**Final Report: SpaceX Rocket Landing Success Prediction**

In this project, I worked on predicting the success of SpaceX rocket landings using data science and machine learning techniques.

SpaceX is a private aerospace company known for launching reusable rockets, and one of the biggest challenges for them is ensuring that rockets land successfully after launch. A successful landing means the rocket can be reused, which significantly reduces costs. Therefore, building a model that predicts whether a landing will be successful or not can provide valuable insights.

I began the project by gathering the dataset through the SpaceX API and CSV data. The dataset included important information such as payload mass, orbit type, launch site, landing type, and whether the landing was successful or not. While working with the data, I faced some issues, such as inconsistent column names and string values in categorical features like “orbit.” I resolved this problem by applying one-hot encoding using pd.get\_dummies(), which transformed categorical variables into numeric values that could be used for machine learning. I also dropped unnecessary columns like flight\_name and rocket\_name that did not add predictive power.

Next, I performed exploratory data analysis (EDA). Using visualization libraries such as Matplotlib, Seaborn, and Plotly, I analyzed relationships between payload mass, orbit types, and landing success. I also experimented with SQL queries to get structured insights, such as counting launches by site. Initially, I faced errors because my dataset was not loaded as a table, but I solved this by reloading the original dataset into a dataframe called df\_copy. Through EDA, I noticed that payload mass and orbit type had a strong influence on landing success.

After EDA, I moved on to preprocessing and feature engineering. All categorical features were encoded into numerical form, and I ensured that no text data was left in the feature set. This step was crucial because machine learning models cannot handle string values directly. For visualization, I also used Folium to create maps for launch sites and Plotly Dash for interactive dashboards. This helped in presenting the data in a more engaging and interactive way, making it easier to communicate insights.

For the predictive modeling part, I trained four machine learning models: Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine (SVM). I split the dataset into training and testing sets and trained each model on the training data. Then, I evaluated them using accuracy, precision, recall, F1-score, and confusion matrices. The results were interesting: Logistic Regression and SVM both achieved the highest accuracy of around 93% with a recall of 100%, meaning they correctly identified all successful landings. Decision Tree and Random Forest performed worse, with lower accuracy and recall, possibly because of overfitting.

To better present my results, I created a performance comparison table and also used Plotly to generate an interactive bar chart where accuracy, precision, recall, and F1-score could be compared across all models. This visualization made it very clear that Logistic Regression and SVM were the best models for this task.

Throughout the project, I faced multiple challenges. I struggled with file not found errors while loading the dataset, spelling mistakes in mapping target values (like “postive” instead of “positive”), SQL errors due to missing tables, and long preprocessing times when working with NLTK stopwords. However, I was able to overcome these challenges by carefully debugging, cleaning data, and optimizing code.

Model Accuracy Precision Recall F1-Score

0 Logistic Regression 0.934783 0.934783 1.000000 0.966292

1 Decision Tree 0.782609 0.945946 0.813953 0.875000

2 Random Forest 0.804348 0.947368 0.837209 0.888889

3 SVM 0.934783 0.934783 1.000000 0.966292

From the results, I learned that Logistic Regression and SVM are both strong candidates for production use in predicting rocket landing success. The high recall rate makes these models particularly useful, since missing a successful landing could be costly in practice. My recommendation is to use either Logistic Regression or SVM for predictive modeling, while also considering adding more features such as weather conditions or rocket version in future iterations to improve performance.

In conclusion, this project gave me valuable hands-on experience with working on a full data science workflow, including data collection from APIs, preprocessing, exploratory data analysis, feature engineering, machine learning, and interactive visualization. Despite the challenges I faced, I was able to build models with strong predictive performance, and I gained a deeper understanding of how data science can be applied to solve real-world aerospace problems.

