Predicting Falcon 9 First Stage Reusability and Launch Cost Analysis

Thippeswamy MN

27-06-2024

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

Project Overview

- In the rapidly evolving commercial space industry, companies like SpaceX have revolutionized space travel by significantly reducing the cost of rocket launches. This has been largely achieved through the reuse of the rocket's first stage, a feat that sets SpaceX apart from other providers. As a new competitor in this space, Space Y, founded by billionaire industrialist Ellon Musk, seeks to enter the market with competitive pricing strategies and advanced predictive capabilities.

Project Goals

This data science project aims to:

1. Predict the Reusability of Falcon 9 First Stage:
    - Utilize machine learning models to forecast whether the first stage of the Falcon 9 rocket will land successfully and be reusable.
    - Leverage public data on SpaceX launches to train and validate these models.
2. Analyze and Estimate Launch Costs:
    - Develop a cost prediction model to determine the price of each rocket launch.
    - Provide a comparative analysis of launch costs between SpaceX and Space Y.

EXECUTIVE SUMMARY

Methodology

1 Data Collection:

- Gather and preprocess public data on SpaceX launches, including parameters such as payload, orbit, mission type, and historical landing outcomes.

2 Feature Engineering:

- Identify key features that influence the reusability of the first stage and the overall launch cost.

3 Model Development:

- Train machine learning models, such as logistic regression, decision trees, and ensemble methods, to predict the likelihood of first stage reusability.
- Develop regression models to estimate the launch cost based on identified features.

4 Dashboard Creation:

- Create interactive dashboards for visualizing predictions and insights.
- Provide stakeholders with tools to make informed decisions regarding launch scheduling and cost management.

EXECUTIVE SUMMARY

Key Findings

1 Prediction Accuracy:

- The developed machine learning models achieved high accuracy in predicting the reusability of the Falcon 9 first stage, with significant factors including mission parameters and payload specifications.

2 Cost Analysis:

- The cost prediction model demonstrated reliable estimations of launch expenses, highlighting the cost advantages of reusing the first stage.

3 Competitive Insights:

- Analysis revealed that SpaceX's ability to reuse the first stage is a critical factor in maintaining lower launch costs compared to competitors..

EXECUTIVE SUMMARY

Business Impact

1 **Optimize Launch Costs:** Implement cost-effective strategies by accurately predicting reusable launches..

2 **Improve Decision-Making:** Use predictive models to plan and execute more efficient and reliable missions.

3 **Gain Competitive Edge:** Position itself competitively in the commercial space market by offering cost-effective and reliable launch services.

Conclusion

This project underscores the potential of data science in transforming the commercial space industry. By predicting the reusability of rocket stages and estimating launch costs, Space X can make strategic decisions that enhance operational efficiency and market competitiveness. As Space X continues to grow, these predictive insights will be instrumental in achieving sustainable success in the commercial space race.

INTRODUCTION

- Data collection and data wrangling methodology
- EDA and interactive visual analytics methodology
- predictive analysis methodology
- EDA with visualization
- EDA with SQL results
- Interactive map with Folium results
- Plotly Dash dashboard
- predictive analysis (classification)

Data collection and data wrangling methodology

- Collecting Data with an API
  - **API Used:** SpaceX REST API
  - **Data Collected:** Information about launches, rockets, payloads, launch and landing specifications, and outcomes
  - Endpoint: api.spacexdata.com/v4/launches/past
  - Request Method: GET request using the requests library
  - **Response Format:** JSON (list of JSON objects)
  - JSON Conversion: Using json_normalize to flatten JSON data into a DataFrame
- Additional Data Collection
  - **Web Scraping:** Using BeautifulSoup to scrape HTML tables from Wiki pages for Falcon 9 launch records
  - **Data Transformation:** Parsing HTML data and converting it into a Pandas DataFrame
- Data Wrangling
  - **Filtering Data:** Remove Falcon 1 launches to focus on Falcon 9
  - **Handling IDs:** Use the API to gather specific data for IDs (Booster, Launchpad, Payload, and Core)
  - Null Values:
    - Replace null values in PayloadMass with the column's mean
    - Leave LandingPad null values as is for future one hot encoding

RESULTS

Conclusion

The process of collecting and wrangling SpaceX launch data involves multiple steps to ensure the data is clean and ready for analysis. By using the SpaceX REST API and web scraping methods, we can gather comprehensive launch data. Key steps include filtering out irrelevant data, handling IDs, and addressing null values. Proper data preparation will enable accurate predictive modeling and analysis to determine whether SpaceX will attempt to land a rocket. This structured approach ensures a reliable dataset, facilitating meaningful insights and decision-making.

EDA and interactive visual analytics methodology

EDA with Visualization Lab

1. Objective:
   - Perform exploratory data analysis using visualizations to understand and interpret the SpaceX launch data.
2. Visualizing Success Rates:
   - Analyze success rates over time.
   - Visualize the improvement in success rates since 2013.
3. Launch Site Analysis:
   - Compare success rates across different launch sites.
   - Use bar charts or pie charts to show success rates for CCAFS LC-40, KSC LC-39A, and VAFB SLC 4E.
4. Payload Mass Analysis:
   - Examine the relationship between payload mass and landing success.
   - Use scatter plots or histograms to visualize payload mass distribution and its impact on success rates.

5. Feature Combination:
   - Combine multiple attributes (e.g., launch site and payload mass) to gain more insights.
   - Use color-coded scatter plots to show combined effects on landing success.
6. Categorical Variables:
   - Convert categorical variables using one hot encoding.
   - Visualize the impact of each category on the landing success.
7. Correlation Analysis:
   - Identify which attributes are correlated with successful landings.
   - Use heatmaps to visualize correlations between different attributes.
8. Machine Learning Preparation:
   - Prepare the dataset for machine learning by ensuring all features are appropriately encoded and scaled.
   - Visualize the prepared dataset to confirm readiness for model training.

RESULTS

Conclusion

In the EDA with Visualization Lab, various visualizations were employed to deeply understand the SpaceX launch data. By analyzing success rates over time and across different launch sites, we identified key trends and patterns. Payload mass was found to be a significant factor, particularly when combined with launch site data. Converting categorical variables using one hot encoding and performing correlation analysis provided further insights into which attributes are most predictive of landing success. This comprehensive visual analysis prepares the dataset for subsequent machine learning tasks, ultimately contributing to the development of a predictive model for SpaceX Falcon 9 landings.

predictive analysis methodology

Predictive Analysis Introduction:

1. Objective::
   - Build a machine learning pipeline to predict the successful landing of the Falcon 9's first stage
2. Steps Involved:
   - Preprocessing:
   - Standardize the data to ensure consistency and improve model performance.
   - Train-Test Split:
     - Split the dataset into training and testing subsets to evaluate the model's performance accurately.
3. Model Training:
   - Train multiple machine learning models:
     - Logistic Regression
     - Support Vector Machines (SVM)
     - Decision Tree Classifier
     - K-Nearest Neighbours(KNN)

4 Hyperparameter Tuning:

- Perform Grid Search to find the best hyperparameters for each algorithm
- Select hyperparameters that optimize model performance.

5 Model Evaluation:

- Use the best hyperparameter values to evaluate models based on training data.
- Determine the model with highest accuracy

6 Confusion Matrix:

- Output the confusion matrix to assess the performance of the best model.

RESULTS

Step 1: Data Preprocessing

Objective: Prepare the data for training by standardizing features and handling missing values.

1. **Data Cleaning**:
    - Handle missing values by imputing or removing them.
    - Remove duplicate entries.
2. **Feature Engineering**:
    - Create new features that may enhance model performance.
    - Transform categorical variables into numerical representations using techniques like one-hot encoding.

3. Standardization

- Standardize the features to have a mean of 0 and a standard deviation of 1, which is essential for algorithms like SVM and K-nearest neighbors.

from sklearn.preprocessing import StandardScaler scaler = StandardScaler() X_scaled = scaler.fit_transform(X)

- Step 2: Train-Test Split
- **Objective:** Split the data into training and testing sets to evaluate model performance on unseen data.

from sklearn.model_selection import train_test_split X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

- Step 3: Model Training and Evaluation
- **Objective:** Train multiple models and evaluate their performance to identify the best model.

1. **Logistic Regression**:

- Train the model using the training set.
- Evaluate using metrics such as accuracy, precision, recall, and F1-score.

from sklearn.linear_model import LogisticRegression lr = LogisticRegression() lr.fit(X_train, y_train) y_pred_lr = lr.predict(X_test)

4. Support Vector Machines (SVM)
    - Train the SVM model.
    - Evaluate using the same metrics as above.

    from sklearn.svm import SVC svc = SVC() svc.fit(X_train, y_train) y_pred_svc = svc.predict(X_test)

5. Decision Tree Classifier
    - Train the Decision Tree model.
    - Evaluate using the same metrics as above.

    from sklearn.tree import DecisionTreeClassifier dt = DecisionTreeClassifier() dt.fit(X_train, y_train) y_pred_dt = dt.predict(X_test)

6. K-Nearest Neighbors (KNN)
   - Train the SVM model.
   - Evaluate using the same metrics as above.

   from sklearn.neighbors import KNeighborsClassifier knn = KNeighborsClassifier() knn.fit(X_train, y_train) y_pred_knn = knn.predict(X_test)

7. Hyperparameter Tuning using Grid Search
   - **Objective:** Use Grid Search to find the best hyperparameters for each model to optimize performance.

   from sklearn.model_selection import GridSearchCV # Example for Logistic Regression param_grid_lr = {'C': [0.1, 1, 10, 100]} grid_search_lr = GridSearchCV(LogisticRegression(), param_grid_lr, cv=5) grid_search_lr.fit(X_train, y_train) best_lr = grid_search_lr.best_estimator_ # Repeat for other models (SVM, Decision Tree, KNN)

6. Model Selection and Evaluation

- **Objective:** Select the best model based on performance metrics and evaluate it on the test set.

1. **Evaluate Best Model**:

- Use the confusion matrix to visualize true positives, false positives, true negatives, and false negatives.

```
from sklearn.metrics import confusion_matrix, accuracy_score # Assuming best_model is determined y_pred_best = best_model.predict(X_test) conf_matrix = confusion_matrix(y_test, y_pred_best) accuracy = accuracy_score(y_test, y_pred_best) print("Confusion Matrix:\n", conf_matrix) print("Accuracy:", accuracy)
```

2. **Visualization:**

- Visualize the confusion matrix using heat map

```
import seaborn as sns import matplotlib.pyplot as plt sns.heatmap(conf_matrix, annot=True, fmt='d') plt.xlabel('Predicted') plt.ylabel('Actual') plt.title('Confusion Matrix') plt.show()
```

Conclusion:

By following these steps, you will build a robust machine learning pipeline to predict the landing success of Falcon 9’s first stage. This process involves data preprocessing, model training, hyperparameter tuning, and thorough evaluation to ensure the best possible predictive performance.

EDA with visualization

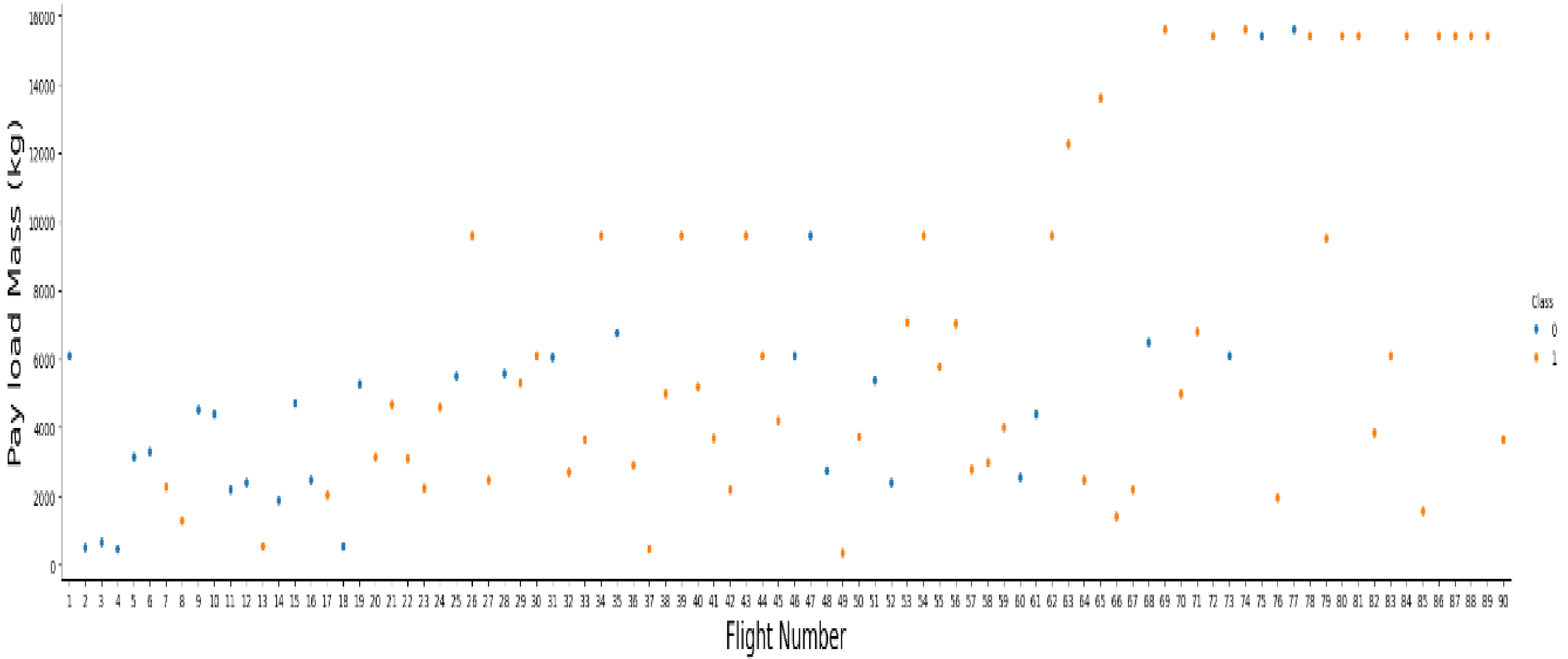
EDA with visualization :

1. Objective::
    - Perform Exploratory Data Analysis (EDA) using visualizations to understand and analyze the SpaceX launch data.
2. Tools and Libraries:
    - Utilize libraries such as Folium and Plotly Dash for interactive visual analytics and dashboard creation.
3. Analyzing Launch Site Geo and Proximities with Folium:
    - Mark the locations of launch sites on an interactive map.
    - Explore the proximity of these launch sites and identify any patterns or insights.
    - Explain the optimal choice of launch site based on the visual analysis.

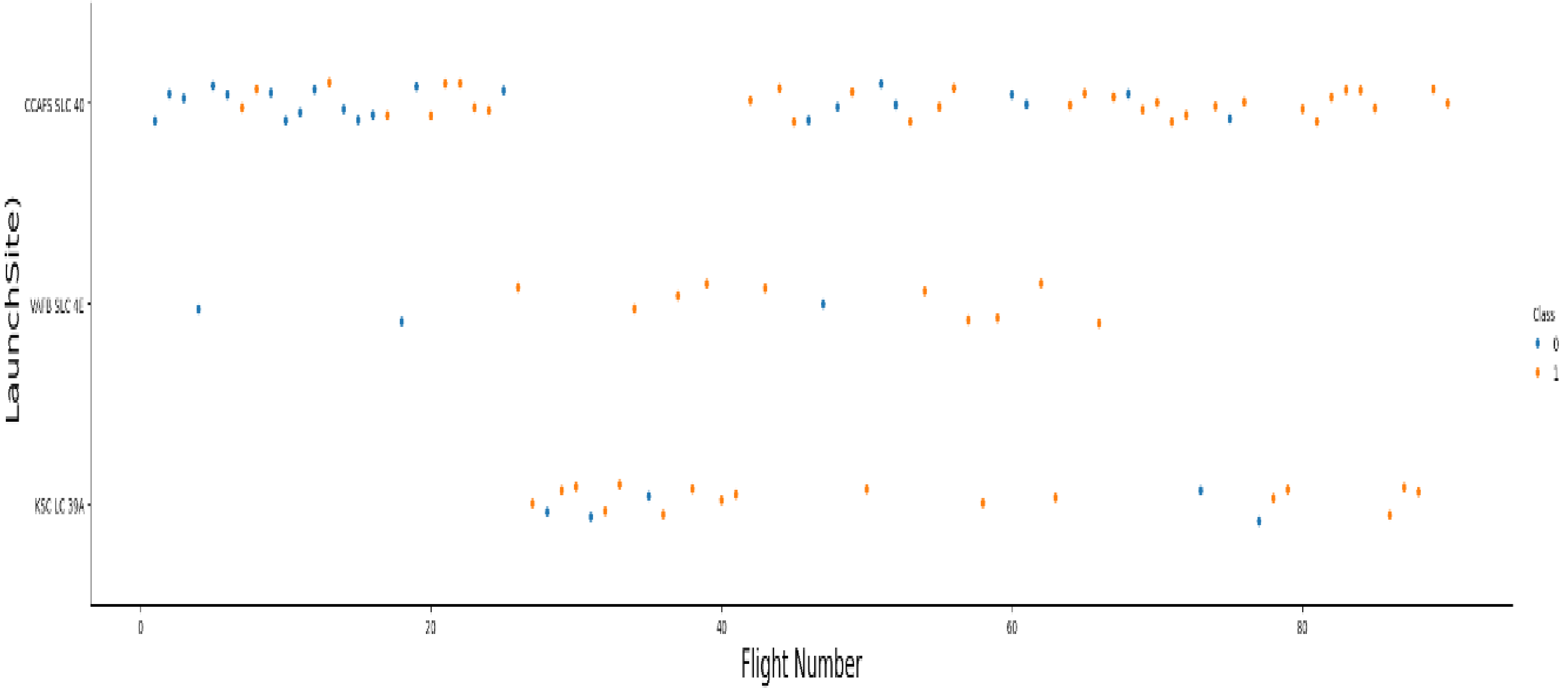
4. Building Interactive Dashboards with Plotly Dash:
    - Create a dashboard application with input components like dropdown lists and range sliders.
    - Incorporate interactive elements such as pie charts and scatter point charts.
    - Enable stakeholders to interact with the data and discover insights more effectively than with static graphs.
5. Features of Interactive Visual Analytics:
    - Interactions like zoom-in, zoom-out, pan, filter, search, and link.
    - Enable users to find visual patterns faster and more effectively.
    - Enhance the storytelling aspect of data presentation through interactive visualizations.
6. Insights and Patterns:
    - Analyze various attributes that correlate with successful landings.
    - Convert categorical variables using one hot encoding to prepare data for machine learning models.
    - Combine multiple features to provide more information and improve analysis.

RESULTS

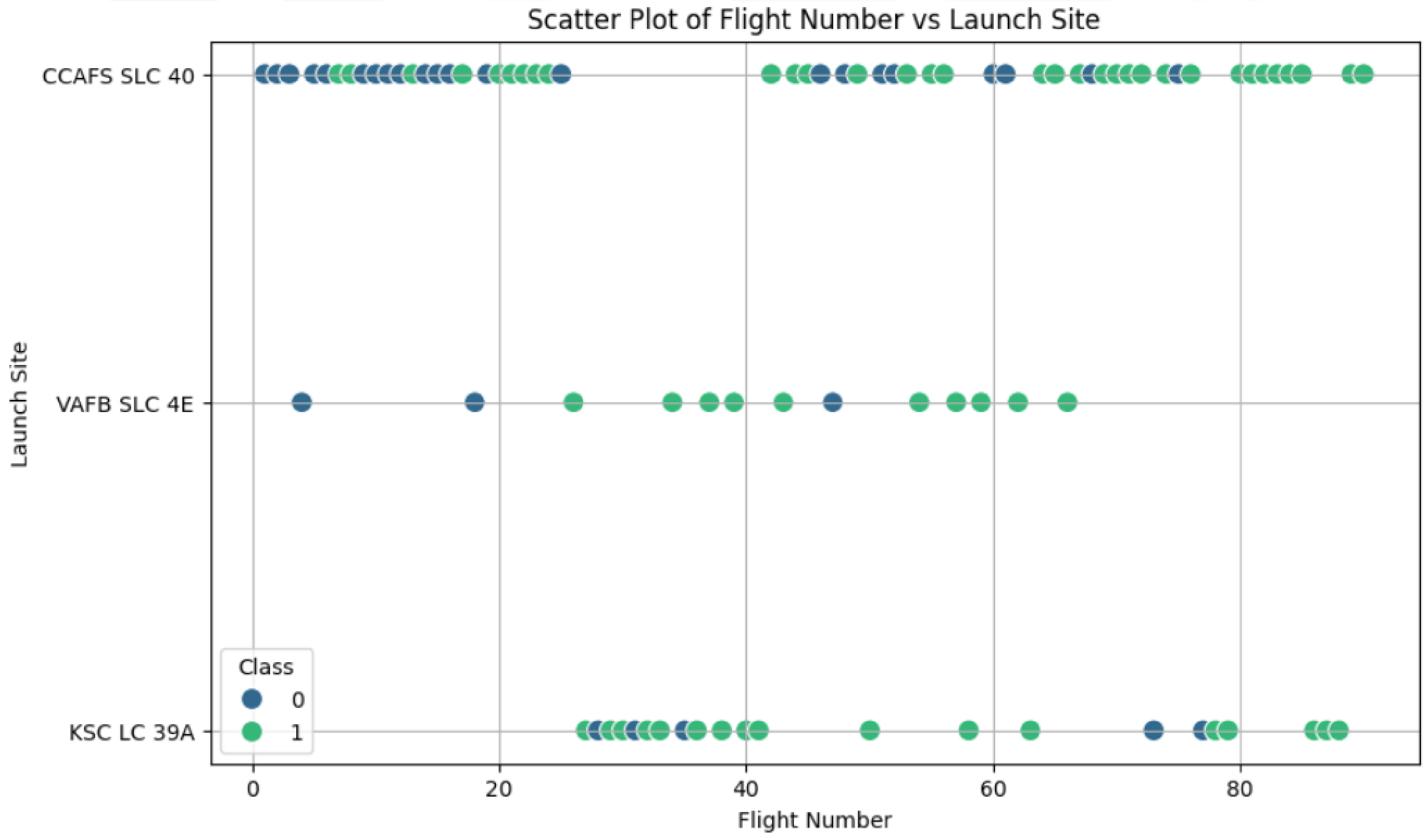
FlightNumber vs. PayloadMass



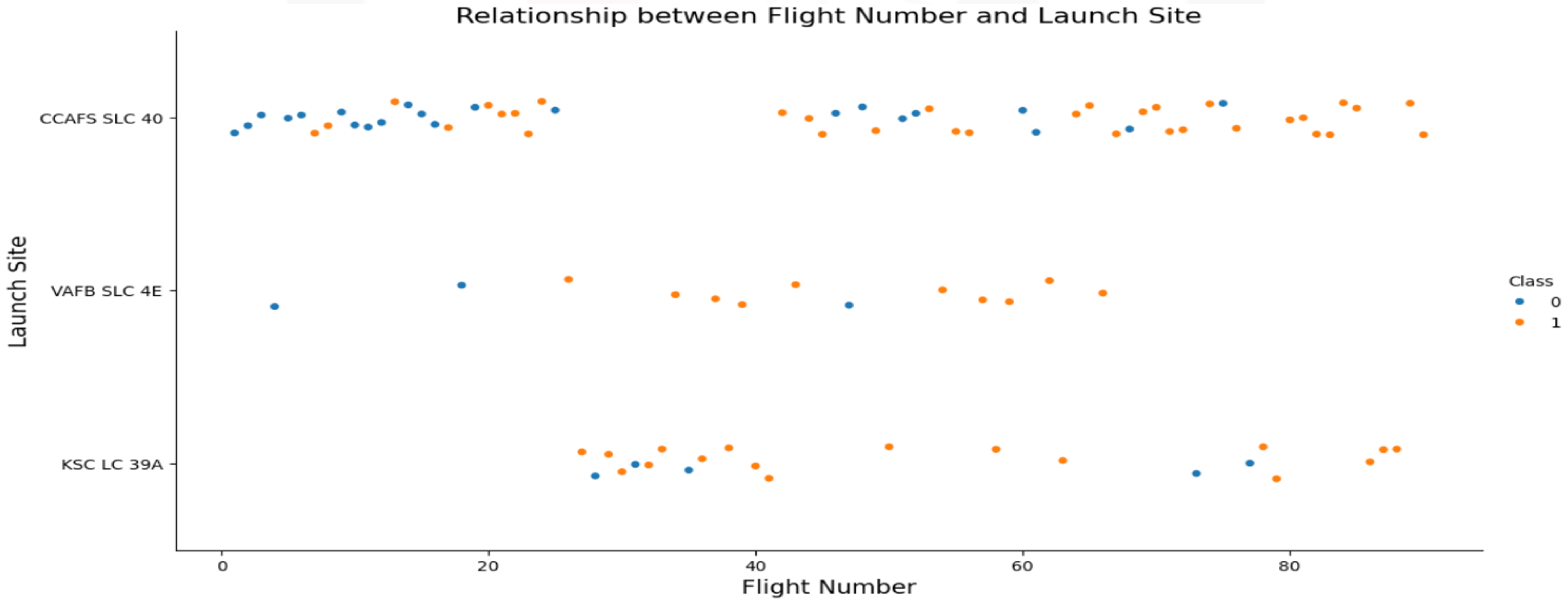
Visualize the relationship between Flight Number and Launch Site



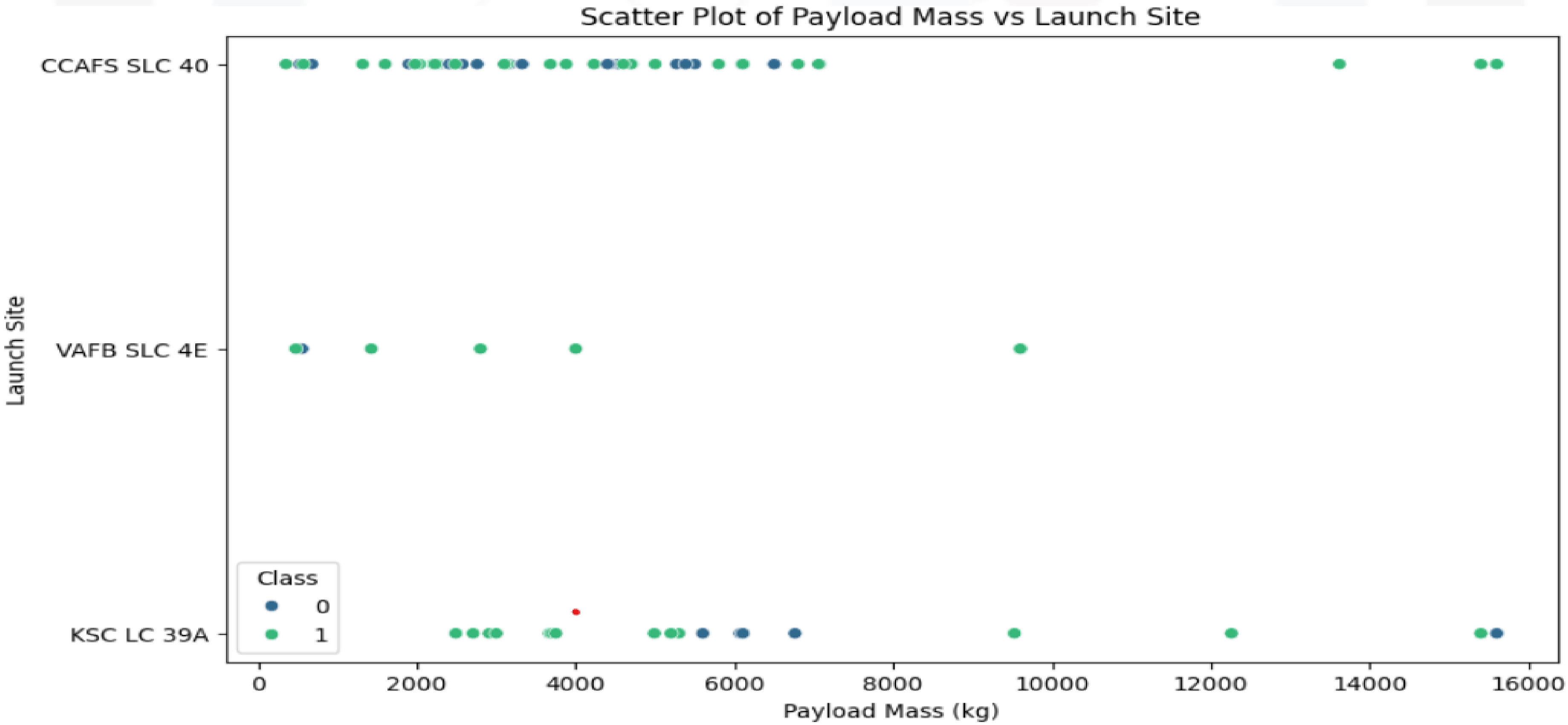
Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value



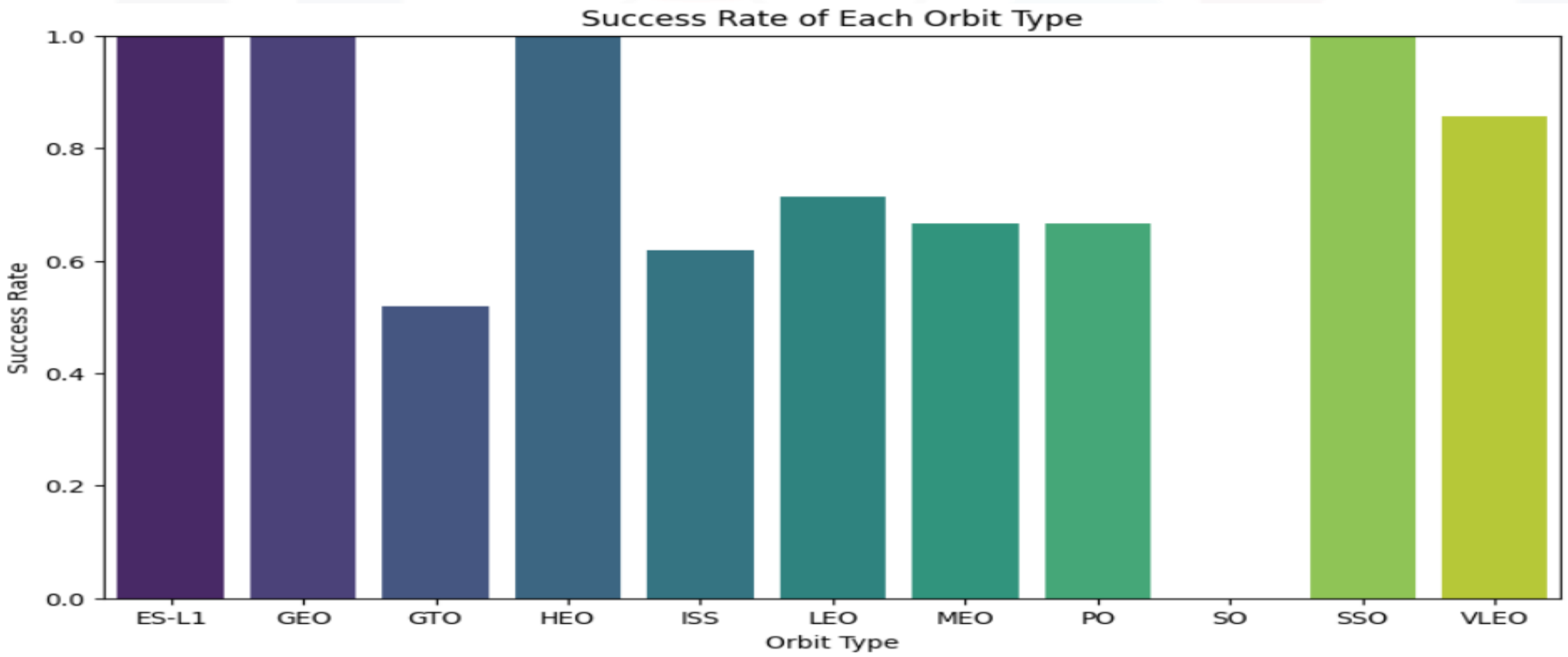
Visualize the relationship between Payload and Launch Site



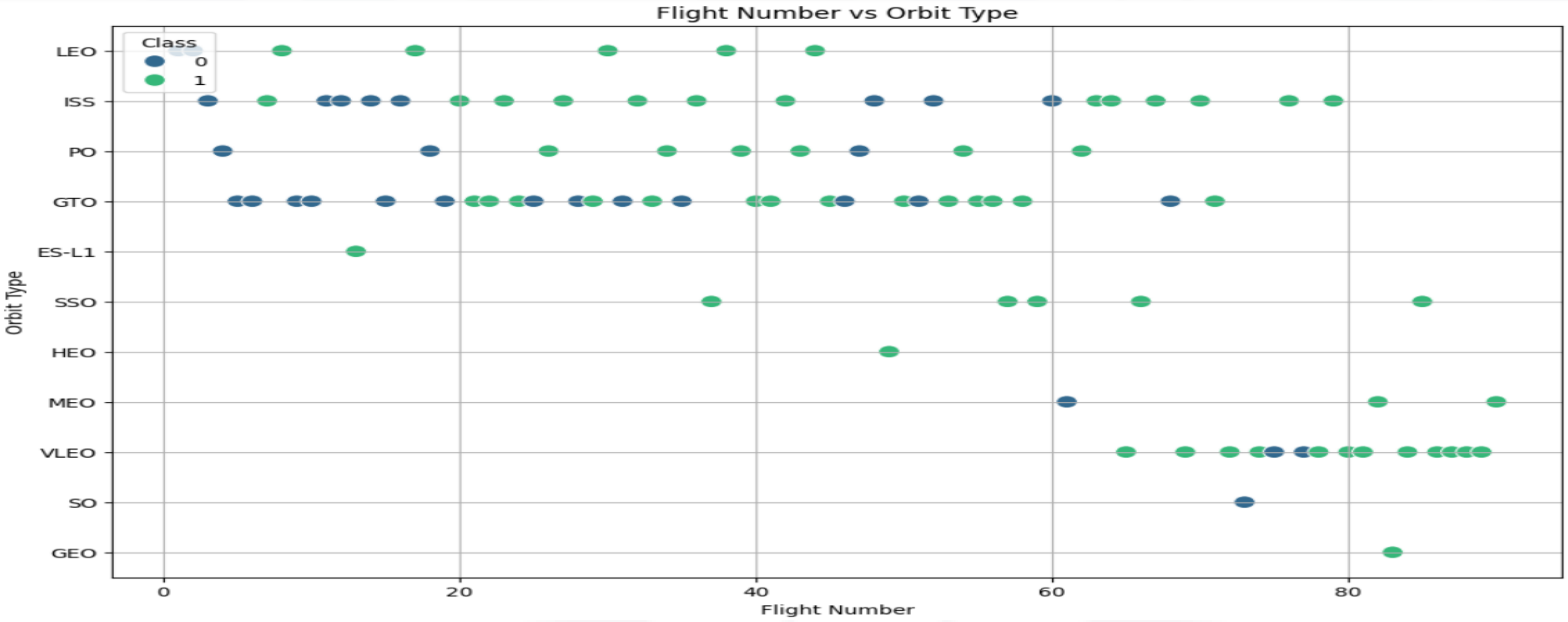
Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value



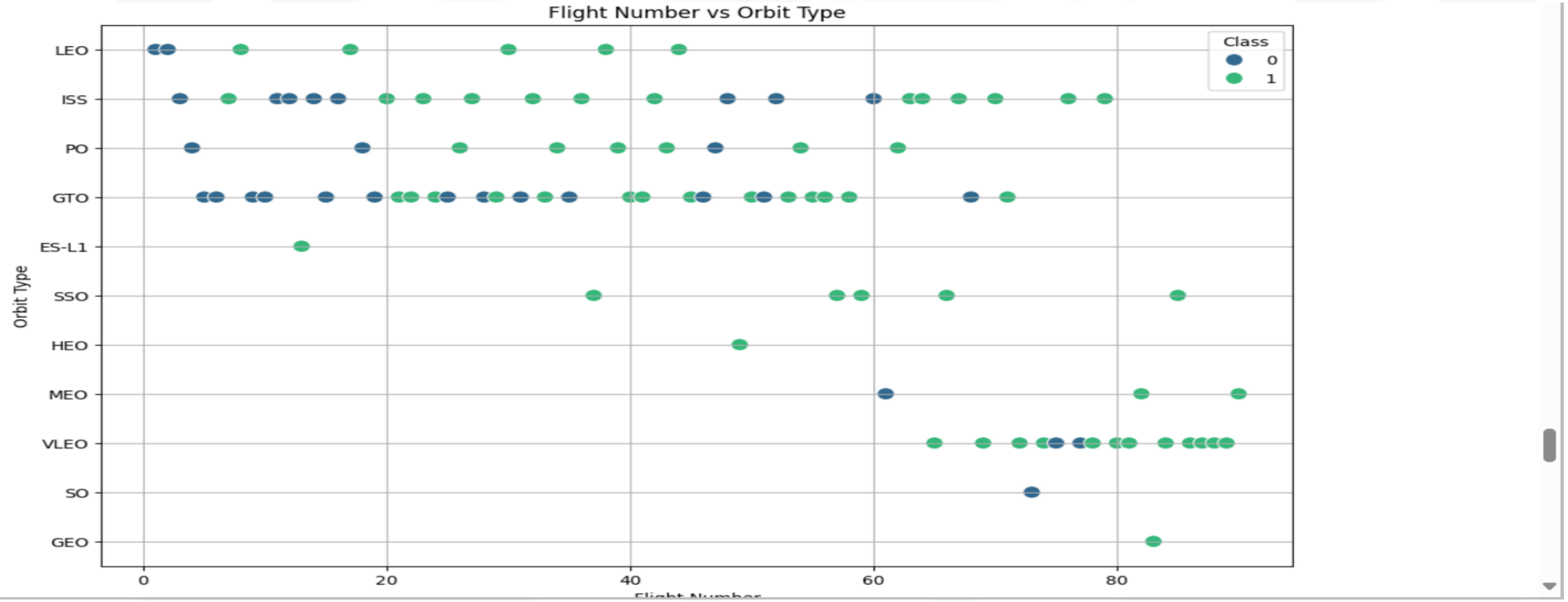
Visualize the relationship between success rate of each orbit type



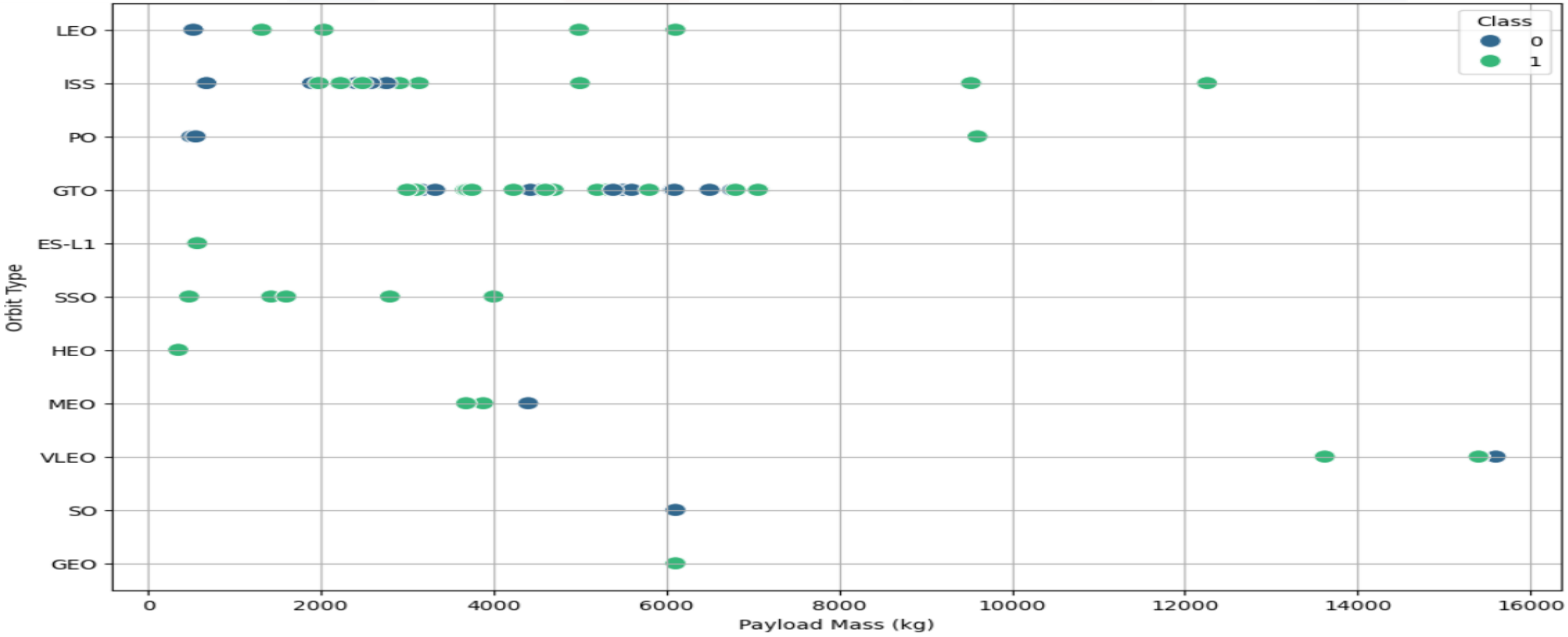
Visualize the relationship between FlightNumber and Orbit type



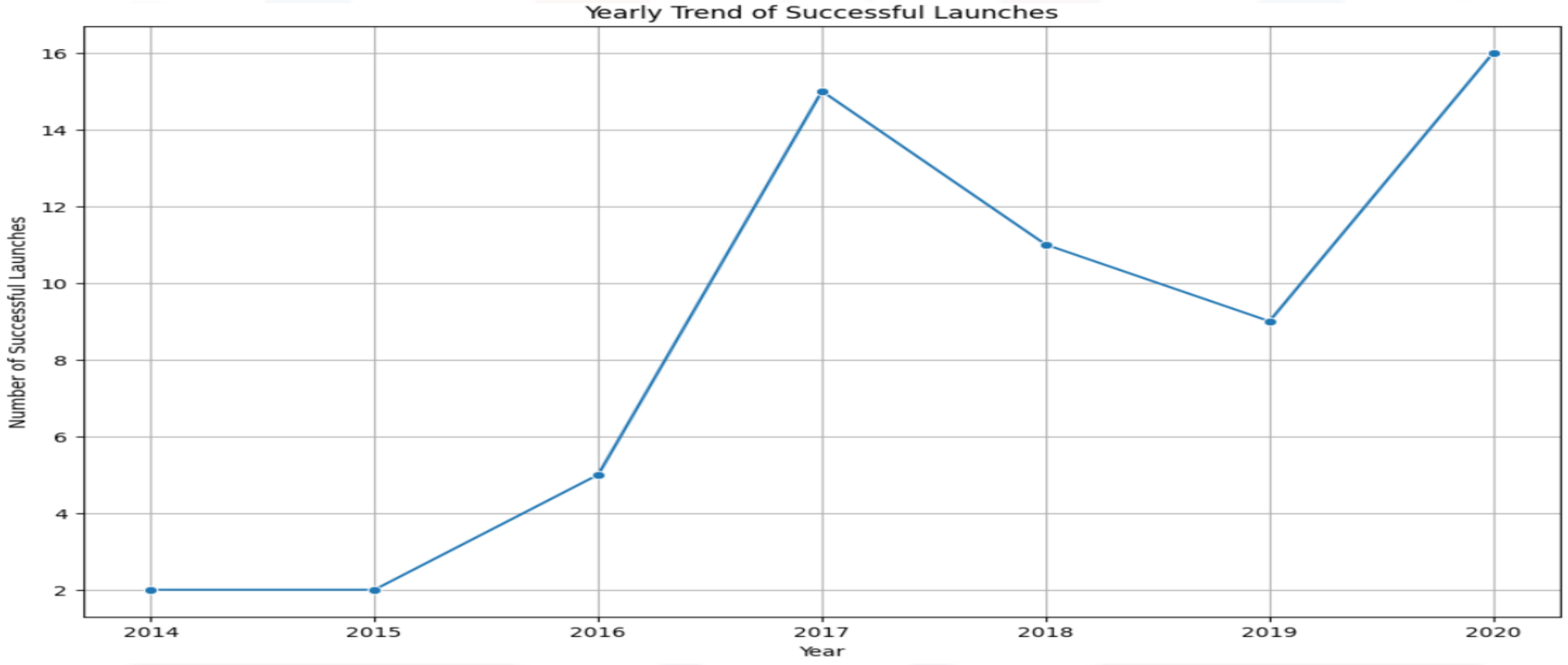
Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value



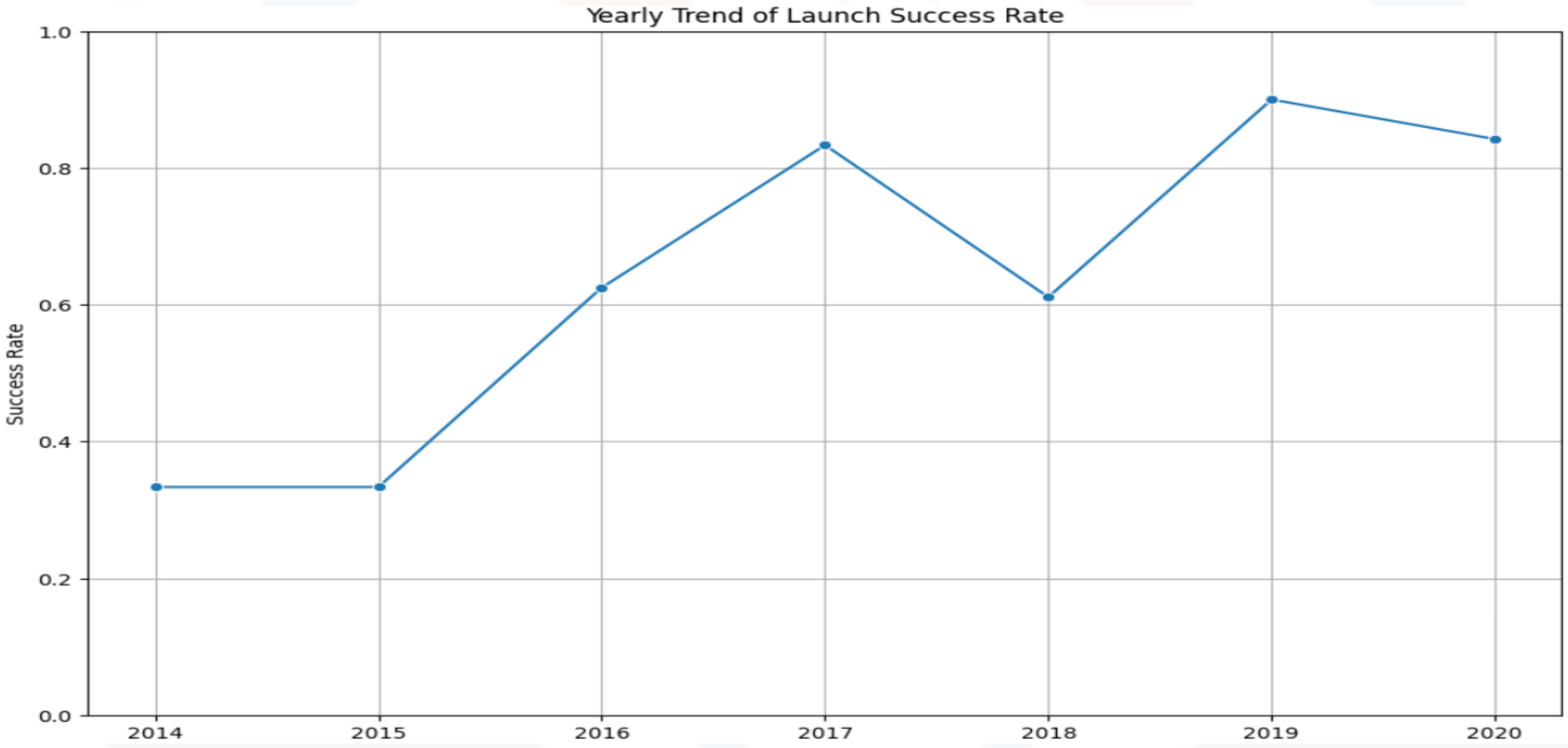
Visualize the relationship between Payload and Orbit type



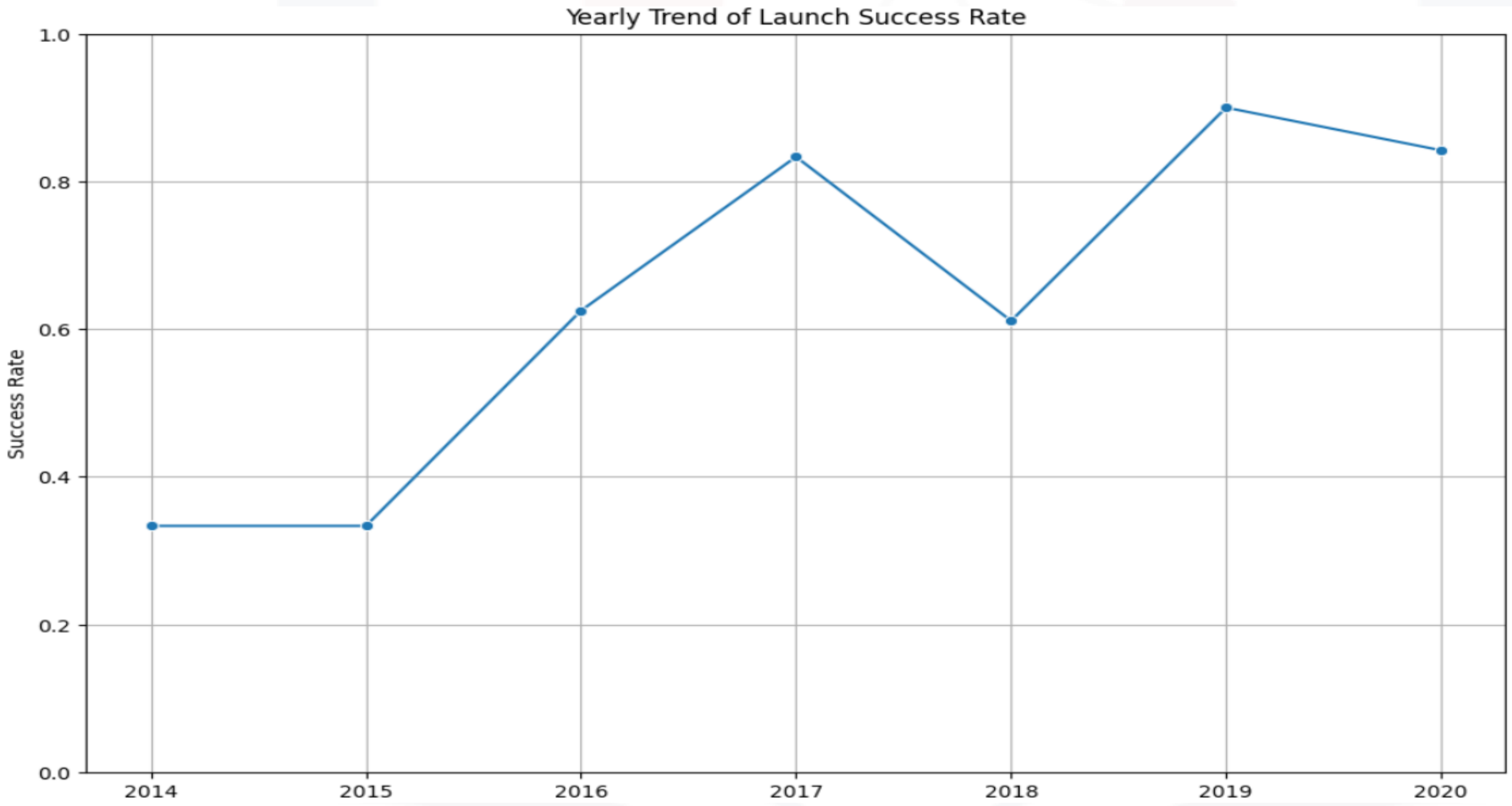
Visualize the launch success yearly trend



Plot a line chart with x axis to be the extracted year and y axis to be the success rate



Plot a line chart with x axis to be the extracted year and y axis to be the success rate



RESULTS

Conclusion

The EDA with Visualization Lab aims to enhance your ability to perform exploratory data analysis using interactive visualizations. By leveraging tools like Folium and Plotly Dash, you can create dynamic and interactive dashboards that allow stakeholders to explore and manipulate data in real-time. This approach not only helps in identifying visual patterns and insights more effectively but also tells a more compelling story through interactive visualizations. The lab prepares you to analyze launch site proximities, determine optimal launch sites, and prepare data for machine learning models, ultimately contributing to the goal of understanding and predicting the success of SpaceX's Falcon 9 first stage landings.

EDA with SQL

EDA with SQL :

1. Objective:
   - Perform Exploratory Data Analysis (EDA) using SQL to query and analyze SpaceX launch data stored in a database.
2. Database and SQL:
   - Use SQL (Structured Query Language) to interact with the database.
   - Retrieve, filter, aggregate, and analyze data using SQL queries.
3. Key SQL Queries and Operations:
   - **SELECT:** Retrieve specific columns from the database.
   - **WHERE:** Filter records based on specified conditions..
   - **GROUP BY:** Group data based on one or more columns.
   - **ORDER BY:** Sort data in ascending or descending order.
   - **JOIN:** Combine rows from two or more tables based on related columns.
   - **Aggregate Functions:** Use functions like COUNT, SUM, AVG, MAX, and MIN to perform calculations on data.

4. Analyzing Launch Data:
    - Query launch data to understand patterns and trends.
    - Identify key metrics such as success rates, launch frequencies, and payload weights.
    - Analyze data based on different launch sites, payload types, and mission outcomes..
5. Data Preparation::
    - Clean and preprocess data to handle missing values and inconsistencies.
    - Create new calculated fields for deeper analysis.
    - Filter out irrelevant data to focus on Falcon 9 launches.
6. nsights and Patterns::
    - Identify correlations between different attributes and successful landings..
    - Determine the impact of launch site, payload mass, and mission type on launch outcomes.
    - Discover trends over time and analyze changes in success rates.

RESULTS

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

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

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

[18]:

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

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

total_payload_mass_
45596

List the date when the first succesful landing outcome in ground pad was acheived

total_payload_mass_
45596

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

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

List the total number of successful and failure mission outcomes

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

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

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

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.

Month_Name	Booster_Version	Launch_Site	Landing_Outcome
January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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.

Month_Name	Booster_Version	Launch_Site	Landing_Outcome
January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Conclusion

The EDA with SQL module equips you with the skills to perform comprehensive data analysis using SQL queries. By interacting with the SpaceX launch data stored in a database, you can efficiently retrieve, filter, and analyze data to uncover valuable insights. Key SQL operations and aggregate functions enable you to calculate metrics, identify patterns, and understand trends in the data. This structured approach to EDA prepares you to clean, preprocess, and analyze data effectively, providing a solid foundation for further predictive analysis and decision-making processes related to SpaceX's Falcon 9 launches.

Interactive Visual Analytics with Folium

EDA with visualization :

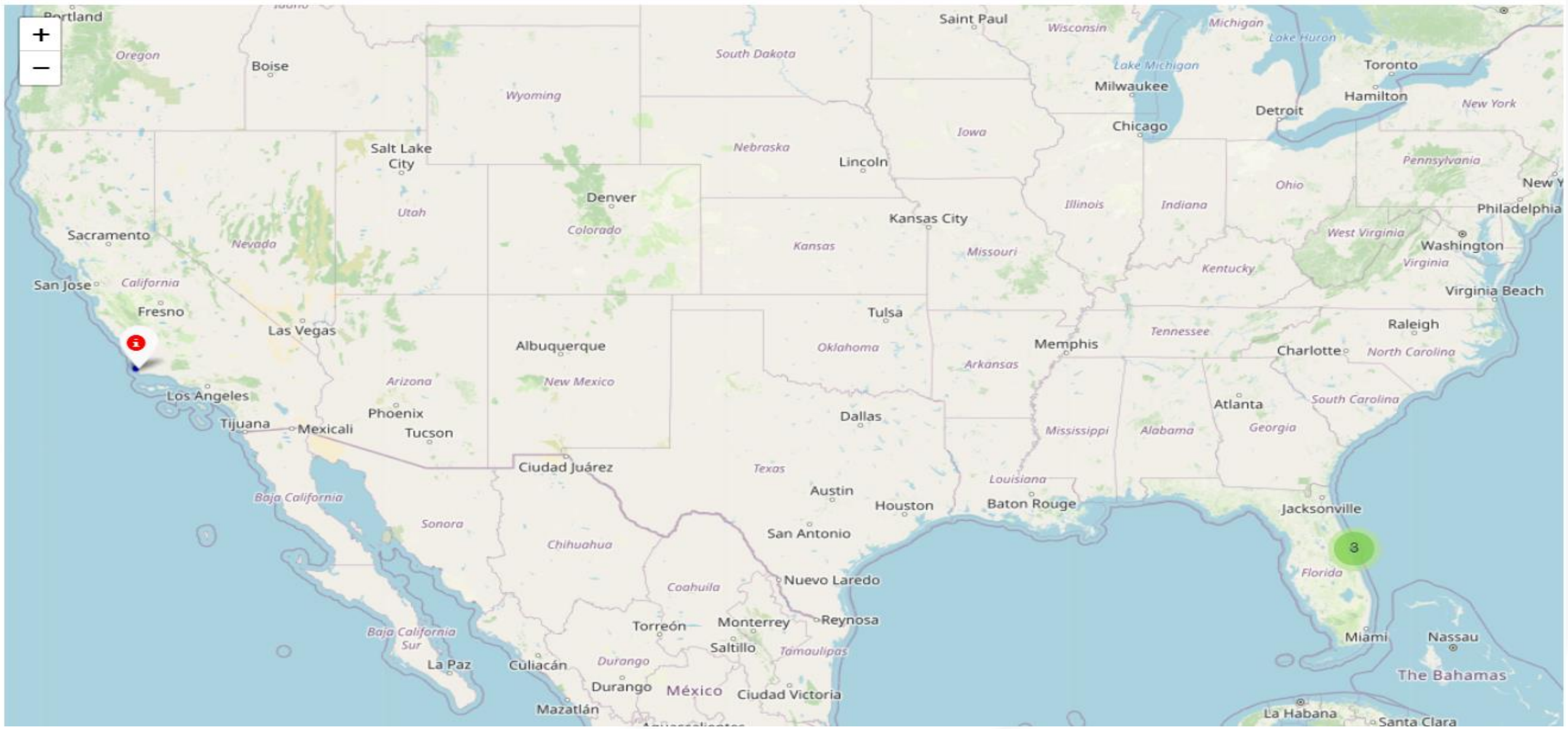
1. Objective::
   - Utilize Folium for interactive visual analytics to analyze SpaceX launch site geographies and proximities..
2. Folium Overview:
   - Folium is a Python library used to create interactive maps.
   - It allows for the easy creation of maps with markers, polylines, polygons, and other visual elements.
3. Interactive Maps:
   - Create an interactive map to visualize launch site locations..
   - Use markers to indicate launch sites and their proximities.
   - Enable zoom-in, zoom-out, and pan functionalities to explore the map dynamically.

4. Analyzing Launch Sites::
    - Plot launch site locations on the map to observe geographical patterns.
    - Identify nearby features and analyze their potential impact on launch operations.
    - Evaluate the distance between launch sites and key landmarks or facilities.
5. Proximity Analysis:
    - Add markers and circles to denote proximity ranges around launch sites.
    - Analyze the spatial distribution of successful and unsuccessful launches.
    - Investigate potential correlations between launch outcomes and geographical factors.
6. Visual Patterns and Insights:
    - Discover visual patterns by exploring the interactive map.
    - Identify optimal launch sites based on proximity analysis and success rates.
    - Use visual insights to inform strategic decisions regarding launch site selection.
7. Customization and Interaction:
    - Customize map appearance, markers, and pop-up information for enhanced visualization.
    - Enable user interaction to filter, search, and link data points for more detailed exploration.

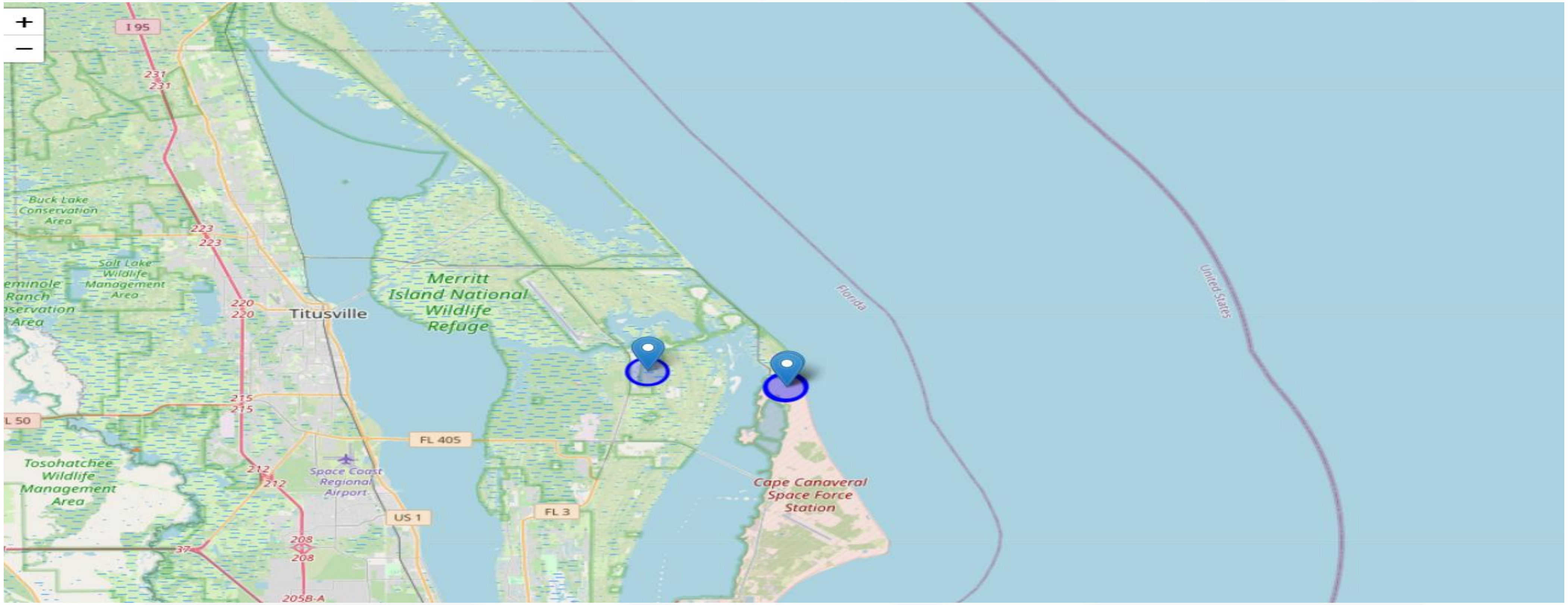
RESULTS

Mark the success/failed launches for each site on the map

[25]:



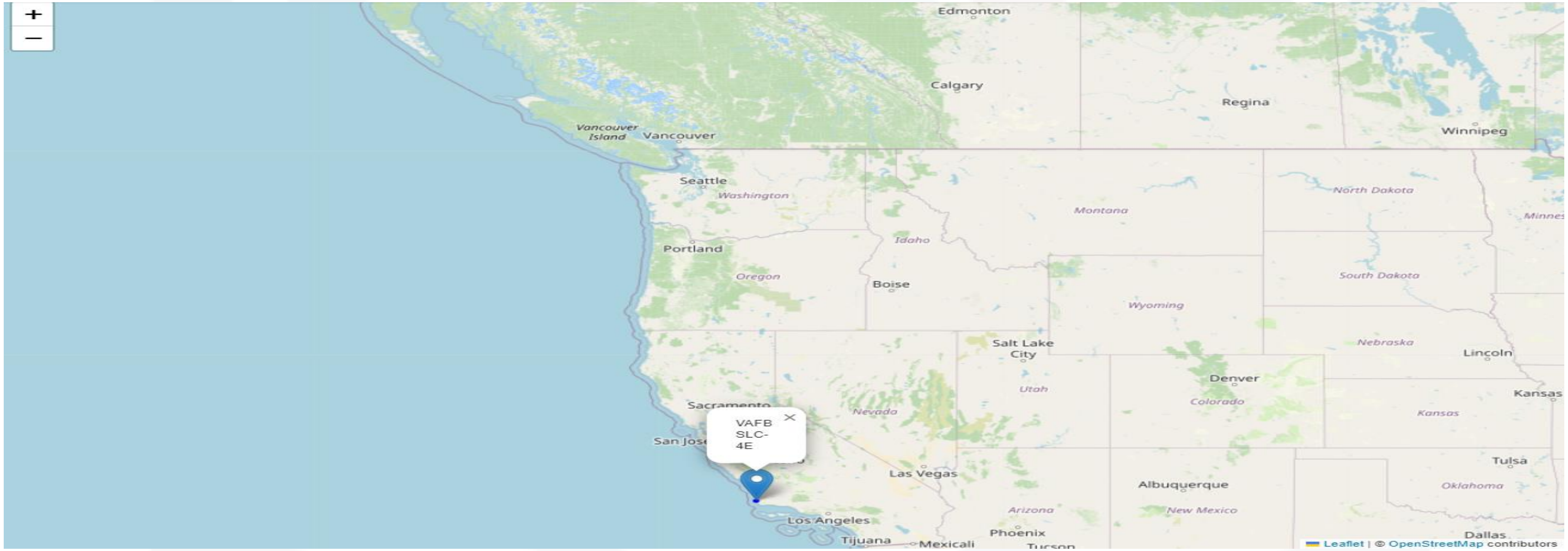
KSC LC-39A,CCAFS LC-40 and CCAFS SLC-40



IBM Developer

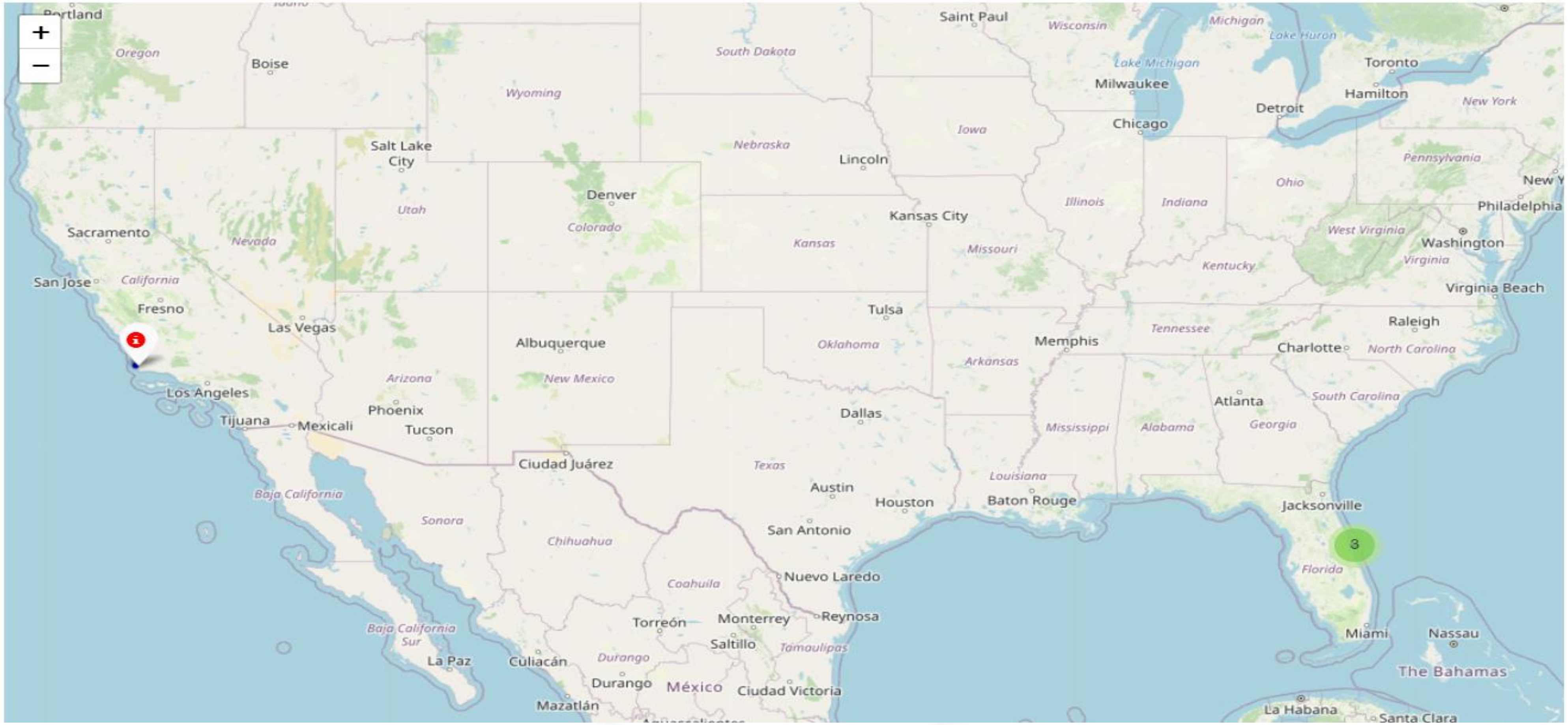
SKILLS NETWORK

VAFB SLC-4E

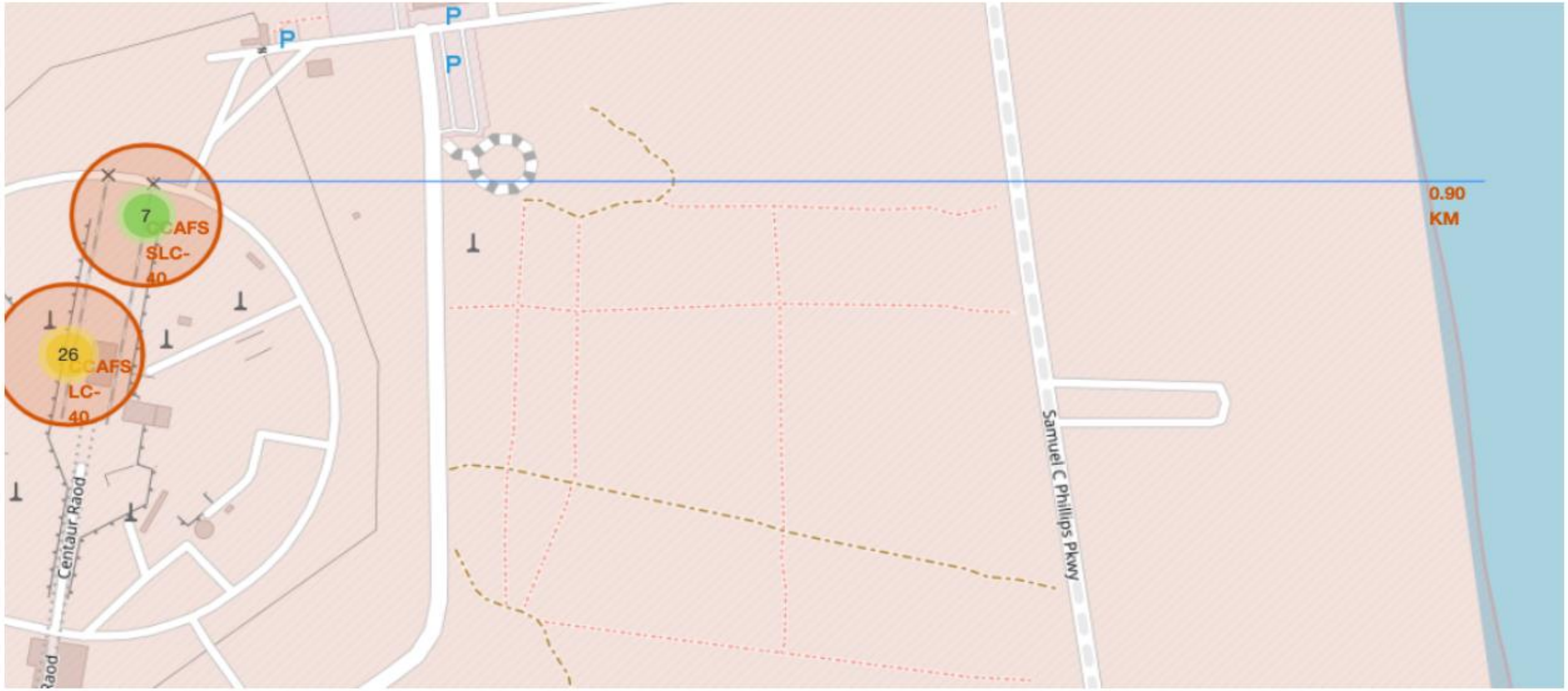

Edmonton
Calgary
Regina
Winnipeg
Vancouver Island
Vancouver
Seattle
Washington
Montana
North Dakota
Portland
Idaho
Oregon
Boise
South Dakota
Wyoming
Nebraska
Salt Lake City
Lincoln
Denver
Utah
Colorado
Kansas
Sacramento
Nevada
VAFB SLC-4E
Las Vegas
Tulsa
Albuquerque
Oklahoma
Arizona
New Mexico
Los Angeles
Phoenix
Tijuana
Mexicali
Dallas
Leaflet | © OpenStreetMap contributors

Mark the success/failed launches for each site on the map

[25]:



Mark the success/failed launches for each site on the map



Conclusion

The Interactive Visual Analytics with Folium module empowers you to leverage interactive mapping tools for comprehensive geographical analysis of SpaceX launch sites. By creating interactive maps with Folium, you can visualize launch site locations, analyze proximities, and identify spatial patterns. This dynamic approach to visual analytics enables you to uncover valuable insights and make data-driven decisions about optimal launch sites and operational strategies. The ability to customize and interact with the map further enhances the exploration and understanding of complex geographical data.

Plotly Dash dashboard

Plotly Dash dashboard

1. Objective::

- Build an interactive dashboard for visualizing and analyzing SpaceX launch data using Plotly Dash.

2 Plotly Dash Overview:

- Plotly Dash is a Python framework for building analytical web applications..
- It allows for the creation of interactive, customizable dashboards with various input and output components.

3 Interactive Components::

- Include input components like dropdown lists and range sliders for user interaction.
- Use these components to filter and manipulate the data dynamically..

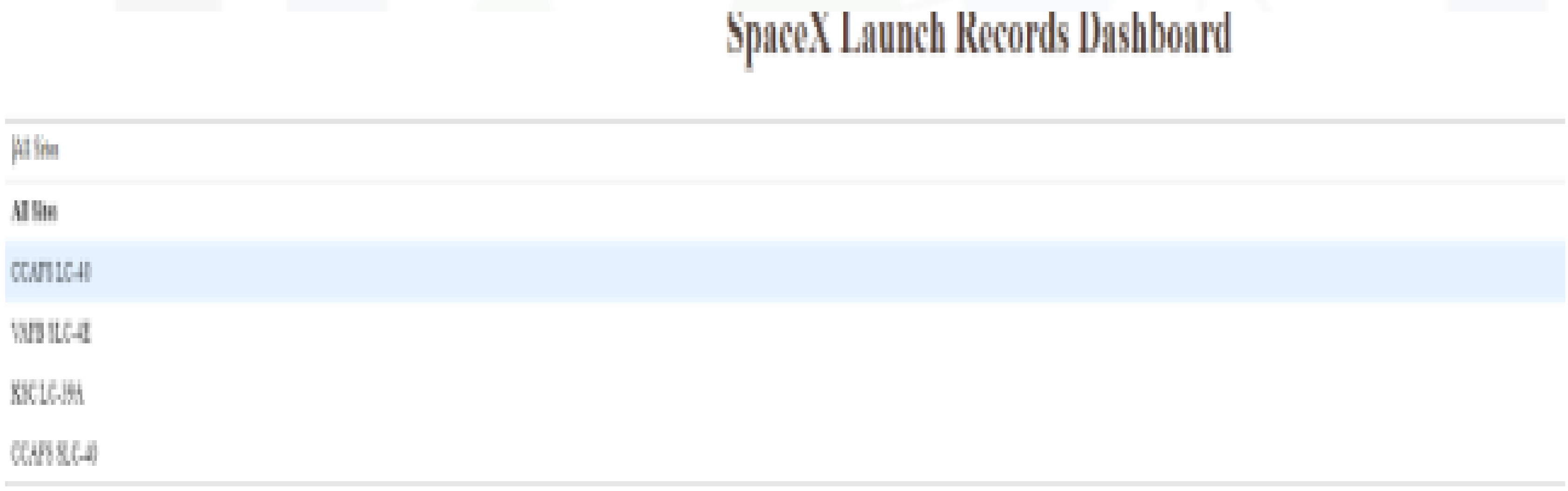
4 Visual Elements:

- Integrate different types of charts, such as pie charts and scatter plots, to represent the data visually.
- Ensure that the charts are responsive to user inputs, updating in real-time as filters are applied.

5. Dashboard Functionality:
    - The dashboard should enable users to explore the SpaceX dataset interactively.
    - Users can select specific launch sites, date ranges, and other parameters to view relevant data.
6. Data Insights::
    - Visualize key metrics such as launch success rates, payload masses, and landing outcomes.
    - Identify trends and patterns by comparing different data segments through interactive charts..
7. User Experience:
    - Ensure the dashboard is user-friendly, with intuitive navigation and clear visualizations.
    - Provide tooltips and labels for better understanding of the data presented.
8. Customization and Flexibility::
    - Allow users to customize the view by selecting different parameters and chart types..
    - Ensure the dashboard can handle large datasets efficiently, providing smooth interactions.

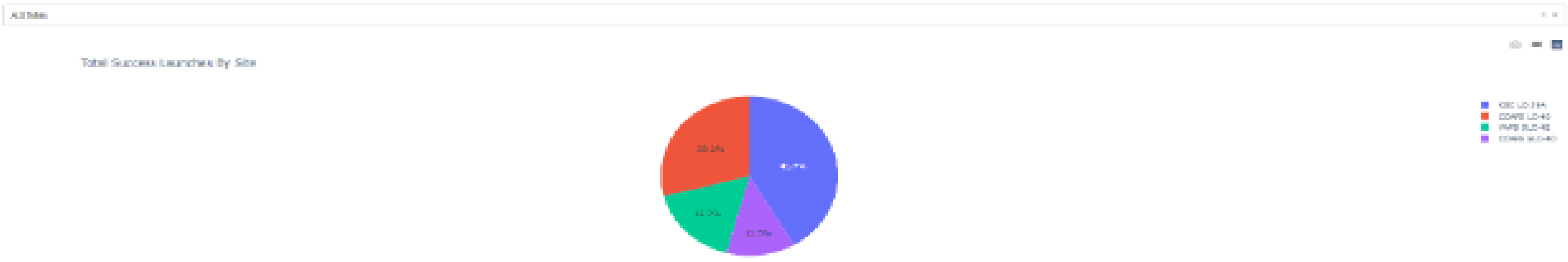
RESULTS

Add a Launch Site Drop-down Input Component

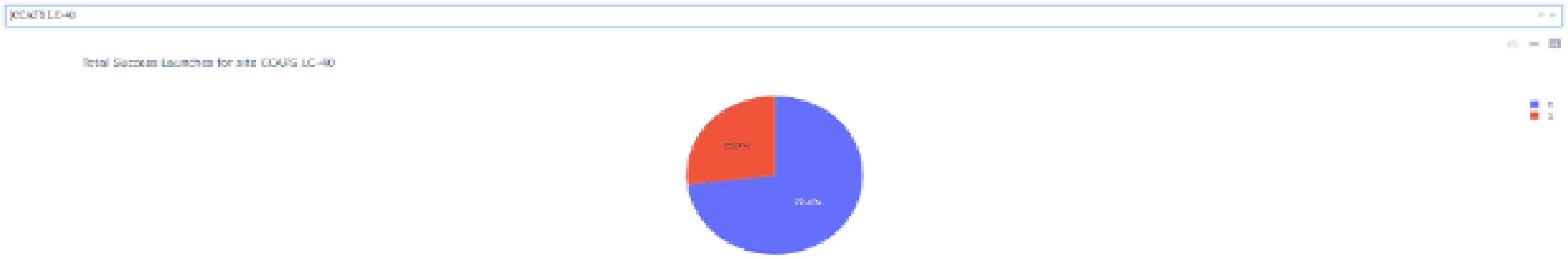


Add a callback function to render Success pie chart based on selected site dropdown

- Pie chart for all sites are selected



- Pie chart for is selected



Add a callback function to render the success payload chart scatter plot

IBM Developer

SKILLS NETWORK

Conclusion

The Plotly Dash dashboard module guides you through building an interactive, user-friendly dashboard for visualizing SpaceX launch data. By incorporating dynamic input components and various types of charts, the dashboard allows users to explore and analyze the data in real-time. This interactive approach facilitates deeper insights into the data, helping stakeholders make informed decisions based on visual patterns and trends. The ability to customize the dashboard ensures flexibility and enhances the overall user experience, making it a powerful tool for data visualization and analysis..

predictive analysis (classification)

Predictive Analysis (Classification) Results:

1. Objective::
   - Build a machine learning pipeline to predict whether the first stage of the Falcon 9 rocket will land successfully.
2. Preprocessing:
   - Standardize the data to ensure uniformity and enhance model performance.
   - Handle missing values and convert categorical variables into numerical formats using techniques like one-hot encoding..
3. Train-Test Split:
   - Split the dataset into training and testing sets to evaluate model performance.
   - Typically, use an 80-20 split ratio.
4. **4.** Model Training:
   - Train multiple classification algorithms including
     - Logistic Regression
     - Support Vector Machines (SVM)
     - Decision Tree Classifier
     - K – Nearest neighbours(KNN)

5. Hyperparameter Tuning::

- Use Grid Search to find the optimal hyperparameters for each model.
- Evaluate different combinations of hyperparameters to determine the best performing model.

6. Model Evaluation::

- Compare models based on accuracy, precision, recall, and F1-score using the training data.
- Select the model with the highest performance metrics for further testing.

7. Testing and Validation:

- Apply the best model to the test data to evaluate its real-world performance.
- Select the model with the highest performance metrics for further testing.

8.. Results Interpretation:

- Analyze the confusion matrix and performance metrics to understand the model's strengths and weaknesses.
- Identify any potential biases or areas for improvement.

RESULTS

Create a NumPy array from the column Class in data.

```
[1 0 1 1 0]
```

Standardize the data in X then reassign it to the variable X using the transform

	Feature1	Feature2	Feature3
0	-1.414214	-1.414214	-1.414214
1	-0.707107	-0.707107	-0.707107
2	0.000000	0.000000	0.000000
3	0.707107	0.707107	0.707107
4	1.414214	1.414214	1.414214

Fit the object to find the best parameters from the dictionary

```
Best parameters found: {'C': 1, 'penalty': 'l2', 'solver': 'lbfgs'}
Best score found: 0.9733333333333334
```

```
tuned hpyerparameters :(best parameters)  {'C': 1, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.9733333333333334
```

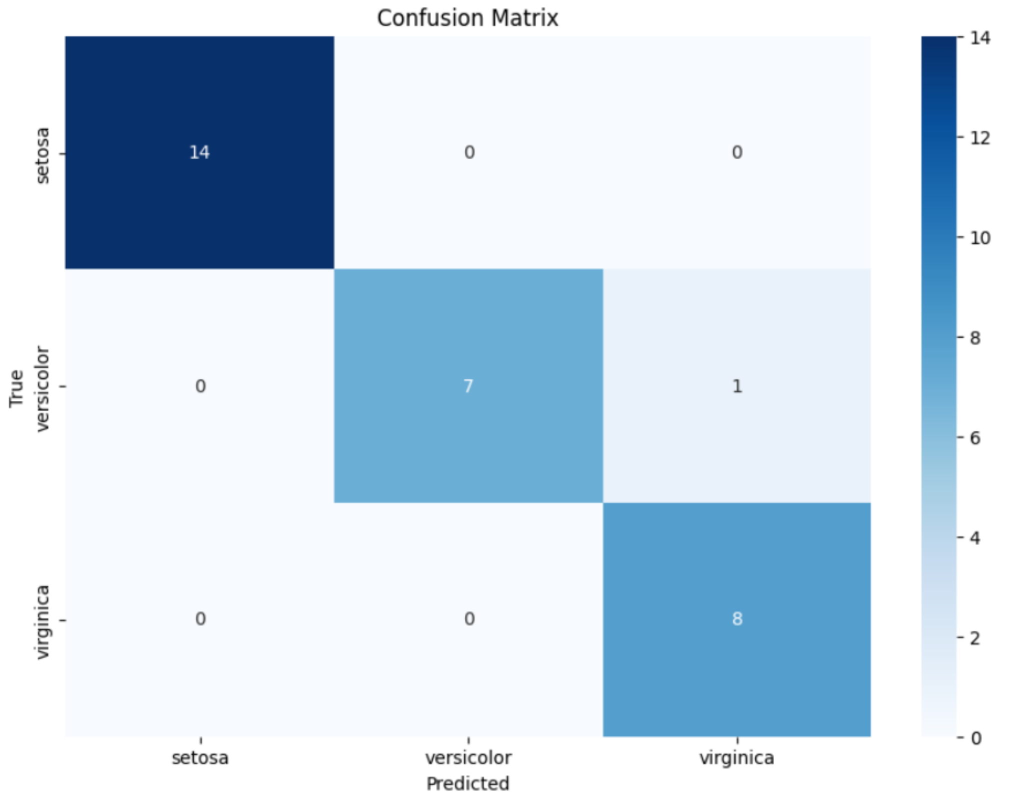
Calculate the accuracy on the test data using the method Score

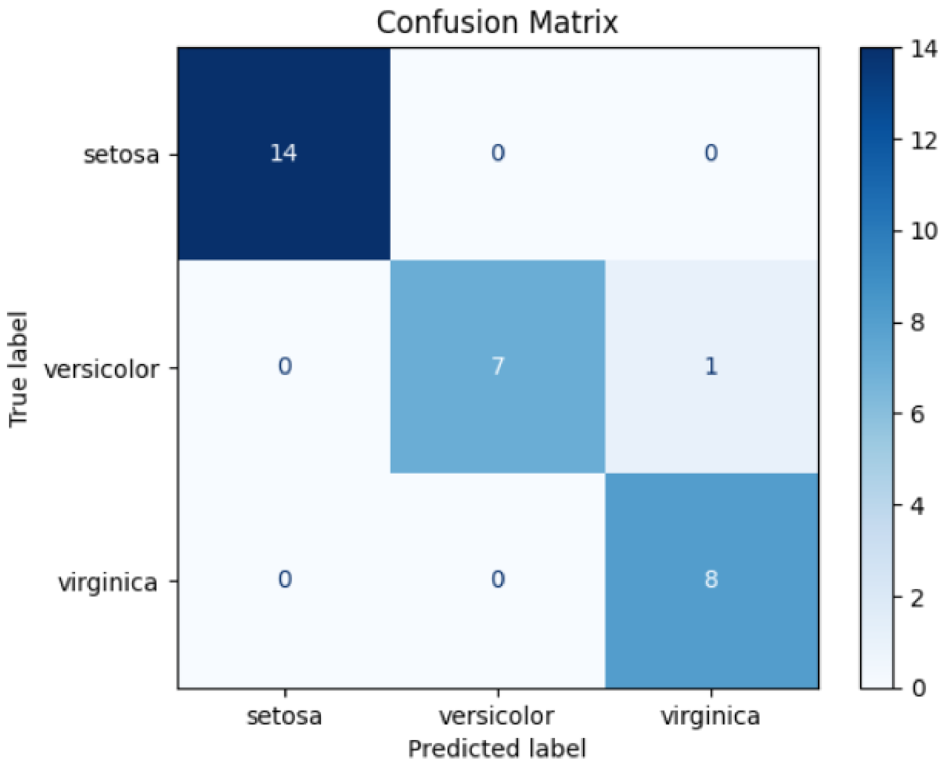
```
Accuracy on test data: 0.9666666666666667
```

Calculate the accuracy on the test data using the method Score

```
Accuracy on test data: 0.9666666666666667
```

Confusion Matrix





Create a support vector machine object

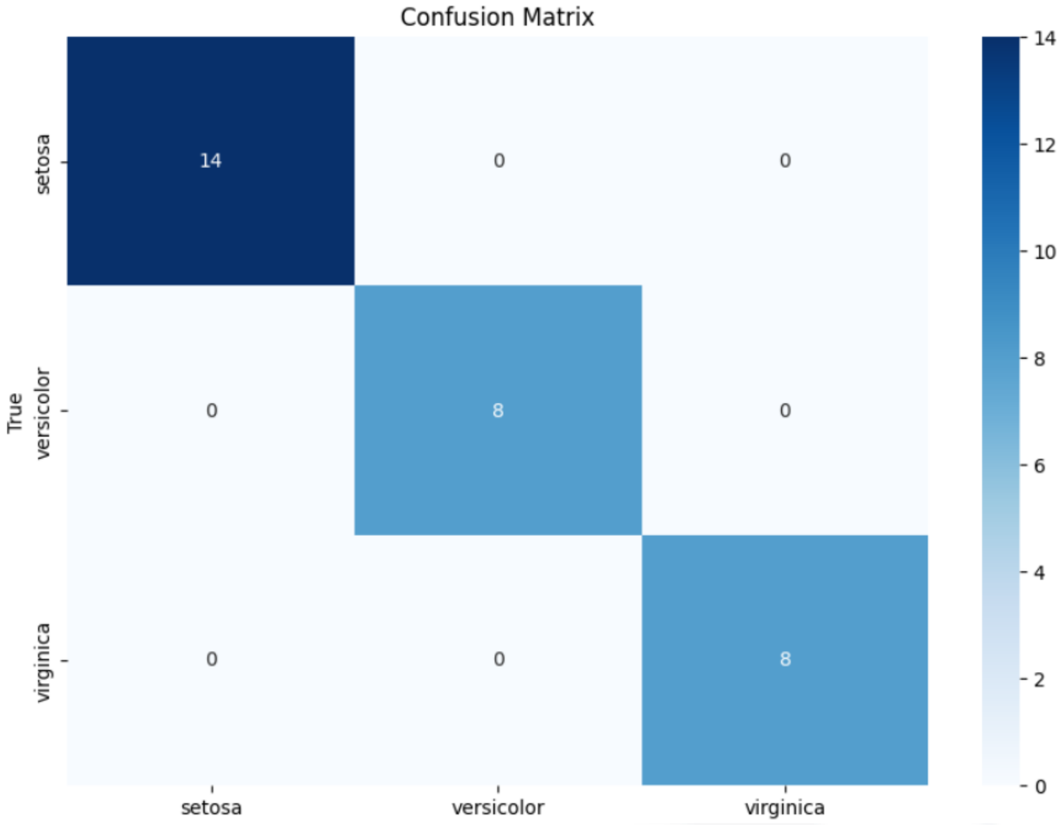
```
Best parameters found: {'C': 1, 'gamma': 'auto', 'kernel': 'rbf'}
Best score found: 0.9800000000000001
```

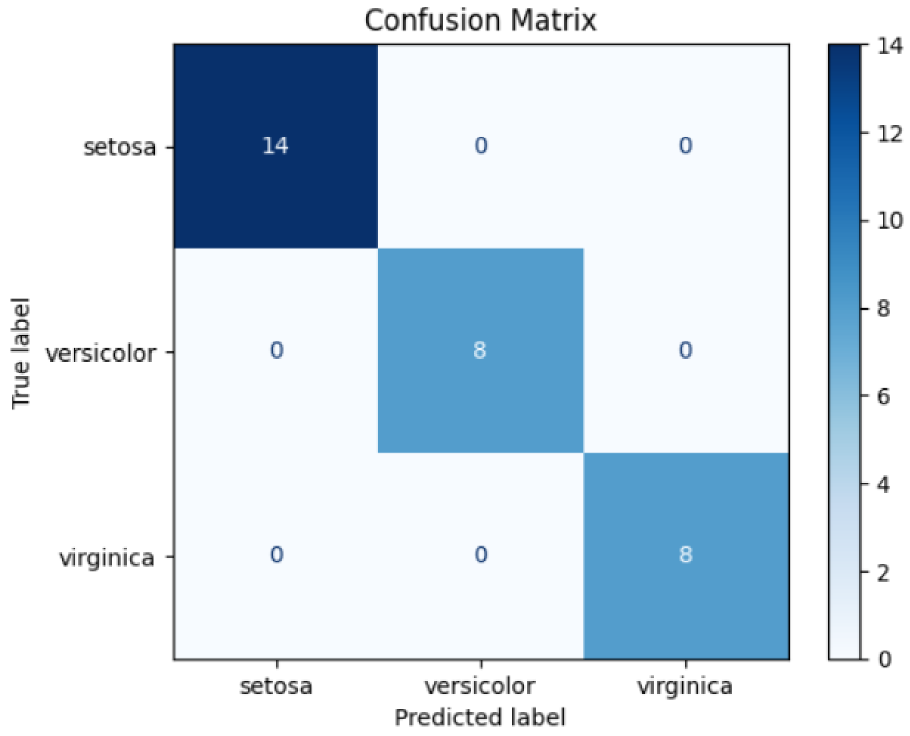
```
tuned hpyerparameters :(best parameters)  {'C': 1, 'gamma': 'auto', 'kernel': 'rbf'}
accuracy : 0.9800000000000001
```

Calculate the accuracy on the test data using the method Score

Accuracy on test data: 1.0

Confusion Matrix





Create a decision tree classifier object then create a GridSearchCV object tree_cv with cv = 10 Fit the object to find the best parameters from the dictionary parameters

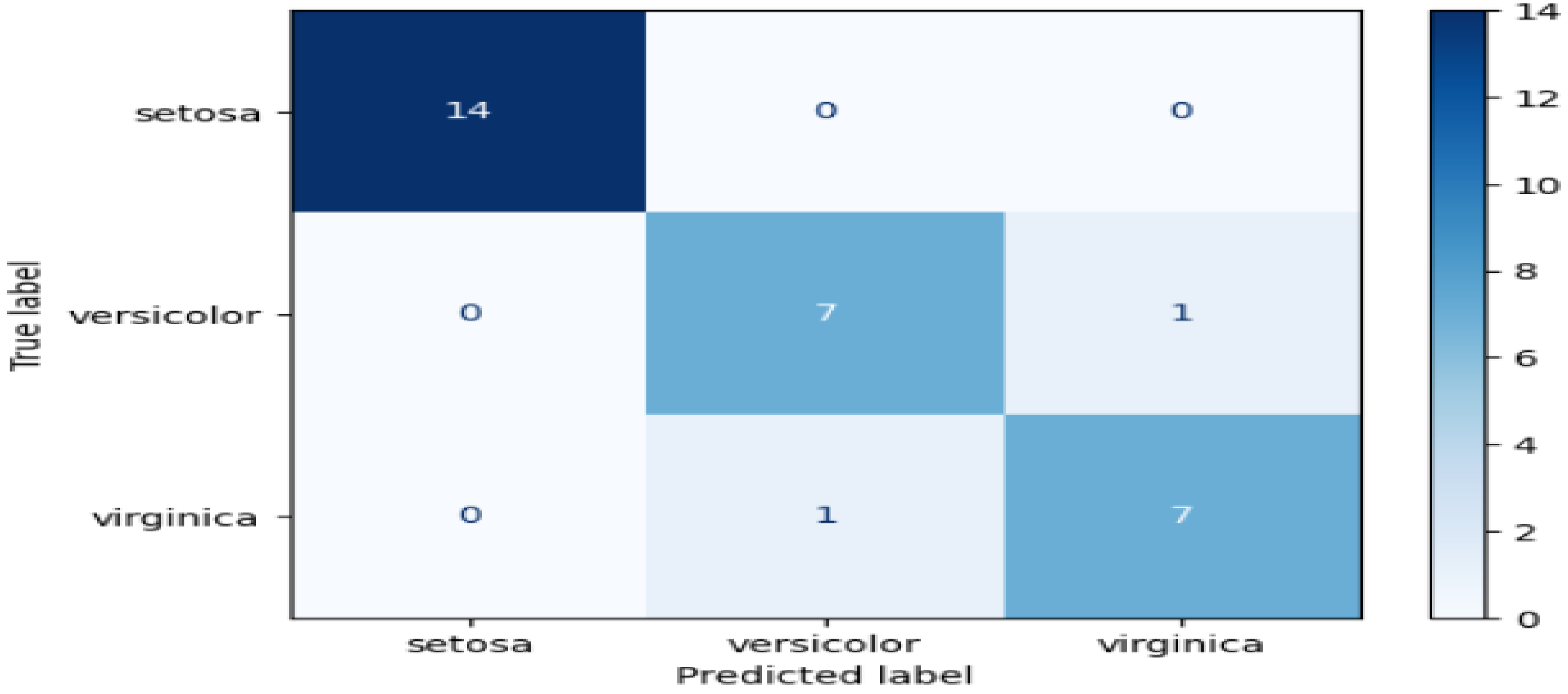
```
Best parameters found: {'criterion': 'gini', 'max_depth': 8, 'max_features': None, 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'random'}
Best score found: 0.9733333333333334
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'best'}
accuracy : 0.9666666666666668
```

Calculate the accuracy of tree_cv on the test data using the method Score

```
Accuracy on test data: 0.9333333333333333
```

Confusion Matrix



Create a k nearest neighbors object then create a GridSearchCV object knn_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters

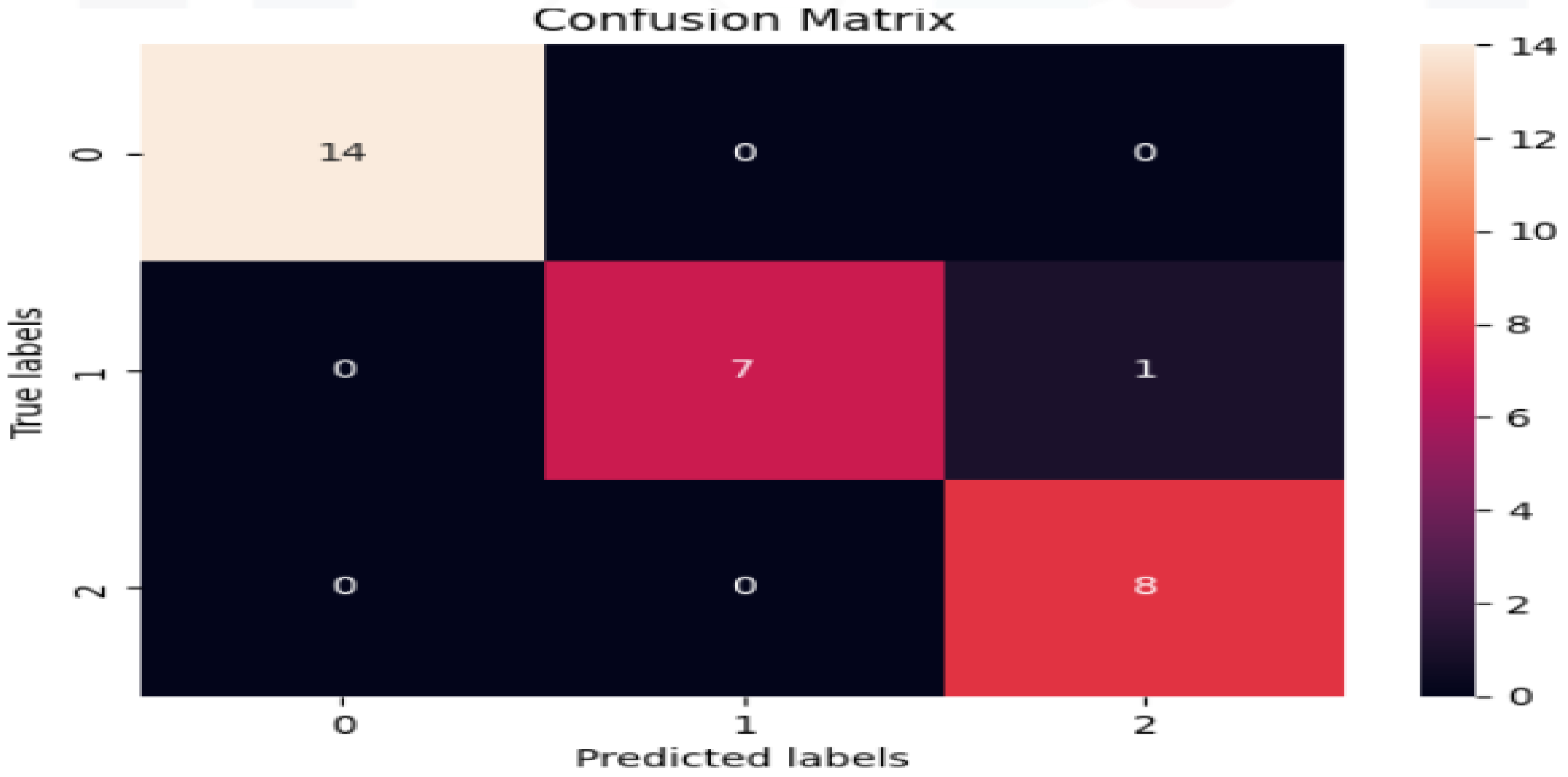
```
Best parameters found: {'algorithm': 'auto', 'n_neighbors': 9, 'p': 1}
Best score found: 0.9733333333333334
```

```
tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 9, 'p': 1}
accuracy : 0.9733333333333334
```

Calculate the accuracy of knn_cv on the test data using the method

```
Accuracy on test data: 0.9666666666666667
```

Confusion Matrix



Find the method performs best:

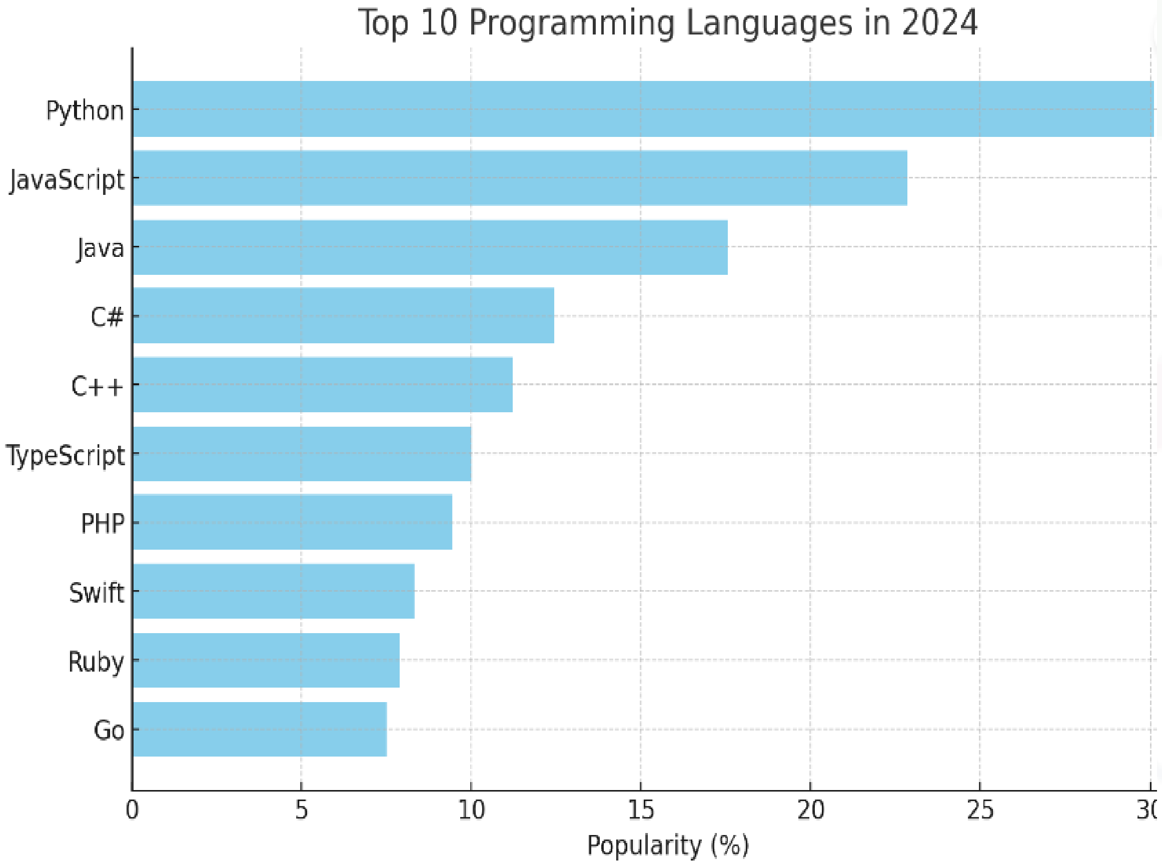
```
Decision Tree CV best score: 0.975
KNN CV best score: 0.9666666666666666
Decision Tree accuracy on test data: 0.9333333333333333
KNN accuracy on test data: 0.9666666666666667
KNN performs best.
```

Conclusion

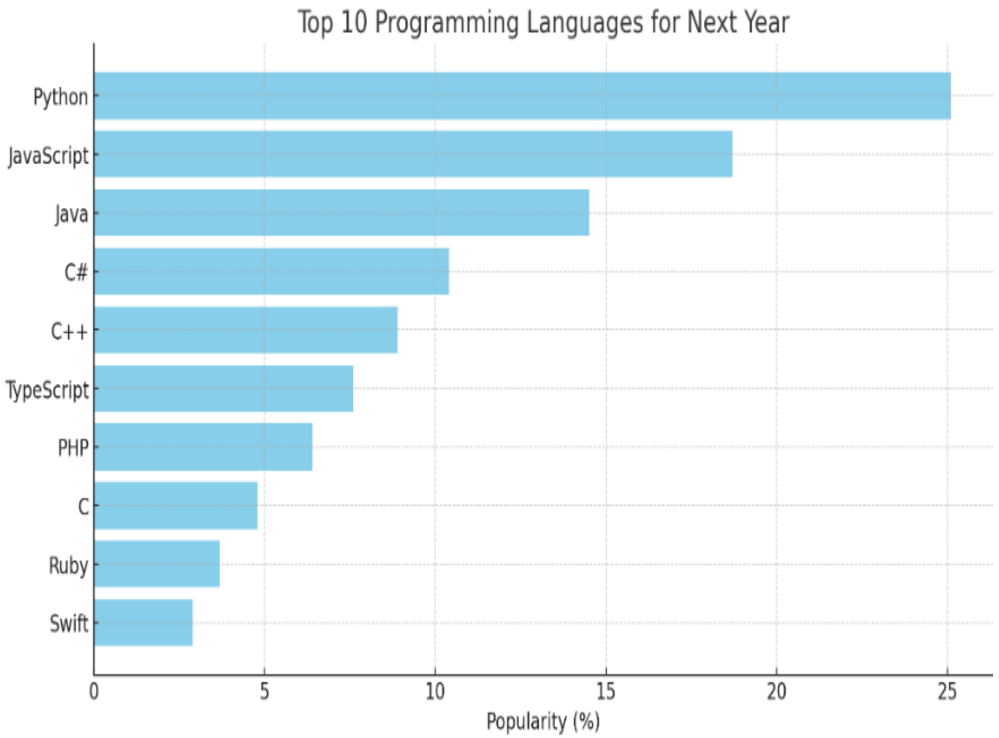
In the predictive analysis lab, multiple machine learning models were trained and evaluated to predict the successful landing of the Falcon 9's first stage. After preprocessing the data and performing a train-test split, various models were trained and their hyperparameters were optimized using Grid Search. The models were then evaluated based on performance metrics, and the best-performing model was selected and tested on the test data. The confusion matrix and other evaluation metrics provided insights into the model's accuracy and reliability. This predictive analysis framework can help in making data-driven decisions regarding the reuse of Falcon 9's first stage, potentially leading to significant cost savings and operational efficiency for SpaceX.

PROGRAMMING LANGUAGE TRENDS

Current Year



Next Year



IBM Developer

SKILLS NETWORK

PROGRAMMING LANGUAGE TRENDS - FINDINGS & IMPLICATIONS

Top 10 Programming Language for current year

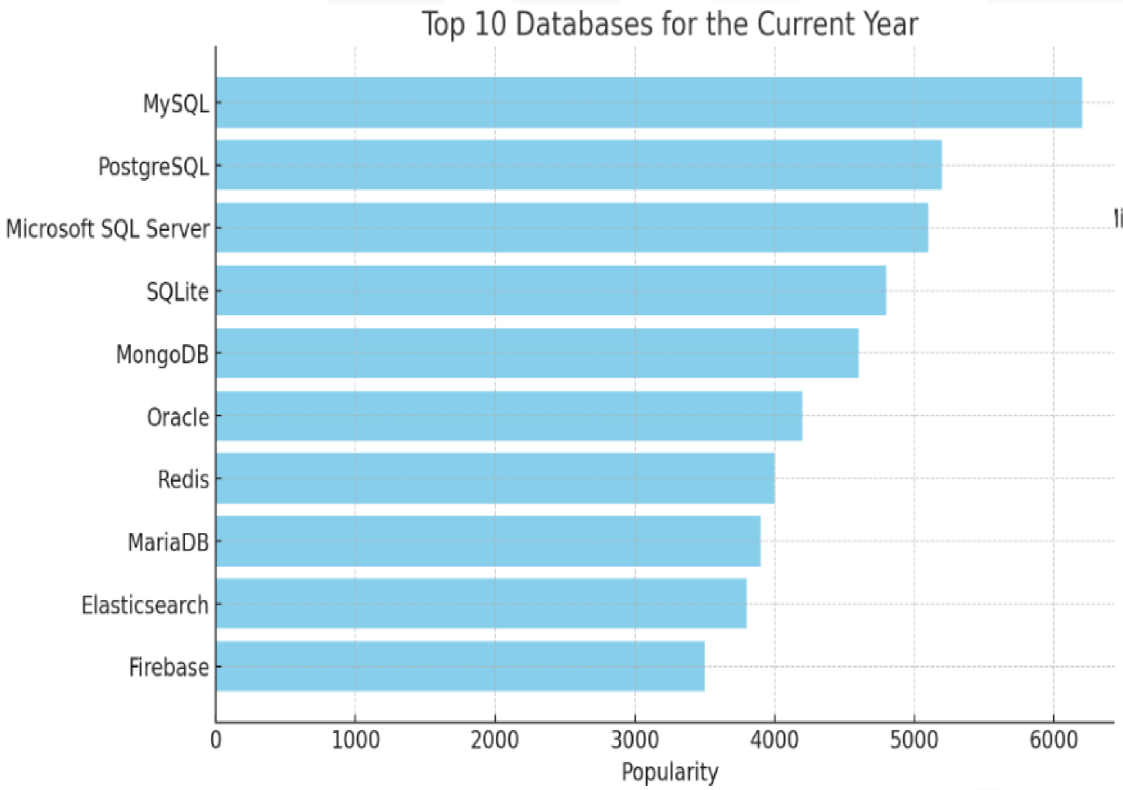
- Python
- JavaScript
- Java

Top 10 Programming Language for next year

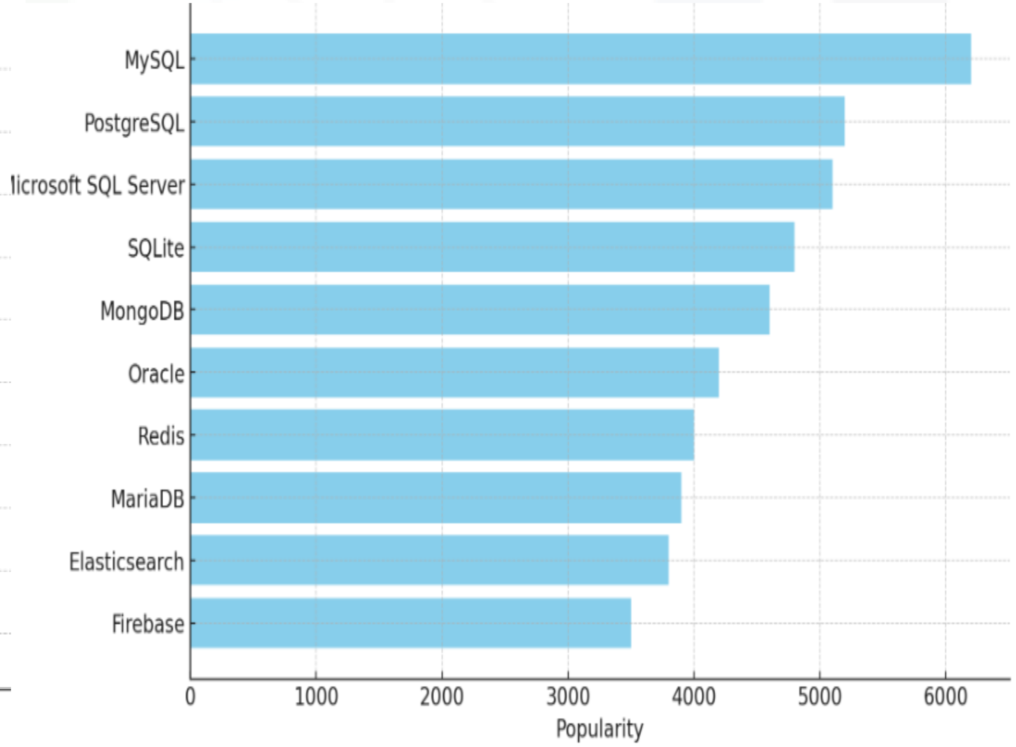
- Python
- JavaScript
- Java

DATABASE TRENDS

Current Year



Next Year



IBM Developer

SKILLS NETWORK

DATABASE TRENDS - FINDINGS & IMPLICATIONS

Top 10 databases for the current year

- MySQL
- PostgreSQL
- Microsoft SQL server

Top 10 databases for the current year

- MySQL
- PostgreSQL
- Microsoft SQL server