DOCUMENTATION

1. Introduction

Provide a brief overview of your project, including the purpose of the machine learning model and its real-world application.

Example:

Purpose: This project aims to predict the likelihood of company bankruptcy using a set of financial features. A FastAPI application has been built to deploy the model for prediction in a real-time scenario.

2. Rationale for Each Step

Data Preparation

- **Data Filtering & Out of Sample Data**:

We filtered out a portion of the raw data to be used for inference during testing. This step ensures that we have a distinct set of data that is not involved in training, providing a fair evaluation for out-of-sample predictions.

- **Null Imputation**:

We applied a null imputation strategy to handle missing data. The approach selected was the median imputation, which is robust to outliers and ensures that the dataset remains intact without dropping rows or columns that contain missing values.

- **Feature Generation**:

We performed feature engineering by adding meaningful features based on domain knowledge (e.g., ratio-based features). This allows the model to better understand the relationship between key variables. Non-linear transformations, such as logarithmic transformations or binning, were also applied to specific features.

- **Train/Test Split**:

We used an 80/20 split for the train/test sets. This ratio ensures that the model is trained on a sufficiently large dataset while still retaining a good portion of data for testing its generalization ability.

Feature Selection

- **Feature Importance**:

We used XGBoost and CatBoost models to assess feature importance. By comparing feature importances across different models, we obtained a more reliable understanding of which features contribute most to the model's predictions.

- **Top Features Selection**:

Based on the feature importance results, we selected the top features that had the most impact on the model's performance. These features were used for model training.

Modeling

- **Benchmarking**:

Several models were benchmarked (RandomForest, XGBoost, and CatBoost). Each model's performance was evaluated based on accuracy, precision, recall, and F1-score. The best-performing model was selected for deployment.

3. Explanation for Skipped Steps (if any)

Skipped Step: Hyperparameter tuning for CatBoost
While hyperparameter tuning for XGBoost was performed using GridSearchCV, we decided to skip hyperparameter tuning for CatBoost as its default parameters provided satisfactory results for our specific use case. The decision was made based on computational efficiency, as further tuning may not yield significantly better results.

---

4. Instructions for Running the Code

Pre-requisites

- Install the necessary libraries:

Running the Training Pipeline

- Place the training pipeline code in a Python file
- Run the training pipeline by executing:
- This will process the data, perform feature engineering, train the model, and save the trained model to a file (trained_model.pkl).

Running the FastAPI Application

- Place the FastAPI code in a separate Python file (e.g., app.py).
- Run the FastAPI server by executing:
    - This starts a local server at http://127.0.0.1:8000.

Testing the API Locally

- You can test the /predict endpoint by sending a POST request with the necessary input data to http://127.0.0.1:8000/predict.

5. Conclusion

This project successfully created a machine learning pipeline for bankruptcy prediction. The FastAPI application allows real-time inference, and the model can be deployed in production for continuous

predictions. Future improvements may include exploring deep learning models or integrating external data sources for more accurate predictions.