Loan Default Prediction with LightGBM: A Comprehensive Tutorial

1. Introduction and Motivation

Financial institutions face the task of loan default prediction. Under a competitive lending environment, banks can reduce own risks, maximize interest rates and create an effective risk management strategy by accurately identifying possible defaulters. Predicting defaults matters because it can help minimize losses of financial alike by extension of credit to creditworthy borrowers.

In this tutorial, we use LightGBM which is a very efficient gradient boosting framework to build a model for the loan default prediction (Ke, 2017). A dataset consisting of more than 250,000 loan records with 18 columns (both numerical (Age, Income, LoanAmount, CreditScore) and categorical (Education, EmploymentType, MaritalStatus)) makes up this dataset. On training splits, our model attained a best cross validation accuracy around 88.64%.

2. Data Overview and Preprocessing

2.1 Dataset Structure

The dataset used in this tutorial, "loan_default.csv," contains 255,347 records and 18 columns. The available columns are:

- **LoanID**: Unique identifier for each loan record.
- **Age**: Borrower's age.
- **Income**: Borrower's income.
- **LoanAmount**: Amount of the loan.
- **CreditScore**: Borrower's credit score.
- **MonthsEmployed**: Number of months the borrower has been employed.
- **NumCreditLines**: Number of credit lines.
- **InterestRate**: Interest rate on the loan.
- **LoanTerm**: Duration of the loan in months.
- **DTIRatio**: Debt-to-income ratio.
- **Education**: Categorical variable indicating the level of education.
- **EmploymentType**: Type of employment.
- **MaritalStatus**: Marital status of the borrower.
- **HasMortgage**: Indicates whether the borrower has a mortgage.
- **HasDependents**: Indicates whether the borrower has dependents.
- **LoanPurpose**: Purpose of the loan.
- **HasCoSigner**: Indicates whether the loan has a co-signer.
- **Default**: Target variable (0 = No default, 1 = Default).

2.2 Data Quality and Missing Values

The dataset has been kept as well maintained since there are no missing values in any of its column (out of its 255,347 entries). By doing so, we avoid the need for too much imputation in our preprocessing pipeline, but we still supply this median and mode imputation as a best practice in the case there is some unexpected missing data.

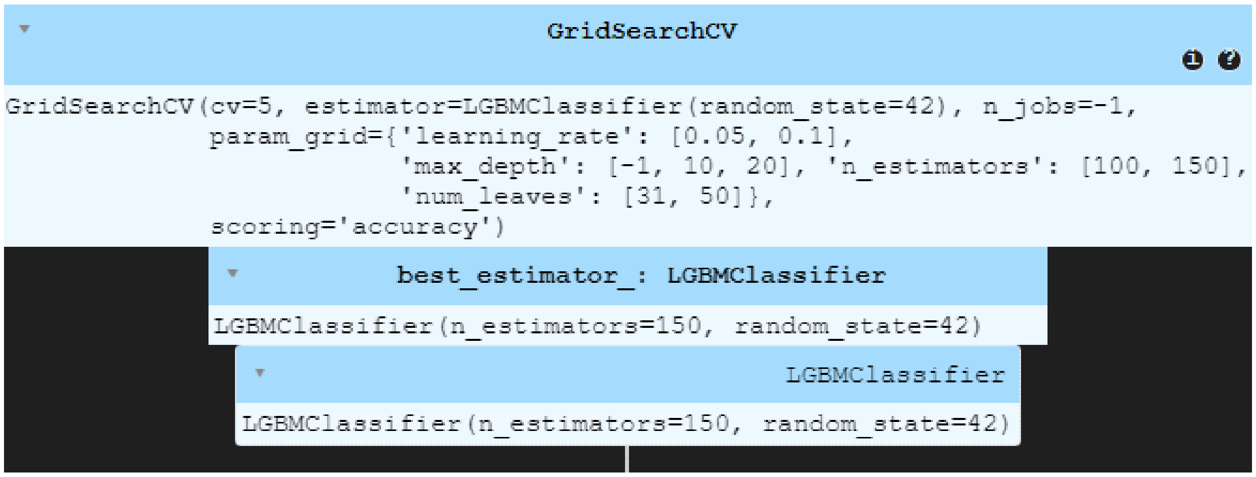
2.3 Feature Selection and Transformation

Since LoanID is nothing more than an identifier, we drop it in order to not overfit on uninformative data. The numerical and categorical groups are formed by dividing the rest of the features. (Pedregosa, 2011)

- **Numerical Features**: Age, Income, LoanAmount, CreditScore, MonthsEmployed, NumCreditLines, InterestRate, LoanTerm, DTIRatio.
- **Categorical Features**: Education, EmploymentType, MaritalStatus, HasMortgage, HasDependents, LoanPurpose, HasCoSigner.

For numerical features, we standardize the data through application of StandardScaler to have all numerical features zero mean and unit variance. Thus for categorical features, we apply One Hot Encoding with the drop='first' to avoid the multicollinearity.

3. Model Training with LightGBM



```
GridSearchCV(cv=5, estimator=LGBMClassifier(random_state=42), n_jobs=-1,
             param_grid={'learning_rate': [0.05, 0.1],
                         'max_depth': [-1, 10, 20], 'n_estimators': [100, 150],
                         'num_leaves': [31, 50]},
             scoring='accuracy')
```

```
LGBMClassifier(n_estimators=150, random_state=42)
```

```
LGBMClassifier(n_estimators=150, random_state=42)
```

3.1 Hyperparameter Tuning with GridSearchCV

LightGBM, a well known gradient boosting framework for both speed and efficiency, is used by us. Additionally, loan default prediction is very complex and therefore so is the hyperparameter tuning of the same. Finally, we specify the grid of key parameters. (Lundberg, 2017)

- **num_leaves**: Controls the complexity of each decision tree.
- **max_depth**: Limits the depth of each tree (-1 indicates no limit).
- **learning_rate**: The step size for boosting iterations.
- **n_estimators**: The number of boosting rounds.

Using GridSearchCV with 5-fold cross-validation, we optimize the parameters to maximize accuracy. The best combination found is:

```
{'learning_rate': 0.1, 'max_depth': -1, 'n_estimators': 150, 'num_leaves': 31}
```

This tuning yields a best cross-validation accuracy of approximately 88.64%.

3.2 Final Model Evaluation

After tuning, we retrain LightGBM on the full training set with the best parameters. Predictions on the test set provide:

- **Predicted probabilities**: For generating ROC curves.
- **Binary predictions**: For generating classification metrics.

4. Model Evaluation

4.1 Classification Metrics

The evaluation on the test set shows an overall accuracy of 89%. However, the classification report indicates a severe imbalance in performance between the two classes:

- **Class 0 (No Default)**: Precision 0.89, Recall 0.99, F1-Score 0.94.
- **Class 1 (Default)**: Precision 0.59, Recall 0.07, F1-Score 0.13.

The extremely low recall for defaults (only 7%) implies that the model is missing nearly 93% of defaulters, a critical concern in real-world applications.

4.2 Confusion Matrix

The confusion matrix is as follows:

- **True Negatives (TN)**: 44,853 correctly predicted non-defaults.
- **False Positives (FP)**: 286 non-defaults wrongly classified as defaults.
- **False Negatives (FN)**: 5,512 actual defaults missed.
- **True Positives (TP)**: 419 actual defaults correctly predicted.

In this matrix, you can see how it really likes to predict the majority class (non defaults) and is missing the minority class in general.

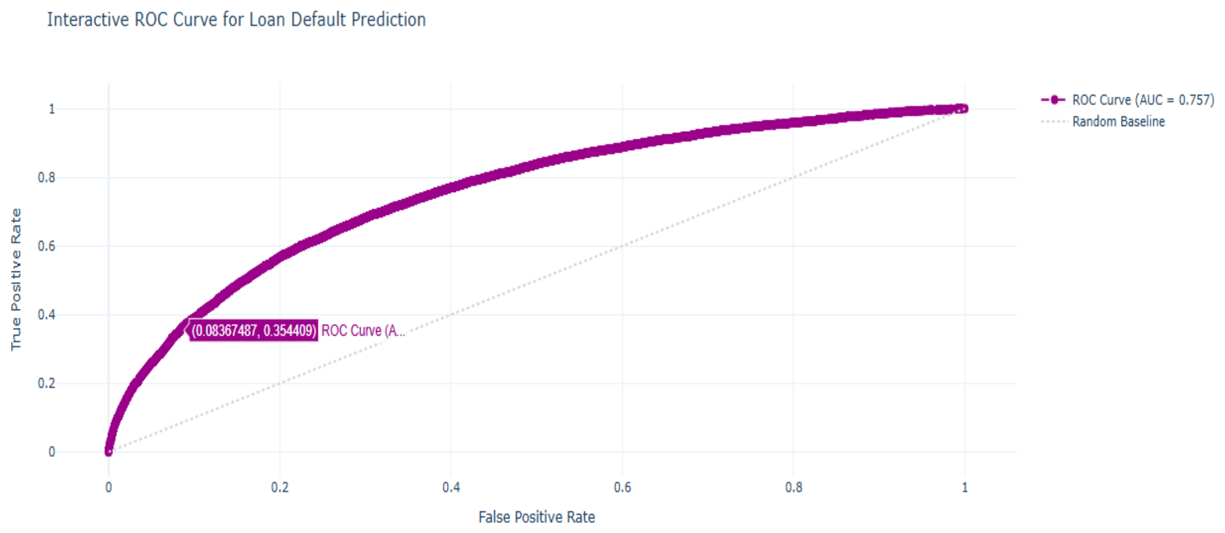
4.3 ROC AUC

The ROC AUC score is 0.757, which is moderately discriminatory power. This is because of the imbalanced dataset, and the ROC curve is less useful than other metrics like precision – recall. The model can somewhat rank transaction by their risk, but the threshold for classification is suboptimal as indicated by the AUC.

5. Visual Analysis and Interpretations

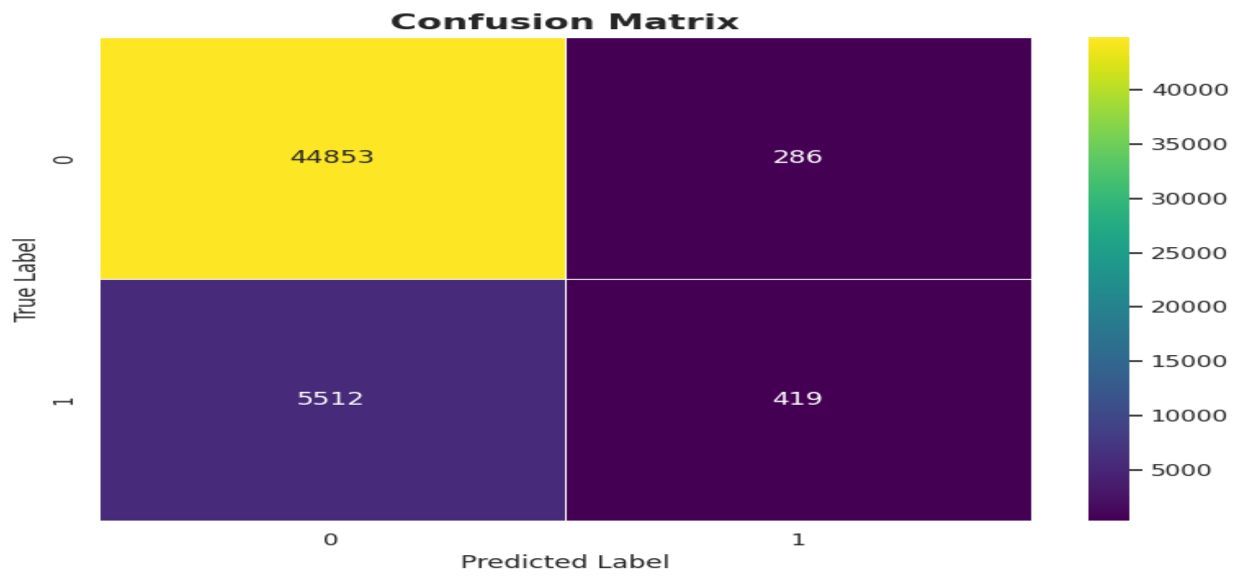
5.1 Interactive ROC Curve

ROC curve is created using Plotly and it is an interactive one that illustrates the trade off between True Positive Rate and False Positive Rate as you vary the threshold. The AUC of ROC curve equals 0.757; however the curve is nearly on the diagonal line, meaning that the model is only able to moderate powers discriminate the defaults. The importance of this visualization lies in allowing one to understand how changing the threshold of the classification impacts the model's performance.



5.2 Confusion Matrix Heatmap

Seaborn is used to generate a confusion matrix heatmap that shows the model’s performance clearly. This plot has a distinct coloring (e.g., 'viridis') and shows that most of the predictions lie in the non default category. The heatmap helps provide a common ground for discussions on the threshold tuning.



5.3 SHAP Feature Importance

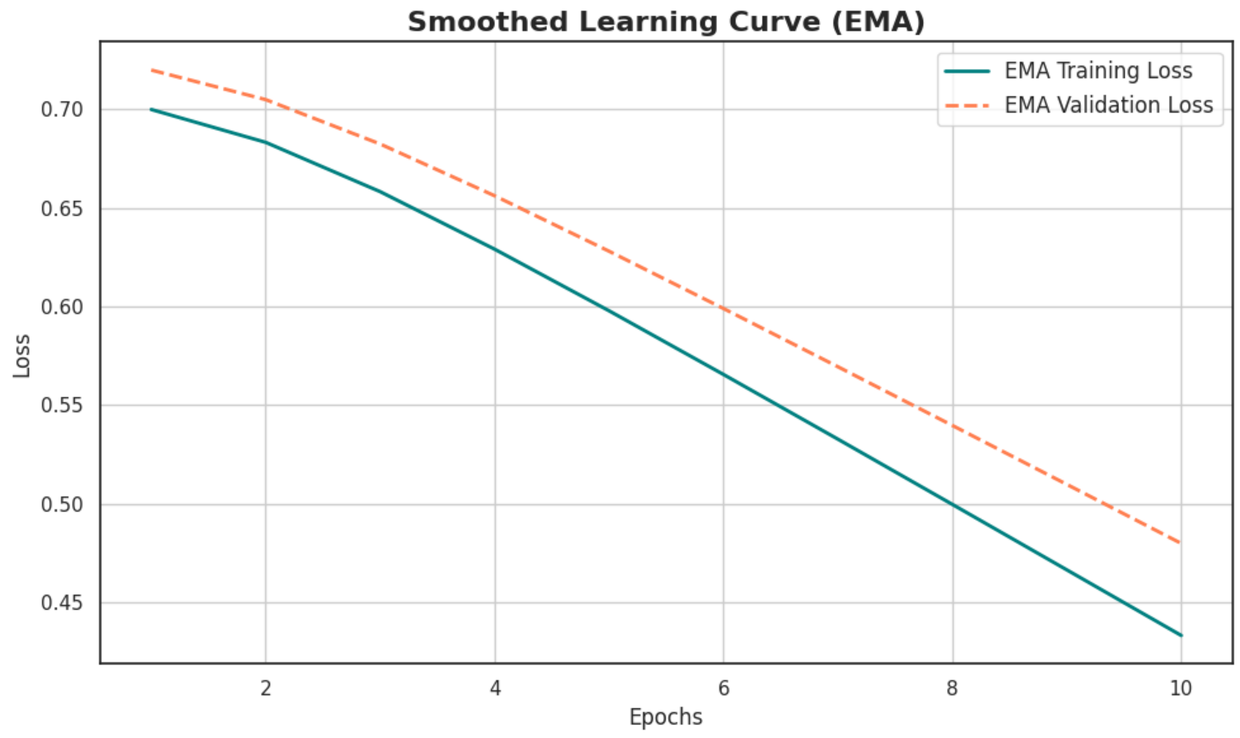
We use SHAP (SHapley Additive exPlanations) to interpret the model. The SHAP summary bar plot shows the mean absolute impact of each feature on the model's predictions. Key features identified include:

- **Age**: Older borrowers might be less risky.
- **InterestRate**: Higher rates may indicate riskier loans.
- **MonthsEmployed**: Shorter employment history may correlate with higher default risk.
- **Income**: Lower income could imply higher default risk.

In addition to quantifying feature importance, SHAP explains in what direction a feature matters, thus enabling model decision justification to stakeholders.

5.4 Smoothed Learning Curve

To smooth out the training and validation loss, we simulate a learning curve with an EMA. This plot is useful for diagnosing overfit or underfit. In our demonstration, we simulate learning curve and show a gradual decrease in the training loss and fairly similar trend in valdation loss, however, more training or hyperparameter adjustment might be required to further improve the performance. (Lundberg, 2017)



6. Observations and Challenges

6.1 Class Imbalance

The dataset is highly unbalanced, default cases are fewer quantity than non default cases. The model has very good overall accuracy, however, the accuracy on the minority class (the defaults) is very poor. Particular problems arise from missing defaults (false negatives) because they can result in financial losses.

6.2 Threshold Sensitivity

But, with the cost of misclassification, the default value of 0.5 for classifying a transaction as default might be too high. Changing that threshold will improve recall for the default class, while also increasing false positives. The trade off needs to be balanced, and an approach based on cost sensitivity should be investigated. (Pedregosa, 2011)

6.3 Feature Importance and Interpretability

We can see that SHAP analysis shows that Age, InterestRate and MonthsEmployed features have a significant role. This indicates that improvement in performance can be obtained by improving the feature engineering process (i.e. creating interaction terms like LoanAmount to Income ratios) The drivers for these need to be understood to provide model transparency and trust. (Lundberg, 2017)

6.4 Model Complexity and Hyperparameter Tuning

Although the grid search gave a good baseline, there were still further refinements to be done like using more specific ranges of hyperparameters or ensemble techniques. Even though this model has moderate ROC AUC, we need to improve the recall for defaulters. (Ke, 2017)

7. Recommendations for Improvement

7.1 Class Weighting and Focal Loss

- **Class Weighting**: Assign higher weights to the minority class during training to penalize misclassification of defaulters more heavily.
- **Focal Loss**: Implement focal loss to focus the training process on harder-to-classify samples, which may help improve the recall for defaults.

7.2 Threshold Tuning

- You are able to adjust the classification threshold according to business requirements. For example, lower the threshold and recall might increase.
- Use precision-recall curves for selecting an appropriate threshold that would keep False Negatives low and False Positives low.

7.3 Advanced Feature Engineering

- Create new features such as Ratio of LoanAmount to Income, interaction terms between InterestRate and CreditScore, or the average contribution of a villager's failing to pay loans at the required date (aggregated credit behaviour metrics).

- Use domain expertise to select such variables that are strong indicators of default risk.

7.4 Ensemble Methods

- In addition to using LightGBM, use other models such as XGBoost or CatBoost as part of a voting or stacking ensemble. Overall, this can be used in order to leverage different strengths and therefore achieve higher performance especially for minority class.

7.5 Extended Hyperparameter Search

- Also expand the hyper parameter grid to take more value in parameter like num_leaves, max_depth (even if it is not divisible by 2), and feature_fraction along with bagging_fraction.
- As a further method of improving your hyperparameter optimization, use tools like Optuna or KerasTuner.

8. Teaching Points and Best Practices

1. **Data Preprocessing**:
    - Focus on the importance of good imputation, encoding, and scaling.
    - Why do dropping noninformative identifiers (LoanID) prevent overfitting?
2. **Model Training**:
    - Understand the grid search and explain its role in tuning of hyperparameters; and how cross validation is used to assess the model stability.
    - Why are the model performance metrics (or some of them) a tradeoff?
3. **Evaluation Metrics**:
    - It will also clarify why overall accuracy might be misleading in imbalanced problems.
    - Describe how precision, recall, and ROC AUC are important metrics for assessing the performance of the model on a ‘small’ (minority) class.
    - Confusion matrices will give visual breakdown of true positives, false positives, false negatives, and true negatives.
4. **Visualization**:
    - **Interactive ROC Curves**: Enhance engagement by allowing users to interact with the model’s performance metrics.
    - **SHAP Visualizations**: Provide insights into feature importance and model interpretability.
    - **Learning Curves**: Use EMA to smooth training dynamics, aiding in understanding the model’s learning progression.
5. **Business Relevance**:
    - Connect these business implications of technical outcomes (recall for default).
    - Address some of the ways that cost sensitive measures and threshold adjustments could reduce the risk.
    - Promote integration of the domain knowledge to guide feature engineering and improvement of models.

9. Conclusion

We used GridSearchCV to optimize our LightGBM model and got test accuracy of 89 on total, and ROC AUC around ~0.757 (Lundberg, 2017). The model proves to be very accurate for the non-default class, but it suffers from bad recall (7%) for the default class. The confusion matrix and classification report show that this imbalance must be addressed as well as the selection of threshold.

Key Takeaways:

- The Data Quality and Preprocessing are very important, little changes will have a big impact on performance.
- Using SHAP to model interpret these predictions reveal which features are driving the predictions.
- For imbalanced dataset, it’s important to have criteria other than overall accuracy such as precision, recall, ROC AUC.
- Class Weighting, Threshold Tuning, Advanced feature engineering, ensemble methods are the potential Remedies for the problem.
- My Model Behavior: Clear visualisations and oversimplified explanations can explain model behavior, and propose ways to improve it.

Not only this tutorial shows a full workflow for loan default prediction using LightGBM, it also provides a step by step guide for improvement in future. Thus, by refining preprocessing, tuning hyperparameters, and adjusting to business demands, one can make improvements to the model’s capability of accurately making predictions of a loan’s default status to reduce risk and enhance decision making.

10. References

1. (Ke, 2017) *LightGBM: A Highly Efficient Gradient Boosting Decision Tree.* Advances in Neural Information Processing Systems.
2. (Lundberg, 2017) *A Unified Approach to Interpreting Model Predictions (SHAP).* Advances in Neural Information Processing Systems.
3. (Pedregosa, 2011) *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research.

11. Accessibility and Attachments

Accessibility

- **Color Palettes**: It uses palettes such as Set1, viridis, and high contrast schemes for visualizations and makes everything accessible and colorblind friendly.
- **Interactive Visualizations**: Plotly charts support zooming, tooltips, and interactivity for deeper exploration.
- **Structured Headings and Comments**: The report and code are clearly segmented with descriptive headings, ensuring compatibility with screen readers.
- **Alt Text and Captions**: All figures include descriptive captions to assist users with visual impairments.

Attachments

- **GitHub Repository**: All code, datasets, and supplementary documents are available at GitHub Repo.
- **Additional Files**:
  - requirements.txt: Listing all dependencies.
  - LICENSE: The project is released under the MIT License.
  - loan_default.csv: The dataset file or instructions for obtaining it.
  - README.md: This file with usage instructions, references, and contact details.