Space X: Falcon9 Success Analysis

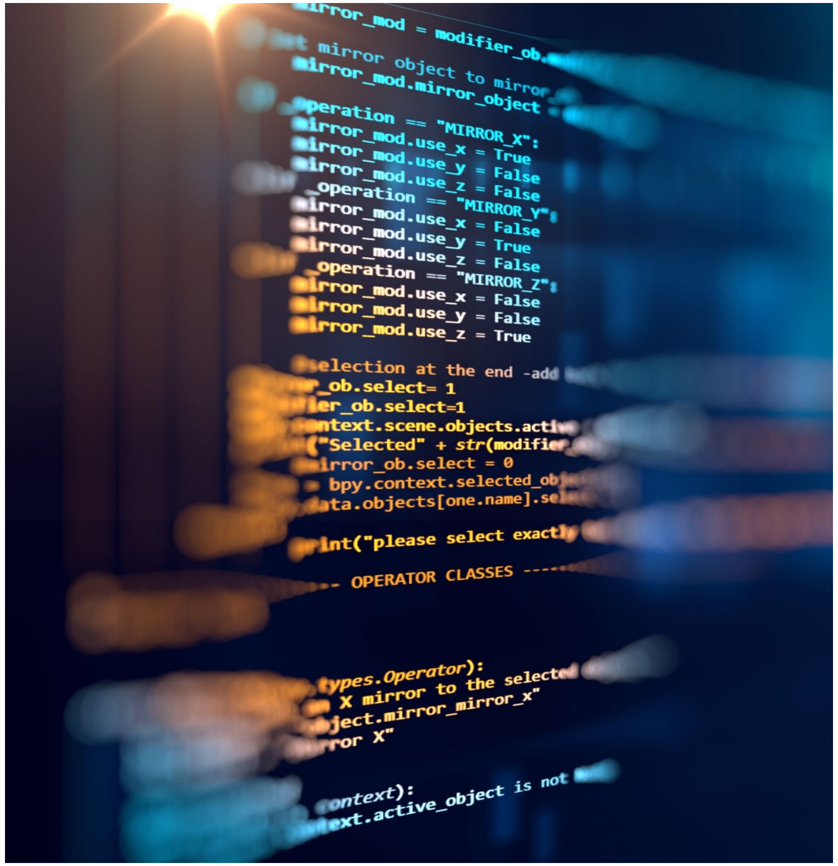
Understanding the factors that lead to Falcon9's successes or failures.

Project Accomplishments

- Identification the most important factors that lead to the success of Falcon 9 rockets given several contributing dimensions.
- Proper utilization of API to extract data from SpaceX database for analysis purposes.
- Proper utilization of BeautifulSoup Library to scrape Falcon9 Wikipedia page for more Falcon9 data.
- Training and Testing of three different machine learning models (SVM, Decision Trees, and Logistic Regression) for most accurate prediction of the success rate of Falcon9 Rocket.
- Use of Matplotlib, Plotly and Folium libraries to visualize the findings in an interactive manner.

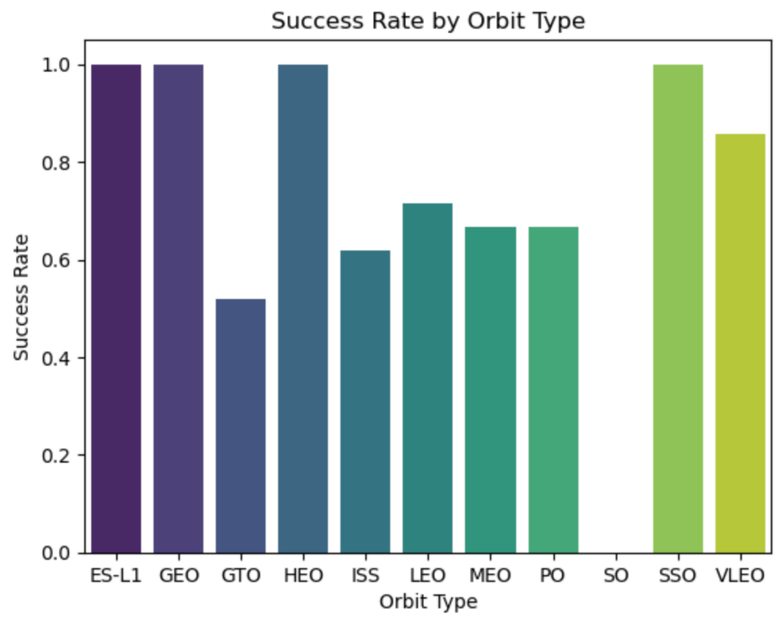
Data Extraction

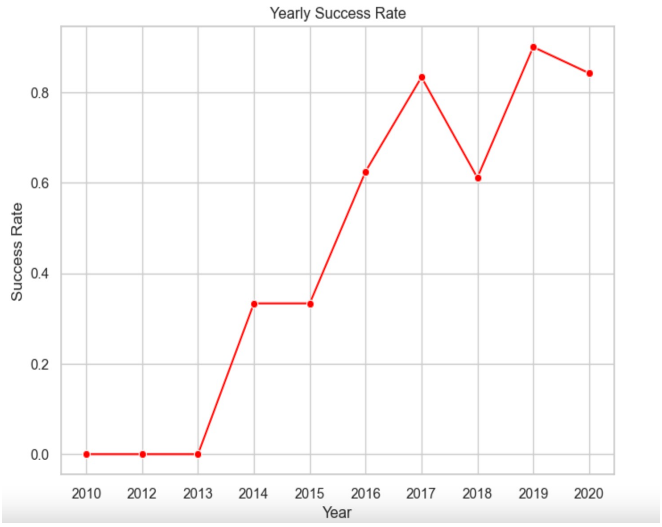
- <u>API Endpoint Definition:</u> I defined the spacex-url variable to specify the API endpoint for retrieving SpaceX launch data. This URL is where I send HTTP requests to get the required information.
- <u>API Data Retrieval Functions:</u> I created several functions, such as getBoosterVersion, getLaunchSite, getPayloadData, and getCoreData. These functions are responsible for making HTTP requests to the SpaceX API by appending specific IDs (e.g., 'rocket', 'launchpad', 'payloads', 'cores') to the API endpoints.
- <u>Data Enrichment:</u> Within these functions, I used the requests.get method to send GET requests to the SpaceX API with the appropriate IDs, retrieved the JSON data from the API response, and parsed it using .json().
- <u>Data Extraction:</u> After retrieving the data from the API, I extracted relevant information, such as booster versions, launch site details, payload data, and core details, from the JSON responses. I stored this extracted data in separate lists (e.g., BoosterVersion, LaunchSite, PayloadMass, etc.).
- <u>Data Compilation:</u> I organized the extracted data into a structured dictionary named launch_dict, which grouped the information by specific categories like flight number, date, booster version, and more.



Data Wrangling and Tidying

- I performed data quality checks on the 'df_api' DataFrame, i.e. data collected using API. I checked for missing values and found that the 'LandingPad' column had a relatively high percentage of missing data (28.89%).
- I transformed the data types of certain columns to ensure they were suitable for analysis. For example, I converted the 'Date' column to a datetime format and ensured appropriate data types for other columns such as 'PayloadMass' and boolean values for columns like 'GridFins.'
- I created a new column called 'Class' in the DataFrame, which categorized the mission outcomes into binary classes (0 or 1) based on whether they were considered successful or not. I used a predefined set of outcomes to make this classification, indicating my ability to create meaningful features for analysis.





EDA

- <u>Data Visualization:</u>

  Visualized correlations between various features using matplotlib and seaborn.

  <u>Yearly Success Trend:</u> Success rates have steadily increased since 2013, reaching a peak in recent years.

  <u>Flight Number vs. Orbit:</u> In LEO and VLEO orbits, success relates to flight numbers, while no clear trend exists in the GTO orbit.

  <u>Orbit Success Rates:</u> ES-L1, GEO, HEO, and SSO orbits have higher success rates than GTO and ISS orbits.

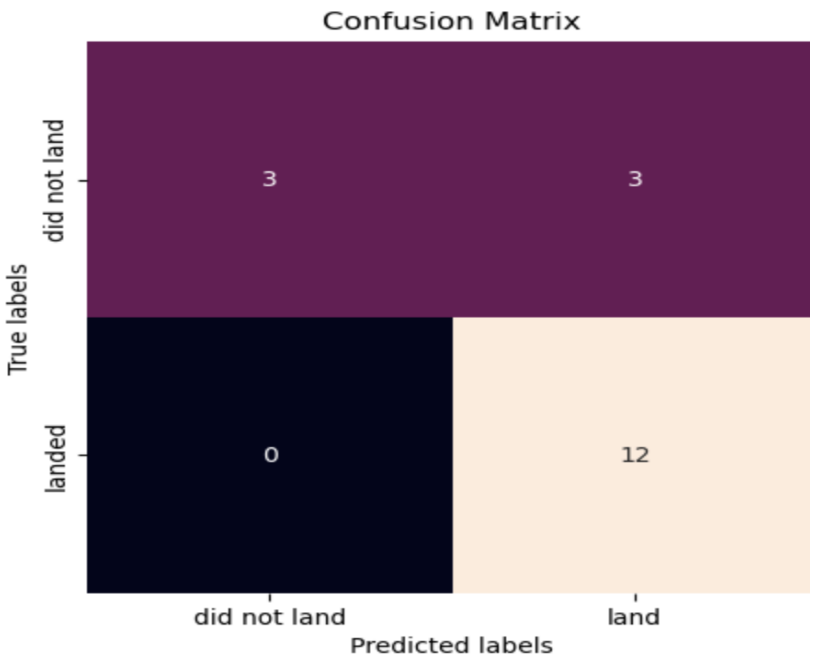
- <u>Data Transformation:</u>

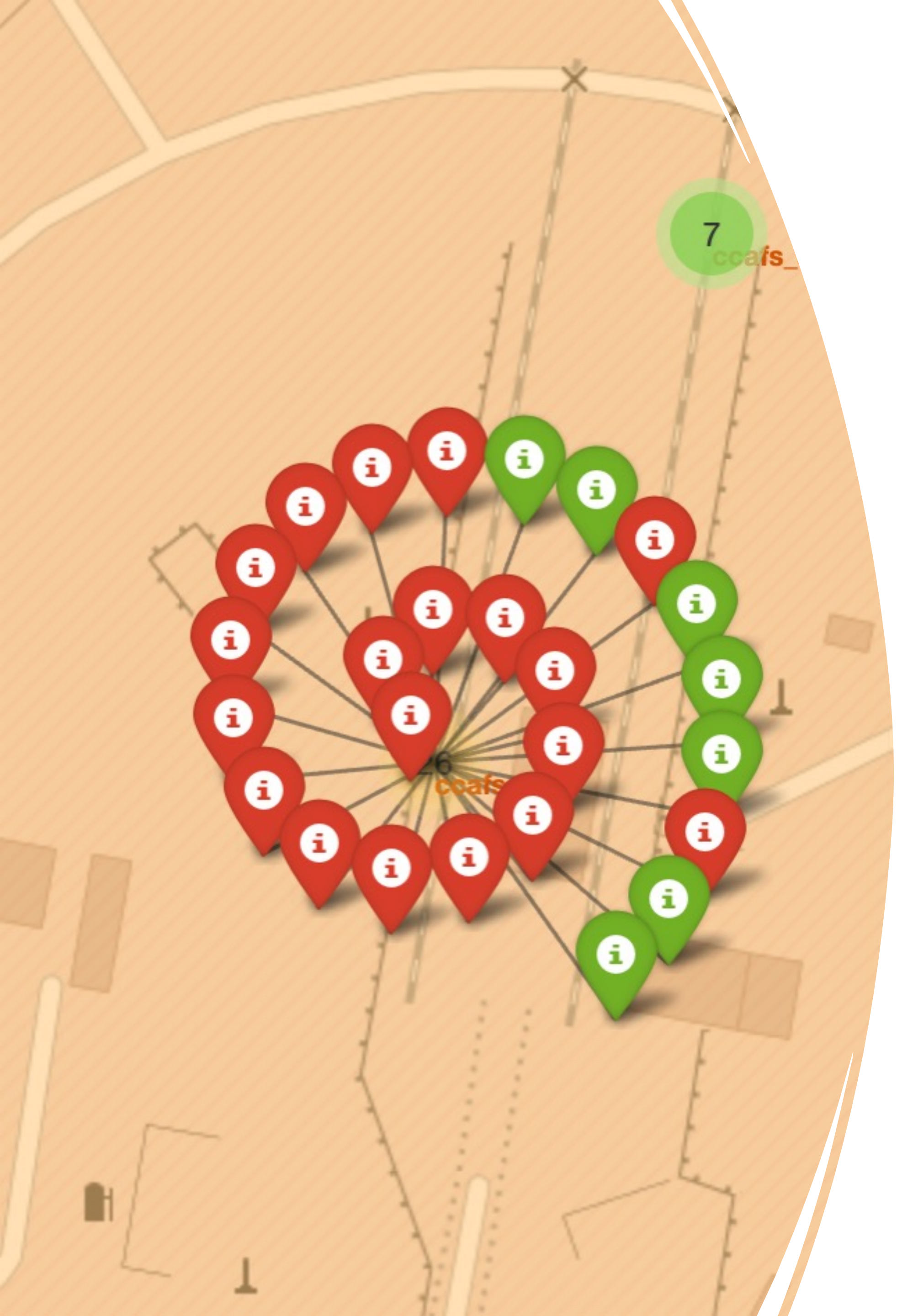
  -Performed one-hot encoding on categorical variables ('Orbit,' 'LaunchSite,' and 'Serial').

  -Numeric columns were appropriately cast to float64 data type, ensuring data compatibility for analysis.

Machine Learning

- **Data Preprocessing:** I standardized the dataset using the StandardScaler, ensuring feature consistency for machine learning.
- **Model Selection & Tuning:** Performed Training, Testing and hyperparameter tuning to optimize Logistic Regression, SVM, Decision Tree, and KNN models through grid search and cross-validation.
- **Model Evaluation:** Evaluated model performance with accuracy scores on training and test data, ensuring robust generalization.
- **Confusion Matrix Visualization:** Created insightful confusion matrix heatmaps for each model, aiding performance assessment.
- **Model Selection & Conclusion:** Selected the SVM model as the best performer with 83.3% accuracy.





Presentation & Visualization

1. Proficient in Data Presentation: Demonstrated the ability to present complex data effectively using Python, Pandas, and Folium to visualize SpaceX launch data.
2. Interactive Mapping: Created an interactive map with Folium, enhancing data exploration by allowing users to zoom, click on markers, and view launch outcomes.
3. Marker Clustering: Employed MarkerCluster to efficiently handle multiple data points at the same location, optimizing the map's performance and readability.
4. Customized Visualization: Utilized custom markers, icons, and popup labels to convey launch site information, success rates, and other details, enhancing the visual appeal and user experience.
5. Data Insights: Effectively showcased launch outcomes with color-coded markers (green for success, red for failure), facilitating quick comprehension of SpaceX mission success rates at different launch sites.

Conclusion

- The project showcases my strong data extraction, cleaning and analysis, machine-learning, visualization, and geospatial mapping skills.
- The interactive SpaceX launch map makes complex data engaging and accessible.
- This demonstrates my proficiency in presentation and visualization techniques, enhancing my ability to communicate insights effectively.