COMP 6721-Project Progress Report: Adult Census Income Analysis

Dina Omidvar
Concordia University
40239883

Niloofar Tavakolian
Concordia University
40220767

Nastaran Naseri
Concordia University
40215694

SeyedehMojdeh Haghighat Hosseini
Concordia University
40171693

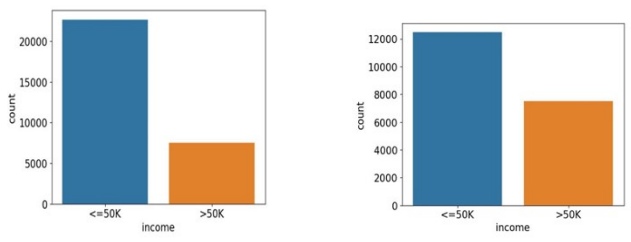
1 INTRODUCTION AND PROBLEM STATEMENT

The purpose of this progress report is to provide an update on the ongoing project focused on the "Adult Census Income" dataset [1]. This dataset encompasses various socio-economic attributes of individuals and holds significant potential for diverse applications. One of these applications includes predicting income levels based on features such as age, education, occupation, and more. The ability to accurately predict income levels can provide valuable insights for targeted marketing, policy planning, and resource allocation.

In light of the dataset's characteristics, we have decided to employ a decision tree classifier as our model of choice for predicting income levels. The decision tree algorithm offers several advantages that make it suitable for our task. Firstly, decision trees can handle both numerical and categorical features, which is crucial given the diverse nature of our dataset. Additionally, decision trees provide interpretability, allowing us to understand the underlying decision-making process and gain insights into the factors influencing income levels.[2]

However, we encountered challenges during the pre-processing phase of the project. The most significant among these challenges was handling missing values in certain features such as workclass and occupation. To ensure data integrity, we addressed these missing values by removing the respective samples from the dataset.[3]

Another notable challenge in this project was the presence of a class imbalance in the target variable. The dataset contained a larger number of samples where income was less than or equal to $50,000 compared to those where income exceeded $50,000. To mitigate this imbalance, we performed downsampling of the majority class, resulting in a more balanced dataset that would facilitate more accurate predictions.[4] Since our dataset consisted of a combination of numerical and categorical data, another challenge we encountered was determining how to handle this mixed data during the training of our decision tree classifier. To address this issue, we initially encoded the categorical features using one-hot encoding [6], transforming them into numerical representations that could be effectively utilized by the decision tree algorithm.



(a) Imbalance dataset **(b) Balanced dataset**

Figure 1: Comparison of datasets [5]

Furthermore, to ensure the integrity of our model evaluation, we took extra measures to verify that all categories within the categorical features were adequately represented in each of the training, testing, and validation sets. This ensured that our decision tree classifier could learn and generalize from the entire range of categorical values, avoiding potential biases or performance disparities.

At this stage, the expected outcome of the project is the development of a robust decision tree classifier capable of accurately predicting income levels based on the provided features. Currently, the model demonstrates an accuracy of 0.79025 on the test set and achieves an accuracy of 0.80225 on the validation set.

2 PROPOSED METHODOLOGIES

As mentioned before, the "Adult Census Income" dataset chosen for this project contains a rich collection of socio-economic attributes that provide valuable insights into individuals' characteristics. The dataset includes features such as age, education, marital status, occupation, and more. Each feature plays a crucial role in understanding the factors that contribute to income levels.

In addition to the commonly understood features, it is worth noting the presence of a few variables that may require further clarification. Firstly, the "fnlwgt" feature represents the final weight assigned to each observation in the census data. This weight is applied to ensure that the sample accurately represents the population and is a critical component in producing unbiased estimates.

Furthermore, the features "capital.gain" and "capital.loss" capture the financial gains and losses, respectively, that individuals have experienced. These features provide insights into the economic activities and investments made by individuals, which can significantly influence their overall income levels.

By leveraging this diverse set of features, we aim to build a robust decision tree classifier capable of accurately predicting income levels. The decision tree algorithm offers interpretability, enabling us to understand the underlying decision-making process and identify the key features driving the predictions.

Throughout the project, special attention will be given to handle any data quality issues, such as missing values or outliers, that may affect the performance of the decision tree classifier. Additionally,

To ensure compatibility with the decision tree classifier, the categorical features in the dataset are encoded using one-hot encoding [6] as mentioned before.

The evaluation of the model's performance will be conducted using appropriate metrics such as accuracy, precision, recall, and F1 score. By comparing the model's predictions against the ground truth labels, we will assess its ability to accurately predict income levels based on the provided features.

Table 1: Evaluation of the model's performance [5]

	Precision	Recall	F1-score	Support
≤50k	0.87	0.81	0.84	2516
>50k	0.72	0.80	0.76	1484
Accuracy			0.81	4000
Macro Avg	0.79	0.81	0.80	4000
Weighted Avg	0.81	0.81	0.81	4000

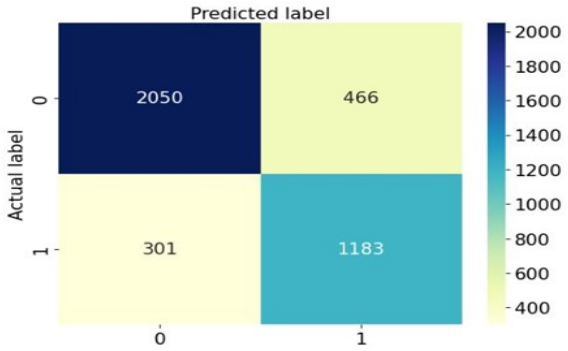


Figure 2: Confusion Matrix [5]

Through these proposed methodologies, we aim to gain valuable insights into the factors influencing income levels and develop a reliable decision tree classifier that can contribute to various applications in marketing, policy planning, and resource allocation.

3 ATTEMPTS AT SOLVING THE PROBLEM

During the training of the decision tree classifier, we encountered a specific challenge related to the handling of categorical features in our dataset. As some categorical feature categories had significantly lower frequencies compared to others within the corresponding columns, there was a risk of certain categories being present only in the test or validation set, while not appearing in the training set. This posed a potential issue as the model would not have been exposed to those categories during the training phase, leading to suboptimal performance on unseen data.

To address this problem and ensure that all categories within the categorical features were adequately represented in each set (i.e., training, testing, and validation), we employed a careful data splitting strategy. We utilized the "stratified" option in the train_test_split function, which ensured that the distribution of categories in the categorical features was maintained across all sets. This approach helped to alleviate the risk of missing categories and ensured that the model had exposure to the entire range of categorical values during training.

By employing this solution, we aimed to enhance the model's ability to generalize and make accurate predictions on unseen data, while also mitigating the potential biases and performance disparities caused by imbalanced category distributions.

Through our initial training of the decision tree classifier, we achieved an accuracy of 0.768 for the test set and 0.78075 for the validation set. However, recognizing the potential for further improvement, we decided to tune the hyperparameters, specifically focusing on the max_depth parameter. By adjusting the max_depth and finding an optimal value, we were able to enhance the performance of the model significantly. With the refined hyperparameters, our decision tree classifier demonstrated improved accuracy, reaching 0.80825 for the test set and 0.8175 for the validation set. This adjustment highlights the importance of hyperparameter tuning in maximizing the model's predictive capabilities and overall performance.[7]

4 FUTURE IMPROVEMENTS

While adjusting the max_depth hyperparameter has proven effective in improving the accuracy of our decision tree classifier, we aim to explore additional techniques to further enhance its performance in the future. One approach we plan to employ is supervised learning classification with a deep learning model. This will involve training a deep neural network to classify data based on labeled examples, allowing us to leverage the power of deep learning for more accurate predictions. In addition to the use of deep learning models, several other techniques can be explored to optimize the performance of our decision tree classifier. These include:

- Feature Engineering: By careful engineering and selecting relevant features, we can provide the decision tree classifier with more informative inputs. In future iterations, we will focus on creating interaction terms, combining related features, and transforming variables to capture complex relationships and improve the model's predictive power.[8]
- Ensemble Methods: Ensemble methods, such as Random Forests or Gradient Boosting, combine multiple decision trees to make predictions. These techniques can help reduce overfitting, enhance generalization, and improve accuracy by leveraging the collective knowledge of multiple models. Incorporating ensemble methods into our classification framework will be a priority in future development.[9]
- Pruning Techniques: Pruning is a technique used to simplify decision trees by removing nodes or branches that do not contribute significantly to the model's performance. By pruning the decision tree based on metrics like information gain or Gini impurity, we can reduce overfitting and improve the model's ability to generalize to unseen data. Exploring and implementing pruning techniques will be crucial to enhance the efficiency and effectiveness of our decision tree classifier.

Furthermore, as we plan to utilize the decision tree classifier within a semi-supervised learning classification framework in the future, finding solutions to further increase its accuracy becomes even more critical. Semi-supervised learning leverages both labeled and unlabeled data to improve classification performance. By incorporating unlabeled data through techniques like self-training or co-training, we can provide additional information and potentially achieve higher accuracy for our decision tree model.

REFERENCES

[1] , "Adult census income," *kaggle*, https://www.kaggle.com/datasets/uciml/adult-census-income?resource=download.

[2] scikit-learn developers, "Decision trees," *scikit-learn documentation*, 2023, https://scikit-learn.org/stable/modules/tree.html.

[3] ——, "Handling missing data," *scikit-learn documentation*, 2023, https://scikit-learn.org/stable/modules/impute.html.

[4] ——, "Resampling methods," *scikit-learn documentation*, 2023, https://scikit-learn.org/stable/modules/classes.html#module-sklearn.utils.

[5] D. Omidvar, N. Tavakolian, N. Naseri, and S. H. Hosseini, "Project github," https://github.com/dinaomidvartehrani/Applied-AI-.git.

[6] J. Brownlee, "Why one-hot encode data in machine learning?" 2020, https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/.

[7] scikit-learn developers, "Hyperparameter tuning," *scikit-learn documentation*, 2023, https://scikit-learn.org/stable/modules/classes.html#module-sklearn.model_selection.

[8] J. Brownlee, "Discover feature engineering, how to engineer features and how to get good at it," 2020, https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/.

[9] scikit-learn developers, "Ensemble methods," *scikit-learn documentation*, 2023, https://scikit-learn.org/stable/modules/classes.html#module-sklearn.ensemble.