CS 6350 Machine Learning
Final Project Midpoint Report
Milena Belianovich
10/26/2023

Midpoint Project Report

Introduction

This project is a part of a Kaggle competition which aims to leverage machine learning for the social good of better understanding income distributions across different demographics. The primary goal is to predict whether a person makes over 50K a year based on census data, allowing for better-targeted social programs and policies.

1. Data Collection and Exploration

The dataset, extracted by Barry Becker from the 1994 Census database, provides a rich set of attributes. These include continuous, categorical, and integer types, with some features having missing values recorded as question marks. The dataset comprises 14 regular attributes and 1 target attribute only present in the training dataset.

During the exploration phase, various statistical and visualization techniques were employed:

- **Descriptive Statistics**: Provided insights into the data's central tendency, dispersion, and shape.
- **Data Visualization**: Enabled to identify patterns, trends, and outliers that could significantly impact the predictive model.

2. Data Preprocessing

To prepare the dataset for the modeling stage, the following systematic approach was planned:

- **Handling Missing Values**: Addressing the instances of missing values, represented by question marks, through imputation.
- **Feature Scaling**: Making sure that numerical features contribute equally to the model's performance by bringing them onto the same scale.
- **One-Hot Encoding**: Transforming categorical variables into a machine-readable format, allowing for a richer representation of data.
- **Train-Validation Split**: Splitting the dataset into a training set for model learning and a validation set for performance evaluation.

Given the dataset, our preprocessing steps involved the following:

- **Handling Missing Values:**
    - Numerical Columns: Missing values in columns like 'age', 'fnlwgt', 'education.num', 'capital.gain', 'capital.loss', and 'hours.per.week' were imputed using the mean strategy.
    - Categorical Columns: Missing values in columns like 'workclass', 'education', 'marital.status', 'occupation', 'relationship', 'race', 'sex', and 'native.country' were imputed with the most frequent values.
    - A unique step was taken to include the missing category 'Holand-Netherlands' in the 'native.country' column by creating a new row with this category. Numerical columns in this row were filled with mean values, and categorical columns were filled with the most frequent values.
- **Encoding:**
    - The 'income' column was converted to a categorical type with specific class names '>50K' and '<=50K'.

    - Categorical variables were then encoded using one-hot encoding to convert them into a machine-readable format. This encoding transformed the categorical data into binary columns, each representing a category in the original data.

- **Data Separation:**
    - The dataset was split into features (X) and target (y). The target column 'income>50K' was converted to numerical codes.

3. Model Selection and Training

Given the nature of the predictive task, RandomForest and XGBoost were picked as the algorithms of choice:

- **RandomForest:**
    - RandomForest is an ensemble learning method that works by constructing multiple decision trees at training time and outputting the mode of the classes for classification. It is robust against overfitting as it averages the result to improve the predictive performance and control over-fitting.
    - The RandomForest model was trained with 100 estimators.
- **XGBoost:**
    - XGBoost, or Extreme Gradient Boosting, is a scalable and accurate implementation of gradient boosting machines. It has proven to be a highly effective machine learning algorithm, frequently outperforming other algorithms in machine learning competitions.
    - The XGBoost model was trained with 1000 estimators and a learning rate of 0.05.

Both models were initially trained using default settings to establish baseline performance.

Evolving of the algorithms:

- RandomForest and XGBoost with **binary results** for predictions (0 and 1):
    - Initially both algorithms were producing binary numbers as results due to using *predict* in the functions. However, even though the average AUC values were high and close to 1, further improvements were possible:
        - Average Random Forest AUC: 0.7761040627963782
        - Average XGBoost AUC: 0.7984041043369097
    - After further exploring predictions in *sklearn,* the goal of improving results could be reached.
- RandomForest and XGBoost with **float results** for predictions (0 to 1):
    - Float numbers were reached by using *predict_proba* instead of *predict* for the prediction models. These changes significantly improved the average AUC:
        - Average Random Forest AUC: 0.902424181861943
        - Average XGBoost AUC: 0.9282713212248513
    - Potential improvements to these numbers are described further in the future plans section.

Plans for the Rest of the Project

The roadmap for the project's next stages is as follows:

- **Hyperparameter Tuning and Model Evaluation**
    - A robust method of GridSearch was intended for the project, although it is not fully implemented in the code for now. The purpose of GridSearch is to systematically work through multiple combinations of parameter tunes, cross-validating as it goes to determine which tune gives the best performance. The models' performance is evaluated based on the Area Under the ROC (AUC) curve with the aim to maximize this value for optimal performance.
    - Hyperparameter tuning will be implemented through GridSearch:
        - **RandomForest**: A grid comprising **n_estimators** and **max_depth** will be utilized to fine-tune the model.
        - **XGBoost**: A comprehensive grid including **n_estimators**, **learning_rate**, **max_depth**, **min_child_weight**, and **subsample** will be employed.
    - The evaluation of models is primarily based on the Area Under the ROC (AUC) curve, with the aim to maximize this value for optimal performance. The AUC provides a comprehensive measure of a model's performance across all possible thresholds, considering both True Positive Rate (TPR) and False Positive Rate (FPR). However, this method is still not fully complete, so finishing its development is a future plan.
- **Cross-Validation:** Use cross-validation to assess the performance of the models more accurately. This can help identify if the model is overfitting or underfitting and make adjustments accordingly.
- **Model Selection:** Try different algorithms and see if one that performs better on the dataset can be found. For example, trying LightGBM, CatBoost, or a deep learning approach.
- **Feature Selection:** Investigate the importance of each feature in the model and consider dropping features that are not contributing much to the model's performance.

Conclusion

In conclusion, it is visible that XGBoost has a higher accuracy of prediction due to the higher average of AUC. However, further improvements can be potentially integrated to increase the accuracy of the predicted values. Nevertheless, following the plan for the future tuning should lead to a satisfactory outcome.

Overall, this project represents a significant step towards effectively employing machine learning in public policy and social programs. By meticulously following the steps of data preprocessing, model selection, and evaluation, it is possible to deliver a robust model that offers valuable insights for more targeted social interventions.