Adult Census Income Level Prediction

TEAM 12: Induni Sandapiumi Nawarathna Pitiyage[1], Sara Campolattano[2]

[1- 2] University of Milano – Bicocca, CdLM Data Science

Keywords

Machine Learning
Adult Income Prediction
Performance Evaluation

Abstract

Income level prediction on adult census data is a crucial task in various socio-economic studies and policy-makings. This paper aims to use various machine learning techniques to predict whether an individual ears more than $50,000 annually. The dataset comprises demographic and socio-economic attributes like age, education, employment, marital status income etc and data related to these attributes extracted from the 1994 Census Bureau database by Ronny Kohavi and Barry Becker.
This study employs set of algorithms including random forest, j48, logistic regression, support vector machines, multi-layer perceptron, Bayes net and NBTree. Feature engineering and feature selection techniques are used to enhance the model performance. Model evaluation is conducted using accuracy, precision, recall and F1 score metrics.

TABLE OF CONTENTS

INTRODUCTION

In a world burdened by disparities, understanding the socio-economic factors of income is crucial for addressing them, and possibly light a spark on poverty to reduce its domain. Predicting individuals' income based on demographic and socio-economic factors can provide useful insights into issues such as wage disparities, access to education, and occupational segregation.
For these reasons, and using the Adult Census Income dataset, in this project we focus on predicting whether an individual earns more than $50,000 annually, making it possible to inform policy makers, drive business decisions and support financial planning. For instance, Governments can utilize these predictions to reform tax policies aimed at reducing income inequality. Additionally, institutions can offer personalised advice to help individuals to achieve their financial goals.

1. DATA EXPLORATION

The Adult Census Income dataset can be found on Kaggle [1], and it was extracted from the 1994 Census Bureau database by Ronny Kohavi and Barry Becker. The original dataset contains information on 32,561 individuals and 15 attributes, comprising five continuous variables and ten categorical variables:

- *Age*, numeric variable representing individuals'age, expressed in years;
- *Workclass*, categorical variable expressing the work type of each individual (e.g., private, self-emoloyed);
- *Fnlwgt*, numeric variable expressing the current population survey (CPS) weights;

- *Education Level*, categorical variable expressing the educational level of each individual (e.g., HS-grad, some college);
- *Education.num*, numeric variable expressing the number of years spent for education for each individual;
- *Marital.status*, categorical variable expressing the marital status of each individual (e.g., divorced);
- *Occupation*, categorical variable expressing the occupation of each individual (e.g., Prof-specialty);
- *Relationship*, categorical variable expressing family relationship (e.g., husband);
- *Race*, categorical variables expressing the ethnicity of each individual (e.g., Asian);
- *Sex*, categorical variables expressing the biological sex of each individual;
- *Capital.gain*, numeric variable expressing the capital gain of each individual;
- *Capital.loss*, numeric variable expressing the capital loss of each individual;
- *Hours.per.week*, numeric variables expressing the amount of weekly working hours for each individual;
- *Native.country*, categorical variable expressing the birth country of each individual;
- *Target*, categorical variable expressing individuals' income class (<=50K, >50K).

From the results of a descriptive analysis, we observed that the individuals' age ranges from 17 to 90, with an average age of approximately 37 years, depicting a slightly skewed distribution. The analysis also showed that most individuals work in the private sector, while only few of them (7) never worked in their lives. As for the educational level, ranging from 1 to 16, we notice that the majority reaches the 9th level, while only 51 individuals (out of 32.561) declare a 1st level of education. Coming to our variable of interest, i.e., the income, we care to highlight that only 24.08% of individuals in the dataset earn more than $50.000 per year, suggesting that the target variable is not balanced. Finally, we discovered that most individuals come from the United States; on the other hand, countries like Columbia, Japan, Italy, and India represent minor records for our sample, as in can be seen in *Figure 1*.

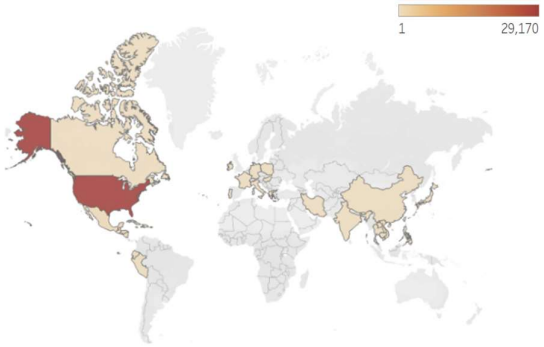

Figure 1: individuals' native countries.

2. DATA PREPROCESSING

To ensure the quality and the reliability of the data, which directly impacts the accuracy and effectiveness of machine learning models, we deemed it necessary to perform some preprocessing techniques on the dataset.

2.1 HANDLING MISSING VALUES

When exploring the dataset, we discovered that, among fifteen features, only three of them present missing values and they are all categorical: *workclass*, *occupation* and *native.country* with 1.835 (5.6%), 1.843 (5.6%), and 583 (1.7%) missing records respectively. As the reason for which these features have missing values in not known, attention must be paid when addressing this issue. Given that the number of NAs for each feature is relatively low compared to the total number of records (32.561), we decided to proceed by replacing the missing value cells with the *most frequent value* for each of the above-mentioned features [2].

2.2 FEATURE TRANSFORMATION

To ensure that the machine learning models we aim to build can leverage all the information contained in each variable, feature transformation was performed. Since all the categorical features in the dataset are non-numeric, we opted for encoding. Specifically:

- One-Hot encoding: it is performed on all the nominal categorical attributes exceeding 2 categories. Therefore, for the categorical features "*workclass*", "*education*", "*marital.status*", "*occupation*", "*relationship*", "*race*" and "*native.country*", the label encoded forms were transformed into One-Hot Encoded Forms. This operation was performed using the One-to-Many Knime [3] node.
- Label Encoding: the categorical feature "*sex*", having only two categories, i.e., male and

female, is already represented in binary form; it is therefore encoded 1 and 0.

After encoding, the dataset comprises 32.561 records and 105 features. Although One-Hot encoding leads to high-dimensional feature space, we intend to perform feature selection to reduce the number of features, therefore addressing the dimensionality issue.

2.3 DATA PARTITIONING

To develop robust and accurate models, we decided to split the dataset into to partitions, allocating 67% of data to partition A, which is intended for training and validation purposes, and 33% of data to partition B, which is intended for model testing, to ensure an unbiased evaluation of the model's performance. Stratified sampling is used to ensure that the target variable, i.e., 'income level,' is adequately represented in each subset.

3. ADDRESSING INBALACED DATASET

While carrying our analysis, we observed that the target variable, i.e., income, is not balanced. In other words, the number of individuals who earn less than $50,000 per year is significantly higher with respect to individuals who earn more than $50,000 per year (*Figure 2*).

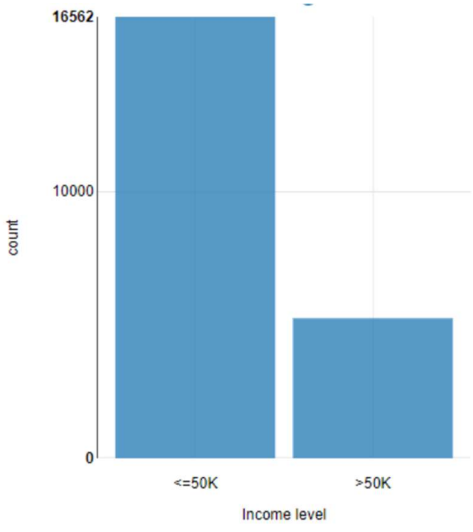

Figure 2: income levels distribution.

Overall, from Figure 2 we can observe that 16,562 individuals earn less than $50,000 per year, while 5253 individuals earn more than $50,000 annually.
To address the class imbalance problem in the dataset, the following resampling techniques are employed to adjust the class distribution, with the goal of enhancing the performance of the machine learning models.

3.1 EQUAL SIZE SAMPLING

The equal size sampling technique was used to randomly reduce the number of samples from the majority class, resulting in equal-sized population groups. After applying this technique to the target variable, i.e., income, each class (<=50K, >50K) contains 5,253 samples, resulting in a total of 10,506 individual records.

3.2 SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) was used to increase the representation of the minority class, enabling machine learning models to better learn its characteristics. After applying the SMOTE technique to the target variable, each class contains 16,562 samples, resulting in a total of 33,124 individual records.

3.3 COST SENSITIVE LEARNING

Cost sensitive learning is another technique used to address the problem of imbalanced data. It assigns different costs to different types of errors made by the classification model.
Assigning cost or penalties for classification errors is a critical task. For this approach, we focused on prioritizing the correct classification of the minority class, i.e., identifying the individuals who earn more than $50,000 annually. Therefore, we decided to assign a higher cost to the misclassification of the minority class (false negatives and false positives) to emphasize their importance (*Table 1*).

		Predicted	
Actual		-1	+1
	-1	0.0	5.0
	+1	10.0	0.0

Table 1: Cost Matrix

For false positives, the cost of misclassification is 5.0, indicating a lower penalty for incorrectly classifying individuals who earn less than $50,000 as high earners. For false negatives, the cost of misclassification is 10.0, implying a higher penalty for failing to identify individuals who earn more than $50,000. In the cost matrix, we placed a higher penalty on false negatives compared to false positives, emphasizing the importance of correctly identifying individuals who earn more than $50,000. As a result, the model is adjusted to be more sensitive to

predicting high-income individuals, as the minority class in this dataset comprises high earners.

3.4 PERFORMANCE EVALUATION

Performance evaluation metrics such as Recall, Precision, F-measure, Accuracy, and AUC (Area Under the Curve) are used to assess the effectiveness of the predictive models. Here is a brief overview of the performance evaluation metrics incorporated in this study:

- ***Recall*** (True Positives Rate or Sensitivity).
  It measures the proportion of the actual positive cases that were correctly identified by the model.

$$Recall = \frac{TP}{TP + FN}$$

- ***Precision***
  Precision measures the proportion of positive predictions that were correct instances.

$$Precision = \frac{TP}{TP + FP}$$

- ***F-measure*** (f1 score)
  F-measure shows the harmonic mean of precision and recall, providing a single metric that balances both measures. It ranges from 0 to 1, where 1 indicates the best possible performance.

$$F\ measure = \frac{2 \times precision \times recall}{precision + recall}$$

- ***Accuracy***
  Accuracy measures the overall correctness of the model's prediction across all the classes.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- ***AUC-ROC (Area Under ROC Curve)***
  This metric quantifies the model's ability to discriminate between negative and positive classes with respect to different threshold. ROC plots the true positive rate (recall) against False positive rate. A value closer to 1, indicates a better performing model.

These metrics provide different insights into the performance of predictive models. The overall correctness of the model is measured by accuracy, while precision and recall are more informative when dealing with imbalanced datasets. For example, a higher precision indicates that the model tends to predict more true positives, resulting in fewer false positives. Conversely, a lower recall suggests that the model misses many actual positives, leading to more false negatives. The F-measure helps to balance the recall and precision metrics, with a high F-measure score indicating a good balance between precision and recall. AUC-ROC assesses the model's performance without being sensitive to class imbalance, providing a comprehensive evaluation of the model's ability to distinguish between classes.

3.5 PERFORMANCE

The primary objective of incorporating various sampling techniques is to determine the most effective methods and understand their performance. The optimal classifiers for predictive analysis can vary based on the dataset's specific characteristics. In this study, we decided to adopt the following three classifiers due to their performance and broad applicability:

1. The Random Forest (RF) which is robust against overfitting and can handle large datasets with high dimensionality.
2. Support Vector Machines (SVMs), since they are effective in handling high-dimensional data, particularly when the number of dimensions exceeds the number of samples.
3. Gradient Boosting Machines (GBMs), as it excels in handling diverse types of data (numerical, categorical, etc.), and their regularization techniques help mitigate overfitting [4].

In *Table 2* it is possible to observe the results obtained from different sampling techniques in relation the models we chose.

		Recall	Precision	F1 Score	Accuracy	AUC
Imbalanced Classes	*RF*	0.59	0.77	0.67	0.86	0.895
	SVM	0.54	0.75	0.62	0.85	0.742
	GBT	0.62	0.78	0.69	0.86	0.921
Equal Size Sampling	*RF*	0.875	0.812	0.842	0.836	0.907
	SVM	0.872	0.784	0.826	0.816	0.816
	GBT	0.869	0.826	0.847	0.843	0.922
SMOTE	*RF*	0.887	0.905	0.896	0.897	0.965
	SVM	0.897	0.803	0.847	0.838	0.838
	GBT	0.897	0.894	0.896	0.895	0.967
Cost Sensitive	*RF*	0.732	0.617	0.670	0.826	0.794
	SVM	0.540	0.750	0.628	0.846	0.742

Table 2: classifiers performance assessment for different sampling methods

Observing the results in *Table 2*, it is possible to affirm that the SMOTE sampling technique yields the best results among all the methods included. Gradient

Boosting Trees demonstrate the highest overall performance, achieving the top AUC score, followed closely by Random Forest. On the other hand, the Support Vector Machine lags, particularly in terms of recall and discriminative ability.

The equal size sampling method also performs well, with Gradient Boosting being the best classifier.

Despite the cost-sensitive method showing an accuracy of 84%, the recall is significantly lower than its precision. This indicates that the model misclassifies many high earners as low earners. Similarly, for imbalanced classes, low values are observed for recall, precision, and F-measure compared to accuracy. This result is expected, as the model tends to favour the majority class over the minority class.

4. PREDICTIVE MODELS

Given that SMOTE and Equal Size Sampling techniques demonstrated the best performance for addressing class imbalance, we will incorporate the dataset along with the following machine learning models, grouped as follows:

- Heuristic Classifiers: J48 and Random Forest (RF) classifiers.
- Regression-Based Classifiers: Logistic Regression (LR) classifier.
- Separation Classifiers: Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP) classifiers.
- Probabilistic Classifiers: BayesNet and NbTree classifiers.

Each type of classifier has its own strengths and weaknesses in solving classification problems [6]. These classifiers will be implemented using Weka (3.7) nodes and further analyzed through holdout and cross-validation methods.

4.1 HOLDOUT

The holdout method is used to evaluate the performance of a model on data that was not used for training. This method involves splitting the available dataset into two parts: 67% of the data is used for training, while the remaining 33% is used as unseen data for testing. In this procedure, the model is trained using only the training data. The trained model is then evaluated on the test data (unseen data) to assess its performance.

4.2 CROSS VALIDATION

Cross-validation is another method for evaluating the performance of machine learning models. Among the various cross-validation techniques, this study incorporates stratified k-fold cross-validation. This technique involves dividing the data into multiple subsets while ensuring that each fold has a similar distribution with respect to the target variable [5].

4.3 FEATURE SELECTION

Feature selection is performed on the datasets obtained using the SMOTE sampling technique and the equal size sampling technique. For feature selection, we decided to adopt the CfSubsetEval method, also known as Correlation-based Feature Subset Evaluation. CfSubsetEval evaluates subsets of features based on their correlation with the target variable, i.e., income. This method is ideal to use with both continuous and categorical variables in the dataset. It combines correlation-based evaluations for continuous variables with information-based evaluations for categorical variables.

4.4 SMOTE ASSESSMENT

Out of 105 features, 28 features were selected for the dataset obtained by SMOTE for all the classifiers: j48, Random Forest (RF), Logistic regression (LR), Support Vector Machine (SVM), Multi-Layer-Perceptron (MLP), BayesNet and NbTree.

4.4.1 Holdout (SMOTE) Assessment

The results can be observed in *Table 3*.

	Recall	Precision	F1 Score	Accuracy	AUC
J48	0.891	0.872	0.881	0.88	0.943
RF	0.894	0.895	0.894	0.895	0.964
Logistic	0.864	0.821	0.842	0.838	0.918
SVM	0.899	0.793	0.842	0.832	0.832
MLP	0.951	0.589	0.727	0.643	0.939
BayesNet	0.811	0.935	0.869	0.877	0.959
NBTree	0.76	0.957	0.847	0.863	0.952

Table 3: Holdout results for SMOTE.

As we can see from the above table, Random Forest (RF) stands out with high scores in all metrics. J48 also performs well with a recall of 0.891, and an AUC of 0.943. On the other hand, Logistic Regression and SVM show moderate performance, while MLP has the highest recall at 0.951 but lower precision and accuracy. BayesNet and NbTree classifiers perform decently, with the latter having the highest precision at 0.957 and an AUC of 0.952. Therefore, it is possible to affirm that Random Forest is the top performer overall, particularly excelling in accuracy and AUC.

4.4.2 Cross-Validation (SMOTE) Assessment

As for cross-validation, we opted for the stratified k-fold cross-validation using 3 folds, where the number of features considered was 28. In Figure 3, 4, and 5 we can observe the results obtained for Recall, Precision and Accuracy, respectively.

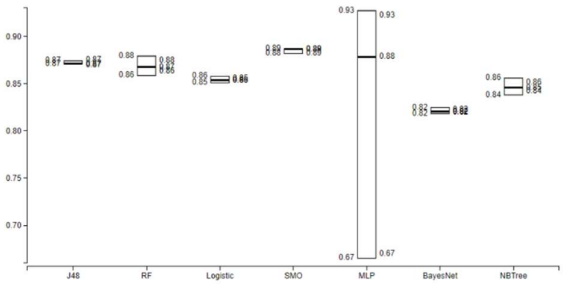

Figure 3: k-folds cross-validation Recall.

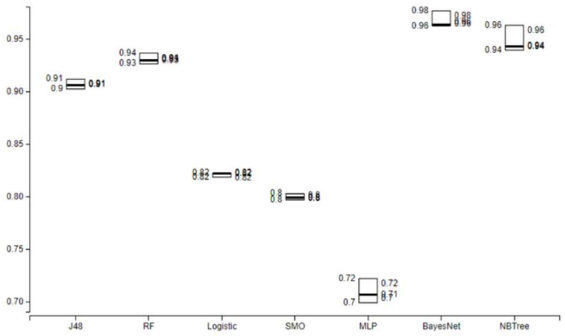

Figure 4: k-folds cross-validation Precision.

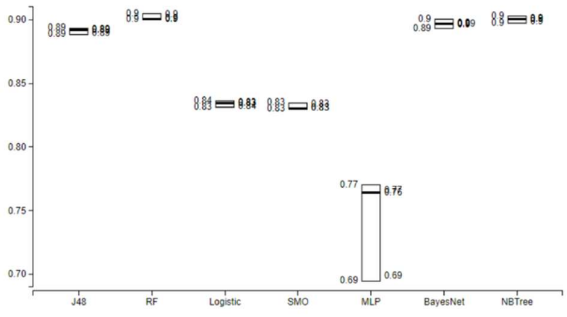

Figure 5: k-folds cross-validation Accuracy.

From the evaluation of several classifiers based on cross-validation, shown in the above graphs, it is possible to observe distinct performance patterns. Specifically, we can see that for:

Recall: Logistic Regression achieves the highest score (0.88), followed closely by J48 and Random Forest (both around 0.87). Multi-Layer Perceptron (MLP) shows highly variable performance (0.67 to 0.93), indicating inconsistency.

Precision: BayesNet demonstrates the highest precision (around 0.98), followed by NBTree (0.94-0.96). J48 and RF perform well with precision around 0.91 and 0.93-0.94, respectively. MLP exhibits the lowest and most variable precision (0.70 to 0.72).

Accuracy: BayesNet, NBTree, and RF achieve the highest accuracy (around 0.90), with consistent performance. J48 also performs well (0.89), while MLP shows lower and more variable accuracy (0.69 to 0.77).

Therefore, we can conclude that Logistic Regression, J48, and Random Forest are consistently reliable across recall, precision, and accuracy metrics. On the other hand, Multi-Layer Perceptron demonstrates the most variability and lower performance across these metrics.

4.5 EQUAL SIZE SAMPLING ASSESSMENT

As for the dataset generated through equal-size sampling, all previously mentioned classifiers select 14 features using the CfSubsetEval method. These attributes appear consistently in both sampling methods: age, capital gain, capital loss, years of education, hours worked per week, executive manager, other services (occupation), having children of one's own in the family, and being the husband in the family.

4.5.1 Holdout (ESS) Assessment

The results can be observed in *Table 4*.

	Recall	Precision	F1 Score	Accuracy	AUC
J48	0.827	0.809	0.818	0.816	0.891
RF	0.896	0.790	0.840	0.829	0.892
Logistic	0.842	0.804	0.822	0.818	0.901
SVM	0.880	0.763	0.818	0.804	0.804
MLP	0.832	0.584	0.686	0.620	0.902
BayesNet	0.867	0.810	0.838	0.832	0.909
NBTree	0.860	0.820	0.839	0.835	0.912

Table 4: Holdout results for Equal Size Sampling

As we can see, the Naive Bayes Tree (NBTree) emerges as the top performer with the highest AUC and strong metrics across the board. Random Forest stands out with the best Recall at 0.896. BayesNet

shows a high AUC of 0.909 and balanced metrics. Logistic Regression excels particularly in AUC (0.901) and maintains solid performance in other metrics. On the other hand, Support Vector Machine (SVM) has a high Recall of 0.880 but a lower Precision of 0.763. Multilayer Perceptron (MLP) demonstrates lower performance compared to the others, especially in Accuracy (0.620) and F1 Score (0.686).

4.5.2 Cross-Validation (ESS) Assessment

We recall that the stratified k-fold cross-validation uses 3 folds, where the number of features considered was 14. The results obtained for Recall, Precision and Accuracy, can be observe in the below graphs.

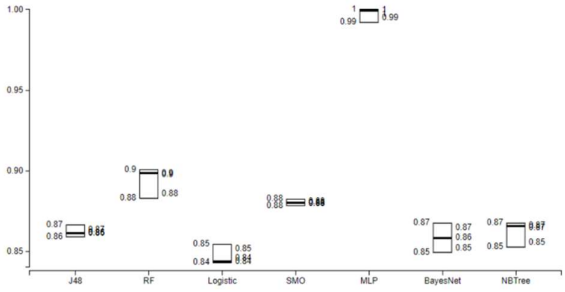

Figure 6: k-folds cross-validation Recall.

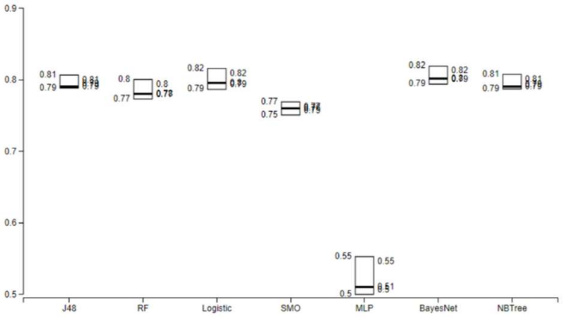

Figure 7: k-folds cross-validation Precision.

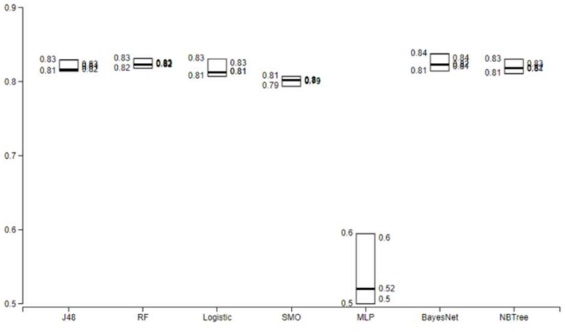

Figure 8: k-folds cross-validation Accuracy.

From *Figure 6*, *7*, and *8*, we can observe that for:

Recall: Random Forest (RF) performs the best with a median of 0.90. SMO (SVM) is also consistent at 0.88. J48, BayesNet, and NBTree have similar recall values around 0.87. Logistic Regression and MLP have the lowest recall at 0.85.

Precision: Logistic Regression and BayesNet lead with a median of 0.82. J48 and NBTree follow at 0.81. RF has a precision of 0.80. SMO shows lower precision at 0.77, and MLP has the lowest precision at 0.55.

Accuracy: J48, RF, Logistic, SMO, BayesNet, and NBTree all show high and similar accuracies between 0.79 and 0.84. MLP has significantly lower accuracy, between 0.5 and 0.6.

Therefore, we can conclude that, overall, Random Forest excels in recall, Logistic Regression and BayesNet lead in precision, and most classifiers show high accuracy except MLP.

5. MODEL VALIDATION

As for model evaluation, also in this case we opted for the Holdout method. Model validation was therefore performed on Partition B, which we recall it consisting of 33% of the original records and not being fed to the models' training process.

In *Table 5* it is possible to observe the results of the evaluation metrics obtained from models' testing.

	Recall	Precision	F1 Score	Accuracy	AUC
J48	0.729	0.655	0.690	0.842	0.894
RF	0.723	0.707	0.715	0.861	0.896
Logistic	0.822	0.580	0.680	0.814	0.902
SVM	0.865	0.543	0.667	0.792	0.820
MLP	0.812	0.439	0.570	0.705	0.898
BayesNet	0.579	0.751	0.654	0.852	0.903
NBTree	0.442	0.821	0.575	0.842	0.898

Table 5: Holdout/Models evaluation on test data

From the above table, it is possible to see that, when predicting unseen data, the Random Forest (RF) model demonstrates the best overall performance with a high F1 score of 0.715, an accuracy of 0.861, and an AUC of 0.896. The Support Vector Machine (SVM) model has the highest recall at 0.865 but falls short in precision at 0.543 and F1 score at 0.667. Logistic Regression also shows strong performance with a recall of 0.822 and an AUC of 0.902, although its precision is lower at 0.580. The J48 model

maintains balanced performance metrics, including an accuracy of 0.842 and an AUC of 0.894. BayesNet achieves the highest AUC at 0.903 and good precision of 0.751 but has a lower recall at 0.579. The Naive Bayes Tree (NBTree) model has high precision at 0.821 but the lowest recall at 0.442. Lastly, the Multilayer Perceptron (MLP) generally performs the worst with the lowest precision at 0.439 and an accuracy of approximately 70%.

From the results obtained by testing the models' performance on previously unseen data, we can conclude that, overall, the Random Forest and BayesNet are the top-performing models.

CONCLUSIONS

As we discussed in the introduction of this paper, disparities in income and educational level are a burden to our society. With the aim of achieving valuable results to inform policy makers, in this study we explored the application of various machine learning algorithms to predict income levels using demographic and socio-economic data from the Adult Census Income dataset. The data preprocessing steps included handling missing values, which were found only for three categorical variables and were dealt with using the most frequent value approach; feature transformation, which comprised variables encoding, and data partitioning, with a 1/3 split ratio. These steps ensured the reliability and quality of the dataset. When conducting the analysis, the class imbalance issue was addressed through SMOTE and equal-size sampling techniques, with the aim to enhance model performance. During the training phase, results demonstrated that different models excelled in various metrics. Random Forest showed the highest recall, making it the best at identifying individuals earning more than $50,000 annually, whereas Logistic Regression and BayesNet achieved the highest precision, indicating their effectiveness in minimizing false positives. Overall, accuracy was high for most classifiers except for the Multilayer Perceptron, which displayed lower performance and consistency across metrics. The model validation using the holdout method confirmed the robustness of Random Forest and BayesNet, with Random Forest achieving the best overall performance on unseen data. BayesNet, with the highest AUC, also showed balanced performance. In conclusion, Random Forest and BayesNet emerged as the top-performing models for income prediction in this dataset. We strongly believe that these findings highlight the importance of selecting appropriate algorithms and preprocessing techniques in building effective predictive models. Although features such as educational level, native country, occupation and number of worked hours per week, resulted in being influential factors for determining whether an individual earns more than $50,000 annually, future work could explore the integration of additional features and advanced ensemble methods to further enhance prediction accuracy and robustness.

REFERENCES

[1] Kaggle (June 2024), Adult Income Census dataset: https://www.kaggle.com/datasets/uciml/adult-census-income

[2] Allison P. D. Missing Data: http:// https://wwwstatisticalhorizonscom/wp-content/uploads/2012/01/Milsap-Allison.pdf

[3] Knime (June 2024). Optimizing KNIME workflows for performance. Retrieved from: Optimizing KNIME workflows for performance | KNIME

[4] Wikipedia, Gradient Boosting: https://en.wikipedia.org/wiki/Gradient_boosting

[5] GeeksforGeeks, Cross Validation: https://www.geeksforgeeks.org/cross-validation-machine-learning/

[6] Deepajothi, S., Selvarajan, S. (2012). A Comparative Study of Classification Techniques On Adult Data Set: https://www.ijert.org/research/a-comparative-study-of-classification-techniques-on-adult-data-set-IJERTV1IS8243.pdf