January - 2023

# ADULT CENSUS INCOME

David Alibrando[1], Igor Schirripa[2], Rossella Longo[3], Tommaso Strada[4]

**Abstract**

This study aimed to predict whether an individual earns more than 50k per year based on socio-economic input attributes using supervised machine learning methods. The performance of several different models was evaluated, with the class imbalance in the dataset addressed through equal size sampling. Feature selection was implemented using both multivariate filter and wrapper methods to identify the most important predictors of income. Classification models were validated using Iterated Holdout and K-fold cross validation methods. The results demonstrated that Bayesian models and tree-based models were the most effective for this task, with performance evaluated in terms of accuracy, precision, recall, F measure, specificity, and AUC.

**Keywords**

Machine Learning — Classification — Feature selection — Supervised learning — KNIME

*[1]82184 - d.alibrando@campus.unimib.it, [2]883614 - i.schirripa1@campus.unimib.it, [3]884261 - r.longo7@campus.unimib.it, [4]829351 - t.strada1@campus.unimib.it*

## Contents

## Introduction and research questions

***Is it possible to determine an individual's annual income based on personal, employment, and social attributes? What are the factors that most influence this decision?***

This project aims to predict whether an individual earns more or less than $50,000 per year based on their personal, work, and demographic characteristics. To do this, we will collect and analyze data on the incomes and characteristics of a sample of individuals, then use this data to build and train machine learning models. We will evaluate the accuracy and robustness of the models using various techniques and make any necessary modifications to improve their performance. Possible applications of this models include aiding in personnel hiring and career planning, offering personalized financial services, and exploring the factors that influence a person's income. It may also be useful for developing policies and programs to reduce income inequalities and promote equal opportunities, or for use in other contexts or sectors to predict the income of similar individuals.

## 1. Dataset Structure

The dataset "adult-census-income", which was obtained from Kaggle[1], consists of 32561 records of American individuals that were extracted from the Census Bureau database in 1994

[1]Adult Census Income - https://www.kaggle.com/datasets/uciml/adult-census-income

by Ronny Kohavu and Barry Becker. These records represent the annual income of these individuals and the characteristics that qualify them to meet the following requirements:

- Age greater than 16 years;
- Adjusted Gross Income (AGI) $> 100$;
- Average weight per class (AFNLWGT) $> 1$;
- Hours worked per week (HRSWK) $> 10$

The 15 attributes present for each class of individuals with their corresponding data type are the following:

- Age: age of individuals, integer.
- Workclass: work sector, string.
- Fnlwgt: weight assigned by the Census Bureau, integer.
- Education: indicates the highest level of education completed, string.
- Education-num: specifies the number of years spent in academia, integer.
- Marital-status: marital status, string.
- Occupation: work sector, string.
- Relationship: type of civil relationship, string.
- Race: Ethnicity, string.
- Sex: gender, string.
- Capital-gain: represents annual capital earned, integer.
- Capital-loss: represents annual capital lost, integer.
- Hours-per-week: indicates hours worked per week, integer.
- Native-country: country of origin, string.
- Income: specifies whether annual net income is above or below $50,000$. This is the target variable for our study, string.

After conducting an exploratory analysis of the data, we discovered that the target class is heavily skewed towards individuals with an annual income of \$50,000. Specifically, there is a 3:1 ratio in favor of this income class. The Pearson correlation matrix also revealed a strong positive linear correlation between the variables ”relationship” and ”sex.” Further analysis using box plots identified outliers in the variables ”Age,” ”Hours per week,” and ”education-num.” Additionally, we identified missing values in the attributes ”Workclass”, ”Occupation”, and ”Native country” with 1836, 1846, and 583 missing values, respectively.

## 2. Preprocessing

To prepare the dataset for analysis, several procedures were carried out using KNIME platform. The main operations are outlined below.

### 2.1 Manipulation

To reduce the number of attributes while preserving the same level of information, a new variable called ”capital net” was created by subtracting the ”capital loss” variable from the ”capital gain” variable using the ’Math formula’ node. This new variable contained some outliers that were later addressed. The target variable was also transformed from a string type to a boolean type (binary) variable, which can only take on two logical values (0 for income $>$ \$50,000 and 1 for income $\leq$ \$50,000). This transformation, carried out using the ’Rule Engine’ node, resulted in the creation of a new variable called ’target’ for easier interpretation.

### 2.2 Missing values

To handle missing values in the ”Workclass”, ”Occupation” and ”Native country” variables, they were replaced with the most frequent values using a specific KNIME node (“missing values node”) configured for this purpose.

### 2.3 Outliers

Outliers in the ”Age”, ”Hours per week”, ”education-num” and ”capital_net” variables were identified using box plots and removed by combining two KNIME nodes. The first node was used to order the values, and the second node was used to remove the corresponding instances. This process was repeated for each variable, and the data distribution was checked before and after the manipulation to verify the effect on other variables of interest. In addition, instances with a value of 17 in the ”Age” variable were removed because they contributed to the outliers in the ”capital_gain” variable. Since the ”capital_net” variable is derived from the ”capital_gain” variable and therefore exhibits a high positive correlation, it was decided to remove the outliers of the latter in order to improve the distribution of ”capital_net.”

### 2.4 Attribute Removal

To avoid perfect correlations, the ”capital_gain” and ”capital_loss” attributes that were used to create the ”capital_net” variable were removed. Additionally, due to high positive linear correlations with the ”education-num” and ”sex” variables, the ”education” and ”relationship” attributes were removed, respectively. The ”fnlwgt” variable was also removed as it was not relevant for the purpose of this analysis. All of these operations were carried out using the ’column filter’ node. Finally, the ”income” attribute was removed and replaced with the ”target” variable.

### 2.5 Class imbalance management

In the final preprocessing phase, the imbalance in the target class identified during the data exploration phase was addressed. Equal size sampling was performed to build samples in which the target variable was evenly distributed and to reduce the size of the dataset. This operation was carried out using the ’equal size sampling’ node.

### 2.6 Data preparation

Before applying and analyzing algorithms, the dataset was prepared for these operations. A Holdout partitioning was performed using the ’partitioning’ node, resulting in three partitions: the training set, the test set, and the validation set

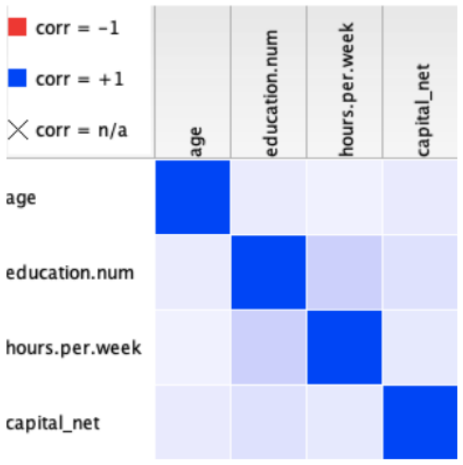

**Figure 1.** Variable's correlation plot

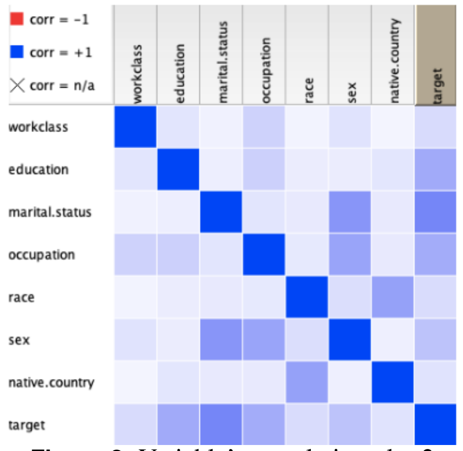

**Figure 2.** Variable's correlation plot 2

(partition B). Finally, the quantitative attributes were normalized to a Gaussian distribution.

### 2.7 Feature selection

To improve the interpretability and efficiency of the models, we used two feature selection techniques: multivariate filter and wrapper. The filter method selects attributes based on CFS, a measure of association based on correlation, before classifier selection. The wrapper method selects the most relevant attributes using a classification model, in this case a random forest. These techniques allow us to reduce the dimensionality of the dataset without reducing accuracy, improving computational costs and inference time. The results of the models with and without feature selection will be discussed in the next section.

## 3. Models

To address the research question, several classification models were implemented.

### 3.1 Probabilistic Models

Probabilistic models estimate the probability that an input belongs to a certain class based on its characteristics, patterns, and relationships. The goal of these models is to accurately classify inputs into predefined classes.

**BayesNet** Two Bayesian Networks were implemented to predict the probability that an input belongs to a certain class using probability distributions and network structure. The two types differ in their algorithms.

**Bayes Net (K2)** The first Bayesian Network type uses the K2 heuristic search algorithm to find the best fit for the input data. It begins with an empty network and adds edges between nodes, selecting the edge that improves the model's fit the most. The search is repeated until no more improvements can be made.

**BayesNet (TAN)** Tree Augmented Naive Bayes is a variant of Naive Bayes that handles relationships between input features and multidimensional data. It has a tree structure, with nodes representing input features, edges representing relationships between features, the root node representing the output class label, and the leaves representing the input features. The tree structure is determined by the information between the input features and the output class.

**Heuristic models** Heuristic models use approximate rules to find a solution that fits the data in a reasonable amount of time. For this work, two types of decision trees were chosen based on their underlying algorithms.

**Decision Tree** A decision tree is a classification model that uses a tree structure to predict an input's belonging to one or more classes. It is built based on the selection of the most relevant features for data classification. The Decision Tree Learner in KNIME was used to build and train a decision tree on input data. It uses an algorithm to select the most relevant features and builds the tree based on them. After training, the decision tree can be used to predict the class belonging to new inputs.

**J48** J48 is an extension of the ID3 algorithm that uses a recursive data division technique to build a decision tree based on the characteristics of the input dataset and a feature selection criterion based on entropy. It also uses a "pruning" technique to improve accuracy.

### 3.2 Regression Models

**Logistic regression** logistic regression is a binary classification model that predicts the probability of an input belonging to one of two possible output classes. It maps input values to probabilities between 0 and 1 using a logistic regression function (sigmoid function). Two types of logistic regression with different underlying algorithms were implemented.

**Logistic Regression** The "Logistic Regression" component in KNIME is an algorithm that optimizes the cost function to train the logistic regression model. It minimizes the classification error of the model.

**Simple logistic** The Simple Logistic component in KNIME implements an ordinary logistic regression algorithm. It maps input values to probabilities and trains the model based on these probabilities using a logistic regression function.

### 3.3 Separation models

Separation models use a separation function to distinguish between two or more possible output classes. This function is typically a mathematical function that maps inputs to the corresponding class.

**SVM (Support Vector Machine)** Two types of Support Vector Machines (SVMs) were implemented. These models use a separation function called the "separation hyperplane" to divide input classes in the input space, aiming to maximize the distance between them (the "margin").

**SPegasos** The Stochastic Primal-Dual Extragradient Method for SVM Optimization is used to train Support Vector Machines in KNIME. It uses a stochastic primal-dual extragradient method to find the optimal separation hyperplane.

**SMO: (Sequential Minimal Optimization)** works sequentially, trying to optimize one pair of support vectors at a time until the optimal separation hyperplane is found.

## 4. Performance Measures

To evaluate the performance of the implemented models, several measures were used, including accuracy, precision, recall, specificity, F1 measure, and AUC.
Accuracy measures a model's ability to accurately classify new records.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \tag{1}$$

The measures chosen were Precision, Recall, and F1. In particular, the latter is defined as the harmonic mean between Recall and Precision, and it was deemed ideal to only show the result obtained from it.
Precision quantifies the number of positive class predictions that actually belong to the positive class. The higher the Precision, the fewer false positives are made.

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

Recall quantifies the percentage of positive class predictions made on all positive examples in the dataset. A high Recall indicates that few positive records are misclassified as the negative class. In fact, Recall is equivalent to the true positive rate.

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

The F1 measure is a metric commonly used in binary classification to evaluate the accuracy of a model's predictions. It is calculated as the harmonic mean of precision and recall, which measure the model's ability to correctly identify positive classes and avoid misclassifying negative records as positive, respectively.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

Specificity is a metric used in binary classification to evaluate a model's ability to correctly classify negative records. It is calculated as the ratio of true negatives (correctly classified negative records) to the total number of negative records. The formula for calculating specificity is as follows:

$$Specificity = \frac{TN}{TN+FP} \tag{5}$$

The ROC curve is a graphical tool used to evaluate the performance of a binary classification model in terms of sensitivity (ability to correctly identify positive cases) and specificity (ability to correctly identify negative cases). It is obtained by plotting the true positive rate (recall) on the Y-axis against the false positive rate on the X-axis at different decision thresholds of the model. An optimal model should have an ROC curve that is as close as possible to the top left point (0,1) of the possible value space, meaning it has the highest true positive rate and lowest false positive rate. The area under the ROC curve (AUC) represents the model's ability to distinguish between the two classes and can be used as a performance evaluation metric. A model with a higher AUC will be more accurate in distinguishing between the two classes than a model with a lower AUC.

## 5. Model validation

The classification models were validated using the Iterated Holdout and 10-fold cross validation methods to analyze the generalization error, and by analyzing the confidence interval on accuracy. The dataset was divided into a training and test partition for model training and evaluation, and a separate partition (B) for more accurate verification of model performance. Validating the models in this way helps reduce the risk of overfitting and accurately estimate generalization.

**Iterated Holdout** Is a method for estimating the generalization error of a machine learning model. It involves fitting the model on the train and test partition, searching for the best hyperparameters, and evaluating the final performance. The process is repeated multiple times to estimate the generalization error through the average of the accuracies, which are then compared to the accuracy of the model on partition B. The filter method estimates are less accurate than those of the wrapper method. The wrapper method also shows that Bayesian (K2) and decision tree-based classifiers have the most accurate estimates.

**K-Fold cross validation** Is a method for estimating a machine learning model's generalization error by dividing the data into K folds, training and testing the model K times, and taking the average performance on the test sets. It is more robust than other methods because it uses the data more efficiently and allows for the identification of potential problems. K-fold cross validation is also more accurate than the iterated holdout method, particularly when using the wrapper feature selection method. In particular, the estimates of the various models using the filter method do not differ significantly from the actual accuracy on partition B, but the wrapper method provides more accurate confirmation, similar to the iterated holdout method, that the Bayesian models have the highest and most accurate estimates compared to the other models, and in this case also similar to decision tree-based models.

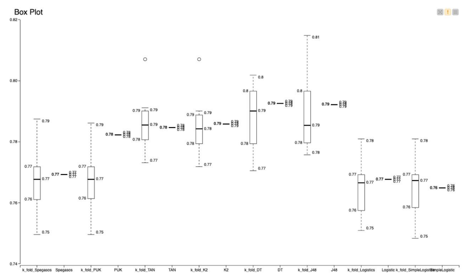

**Figure 3.** Wrapper method gen error with k-fold cross validation

**Confidence interval** To validate the accuracy of the models, a confidence interval is calculated using the accuracy of the model on the train and test set partition as an estimate of the model's accuracy on the population. The interval is calculated using the following formula:

$$\left(\frac{acc+\frac{Z^2_{1-\frac{a}{2}}}{2\cdot N}-Z_{1-\frac{a}{2}}\cdot\sqrt{\frac{acc}{N}-\frac{acc^2}{N}+\frac{Z^2_{1-\frac{a}{2}}}{4\cdot N^2}}}{\left(1+\frac{Z^2_{1-\frac{a}{2}}}{N}\right)},\frac{Z^2_{1-\frac{a}{2}}+Z^{2\cdot N}_{1-\frac{a}{2}}\cdot\sqrt{\frac{acc}{N}-\frac{acc^2}{N}+\frac{Z^{4\cdot N^2}_{1-\frac{a}{2}}}{Z^2}}\Bigg)}{\left(1+\frac{Z^N_{1-\frac{a}{2}}}{N}\right)}\right)$$

In both feature selection methods, the actual accuracy values on partition B are similar to the calculated confidence interval range, indicating that the estimates are accurate. Confidence intervals can help evaluate a model's performance, including the uncertainty of its predictions.

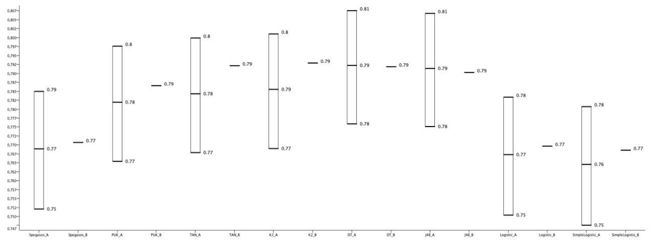

**Figure 4.** Wrapper method confidence interval

**Test vs Partition B** Finally, to validate the models, a comparison was made between the accuracy values obtained on the test set and partition B, in order to highlight any positive or negative changes in the models' performance. In the filter method for variable selection, we can see that the accuracy of support vector machine-based models is quite unstable, tending to decrease on partition B.
This shows, as previously demonstrated, a high generalization error that has led to overfitting of the models. In contrast, for the wrapper method, results are more stable for most models and there is a slight increase in accuracy on partition B for almost all classifiers.

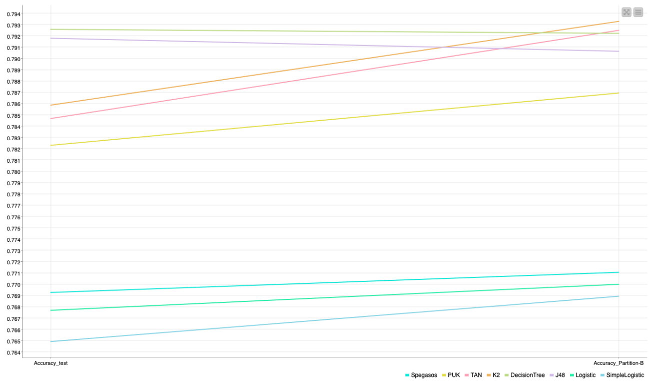

**Figure 5.** Wrapper method test vs. partition B

## 5.1 Results interpretation

All models perform well in the binary classification task, with accuracy and AUC reaching 80% and 90%, respectively, in both feature selection methods.

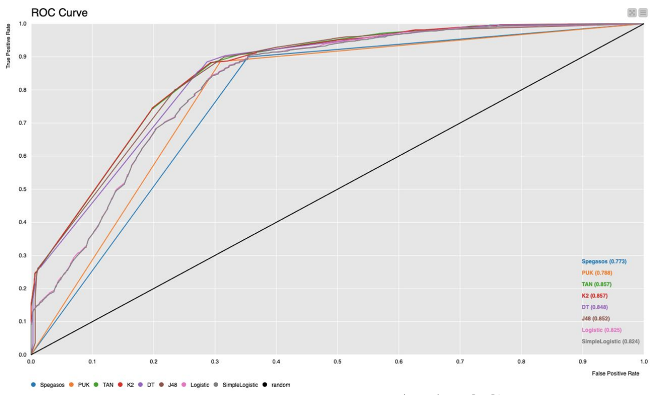

**Figure 6.** Wrapper method ROC

Bayesian classifiers K2 and Tan perform exceptionally well in both the filter and wrapper methods, followed by decision tree-based classifiers. These models have excellent F-measure values, precision, recall, and specificity.

The low generalization errors and confidence intervals confirm the strong performance of Bayesian and decision tree-based classifiers. While the filter method produced the highest results, the wrapper method was selected as the best feature selection method due to its stable and accurate model validation results using only 3 variables, which are as effective as the filter method using 5 variables but with less computational cost, time, and memory space.

| MODEL | Recall | Prec. | Acc. | Spec. | F1 | AUC |
|---|---|---|---|---|---|---|
| K2 | 0,706 | 0,861 | 0,793 | 0,883 | 0,776 | 0,857 |
| TAN | 0,706 | 0,860 | 0,792 | 0,882 | 0,775 | 0,857 |
| DT | 0,721 | 0,846 | 0,792 | 0,865 | 0,779 | 0,848 |
| J48 | 0,685 | 0,875 | 0,791 | 0,899 | 0,769 | 0,852 |
| PUK | 0,691 | 0,861 | 0,787 | 0,886 | 0,767 | 0,788 |
| Spegasos | 0,646 | 0,869 | 0,771 | 0,900 | 0,741 | 0,773 |
| Logistic | 0,650 | 0,862 | 0,770 | 0,893 | 0,741 | 0,825 |
| SL | 0,650 | 0,860 | 0,769 | 0,892 | 0,740 | 0,824 |

**Table 1.** Performance of different models (wrapper)

## 6. Conclusion

Our analysis has successfully confirmed the first objective outlined in the appendix, which was to identify the factors that have the greatest impact on income in 1994. Specifically, we found that age, marital status, and net capital were the key factors that influence income level during this time period. In conducting this analysis, we utilized a number of classification models, including Bayesian models and tree-based models that employed the wrapper method of feature selection. These models proved to be effective, with strong and reliable performance. It is worth noting that, in our analysis, we did not find significant influence of race and gender on income level. While these factors are often associated with inequalities, they did not play a significant role in determining income in this particular study. This finding suggests the importance of considering other factors, such as age, marital status, and net capital, when examining income disparities and developing policies to address them. Looking ahead, it would be interesting to conduct similar analyses using more recent data in order to understand how these factors have changed over time and to determine if they continue to be significant in influencing income. Additionally, it may be beneficial to incorporate cluster analysis as a future development in order to group individuals with similar characteristics and examine how these clusters differ in terms of income level. Such an analysis would provide valuable insights into current social income disparities and could inform the development of public policies and support programs aimed at reducing inequalities and promoting equal opportunity. In addition, it would be worthwhile to expand the scope of this analysis to include a wider range of variables in order to gain a more comprehensive understanding of the factors that influence income.

## References

1. https://www.knime.com/

2. Importance normalization: https://medium.com/@urvashilluniya/why-data-normalization-is-necessary-for-machine-learning-models-681b65a05029

3. Importance and types of feature selection - https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/

4. Census bureau - https://www.census.gov/en.html

5. Top 6 models for classification task - https://towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501

6. Class Imbalance problem - Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In Proc.of the Int'l Conf.on Artificail Intelligence, volume 56. Citeseer,2000