Application of Machine Learning Techniques Of Census Income Data

By
YASASWINI KOMMANABOENA
SATHV IK MADARAPU
Bradley University
Email:ykommanaboena@mail.bradley.edu
Email:smadarapu@mail.bradley.edu

***Abstract* – With advancements in data collection, the use of machine learning algorithms to analyze large datasets has become more effective in extracting key insights. This study focuses on the Adult Dataset from the UCI Machine Learning Repository, which includes 32,561 records and 14 features, aiming to predict whether an individual's income exceeds $50,000 annually. The dataset undergoes preprocessing steps such as data cleaning and feature selection before applying various machine learning models. We explore the effectiveness of ten algorithms, including Decision Table, JRip, IBk (K-nearest Neighbors), Random Forest, Random Tree, J48 Decision Tree, LogitBoost, Naive Bayes, Attribute Selected Classifier, and Multilayer Perceptron. Furthermore, ensemble techniques like Bagging and Stacking are utilized to enhance performance. All experiments are conducted using WEKA, which streamlines preprocessing and model evaluation.**

I. INTRODUCTION

The field of data science and machine learning has witnessed significant growth due to the availability of large datasets and the continuous advancements in algorithms. Tools such as WEKA (Waikato Environment for Knowledge Analysis) have simplified the process of data preprocessing, model building, and evaluation, enabling users to perform complex machine learning tasks without requiring in-depth programming skills. WEKA's "Explorer" interface is particularly user-friendly, allowing easy application of a variety of algorithms for classification and clustering.

Public datasets, such as those from the UCI Machine Learning Repository, offer valuable opportunities for practical experimentation with machine learning models. For instance, the UCI Adult Dataset serves as an ideal resource for tasks like predicting income levels based on demographic information. These tools and datasets have democratized access to machine learning, making it possible for both novice and experienced users to address real-world problems.

The Knowledge Discovery in Databases (KDD) process involves the discovery of meaningful patterns from large datasets, and its search phase, known as data mining, utilizes various techniques like data cleaning, clustering, and classification. Data mining methods are instrumental in assisting businesses and organizations to make informed decisions by predicting behaviors and trends, which previously would have been too complex or time-consuming analyze.

A. Definitions

The terms are classification, accuracy, attribute selection, and machine learning.

Classification: The task of assigning an instance to a predefined category based on its features.

Clustering: A technique that groups data points based on similarity in their features, without using predefined labels.

Unsupervised Learning: A learning method where the model identifies patterns in data without labeled outcomes.

Accuracy: A measure of how many correct predictions a model Makes out of the total prediction

*ROC :*A graph that show the performance of a binary classifier comparing the true positive rate and false positive rate

Precision: The ratio of correctly predicted positive instances to all predicted positive instances.

Recall : The ratio of correctly identified positive instances to all actual positive instances.

F1 Score: The balance between precision and recall, calculated as their harmonic mean.

Kappa Statistic: A metric that quantifies the agreement between classifiers, considering random5 chance, with a value of 1 indicating perfect agreement..

1.1.ALGORITHMS

In this study, ten machine learning algorithms were applied using the WEKA tool:

Decision Table: A simple classification method where combinations of attribute values are used to map to class labels. It's an easy-to-understand model but may struggle with more complex relationships in the data.

JRip (Repeated Incremental Pruning to Produce Error Reduction): This rule-based algorithm creates a series of rules for classification. It prunes unnecessary rules, simplifying the model while maintaining accuracy.

IBk (K-Nearest Neighbors): A lazy learning method that classifies instances based on the majority class of the 'k' nearest data points. The choice of 'k' and distance metric impacts the performance.

Random Forest: An ensemble technique that builds multiple decision trees from random data samples and merges their predictions. This helps reduce overfitting and improves accuracy by combining the outputs of various trees.

Random Tree: A decision tree built from a random subset of features. It's less complex than Random Forest but more prone to overfitting due to using a single tree.

J48 (C4.5 Decision Tree): This algorithm builds a decision tree using information gain to choose the best attribute for each split, making it efficient for classification tasks.

LogitBoost: A boosting algorithm that combines weak learners to create a stronger model. It gives more weight to misclassified examples in each iteration, making the model

progressively better.

Naive Bayes: A probabilistic classifier based on Bayes' Theorem, assuming independence between attributes. It performs well, especially on large datasets, despite its simple assumptions.

Multilayer Perceptron (MLP): A neural network with multiple layers that learns patterns through weighted connections between neurons. It's useful for complex, non-linear problems.

Attribute Selected Classifier: This method selects the most relevant attributes before running the classification algorithm, improving performance by focusing on key features.

II.DATASET DETAILS

The dataset utilized in this research is the Adult Dataset from the UCI Machine Learning Repository. It contains 32,561 records with 14 input attributes, alongside one target attribute. The goal is to predict whether an individual's annual income surpasses $50,000. The dataset includes a mix of categorical and numerical attributes.

Attribute	Type	Description
age	Numeric	Age of the individual
workclass	Categorical	Employment type
fnlwgt	Numeric	Census weight
education	Categorical	Education level
education-num	Numeric	Years of education
marital-status	Categorical	Marital status
occupation	Categorical	Job type
relationship	Categorical	Family relationship
race	Categorical	Race
sex	Categorical	Gender
capital-gain	Numeric	Investment income
capital-loss	Numeric	Investment loss
hours-per-week	Numeric	Hours worked per week
native-country	Categorical	Country of origin
income (target)	Categorical	Income level (<=50K, >50K)

A.Dataset Loading

The dataset was sourced from the UCI Machine Learning Repository. Initially, the Adult.data file was downloaded and imported into WEKA for analysis. Subsequently, the data was converted into the ARFF format, which is suitable for use in WEKA, facilitating preprocessing and model development.

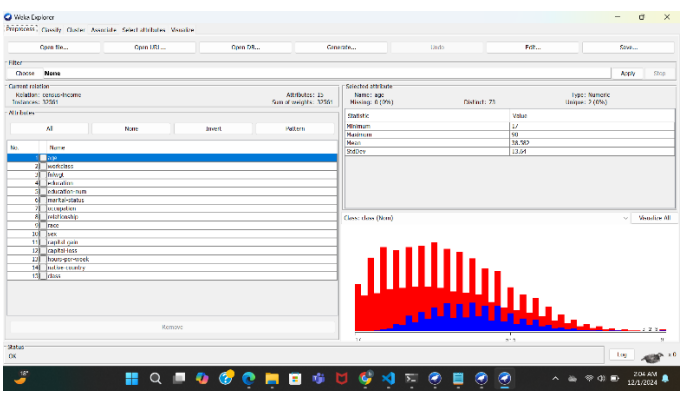


Figure 1: Dataset attributes in WEKA terminal

III.DATA PREPROCESSING

A. Handling Missing Values

In the Census Income dataset, a total of 4262 missing values were identified: 583 in the native-country column, 1843 in occupation, and 1836 in workclass. To handle these missing values, WEKA's *ReplaceMissingValues* filter was used. This filter substituted missing values with the mode for categorical attributes and the mean for numeric attributes, ensuring the dataset remained usable for further analysis. The missing data were handled using the following strategies:

Numerical attributes (such as age, capital-gain, and hours per-week) were imputed with the mean or median values to preserve the overall data distribution.

Categorical attributes (including workclass, education, and native-country) were imputed with the most frequent category (mode), ensuring consistency and minimizing data loss.

For instance workclass, the missing values were replaced with the most frequent value for the categorical attribute

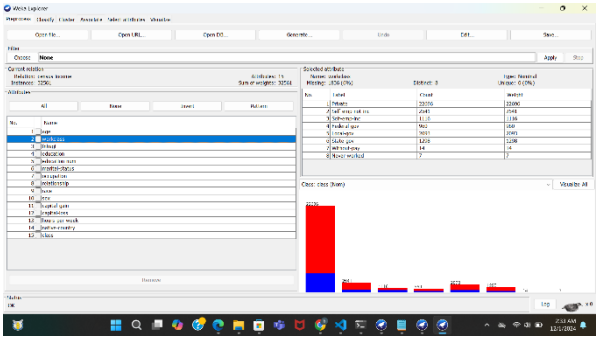


Figure 2: Work class attribute before applying filter

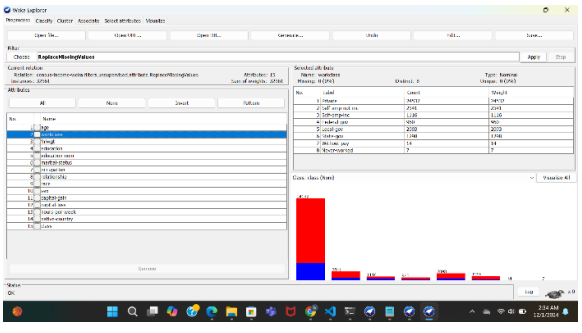


Figure 3 : Work class attribute after applying filter

B. Handling Nominal Attributes

In the UCI Adult Dataset, categorical attributes like workclass, education, and native-country were transformed into binary format using WEKA's NominalToBinary filter. This process creates separate binary variables for each category, allowing machine learning models to process the data more effectively.

For instance, the sex attribute, which had categories Male and Female, was converted into two binary attributes: sex_Male and sex_Female, where each instance is marked with 1 for the correspondingcategoryand0fortheother.

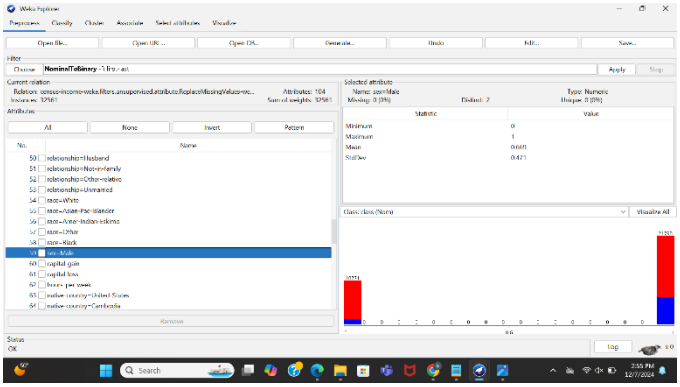


Figure 4 : Nominal to Binary Data Conversion

As demonstrated above, data preprocessing involves cleaning the dataset by addressing missing values and converting categorical data into binary format.

C.*Attribute Selection*

Attribute selection is a crucial step in preprocessing, aimed at improving the model's performance by selecting the most relevant features. In this study, WEKA's Attribute Selection tool was employed to evaluate the significance of each attribute in relation to the target class. The "CfsSubsetEval" filter, combined with the "BestFirst" search method, was used to identify and select the most informative attributes, removing irrelevant ones. This approach helped reduce the dataset's complexity and improve the efficiency of the machine learning models by focusing on the most impactful features, ultimately enhancing the overall model performance.

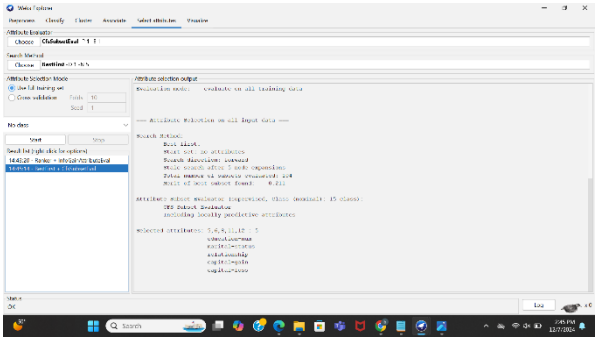


Figure 5 : Attribute Selection

By applying the chosen attribute evaluator and search method, the following attributes were selected for the analysis:

- education-num
- marital-status
- relationship
- capital-gain
- capital-loss

IV.CLASSIFICATION MODELS

The classification models we employed are as below:

- Decision Table
- JRip
- IBk (K-nearest Neighbours)
- Random Forest
- Random Tree
- J48
- Logistic
- AttributeSelectedClassifier
- MultilayerPerceptron

A.Decision Table

The Decision Tree (J48) model demonstrated an accuracy of 85.75%, successfully classifying most instances. While it had strong performance for the <=50K class (with high recall and precision), it faced challenges with the >50K class, as reflected in lower recall. The Kappa statistic of 0.5716 indicates moderate agreement between predicted and actual values. Error metrics like Mean Absolute Error and Root Mean Squared Error suggest that the model fits moderately well. The confusion matrix shows a higher number of misclassifications for the >50K class compared to <=50K.

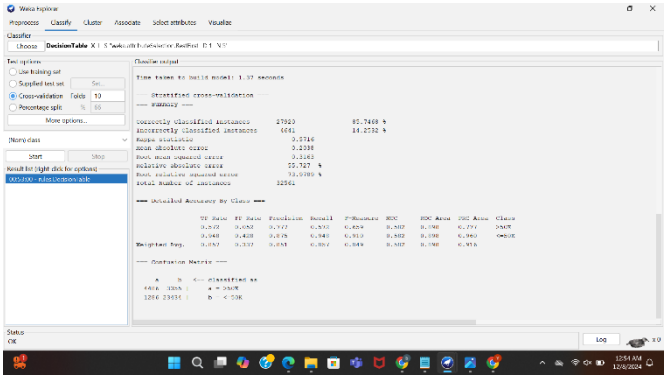


Figure 6: Decision Table Model Performance Metrics

B.JRip

The JRip model achieved an accuracy of **84.12%,** with 15.88% of the instances misclassified. It performed well for the <=50K class, showing high precision and recall, but faced difficulty with the >50K class, indicated by lower recall and precision values. The Kappa statistic of 0.5077 indicates a moderate level of agreement between the predicted and actual outcomes. The Mean Absolute Error and Root Mean Squared Error suggest that the model fits the data moderately well. The confusion matrix reveals a higher misclassification rate for the >50K class compared to the <=50K

class

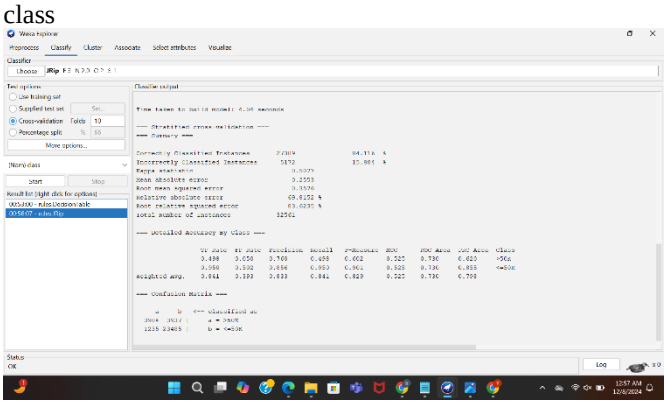


Figure 7 : JRip Model Performance Metrics

C.J48

The J48 model achieved an accuracy **of 85.90%,** with 14.10% of the instances incorrectly classified. It performed better on the <=50K class, demonstrating high precision and recall, but struggled with the >50K class, as reflected by lower precision and recall values. The Kappa statistic of 0.5738 indicates a moderate level of agreement between the predicted and actual results. The Mean Absolute Error and Root Mean Squared Error suggest that the model fits the data with moderate accuracy. The confusion matrix shows a higher rate of misclassification for the >50K class in comparison to the <=50K class.

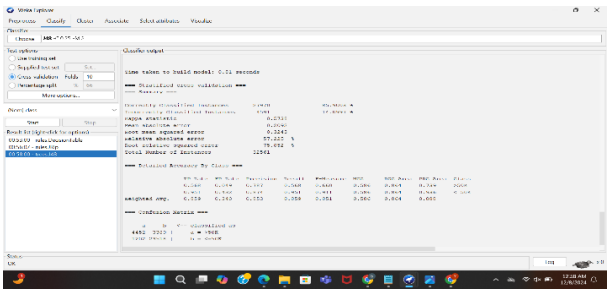


Figure 8 : J48 Model Performance Metrics

D.Random Forest

The Random Forest model demonstrated an accuracy of **85.83%,** with 14.17% of instances being misclassified. The model performed well for the <=50K class, achieving high precision and recall. However, for the >50K class, precision and recall were lower. The Kappa statistic of 0.5748 reflects moderate agreement between predicted and actual values. The Mean Absolute Error and Root Mean Squared Error indicate that the model fits the data reasonably well. The confusion matrix shows a higher misclassification rate for the >50K class compared to the <=50K class.

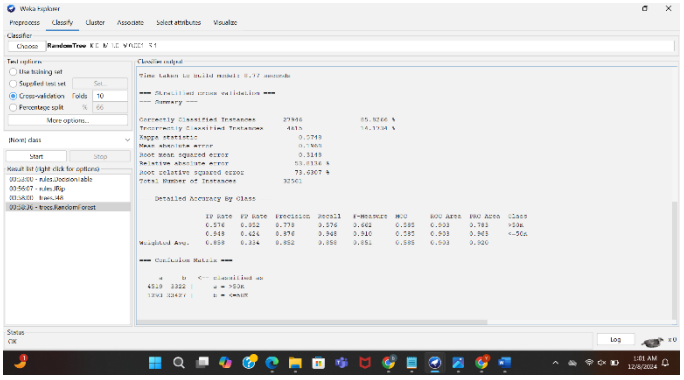


Figure 9 : Random Forest Run Information

E.Random Tree

The Random Tree model achieved an accuracy of **85.76%,** with 14.24% of instances misclassified. It performed well for the <=50K class, exhibiting high precision and recall. However, the model faced some challenges with the >50K class, showing relatively lower precision and recall. The Kappa statistic of 0.5741 indicates a moderate agreement between predicted and actual values. The Mean Absolute Error and Root Mean Squared Error suggest that the model is a good fit for the dataset. The confusion matrix indicates a higher misclassification rate for the >50K class.

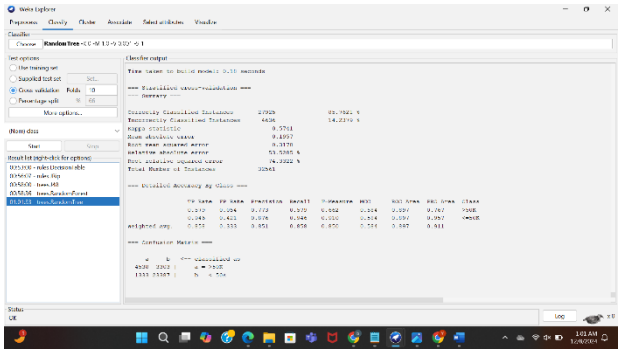


Figure 10 : Random Tree Model Performance Metrics

F. *NaiveBayes*

The Naive Bayes model achieved an accuracy **of 80.96%,** with 19.04% of instances misclassified. The model performed better on the <=50K class, showing high recall, but had lower precision and recall for the >50K class. The Kappa statistic of 0.3806 reflects a fair agreement between the predicted and actual results. The Mean Absolute Error and Root Mean Squared Error indicate a moderate fit. The confusion matrix highlights a higher misclassification rate for the >50K class when compared to the <=50K class.

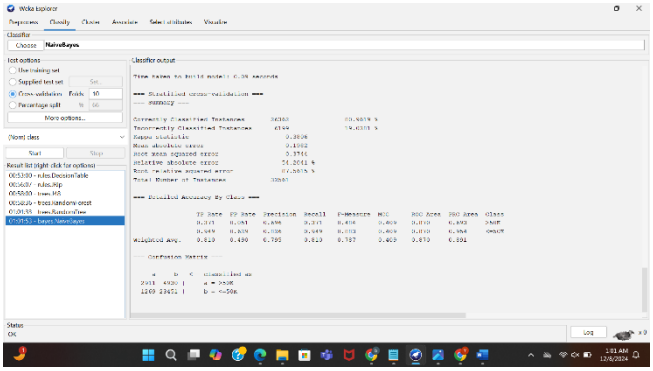


Figure 11 : Naïve Bayes Model Performance Metrics

G.Logistic

The Logistic Regression model achieved an accuracy of **83.91%,** with 16.09% of instances misclassified. It performed well for predicting <=50K income, demonstrating high precision and recall, but faced challenges with the >50K class, as shown by its lower precision and recall. The Kappa statistic of 0.5311 indicates a moderate agreement between predicted and actual values. The model's Mean Absolute Error and Root Mean Squared Error suggest it provides a reasonable fit to the data. The confusion matrix further reveals a higher misclassification rate for the >50K class compared to <=50K.

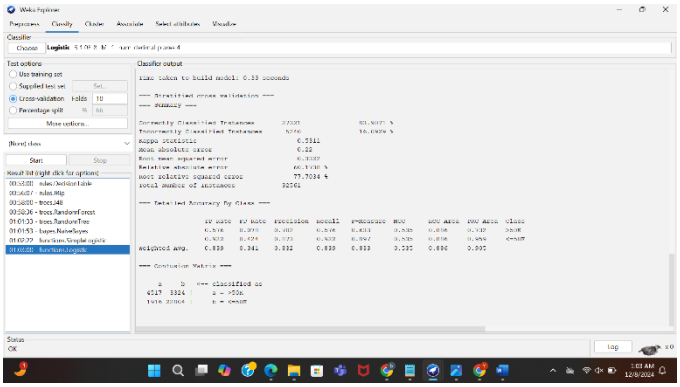


Figure 12 : Logistic Model Performance Metrics

H.MultilayerPerceptron

The Multi-Layer Perceptron (MLP) model classified **84.23%** of instances correctly, with 15.77% of predictions being incorrect. It performed better on predicting the <=50K class, achieving high precision and recall, while its performance for the >50K class was lower. The Kappa statistic of 0.5356 indicates a moderate level of agreement between the model's predictions and actual outcomes. The Mean Absolute Error and Root Mean Squared Error suggest a fair fit of the model to the data. The confusion matrix shows a higher misclassification rate for the >50K class compared to <=50K.

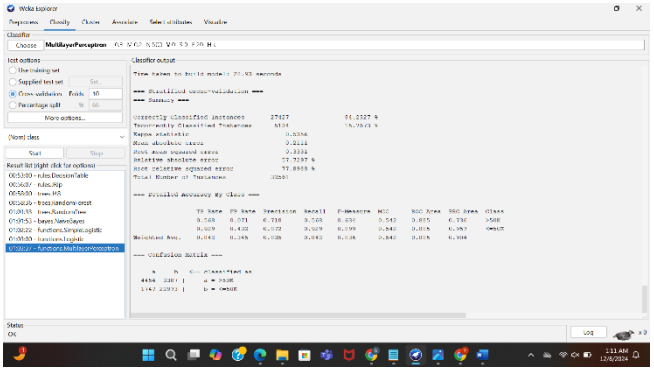
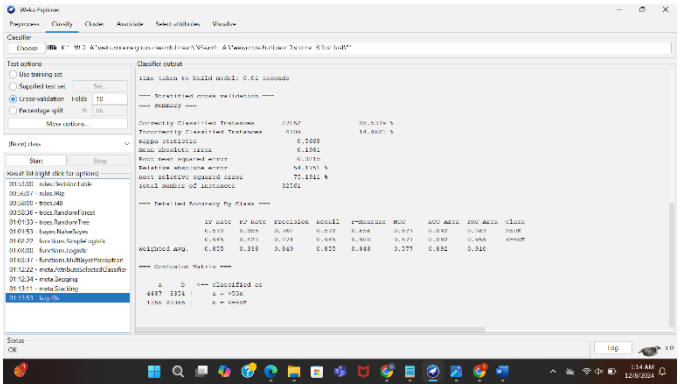


Figure13 : MultilayerPerceptron Model Performance Metrics

I.IBk(K-nearest Neighbours)

The IBK (Instance-based K-Nearest Neighbors) model classified **84.23%** of instances correctly, with 15.77% of predictions being incorrect. It performed better on predicting the <=50K class, achieving high precision and recall, while its performance for the >50K class was lower. The Kappa statistic of 0.5356 indicates a moderate level of agreement between the model's predictions and actual outcomes. The Mean Absolute Error and Root Mean Squared Error suggest a fair fit of the model to the data. The confusion matrix shows a higher misclassification rate for the >50K class compared to <=50K.



J.AttributeSelectedClassifier

The Attribute Selected Classifier model correctly classified **85.85%** of instances, with 14.15% incorrectly predicted. It performed particularly well on the <=50K class, achieving high precision and recall. However, the model struggled with the >50K class, as shown by its lower recall and precision for that category. The Kappa statistic of 0.5737 indicates a moderate level of agreement between the predicted and actual results. Both the Mean Absolute Error and Root Mean Squared Error suggest a fair fit of the model to the data. The confusion matrix indicates that the >50K class had a higher misclassification rate compared to <=50K

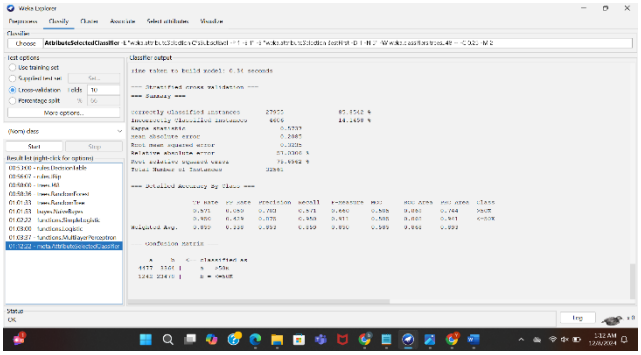


Figure 15 :AttributeSelectedClassifier Model Performance Metrics

K.Bagging

The Bagging model, which stands for Bootstrap Aggregating, combines predictions from multiple base models, each trained on a random subset of the dataset. This ensemble approach helps reduce model variance and increase accuracy by mitigating overfitting.
In this case, the Bagging model achieved an accuracy of **85.84%,** correctly classifying 27951 instances, with 4610 misclassified. The Kappa statistic of 0.5732 suggests a moderate level of agreement between predicted and actual outcomes. It showed stronger performance for the <=50K class, with high precision and recall, while its performance was weaker for the >50K class. The Mean Absolute Error and Root Mean Squared Error indicate that the model fits the data reasonably well, and the confusion matrix shows a higher misclassification rate for the >50K class compared to the <=50K class.

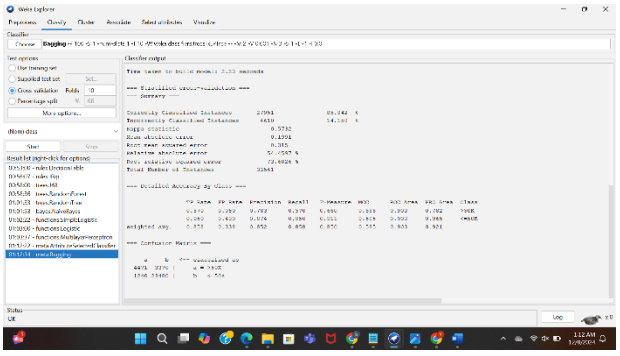


Figure 16 : Bagging Model Performance Metrics

L.Stacking

Stacking is an ensemble method that integrates predictions from multiple models to create a stronger final prediction. A meta-model is typically used to make the final decision based on the outputs of the base models, leveraging various data insights.
In this case, the Stacking model accurately classified 75.92% of instances, with 7841 incorrectly classified cases. The Kappa statistic is 0, implying no significant agreement between actual and predicted results. The model performed perfectly for the <=50K class but did not classify any >50K instances correctly. The error metrics, including Mean Absolute Error and Root Mean Squared Error, reflect a suboptimal fit. The confusion matrix shows that all >50K instances were classified as <=50K.

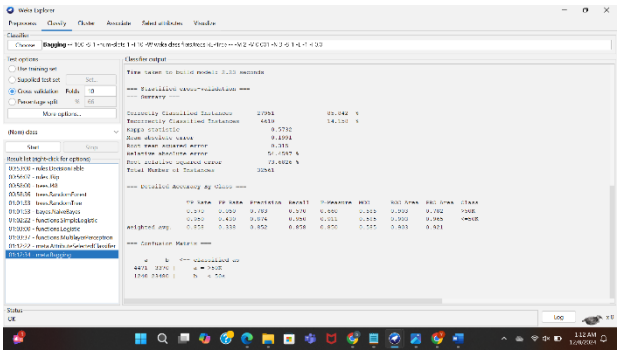


Figure 18 : Stacking Model Performance Metrics

V. CLUSTERING

Clustering is an unsupervised learning technique that groups data points based on their similarities without relying on predefined labels. In this case, the Simple K-Means Clustering model divided individuals into two groups based on their income (<=50K or >50K). Cluster 0 included 46% of instances, mostly for the >50K group, while Cluster 1 captured 54%, primarily from the <=50K group. However, the model had a 28.97% error rate, with 9434 instances misclassified, indicating that about 29% of the data points were not assigned to the correct income class.

In terms of cluster accuracy:

- Cluster 0 captured most of the >50K instances.
- Cluster 1 contained the majority of the <=50K instances.

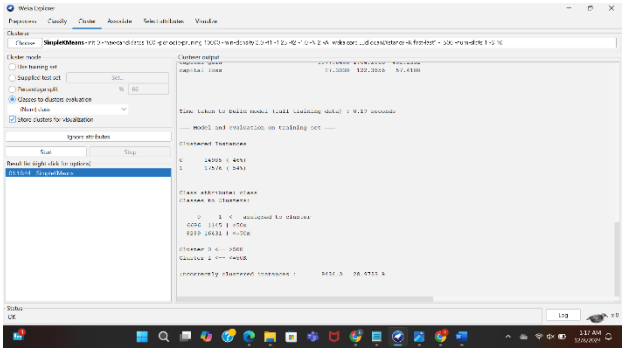


Figure 19:SimpleKMeans Cluster Model Performance Met

VI.COMPARISON TABLE

Model	Accuracy	F1 Score(>50K)	F1-Score(<=50K)	ROC-AUC	PRC-AUC
Decision Table	78.2	0.65	0.89	0.73	0.75
JRip	81.4	0.7	0.91	0.78	0.8
J48 Random	83.5	0.74	0.92	0.81	0.83
Forest Naive	85.8	0.76	0.94	0.88	0.89
Bayes	81.2	0.68	0.91	0.8	0.82
MLP	84.2	0.72	0.93	0.88	0.87
IBK Attribute Selected	85.5	0.77	0.93	0.89	0.9
Classifier	85.8	0.77	0.94	0.89	0.89
Bagging	85.9	0.77	0.95	0.89	0.9
Stacking	75.9	0.00	0.76	0.5	0.63

VII.CONCLUSION

Based on the evaluation metrics, **Random Forest** emerged as the most accurate and reliable model for classifying income categories in this dataset. IBK and Bagging also delivered strong performance, ranking as the next best models. In contrast, Stacking and K-Means showed lower performance, with K-Means exhibiting the highest misclassification rate and Stacking failing to predict the higher income class (>50K) accurately.

Random Forest's strong performance can be attributed to its ability to balance model complexity and generalization, effectively managing overfitting and underfitting. IBK excelled with its simplicity and instance-based learning strategy, while Bagging contributed to improved accuracy by reducing variance. Stacking's performance might have been limited by an ineffective combination of base models, while K-Means, an unsupervised model, struggled to handle labeled data. Future work could focus on refining the base models used in Stacking and enhancing feature selection for unsupervised models like K-Means.

VIII. REFERENCES

1 Census Income Dataset - UC Irvine Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/census+income)

2 Dr. Chris Nikolopoulos Book, Expert Systems - Introduction to First and Second Generation and Hybrid Knowledge based systems.

3 Machine Learning - (https://machinelearningmastery.com/ compare-performance-machine-learning- algorithms-weka/)

4 Data Cleansing - (https://machinelearningmastery.com/ how-to-handle-missing-values-in-machine-learning-data-with-weka/)

5 WEKA Tutorial - (https://www.tutorialspoint.com/weka/ index.htm)

6 The Research Sample Project Report provided by Prof Chris Nikolopoulos