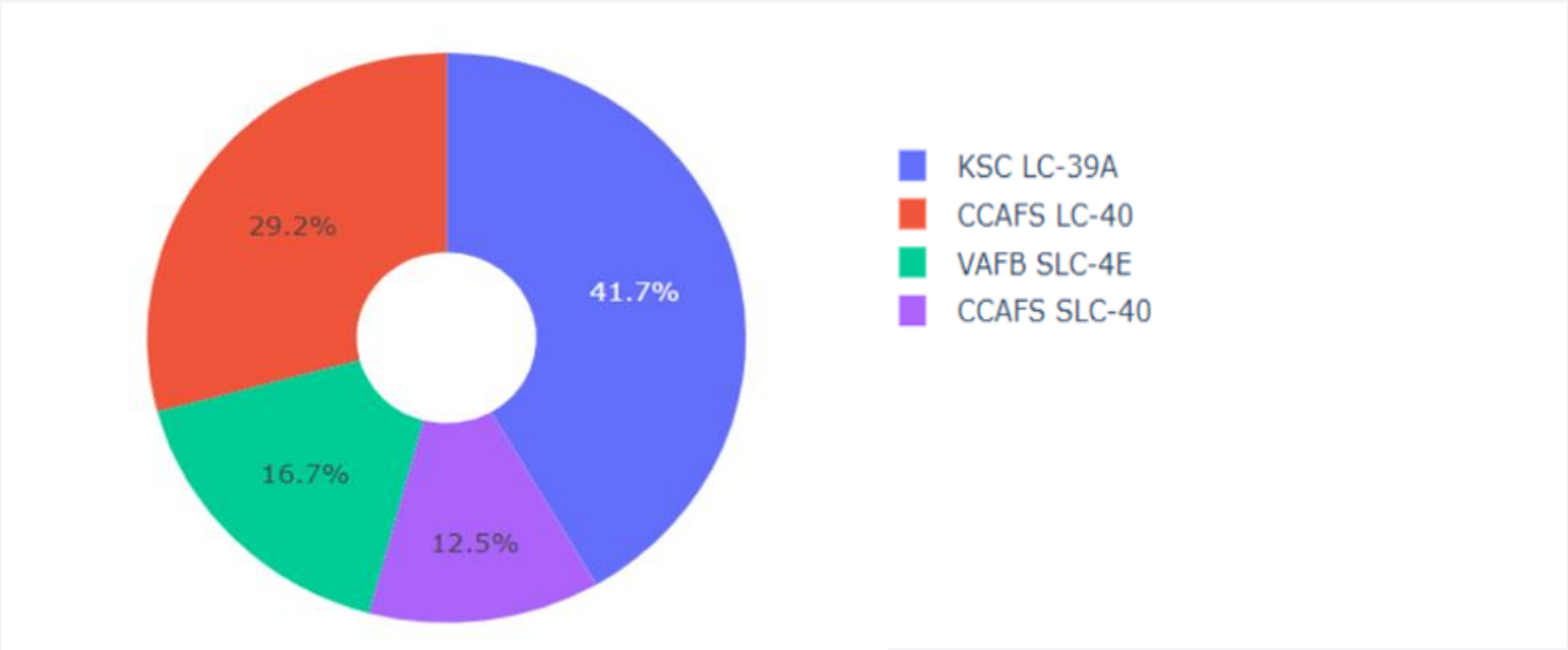
Launch Site distance to landmarks

- Folium was used to check the distance of the landmarks to the location of the launch sites as shown in the images below.


Section 4
Build a Dashboard with Plotly Dash

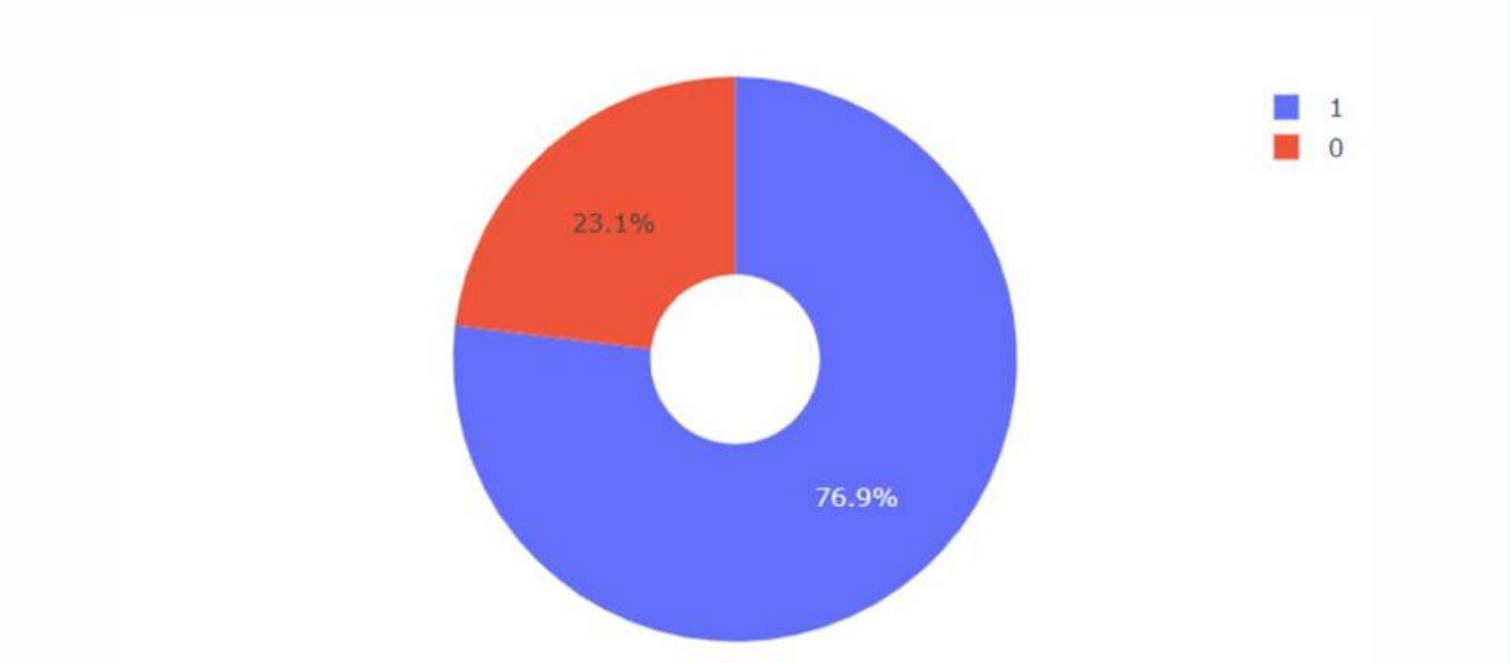
Percentage of successful launch each sites

- Pie chart indicating the success of percentage achieved by each launch sites:



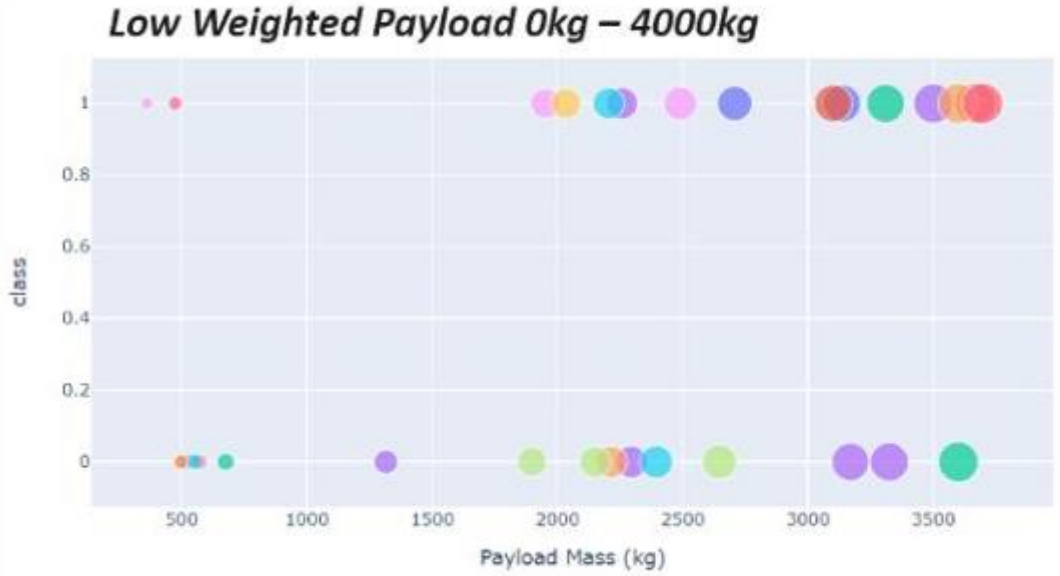
Highest Launch Success Ratio

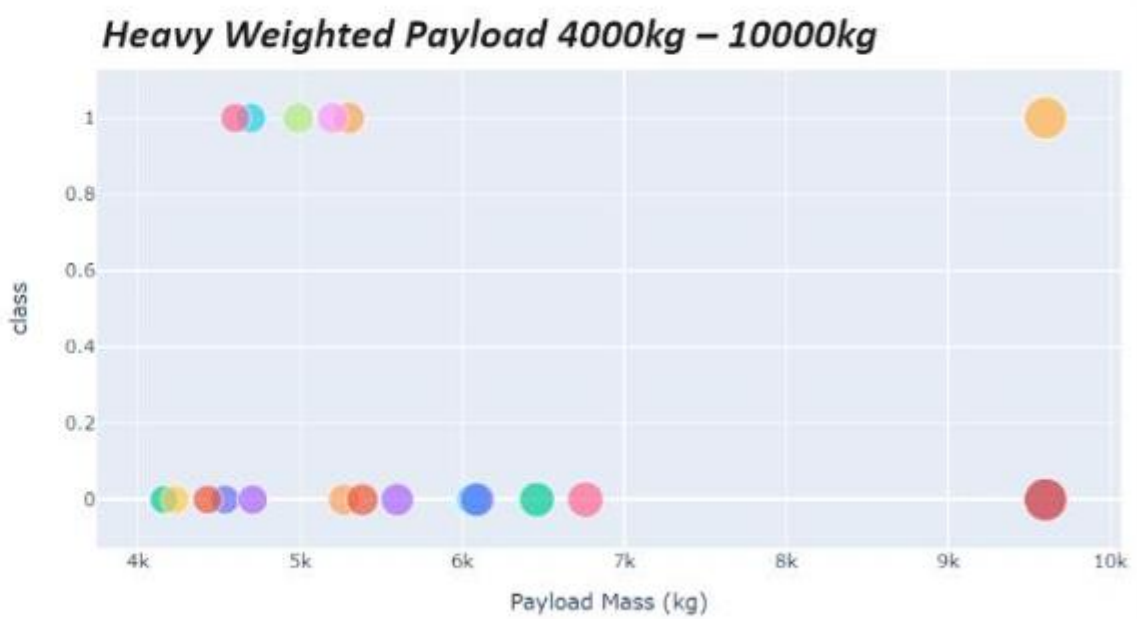
- Pie chart showing the launch site with the highest success ratio. KSC LC-39A achieved the highest success rate of 76.9% and failure rate of 23.1%.



Scatter plot for Payload with Launch Outcomes

- We can take note from the scatter plot below that the success rates of rocket launch is heavily dependent on the weight of the payload. Lower the payload higher is the success rate.





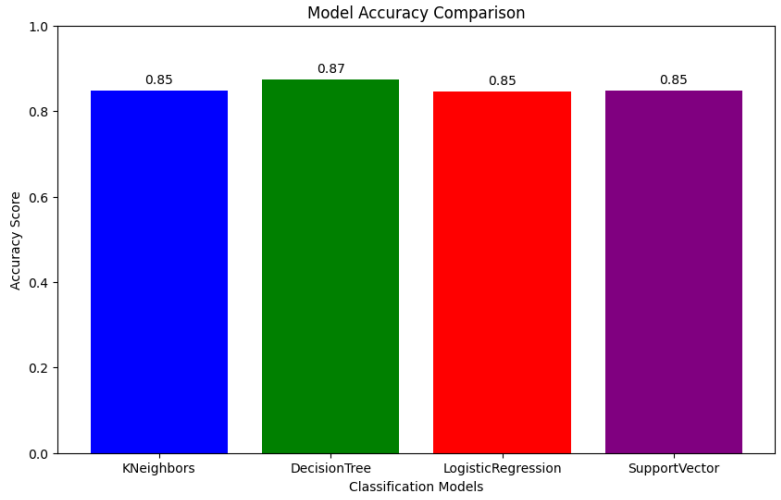
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Decision tree has the highest classification accuracy of 0.87.

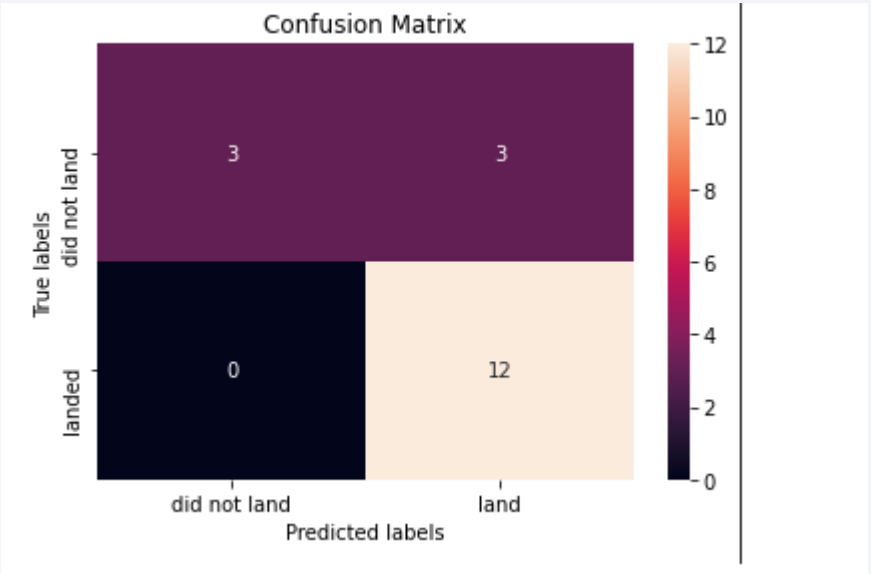
[38]:
```
models = {
    'KNeighbors': knn_cv.best_score_,
    'DecisionTree': tree_cv.best_score_,
    'LogisticRegression': logreg_cv.best_score_,
    'SupportVector': svm_cv.best_score_
}

# Plotting the bar chart
plt.figure(figsize=(10, 6))
plt.bar(models.keys(), models.values(), color=['blue', 'green', 'red', 'purple'])
plt.xlabel('Classification Models')
plt.ylabel('Accuracy Score')
plt.title('Model Accuracy Comparison')
plt.ylim(0, 1)  # assuming accuracy scores are between 0 and 1
for i, (model, score) in enumerate(models.items()):
    plt.text(i, score + 0.01, f'{score:.2f}', ha='center', va='bottom')

plt.show()

# Identifying the best model and printing its details
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': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

- The confusion matrix for the decision tree classifier indicates that the model can differentiate between the various classes. However, a significant issue is the number of false positives, meaning that the classifier incorrectly labels unsuccessful landings as successful ones.



Conclusions

1. **Flight Frequency and Success Rate**: There is a positive correlation between the number of flights at a launch site and the success rate of launches from that site. In other words, launch sites with higher flight frequencies tend to have higher success rates. This may be due to increased experience, refined processes, and improved infrastructure that come with handling more launches.
2. **Temporal Trend in Launch Success**: The success rate of launches has shown a significant upward trend starting from 2013 through to 2020. This improvement could be attributed to technological advancements, better project management, more rigorous testing procedures, and cumulative experience over the years.
3. **Success Rates by Orbit Type**: Specific orbit types such as ES-L1, GEO, HEO, SSO, and VLEO have exhibited higher success rates compared to other orbits. These orbits might benefit from more specialized technology and processes, contributing to their higher success rates. Each orbit type has its unique challenges, but the data indicates these particular orbits are managed more effectively.
4. **KSC LC-39A Performance**: The Kennedy Space Center Launch Complex 39A (KSC LC-39A) has achieved the highest number of successful launches compared to other launch sites. This site likely benefits from superior infrastructure, extensive experience, and possibly a higher volume of launches, all contributing to its success.
5. **Optimal Machine Learning Model**: Among the various machine learning algorithms evaluated, the Decision Tree classifier has proven to be the most effective for this task. This model demonstrated the highest accuracy in predicting the outcomes of the launches, making it a valuable tool for analyzing and forecasting launch success. The Decision Tree classifier's ability to handle complex, non-linear relationships in the data likely contributes to its superior performance

Thank you!