**Project Overview**

This project focuses on building machine learning models to predict whether an individual will default on a loan using the **German Credit Dataset** from the UCI Machine Learning Repository. The dataset includes demographic, financial, and credit history features and poses a binary classification problem:

* **Class 0**: No Default
* **Class 1**: Default

The goal is to develop a robust credit risk assessment model that can assist financial institutions in decision-making processes. After evaluating multiple models, **Logistic Regression** was finalized as the best-performing model, offering interpretability and strong performance.

**Dataset Summary**

The **German Credit Dataset** contains:

* **Instances**: 1,000
* **Features**: 20 (both numerical and categorical)
* **Target Variable**: Binary classification of credit risk:
  + 1 (Good Credit Risk)
  + 2 (Bad Credit Risk)

**Key Features**

* **Numerical**: Age, Duration of credit, Credit amount, Number of existing credits, etc.
* **Categorical**: Checking account status, Employment type, Housing situation, etc.

**How to use the Metadata?**

The metadata provides valuable insights that will guide the preprocessing and modelling steps:

Encoding Categorical Variables:

We’ll need to encode categorical variables (e.g., credit\_history, purpose, housing) into numerical values. For binary categorical variables (like telephone and foreign\_worker), use binary encoding (0/1). For multiclass categorical variables (like status\_of\_existing\_checking\_account or credit\_history), use one-hot encoding. Numerical Variables:

Scaling: Numerical features such as duration, credit\_amount, and age will need to be scaled using a method like StandardScaler (to have a mean of 0 and standard deviation of 1). Cost Matrix:

The cost matrix suggests that misclassifying a bad credit risk as good is more costly. This should be considered when tuning the model's parameters and during evaluation (e.g., using a weighted loss function or adjusting class weights). Feature Types:

Based on the feature types (categorical and numerical), we can decide on the appropriate preprocessing steps: Categorical: One-hot encoding, label encoding, or ordinal encoding. Numerical: Scaling or normalization.

**Models and Approach**

**Models Implemented**

1. **Logistic Regression** (Finalized Model)
2. **Random Forest**
3. **XGBoost**

**Optimization Technique**

**Bayesian Optimization** was employed to fine-tune hyperparameters, ensuring efficient exploration of the search space while balancing performance and computational cost.

**Evaluation Metrics**

* **Accuracy**: Measures overall prediction correctness.
* **Precision**: Evaluates true positive rates.
* **Recall**: Measures the ability to identify positive cases.
* **F1-Score**: Balances precision and recall.
* **ROC AUC**: Assesses the model's discrimination ability.

**Final Model - Logistic Regression**

**Logistic Regression** was selected for its interpretability and competitive performance.

**Final Hyperparameters**

* **C (Regularization strength)**: 100
* **Penalty**: L1 (encourages sparsity)
* **Solver**: SAGA (efficient for large datasets)
* **Max Iterations**: 100

**Performance Metrics (Test Set, Threshold: 0.4)**

| **Metric** | **Class 0 (No Default)** | **Class 1 (Default)** |
| --- | --- | --- |
| Precision | 0.87 | 0.63 |
| Recall | 0.82 | 0.71 |
| F1-Score | 0.85 | 0.67 |
| ROC AUC | 0.82 |  |

| **Confusion Matrix** | **Predicted: No Default** | **Predicted: Default** |
| --- | --- | --- |
| Actual: No Default | 116 | 25 |
| Actual: Default | 17 | 42 |

**Threshold Adjustment**: A decision threshold of 0.4 was used to prioritize identifying default cases, trading off precision for improved recall.

**Key Insights**

* **Random Forest and XGBoost** were strong contenders but slightly underperformed compared to Logistic Regression.
* **Logistic Regression** strikes a balance between interpretability and performance, making it ideal for a production-ready credit risk model.
* Bayesian Optimization significantly improved hyperparameter tuning efficiency across all models.

**Deployment**

The Logistic Regression model is ready for deployment in real-time credit scoring applications. It can:

1. **Predict Loan Default Probability**: Help banks assess credit risk efficiently.
2. **Guide Loan Decisions**: Inform loan approval processes, set credit limits, or determine loan terms.

**Deployment Considerations**

* **Monitoring**: Continuously evaluate the model's performance to detect any drift in feature distributions.
* **Updates**: Retrain the model periodically with updated data to maintain accuracy.
* **Integration**: Deploy as a web service for real-time or batch processing.

**Future Work**

1. **Model Enhancements**:
   * Explore ensemble approaches combining Logistic Regression with Random Forest or XGBoost.
   * Experiment with advanced boosting methods for potential performance gains.
2. **Feature Engineering**:
   * Create interaction terms or test alternate encodings to capture more complex relationships.
   * Perform feature selection to further refine the model.
3. **Handle Class Imbalance**:
   * Apply SMOTE or adjust class weights to better address minority class (defaults).
4. **Cross-validation**:
   * Use extensive cross-validation to validate robustness and guard against overfitting.

**Summary**

This project demonstrates the application of machine learning to predict loan defaults using the German Credit Dataset. By evaluating multiple models and leveraging Bayesian Optimization, **Logistic Regression** was identified as the optimal solution. The final model balances performance and interpretability, making it well-suited for deployment in the financial sector.

**Authors**

* **Developed by**: Krisha Jain
* **Date**: 20th November 2024

Feel free to explore the repository and reach out with any questions or suggestions!