Executive Summary
LoanGuard: Enhancing Loan Default Prediction and Credit Limit Optimization

Mandela Bright Quashie, Omeiza Olumuye, Reid Harris

December 2, 2024

Project Overview

LoanGuard is a user-friendly web application designed to assess the risk of loan default for applicants. By analyzing various financial and credit-related factors, the app provides users with an informed risk assessment, aiding both lenders and borrowers in making prudent financial decisions. Leveraging a substantial dataset from Lending Club, the project aims to develop robust machine learning models that accurately identify high-risk loan applicants and recommend appropriate credit limits, thereby mitigating financial risks and enhancing decision-making processes.

Objectives

1. **Predict Loan Default Risk:** Develop a predictive model to classify loan applications as likely to be "Fully Paid" or "Charged Off," enabling lenders to make informed lending decisions.
2. **Optimize Credit Limits:** Create a regression model to recommend optimal credit limits for borrowers based on their financial profiles and loan characteristics.

Methodology

1. Exploratory Data Analysis (EDA)

Before initiating data wrangling, a comprehensive **Exploratory Data Analysis (EDA)** was conducted to understand the underlying patterns, distributions, and relationships within the dataset. This phase included:

- **Data Visualization:** Utilized histograms, box plots, and scatter plots to visualize the distribution of key numerical features and identify potential outliers.
- **Correlation Analysis:** Assessed the relationships between features using correlation matrices to detect multicollinearity and inform feature selection.
- **Missing Values Exploration:** Identified patterns in missing data to determine appropriate imputation or removal strategies.
- **Feature Importance Insight:** Gained preliminary insights into which features might significantly influence loan outcomes, guiding subsequent feature engineering efforts.

The insights derived from EDA were instrumental in shaping the data wrangling and feature engineering processes, ensuring a data-driven approach to preparing the dataset for modeling.

2. Data Wrangling and Cleaning

- **Irrelevant Columns Removal:** Eliminated non-essential columns such as unique identifiers (`id`, `member_id`), URLs, and address details to streamline the dataset.
- **Target Variable Filtering:** Focused analysis on critical loan statuses by retaining only "Fully Paid" and "Charged Off" records, resulting in a balanced subset of approximately 1.35 million records.
- **Single-Value Columns Elimination:** Dropped columns exhibiting no variability (e.g., `pymnt_plan`, `policy_code`) to prevent redundancy and reduce dimensionality.
- **Handling Missing Values:**
  - *Dropping Columns with Excessive Missing Data:* Removed columns with more than 10% missing values to maintain data integrity.
  - *Critical Missing Values Treatment:* Discarded records missing essential information (`emp_length`, `last_credit_pull_d`, `last_pymnt_d`) to ensure the reliability of subsequent analyses.
  - *Imputation:* Applied mean imputation for remaining numerical columns with missing values, ensuring continuity in the dataset without introducing significant bias.
- **Feature Engineering:**
  - *Term Conversion:* Transformed the `term` column from textual to numerical format (e.g., "36 months" to `36.0`) to facilitate quantitative analysis.
  - *Employment Length Processing:* Extracted numerical representations from the `emp_length` column, substituting ambiguous entries (e.g., "¡ 1 year") with appropriate numerical values (`0.5`).
  - *Date Components Extraction:* Segmented date-related columns into separate month and year numerical features to enhance temporal feature representation.
- **Categorical Variables Encoding:** Employed one-hot encoding to convert categorical variables (e.g., `grade`, `home_ownership`) into numerical formats, enabling their effective utilization in machine learning models.
- **Feature Reduction Using Logistic Regression:**
  - *Purpose of Feature Reduction:* Streamlined the model by retaining only the most impactful predictors, reducing computational complexity, and preventing overfitting.
  - *Logistic Regression as a Tool for Feature Selection:* Implemented L1 (Lasso) regularization to penalize the absolute size of coefficients, encouraging sparsity and effectively driving less important feature coefficients to zero.
  - *Outcome of Feature Reduction:* Enhanced model simplicity and improved predictive performance by eliminating redundant or less informative features.
- **Outlier Removal:** Identified and excluded outliers beyond the 0.5th and 99.9th percentiles in numerical features to prevent skewed model training and ensure robust performance.
- **Target Variable Encoding:** Transformed the `loan_status` column into a binary `loan_outcome` variable (`0` for "Fully Paid" and `1` for "Charged Off") to streamline classification tasks.

3. Data Processing and Balancing

- **Train-Test Split with Stratification:** Divided the dataset into training (70%) and testing (30%) sets using stratified sampling to preserve the original class distribution, ensuring that both sets accurately represent the underlying population.
- **Feature Scaling:** Applied **MinMaxScaler** to rescale numerical features between `0` and `1`, standardizing the data to improve model convergence and performance.
- **Addressing Class Imbalance:** Utilized **RandomUnderSampler** from the **imbalanced-learn** library to balance the training data by undersampling the majority class ("Fully Paid"). This strategy ensures that the model does not become biased towards the majority class, enhancing its ability to accurately predict the minority class ("Charged Off").

4. Model Development and Evaluation

- **Model Training:** Trained multiple machine learning models, including **Logistic Regression**, **Naive Bayes**, **Decision Tree**, **Random Forest**, **Gradient Boosting**, and **XGBoost**, on the balanced and scaled training dataset to predict loan default risks.
- **Performance Evaluation:** Assessed model performance using metrics such as **Accuracy**, **Precision**, **Recall**, **F1-Score**, and **ROC AUC** on the untouched test dataset to gauge real-world applicability and robustness. Cross-validation was employed to ensure the reliability of performance estimates.

5. Integration into the Streamlit Application

- **User Interface Design:** Designed an intuitive sidebar for user inputs, allowing users to enter financial details such as current balance, number of past delinquencies, and credit limits. The main display area showcases risk assessment results and feature importance visualizations.
- **Real-Time Risk Assessment:** Upon entering their financial details, the app processes the inputs using the trained XGBoost model to predict the risk category—'High', 'Moderate', or 'Low'—and displays the results with clear indicators and explanations.
- **Feature Importance Visualization:** Incorporated interactive charts to present feature importance, helping users understand which financial factors most significantly influence their risk assessment.
- **Model Training Interface:** Provided functionality within the app for administrators to retrain the XGBoost model directly from the interface, ensuring the model remains updated with new data.

6. Deployment and Continuous Improvement

- **Deployment Strategy:** Deployed the application using platforms compatible with Streamlit, such as Streamlit Sharing and cloud services like AWS or GCP, ensuring accessibility and scalability.
- **Monitoring and Maintenance:** Implemented monitoring systems to track model performance over time and incorporated user feedback mechanisms to facilitate ongoing enhancements and feature additions.
- **Security and Data Privacy:** Ensured that user data is handled securely by implementing measures to protect sensitive financial information and adhering to relevant data protection regulations.

Tools and Technologies

- **Programming Language:** Python
- **Libraries and Frameworks:**
  - **Data Manipulation:** Pandas, NumPy
  - **Machine Learning:** Scikit-learn, XGBoost
  - **Handling Imbalanced Data:** Imbalanced-learn
  - **Serialization:** Joblib
  - **Web Application:** Streamlit
- **Environment Management:** Anaconda (Conda environments)
- **Version Control:** Git and GitHub

Modeling Outcomes

The project evaluated the performance of six machine learning models, each demonstrating varying degrees of accuracy and efficiency:

1. Logistic Regression

- **Mean Accuracy:** 99.65%
- **ROC AUC:** 0.9949
- **Confusion Matrix:**

```
[[275507    701]
[   498  64104]]
```

- **Remarks:** Exhibited high accuracy and excellent ability to distinguish between classes.

2. Naive Bayes

- **Mean Accuracy:** 94.72%
- **ROC AUC:** 0.8895
- **Confusion Matrix:**

```
[[271358   4850]
[ 13139  51463]]
```

- **Remarks:** While moderately accurate, it showed lower performance compared to other models, particularly in recall for the minority class.

3. Decision Tree

- **Mean Accuracy:** 99.57%
- **ROC AUC:** 0.9958
- **Confusion Matrix:**

```
[[275020   1188]
[   269  64333]]
```

- **Remarks:** Demonstrated high accuracy and strong classification capabilities with minimal misclassifications.

4. **Random Forest**

   - **Mean Accuracy:** 99.85%
   - **ROC AUC:** 0.9970
   - **Confusion Matrix:**

     ```
     [[276050    158]
     [   346  64256]]
     ```

   - **Remarks:** Achieved the highest accuracy and ROC AUC among the evaluated models, indicating superior performance and reliability.

5. **Gradient Boosting**

   - **Mean Accuracy:** 99.64%
   - **ROC AUC:** 0.9962
   - **Confusion Matrix:**

     ```
     [[275851    357]
     [   405  64197]]
     ```

   - **Remarks:** Exhibited excellent accuracy and ROC AUC, closely trailing the Random Forest model in performance.

6. **XGBoost**

   - **Mean Accuracy:** 99.92%
   - **ROC AUC:** 0.9994
   - **Confusion Matrix:**

     ```
     [[276194     14]
     [    73  64529]]
     ```

   - **Remarks:** Delivered outstanding accuracy and ROC AUC, making it the top-performing model for loan default prediction in this project.

Outcomes and Impact

- **Data Integrity and Quality:** Through meticulous data cleaning, feature engineering, and outlier removal, the project ensured a high-quality dataset primed for accurate model training.
- **Balanced and Representative Models:** Addressing class imbalance through undersampling led to models that are better equipped to predict minority class instances, reducing the risk of overlooking high-risk loan applicants.
- **Superior Model Performance:** The **XGBoost** model emerged as the most effective, achieving an impressive **99.92% accuracy** and a **ROC AUC of 0.9994**, indicating exceptional discriminative ability between loan outcomes.
- **Scalable and User-Friendly Deployment:** The development of the **LoanGuard App** via Streamlit offers an accessible platform for lenders to leverage predictive analytics in real-time, enhancing decision-making processes and operational efficiency.
- **Future-Proofed Analytical Pipeline:** By adhering to best practices in data processing and maintaining a modular, well-documented codebase, the project ensures ease of maintenance, scalability, and adaptability to future requirements or data updates.

Conclusion

LoanGuard successfully harnessed advanced data processing techniques, comprehensive exploratory analysis, and state-of-the-art machine learning methodologies to deliver predictive insights essential for mitigating financial risks in lending practices. The structured approach—from EDA and data wrangling to model training and deployment—ensures that lending institutions can make informed, data-driven decisions, ultimately fostering a more secure and efficient lending ecosystem.

Next Steps

1. **Model Enhancement:** Continuously refine and optimize machine learning models by incorporating additional features, experimenting with different algorithms, and tuning hyperparameters to boost predictive performance.

2. **Integration with Real-Time Systems:** Integrate the **LoanGuard App** with existing lending platforms to enable seamless real-time predictions and decision-making.

3. **Monitoring and Maintenance:** Implement monitoring systems to track model performance over time, ensuring sustained accuracy and reliability, and schedule regular updates to accommodate new data trends.

4. **Expansion of Use Cases:** Explore additional applications such as identifying fraudulent loan applications, segmenting borrowers for targeted marketing, and enhancing customer relationship management based on predictive insights.

5. **Stakeholder Training and Support:** Provide training sessions and documentation for lending institution staff to effectively utilize the **LoanGuard App** and interpret model predictions.