Machine Learning Project Documentation

■ Overview

This project focuses on building a machine learning model for data-driven prediction tasks. The goal is to develop an end-to-end pipeline — from data collection to model deployment — including preprocessing, EDA, feature engineering, model selection, and evaluation.

■ Dataset Description

• Source: [Add source link or dataset name] • Size: [No. of records, columns] • Target: [Target column name] • Features: [Brief description of important features]

■■ Pipeline Summary

1■■ Data Collection: Loaded dataset using pandas and verified data types, missing values, and duplicates. 2■■ Data Cleaning & Preprocessing: Lowercased text, removed URLs, HTML tags, stop words, and special characters. Applied lemmatization and filtered reviews between 3–100 words. 3■■ Visualization: Used matplotlib & seaborn for distribution plots, word counts, heatmaps, and class balance visualization. 4■■ Balancing Strategy: Used RandomOverSampler (imbalanced-learn) to create balanced classes. 5■■ Feature Engineering: Applied TF-IDF vectorization (max_features=5000) for text representation. 6■■ Train-Test Split: Used sklearn's train_test_split (80–20 split) with stratification. 7■■ Model Training: Trained DecisionTree, RandomForest, Logistic Regression models and compared metrics. 8■■ Model Evaluation: Calculated R^2, MAE, and MAPE; Random Forest performed best ($R^2 \approx 0.935$).

■ Libraries Used

• pandas • numpy • scikit-learn • nltk • matplotlib • seaborn • imbalanced-learn • joblib

■ Notes & Decisions

• Lemmatization chosen over stemming for better context retention. • TF-IDF preferred over Bag-of-Words for better feature weighting. • Random Forest selected as final model for its superior accuracy and interpretability. • Used train-test split ratio of 80–20 with stratification to ensure class balance.

■ Results & Visuals

• Added confusion matrix, accuracy charts, and feature importance plots. • Model saved as 'best_model.pkl' in /models/ directory.

■ Future Work

• Integrate GridSearchCV for tuning. • Add Flask/Streamlit app for deployment. • Implement model interpretability (SHAP/LIME). • Improve documentation and add reproducibility details.

■ GitHub Repository

Repository Link: [Add your GitHub repository URL here]