CS 6350 Machine Learning
Final Project Final Report
Milena Belianovich
12/17/2023

Final Project Report

Github Repository
https://github.com/milenabel/CS6350-ML2023/tree/main/Kaggle

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 of this project is to predict whether a person makes over 50K a year based on census data, allowing for better-targeted social programs and policies.

Problem Definition and Motivation

The chosen problem is to predict whether an individual's income exceeds $50,000 per year based on 1994 U.S. Census data. This challenge is not only a valuable exercise in applied machine learning but also holds substantial socio-economic importance. Accurate predictions can inform and influence public policies, particularly in the realms of social welfare and economic planning. Understanding income distributions aids in the development of targeted social programs, thereby potentially reducing inequality and enhancing societal well-being.

The decision to utilize machine learning techniques for this problem is driven by the nature of the dataset and the complexity of the task. Machine learning's capacity to analyze large datasets and uncover non-linear relationships makes it exceptionally suited for this purpose. Traditional statistical methods might struggle with the volume and diversity of data, whereas machine learning algorithms can efficiently process and extract meaningful patterns from the data. This efficiency is critical in addressing the multifaceted nature of income prediction, which is influenced by a multitude of demographic and socio-economic factors. Thus, machine learning not only offers a methodological advantage but also enhances the potential impact of the analysis.

Problem Solution

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 produced 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
  - Further improvements to these numbers are described in model development section.

4. Model Development

The project advanced from using RandomForest and XGBoost with basic configurations to enhanced models through hyperparameter tuning and cross-validation, significantly refining our predictive capabilities.

RandomForest and XGBoost - Enhanced Approach:

- **Hyperparameter Tuning**: extensive hyperparameter tuning was conducted for both RandomForest and XGBoost models using GridSearch. For RandomForest, the parameters such as n_estimators, max_depth, min_samples_split, and min_samples_leaf were optimized. In the case of XGBoost, parameters like n_estimators, learning_rate, max_depth, min_child_weight, and subsample were fine-tuned. This systematic exploration through various parameter combinations is aimed to optimize the models' performance.
- **Cross-Validation Integration**: to ensure the models' robustness and generalizability as well as prevent overfitting, cross-validation was integrated into the GridSearch process. This approach provided a more rigorous and more comprehensive assessment of the models' performance, reducing the risk of overfitting and ensuring the reliability of the results.

- The models were evaluated based on the Area Under the Receiver Operating Characteristic (ROC) curve (AUC), which offers an insightful measure of performance by considering both True Positive Rate (TPR) and False Positive Rate (FPR). These changes **slightly improved** the average AUC:
    - Average Random Forest AUC: 0.9183640913203484
    - Average XGBoost AUC: 0.9297907285514591

Experimental Results

RandomForest: The RandomForest model, initially showing an AUC of 0.776, witnessed a substantial improvement in performance after applying hyperparameter tuning and cross-validation, ensuring more reliable and consistent predictions. Here are the steps of improvement and their respective results:

- Binary results for predictions (0 and 1):
    - Average Random Forest AUC: 0.7761040627963782
- Float results for predictions (0 to 1):
    - Average Random Forest AUC: 0.902424181861943
- GridSearch using float results for predictions (0 to 1):
    - Average Random Forest AUC: 0.9183640913203484

XGBoost: Similarly, the XGBoost model, starting with an AUC of 0.798, also demonstrated significant gains in predictive accuracy post-enhancements. The fine-tuning of parameters and validation approach contributed to a more robust model capable of capturing complex patterns in the data. Here are the steps of improvement and their respective results:

- Binary results for predictions (0 and 1):
    - Average XGBoost AUC: 0.7984041043369097
- Float results for predictions (0 to 1):
    - Average XGBoost AUC: 0.9282713212248513
- GridSearch using float results for predictions (0 to 1):
    - Average XGBoost AUC: 0.9297907285514591

Plans for Potential Project Continuation

- **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.

Conclusion

In conclusion, it is visible that XGBoost has a higher accuracy of prediction due to the higher average of AUC. The implementation of GridSearchCV and Cross-Validation led to a noticeable improvement in the models' AUC scores. The refined models not only elevated the models' predictive performance and demonstrated enhanced predictive capabilities, underscoring the effectiveness of these advanced machine learning techniques, but also provided insights into their behavior across different scenarios, paving the way for more accurate and reliable predictions in income prediction tasks.

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, model development, and evaluation, it is possible to deliver a robust model that offers valuable insights for more targeted social interventions.