Supervised Learning Reveals If Your Salary Exceeds $50,000 or Not

Chas Hamel
CS 5033 Spring 2024

1. Introduction

To demonstrate the implementation of Supervised Learning (SL), multiple Linear Regressions Models using the Adult Census Income Dataset **(UCI Machine Learning Repository, 1996)** were developed. After the models were developed, each are optimized using hyperparameter tuning and compared against each other to determine which is best at predicting if an individuals salary exceeds $50,000 or not given various socioeconomic inputs.

The Adult Dataset
The Adult Census Income Dataset contains many categorical and numerical features that describe socioeconomic traits of $n = 48,842$ adults such as age, education level, work class, marital status, occupation, etc. The target variable is income, which is a binary value of $\leq \$50k$ or $<\$50k$.

2. Data Preparation & Exploration

Prior to splitting the dataset to train, test and validate each model, processing occurred to ensure that all categorical variables are converted to numerical values using one-hot encoding. Additionally, as shown in **Figure 1**, there are multiple missing variables within the dataset that need to be handled. As each row of the dataset is a unique input, the removal of these will not impact the remaining data other than reducing the size of the dataset. For this, each variable with missing data makes up less than 6% of the total dataset.

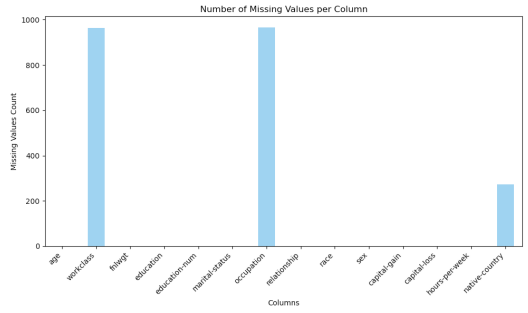

Figure 1. Adult Census Income Dataset Missing Values

Continuing with data exploration, the most frequent categorical values for the hypothetical most important variables which dominate the dataset based on percentage, as shown in **Table 1** where the Variable, Value, and percentage of dataset can be seen, potentially giving bias to the outcome of the models.

Variable	Entry	% Data
Work Class	Private	73.89
Relationship	Husband	41.32
Race	White	85.98
Country	United States	91.18

Table 1. Dominating Categorical Entries

After preparing and splitting the data into a test and training set, a Decision Tree, K Nearest Neighbor (KNN), and Random Forest algorithms are implemented.

3. Model Implementation

After exploring the data and preparing the dataset, implementation of Decision Trees, K-Nearest Neighbors, and Random Forest occurred.

For implementation, the data was randomly split into a training and testing subset, with 80% of the data in the training and 20% in the test sets.

The three model types are all implemented without hyperparameter tuning and using standard modeling from the sklearn python package. After training each model, initial performance via the sklearn.metrics.accuracy_score package is used to determine initial model accuracy. **Table 2** below shows the model accuracy for each.

Model	Accuracy
Decision Tree	81.19
K-Nearest Neighbor	76.08
Random Forest	85.51

Table 2. Initial Model Accuracy

After initial modeling, hyperparameter tuning of each model occurred.

Using Grid Search with a Cross Validation of 5 folds on the Decision Tree model, a max depth and min samples split was established and the optimal parameters of max depth

equal to 9 and minimum samples required to split a node is set to 10.

Similarly for K-Nearest Neighbors, a Grid Search with Cross Validation using 5 folds was performed to establish the optimal nearest neighbor value. From this, it is determined that nearest number of neighbors to be considered during prediction is 11.

Finally, using a similar Grid Search with Cross Validation with 5 folds on the Random Forest model, a max depth is not set thus each tree is filled until each tree contains only one class of samples. The minimum number of samples per leave is set to 2, with the minimum samples required to split a node is set to 6. Lastly, the number of trees, or n estimators, is set to 100.

Table 3 below shows the accuracy comparison as well as the percentage performance increase versus an untuned model, for these three models using the tuned hyperparameters.

Model	Accuracy	% Increase
Decision Tree	85.53	5.21
K-Nearest Neighbor	78.34	2.88
Random Forests	86.15	0.74

Table 3. Post Tuning Model Accuracy

4. Results

When comparing the Model Accuracy previously depicted in **Table 3**, both the Decision Tree and Random Forest models outperform the K-Nearest Neighbor model, with each model seeing an increase in overall accuracy after hyperparameter tuning compared to the initial model implementation. The under performance of the K-Nearest Neighbor model compared to the other models could be a result of the models sensitivity to the Curse of Dimensionality, or, "as the number of features in [the] data increases, the effectiveness of KNN drops."[2]

Stepping beyond the Accuracy measurements, a Confusion Matrix and Classification Report are generated to aid in understanding the overall precision and recall ability of the models, as calculated by the True Positive, True Negative, False Positive, and False Negative Values from the Confusion Matrix. **Table 4** below shows the Weighted Average of the precision, recall, and f-1 score for each of the three models. As expected, similarly to the accuracy scores, the Random Forest model performs slightly better overall in comparison to Decision Trees or K-Nearest Neighbors.

Using the Confusion Table and Classification Report, Receiver Operating Characteristic (ROC) curves, **Figure 2**, are generated which depict the Random Forest and Decision Tree models to have an overall high performing classifier with an Area Under the Curve (AUC) near 1.0 while the K-Nearest Neighbor has a classifier which is close to the random predictor with an AUC of 0.66.

Model	Precision	Recall	F-1 Score
Random Forest	0.86	0.86	0.86
K-Nearest Neighbor	0.77	0.78	0.74
Decision Tree	0.85	0.86	0.85

Table 4. Classification Report - Weighted Avg

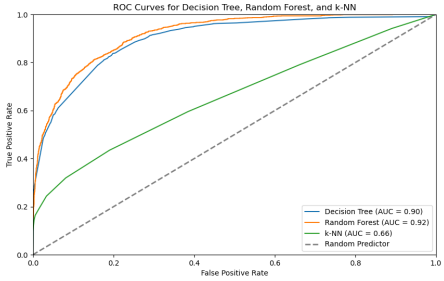

Figure 2. ROC Curve

Additional to the overall accuracy of the models, the computational time of each algorithm varies significantly. As depicted in **Table 5**, although the Random Forest model performed with slightly more accuracy that its Decision Tree counterpart (+4.44%), the Random Forest model takes 98.9% longer (432.7 seconds) than the Decision Tree model and 88.1% (385.2 seconds) longer than the K-Nearest Neighbor model.

Model	Comp. Time (sec)	Delta (%)
Random Forest	437.2	–
K-Nearest Neighbor	52.0	-88.1
Decision Tree	4.5	-98.9

Table 5. Computational Time per Model Type

Given the slight increase in performance of the Random Forest model compared to the other two models, 0.62% compared to Decision Tree and 7.81% compared to K-Nearest Neighbors, the significant increase in computational time shall outweigh the accuracy improvement especially for large datasets.

Moving beyond the performance of the models, Feature Importance from the dataset within the Decision Tree and Random Forest models was evaluated. As shown in **Figure 3** and **Figure 4**, the top 10 dataset features driving predictions of if someone's income is greater than $50k or not can be explored.

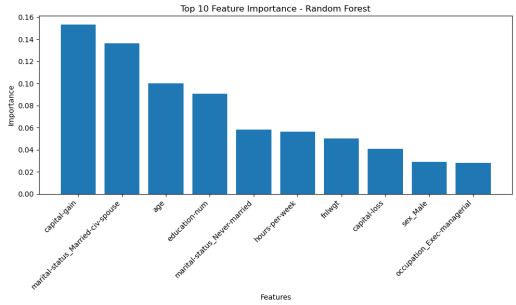

Figure 3. Feature Importance - Random Forest

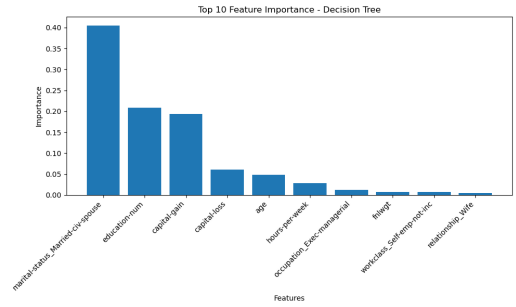

Figure 4. Feature Importance - Decision Tree

Focusing solely on importance greater than 10%, both models identify a persons Marital Status of "Married / Civil Spouse" and Capital Gains value of being greater than 0 as significant drivers of if a person makes greater than $50k or not. Additionally, the Random Forest model identifies greater than 10% importance to a persons age having an impact while the Decision Tree model identifies a persons educational achievements. It shall be noted though, that although these features are identified as strongly important for the specific model, both identify these features as well but with less than 10% overall importance.

5. Summary

In summary, the Decision Tree and Random Forest models outperform the K-Nearest Neighbor model when using the Adult Census Income dataset from the UCI Machine Learning Repository. A relative significant increase in the area under the ROC curve for the Decision Tree and Random Forest models compared to the K-Nearest Neighbor show the models classifier performance in comparison to a random predictor. Between the Decision Tree and Random Forest models, performance was similar enough that a determining factor on which model to be used comes down to computational time, where the Decision Tree model truely shines.

Specific to the dataset, these high performing models identify similar important features which drive if a person makes more than $50k a year or not where if a person is married, or in a civil relationship, as well as has capital assets, there is a high change that the annual income exceeds this threshold.

6. Literature Review

A Statistical Approach to Adult Census Income Level Prediction In the publishing, A Statistical Approach to Adult Census Income Level Prediction [3] Chakrabarty and Biswas discuss the use of machine learning to predict income level using the same Adult Income Census dataset as described in this paper. Many of their approaches were similar in morphing the dataset to prepare for model training, but, the took a more statistical approach to understanding the dataset than I did. Additionally, in modelling, Chakrabarty and Biswas use a Gradient Boosting Classifier model which achieved a 88.73% accuracy rating, which can be considered similar to the 86.15% accuracy rating achieved by the Random Forest model in this project.

Centre for Data Ethics and Innovation - Adult Dataset The Centre for Data Ethics and Innovation in the United Kingdom uses the Adult Census Income dataset to investigate the potential for prediction bias in Machine Learning models. In their report, the Centre takes the view point of being a Finance Company looking to approve credit for high income earners. Similarly to what I describe in section 2 of this report, the possibility of income bias in this dataset is very high as income potential is, in some cases, proportional to hours worked, sex, and race. To combat this potential bias, the Centre for Data Ethics and Innovation recommends to ensure data is collected and representative to the demographic being studied, as well as ensuring correct representation of all classes between sex, race, etc.

Retiring Adult: New Datasets for Fair Machine Learning In the paper "Retiring Adult: New Datasets for Fair Machine Learning" by Ding et al., the authors argue that the UCI Adult dataset may present several limitations when it comes to fairness and its representation of the population. The dataset is derived from census data collected in the 1990s, which might not reflect societal and demographic trends. This creates risks of biased outcomes when building machine learning models, as the dataset may not account for changes in population dynamics, labor markets, or societal structures. Ding's new dataset addresses these issues by offering a more current and diverse representation of the population. These improvements provide an improved foundation for training machine learning models, reducing the likelihood of bias based on outdated or skewed data. By introducing a dataset that aligns more closely with today's population, the authors aim to promote fairer outcomes in machine learning applications and contribute to the development of more robust, equitable models.

7. References

1. Becker,Barry and Kohavi,Ronny. (1996). Adult. UCI Machine Learning Repository. https://doi.org/10.24432/C5XW20.

2. https://www.geeksforgeeks.org/knn-vs-decision-tree-in-machine-learning/

3. N. Chakrabarty and S. Biswas, "A Statistical Approach to Adult Census Income Level Prediction," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2018, pp. 207-212, doi: 10.1109/ICACCCN.2018.8748528. keywords: Training;Boosting;Decision trees;Predictive models;Logistics;Vegetation;machine learning;data mining;income equality;Classification;Gradient Boosting Classifier,

4. "Finance - the Adult Dataset." Machine Learning Bias Mitigation, cdeiuk.github.io/bias-mitigation/finance. Accessed 6 May 2024.

5. Ding, F., Hardt, M., Miller, J., Schmidt, L. (2021). Retiring Adult: New Datasets for Fair Machine Learning. In A. Beygelzimer, Y. Dauphin, P. Liang, J. Wortman Vaughan (Eds.), Advances in Neural Information Processing Systems. OpenReview. https://openreview.net/forum?id=bYi_2708mKK.