Final Report: Predicting Yearly Income Levels Using Machine Learning

Hosoo Lee
https://github.com/Michael053000/ML2024F.git

December 13, 2024

1 Problem Definition and Motivation

Introduction
The primary objective of this project is to predict whether a resident's yearly income exceeds $50,000 based on attributes derived from the 1994 Census database. This binary classification problem is not only a standard benchmark in the machine learning community but also holds significant real-world implications in domains such as policy making, social research, and targeted marketing.

Importance of the Problem
Understanding income levels is critical for government agencies and businesses to allocate resources effectively, implement tailored interventions, and design policies that address socio-economic disparities. Income prediction can also support financial institutions in credit scoring, loan approvals, and risk assessments. As such, developing an accurate predictive model for income classification provides a valuable tool for decision-making processes across sectors.

Motivation for Using Machine Learning
The complexity of the dataset, which includes both categorical and numerical features, coupled with the presence of missing data and nonlinear relationships, makes traditional statistical methods inadequate for achieving high accuracy. Machine learning (ML) techniques offer the flexibility to handle these challenges by leveraging powerful algorithms that can capture complex patterns in the data. Moreover, the availability of scalable ML libraries and efficient computational resources makes this approach highly feasible.

Furthermore, the availability of open-source tools such as Python libraries (e.g., Scikit-learn, XGBoost, and LightGBM) has democratized the use of advanced machine learning techniques, making it easier to experiment, optimize, and deploy models with minimal overhead. The adaptability of machine learning approaches allows for handling imbalanced datasets, intricate dependencies between features, and the ability to generalize across unseen data, making it the most appropriate choice for this classification task.

2 Solution

2.1 Overview of Models Used

To address this problem, we employed multiple machine learning algorithms, including:

- **Random Forest Classifier**: A robust ensemble learning method based on decision trees, well-suited for handling heterogeneous data types and missing values. It provides insights into feature importance and is relatively easy to interpret.
- **XGBoost Classifier**: A gradient boosting algorithm known for its efficiency and superior performance in structured data tasks. XGBoost incorporates regularization techniques to prevent overfitting and provides highly optimized parallel tree boosting.
- **LightGBM Classifier**: A highly optimized gradient boosting framework designed for speed and accuracy, especially with large datasets. It uses a histogram-based algorithm for faster computation and better memory usage.

2.2 Data Preprocessing

The dataset comprises 14 attributes, including both numerical and categorical variables. Key preprocessing steps included:

- **Handling Missing Values**: Missing numerical features were imputed using the mean, while missing categorical features were imputed with the most frequent value in the respective column.
- **Encoding Categorical Features**: One-hot encoding was applied to categorical variables to ensure compatibility with machine learning models. This step was essential for gradient boosting models to process categorical data effectively.
- **Scaling and Normalization**: While scaling is less critical for tree-based methods, it was applied to ensure consistency across numerical attributes, especially when combining predictions from multiple models.

2.3 Model Selection and Evaluation

Initially, individual models were trained and evaluated to assess their baseline performance:

- **Random Forest Classifier**: Provided robust predictions with an AUC-ROC of 0.910.
- **XGBoost Classifier**: Achieved the highest public score (0.92719) among the individual models, demonstrating its effectiveness in handling structured data.
- **LightGBM Classifier**: Tuned using Bayesian optimization, LightGBM achieved a private score of 0.92731, slightly outperforming XGBoost in the private evaluation.

Although an ensemble approach was considered, it was not submitted for evaluation due to resource constraints and focus on individual model optimization. Instead, weighted averages of predictions were tested offline, indicating potential for further improvement.

2.4 Hyperparameter Optimization

Hyperparameters for each model were tuned using the following approaches:

- **Grid Search**: Conducted for Random Forest to optimize the number of estimators, maximum depth, and minimum samples required to split a node.

- **Bayesian Optimization**: Used for LightGBM to efficiently explore hyperparameter spaces with fewer iterations, focusing on learning rate, maximum depth, and number of leaves.
- **Randomized Search**: Performed for XGBoost to identify optimal hyperparameters such as learning rate, subsample ratio, and maximum depth.

Each tuning process involved five-fold cross-validation to ensure robust evaluation and avoid overfitting during hyperparameter selection.

3 Experimental Results

3.1 Evaluation Metrics

The performance of the models was evaluated using the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). This metric was chosen for its robustness in assessing the trade-off between true positive and false positive rates across thresholds. Additionally, AUC-ROC provides a comprehensive measure of model performance without relying on a specific classification threshold, making it ideal for this binary classification task.

3.2 Results Summary

- **Random Forest**: Public AUC-ROC = 0.91579, Private AUC-ROC = 0.91711
- **XGBoost**: Public AUC-ROC = 0.92719, Private AUC-ROC = 0.92800
- **LightGBM**: Public AUC-ROC = 0.92639, Private AUC-ROC = 0.92731

3.3 Key Observations

- **Incremental Improvements**: The progression from Random Forest to XGBoost and LightGBM demonstrates the value of using gradient boosting methods for structured data.
- **Importance of Hyperparameter Tuning**: Bayesian optimization for LightGBM and randomized search for XGBoost provided critical performance boosts, leading to near-optimal models.
- **Potential for Ensembles**: While ensemble learning was explored offline, the highest submitted scores were obtained through individual model optimization, indicating that further ensemble tuning could yield improvements.

4 Future Work

4.1 Extended Feature Engineering

Given more time, additional feature engineering steps could be implemented, such as:

- Creating interaction terms between significant features (e.g., age and education level).
- Clustering related features to reduce dimensionality and noise.
- Incorporating domain-specific transformations, such as categorizing continuous variables like age into meaningful age groups.

4.2 Advanced Model Techniques

- **Stacking Ensembles**: Implementing a meta-model to combine predictions from LightGBM, XGBoost, and Random Forest could provide additional accuracy improvements.
- **Neural Networks**: Exploring deep learning architectures for automatic feature extraction and capturing non-linear relationships.

4.3 Real-Time Deployment

Integrating the model into real-time applications for dynamic predictions, such as in financial services or targeted marketing, would extend its practical utility.

5 Conclusion

This project successfully applied machine learning techniques to predict yearly income levels with high accuracy. Individual models, particularly LightGBM and XGBoost, demonstrated strong performance, achieving public and private AUC-ROC scores above 0.926. Through hyperparameter tuning and careful model evaluation, the project highlights the power of machine learning in addressing socio-economic prediction challenges. Future efforts could focus on ensemble methods and real-world deployments to maximize impact.