Predicting Annual Income of Individuals using Classification Techniques

Soumen Sikder Shuvo
Department of ECE
Stevens Institute of Technology
Hoboken, NJ, USA
sshuvo@stevens.edu

Janmejay Mohanty
Department of Computer Science
Stevens Institute of Technology
Hoboken, NJ, USA
jmohanty@stevens.edu

Darsh Patel
Department of Computer Science
Stevens Institute of Technology
Hoboken, NJ, USA
dpate154@stevens.edu

***Abstract*—This project aims to predict the annual income of individuals using classification techniques. We used the Adult Dataset, which contains demographic and socio-economic features such as age, education, occupation, work class, marital status, race, gender, and income. We applied various classification techniques such as logistic regression, decision tree, random forest, SVM, neural network, and ensemble methods. We experimented with hyperparameter tuning and feature engineering to improve the performance of our models. The major contribution of this work is to provide insights into the important features that impact a person's annual income and compare the performance of various classification techniques.**

I. Introduction

The ability to accurately predict a person's annual income is an important task that has numerous real-world applications, such as determining loan eligibility, assessing creditworthiness, and evaluating job candidates. In this project, we aim to predict the annual income of individuals using various classification techniques. We will use the Adult dataset, which contains demographic and socio-economic features such as age, education, occupation, work class, marital status, race, gender, and income, and has been widely used in research and machine learning.

Our approach involves developing binary classification models that predict whether an individual's income is above or below $50K per year using different classification techniques, including Logistic Regression, Decision Tree, Random Forest, Support Vector Machines, and Neural Networks. We will also experiment with ensemble models that combine the predictions of multiple classification algorithms and hyperparameter tuning to improve overall performance. Additionally, we will extract new features from the dataset and use these features to train a classification model.

The outcomes of this project will help us to evaluate the effectiveness of different classification techniques and ensemble models for predicting annual income, and identify which techniques are most effective. The results may also provide insights into the factors that contribute to higher incomes and inform policies and decision-making in various fields.

II. Related Work

Several studies have investigated classification techniques and feature selection methods in the same dataset. These studies provide insights into the strengths and weaknesses of different algorithms and emphasize the importance of data preprocessing for successful prediction.

One study compares classification accuracy provided by different classification algorithms, including Naïve Bayesian, Random Forest, Zero R, and K-Star on the census dataset [1]. Here the authors present a comprehensive review of these algorithms on the dataset and discuss their strengths and weaknesses in terms of classification accuracy. It also provides an introduction to data mining and classification techniques.

Another study focuses on feature selection techniques in data mining and compares the effectiveness of Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Genetic Algorithm in terms of the number of selected feature subsets [2]. These techniques were applied to K-Means clustering on the adult dataset, and the study was carried out in terms of accuracy and execution time. The paper emphasizes the importance of feature selection in data preprocessing for successful data mining.

Finally, a study uses machine learning and data mining to address economic inequality in the United States by predicting whether a person's income falls into a category greater than 50K Dollars or less equal to 50K Dollars [3]. This achieved an accuracy of 88.16% using the Gradient Boosting Classifier Model and emphasizes the importance of using machine learning techniques to tackle this issue.

III. Our Solution

In this section we introduced some well known Machine Learning Algorithms to our dataset and observed the accuracy of those algorithms.

A. Description of Dataset

The Adult Income dataset is a publicly available dataset from the UCI Machine Learning Repository, and it contains information about individuals from the 1994 United States Census. The dataset consists of 48842 rows and 15 columns. In Table no I we can see all the features and their types.

In figure 5, we can see that some features has significant influence on the target values, such as in marital-status or relationship. On the other hand, we can see that the native-country feature do not have any significant influence on target value we need to classify. These gives us an intuitive direction on which features are better to use.

TABLE I. FEATURES NAMES AND TYPE

Categorical Features	Continuous Features
workclass	age
education	fnlwgt
marital-status	education-num
occupation	capital-gain
realtionship	capital-loss
race	hour-per-week
gender	
native-country	

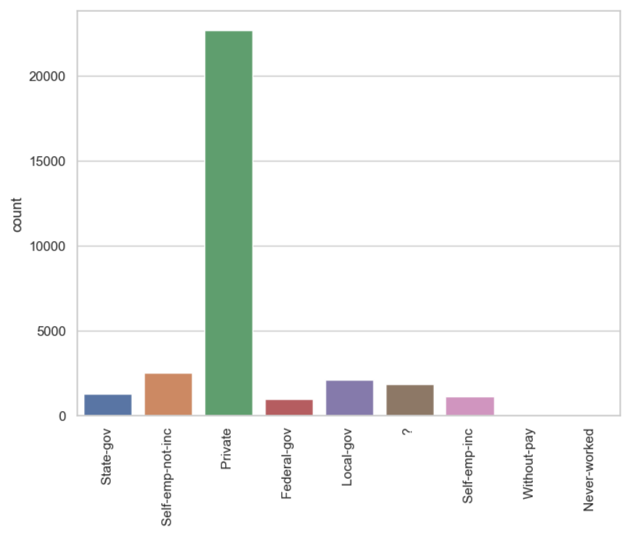


Fig. 1. The number of incidents for workclass feature

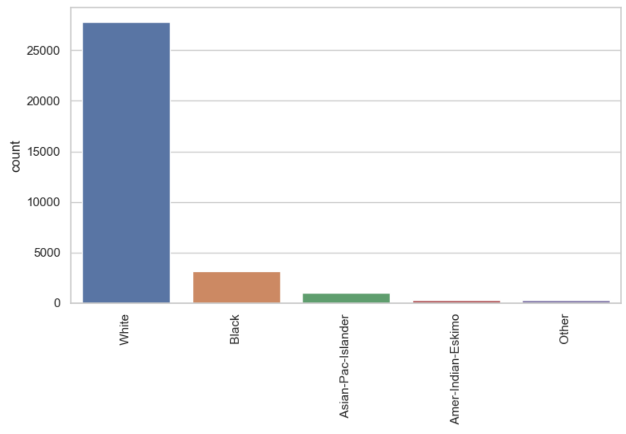


Fig. 2. The number of incidents for race feature

The goal of this dataset is to predict whether an individual's income level is above or below $50,000 based on their demographic and socioeconomic features. It is a commonly used dataset for tasks such as classification, data analysis, and machine learning modeling.

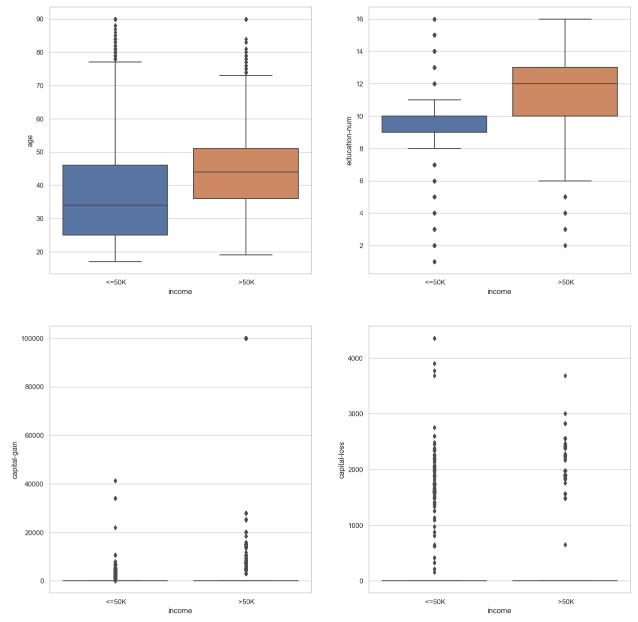
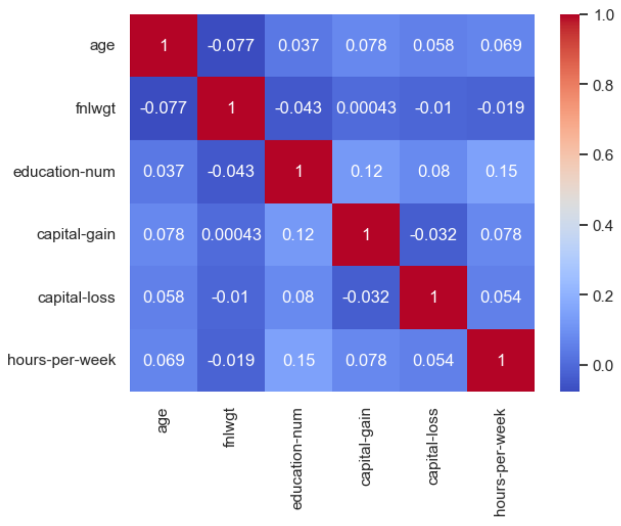


Fig. 3. Relationship between features and target



	age	fnlwgt	education-num	capital-gain	capital-loss	hours-per-week
age	1	-0.077	0.037	0.078	0.058	0.069
fnlwgt	-0.077	1	-0.043	0.00043	-0.01	-0.019
education-num	0.037	-0.043	1	0.12	0.08	0.15
capital-gain	0.078	0.00043	0.12	1	-0.032	0.078
capital-loss	0.058	-0.01	0.08	-0.032	1	0.054
hours-per-week	0.069	-0.019	0.15	0.078	0.054	1

Fig. 4. Heatmap of Correlation between features

B. Machine Learning Algorithms and Implementation Details

1) Decision Tree: A decision tree is a type of algorithm used for supervised learning in which the data is recursively split into subsets based on the most informative features for predicting the target variable. It can be a robust algorithm for binary classification tasks, especially when the data has a complex structure or there are many features to consider. It can handle both categorical and continuous data, and it can also be used to identify important features in the data. So, naturally, it becomes a very convenient model that we can apply to classify our data.

We used the scikit-learn library to build a decision tree model. We split our dataset into training and testing sets, with 80% of the data used for training and 20% for testing. We experimented with different hyperparameters of the decision

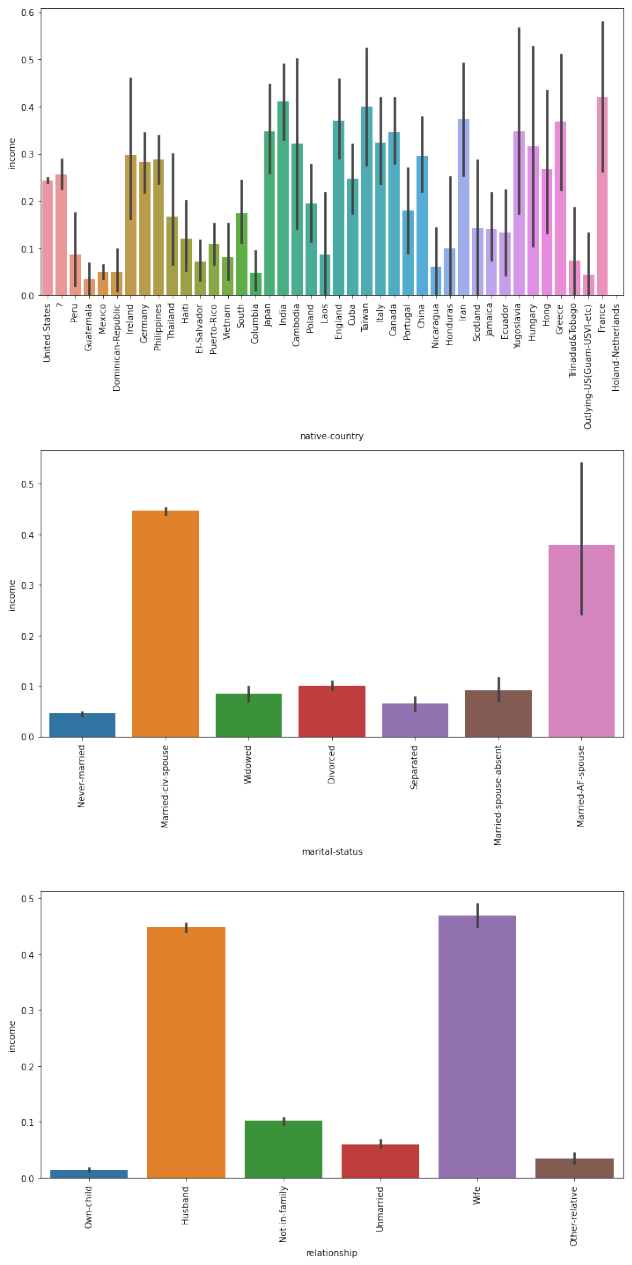


Fig. 5. Probabilty of Target Value at Different Features

tree algorithm such as the maximum depth of the tree, the minimum number of samples required to split a node, and the minimum number of samples required to be at a leaf node. After tuning these parameters, we found that the optimal parameters were a maximum depth of 9, a minimum number of samples required to split a node of 2, and a minimum number of samples required to be at a leaf node of 4.

Our decision tree model achieved an accuracy of 85.6% and an f1 score of 68.7% on the test set, which is a promising result. We also visualized the decision tree using the graphviz library to gain insights into the most important features that contributed to the prediction of the target variable. In Table II, we have a comparison of accuracy on different depth levels. We can see the confusion matrix of the best f1-scored parameters in Table III.

TABLE II. ACCURACY ON DIFFERENT MAX DEPTH

max_depth	5	6	7	8	9
accuracy	85.17%	85.33%	85.56%	85.58%	85.65%

TABLE III. CONFUSION MATRIX OF DECISION TREE

	Predicted 0	Predicted 1
True 0	6826	595
True 1	809	1539

2) Random Forest: Random Forest is a type of ensemble learning algorithm that is an extension of the decision tree algorithm. It builds multiple decision trees and combines their results to make a final prediction. Each decision tree in a random forest is trained on a random subset of the training data and a random subset of the features. This helps to reduce overfitting and increase the generalization performance of the algorithm.

Random Forest is widely used for binary classification tasks because it can handle both categorical and continuous data and can automatically identify important features in the data. It also has a lower risk of overfitting compared to a single decision tree because it averages the predictions of multiple trees.

We used same data format used in Decision Tree and Random Forest model from scikit-learn. We used Grid Search of scikit-learn to find the best parameters for Random Forest model. The best result was achieved using 300 estimators and 6 maximum features. Here we got an accuracy of 85.59% and an f1 score of 68.8%. In Table IV, we can see the confusion matrix of Random Forest Model and it is quite an improvement from the Decision Tree.

3) SVM: SVM or Support Vector Machine is a type of supervised learning algorithm used for both classification and regression tasks. It works by finding a hyperplane in a high-dimensional space that best separates the data into different classes.

SVM is often used for binary classification tasks because it naturally finds the best boundary between the two classes. It is also effective when there are many features, both categorical and continuous, and when the decision boundary is complex. However, sometimes it can be sensitive to the choice of hyper-parameters, such as the kernel function and the regularization parameter, and can be computationally expensive for large datasets.

We processed our data and then used SVM with rbf kernel type. We got high accuracy in SVM but the F1 score was not

TABLE IV. CONFUSION MATRIX OF RANDOM FOREST

	Predicted 0	Predicted 1
True 0	6982	492
True 1	815	1480

TABLE V. CONFUSION MATRIX OF SVM

	Predicted 0	Predicted 1
True 0	5688	1733
True 1	1739	609

TABLE VI. TEST ACCURACY OF NAIVE BAYES

var_smoothing	1e-9	1e-8	1e-7
Accuracy	79.39%	79.34%	79.37%
var_smoothing	1e-6	1e-5	1e-4
Accuracy	79.4%	80.3%	79.67%

upto the mark. We can observe the confusion matrix of SVM in Table V.

4) Naive Bayes: Naive Bayes is a probabilistic machine learning algorithm that is commonly used for classification tasks. It is based on Bayes' theorem, which is a fundamental principle in probability theory. Naive Bayes assumes that the features are independent of each other, which means that the presence or absence of one feature does not affect the probability of another feature.

Naive Bayes works by calculating the probability of a class given a set of features. It uses Bayes' theorem to calculate the posterior probability of each class, which is the probability of the class given the features. The class with the highest posterior probability is selected as the predicted class.

$$P(class/features) = \frac{P(class) * P(features/class)}{P(features)}$$

The Adult dataset is a classification problem, where the goal is to predict whether an individual earns more or less than $50,000 per year based on their demographic and other attributes. Naive Bayes is a good choice for this dataset because it is computationally efficient, can handle large datasets with many features, and is resistant to overfitting.

We performed a grid search to find the best hyperparameters for our model. We experimented with different values of the hyperparameter *"var_smoothing"*, which controls the amount of smoothing applied to the probability estimates. After tuning this parameter, we found that the optimal value was 1e-05.

Our Naive Bayes model achieved an accuracy of 80.9% on the test set, which is a good result. Although it is not as accurate as the decision tree model, it is a simple and efficient algorithm that works well for high-dimensional datasets. We did not perform feature selection or feature engineering, which could potentially improve the accuracy of the model. In conclusion, the Naive Bayes algorithm with the optimal hyperparameter achieved a decent accuracy on the test set. It is a useful algorithm for classification tasks, especially for high-dimensional datasets. We can see the Test Accuracy Table VI and Test Score Graph 6

5) ANN: An artificial neural network (ANN) is a type of machine learning algorithm that works by learning a set of weights that determine the strength of connections between

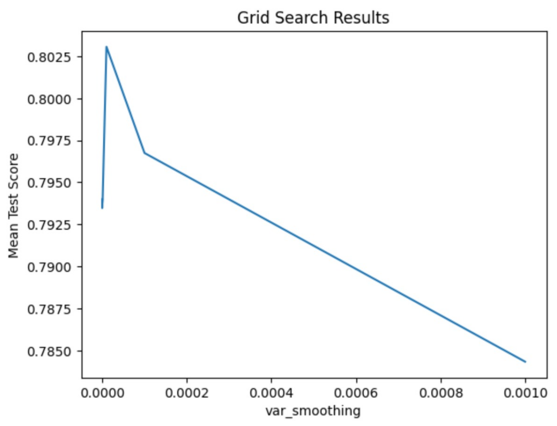


Fig. 6. Test score of Hyperparameter Tuning on Naive Bayes

nodes. During training, the ANN adjusts these weights to minimize the difference between predicted and actual outputs which is called backpropagation. Backpropagation computes the gradient of the loss function with respect to the weights and biases and uses the gradients to update the weights and biases. Optimization algorithms such as stochastic gradient descent are commonly used for this purpose. Architecture of ANN 7

In addition to weight adjustment, ANNs use activation functions to introduce nonlinearity in the node's output. An activation function determines a node's output based on the weighted sum of its inputs. The most commonly used activation functions in ANNs include the sigmoid function, ReLU (Rectified Linear Unit), and softmax function and binary crossentropy loss funtion.

$$CrossentropyLoss = -(y\log(p) + (1-y)\log(1-p))$$
$$Sigmoid = \sigma(z) = \frac{1}{1+e^{-z}}$$
$$Relu(z) = max(0, z)$$

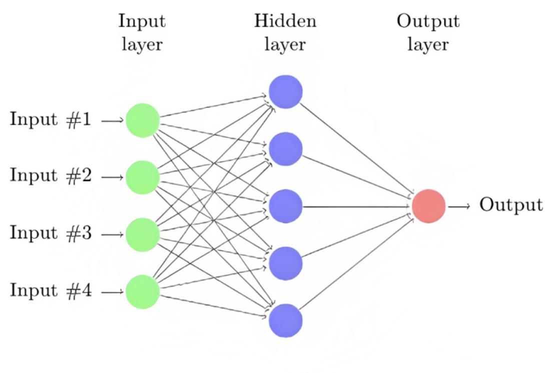


Fig. 7. Architecture of Neural network

The Adult dataset is aimed at solving a classification problem, which involves determining if an individual earns more or less than $50,000 annually based on their demographic and other attributes. ANNs are a suitable choice for this dataset because of their ability to recognize intricate patterns in the data and their capability to handle a broad range of data types. ANNs have demonstrated high accuracy when used on the Adult dataset, especially when used together with feature engineering methods such as normalization and one-hot encoding.

Using the Keras library with TensorFlow backend. We built a neural network model with two hidden layers of 64 and 32 nodes and an output layer with a sigmoid activation function for binary classification. We trained the model with two different optimizers: Stochastic Gradient Descent (SGD) and Adaptive Moment Estimation (Adam).

After training the model with both optimizers, we evaluated the performance using accuracy, precision, recall, and F1-score metrics. The model trained with Adam optimizer achieved an accuracy of 86.1%, On the other hand, the model trained with SGD optimizer achieved an accuracy of 78.4%.

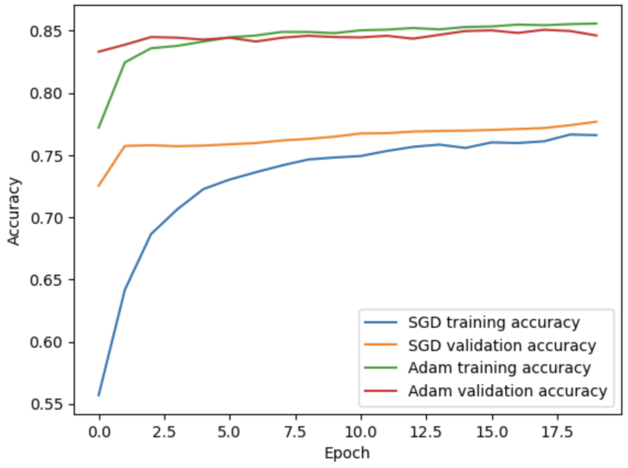


Fig. 8. Comparison of Accuracy of SGD and ADAM

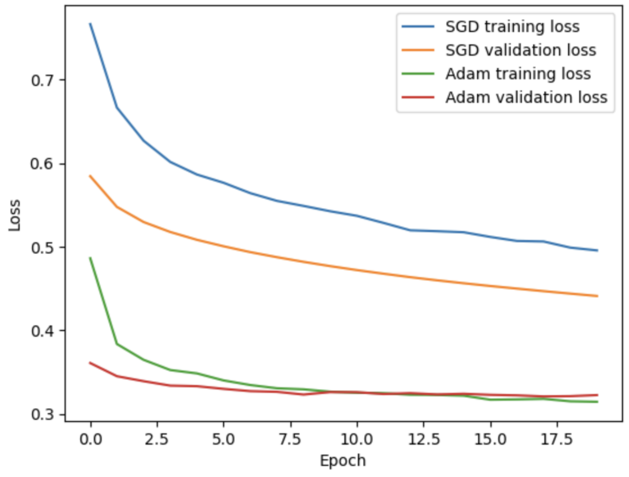


Fig. 9. Comparison of Loss of SGD and ADAM

We also performed Hyperparameter tuning using gridsearch and parameters were set to batch size: 32,64,128, epochs: 10,20,30, optimizer Adam with 0.001 and 0.01 learning rate and dropout rate set to 0.2,0.3,0.5. Fitting 3 folds for each of 54 candidates, totalling 162 fits. The best performance we got was for batch size': 128, 'dropout rate': 0.3, 'epochs': 20, with

TABLE VII. EVALUATION METRICS

	SGD	Adam
Accuracy	0.795	0.860
Precision	0.771	0.759
Recall	0.177	0.594
F1-Score	0.288	0.666

Accuracy of 85.36%. We can see the Evaluation Metrics Table VII, and Comparison Figure 8, 9 and Test Figure 10

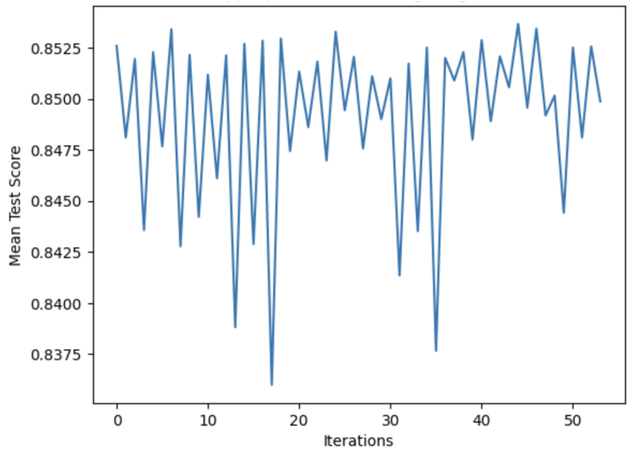


Fig. 10. Test Score through iterations in Hyperparameter Tuning

6) Logistic Regression: The logistic model is a statistical model that models the probability of an event taking place by having the log-odds for the event be a linear combination of one or more independent variables. In this type of regression analysis, logistic regression estimates the parameters of a logistic model. Logistic function is responsible for converting log-odds to probability. Logit is unit of measurement for log-odds scale.

Logistic Regression is widely used in various fields such as machine learning, medical fields, and social sciences.

General Logistic Regression expression,

$$P(y=1|x) = \frac{1}{1+e^{-(w^T x)}}$$

We use logistic regression for this given dataset because it helps us to identify the data anomalies. Also, it determines the relationship between two or more data factors which is our aim to get for this dataset.

We used the sklearn library to build a logistic regression model. We had used a label encoder to convert the data frame values into categorical values for logistic regression. We had taken features with respect to target income for predicting probability of income classifying as . We further split the datasets into 20:80 ratio, i.e., 20% test data and 80% train data. We used default parameters for our model to get an accuracy around 79% which seems to be low. Also, we got a classification report for our model shown below.

Furthermore, we have performed hyperparameter tuning and ended up getting best performance for regularization 'L1',

TABLE VIII. CONFUSION MATRIX OF LOGISTIC REGRESSION

	Predicted 0	Predicted 1
True 0	4658	284
True 1	849	722

TABLE IX. CLASSIFICATION REPORT OF LOGISTIC REGRESSION

	Precision	Recall	F1-score	Support
0	0.85	0.94	0.89	4942
1	0.72	0.46	0.56	1571
Accuracy			0.83	6513
Macro avg	0.78	0.70	0.73	6513
weighted avg	0.81	0.83	0.81	6513

optimizer 'liblinear' and inverse regularization strength '10' with an Accuracy rate of 83%. We can see the Confusion Matrix Table VIII and Classification Report Table IX

7) Ensemble Modeling: Ensemble modelling is a technique in which various multiple models are created to predict an outcome, either by using various modeling algorithms or using different training datasets. For this project and given dataset we have use the different modeling algorithms approach. We have use three different algorithms for ensemble modeling which are Decision Tree, Random Forest, and Logistic Regression with the help of Voting Classifier.

We use ensemble modeling method over a single model for two main reasons for our project dataset. Firstly, an ensemble can make better predictions and achieve better performance than any single running model. Secondly, an ensemble reduces the spread or dispersion of the predictions and model performance.

Furthermore, we have performed hyperparameter tuning and ended up getting best performance for Decision Tree max depth '10', Logistic Regression inverse regularization strength '1.0' and Random Forest estimator '200' with an Accuracy rate of 86%. We can see the Confusion Matrix Table X and

TABLE X. CONFUSION MATRIX OF ENSEMBLE MODELING

	Predicted 0	Predicted 1
True 0	4716	226
True 1	687	884

TABLE XI. CLASSIFICATION REPORT OF ENSEMBLE MODELING

	Precision	Recall	F1-score	Support
0	0.87	0.95	0.91	4942
1	0.80	0.56	0.66	1571
Accuracy			0.86	6513
Macro avg	0.83	0.76	0.79	6513
weighted avg	0.85	0.86	0.85	6513

Classification Report XI

8) Feature Engineering: One-hot encoding is a technique used to convert categorical values into numerical values that can be used in machine learning algorithms. In one-hot encoding, each categorical value is replaced by a binary vector that has a 1 in the position corresponding to the value and 0s elsewhere.

Using one-hot encoding has several advantages for binary classification tasks. It helps the machine learning algorithm to better distinguish between the different categories and prevents the algorithm from assigning any inherent ordering or hierarchy to the categories. It also allows the algorithm to better capture any nonlinear relationships between the categorical variable and the target variable.

We use One-hot encoding and use the converted data in Decision Tree and Random Forest to observe the outputs. We get around 85.77% test accuracy for decision tree and 85.73% for Random Forest. We can see the confusion matrix in Table XII and XIII.

TABLE XII. CONFUSION MATRIX OF DECISION TREE WITH ONE-HOT ENCODING

	Predicted 0	Predicted 1
True 0	6999	422
True 1	968	1380

TABLE XIII. CONFUSION MATRIX OF RANDOM FOREST WITH ONE-HOT ENCODING

	Predicted 0	Predicted 1
True 0	6952	522
True 1	901	1394

We also used Discretization as feature engineering. It involves essentially taking a set of values of data and grouping sets of them together in some logical fashion. This could help prevent data from overfitting but comes at the cost of loss of granularity of data. As we previously seen in Figure 5, some features have more influence on target value and we can merge those to make new features. In our dataset, we merged marital-status and relationship features and used that data to evaluate Decision Tree and Random Forest. We got slight improvement in both models. The confusion matrices can be seen in Table XIV and XV.

IV. COMPARISON

After evaluating several Machine Learning Models, the results are presented in Table XVI. Here we can see that the ANN model with ADAM optimizer achieved the highest accuracy. However, Adult dataset has a highly imbalanced target

TABLE XIV. CONFUSION MATRIX OF DECISION TREE USING DISCRETIZATION

	Predicted 0	Predicted 1
True 0	6874	547
True 1	857	1491

TABLE XV. CONFUSION MATRIX OF RANDOM FOREST USING DISCRETIZATION

	Predicted 0	Predicted 1
True 0	6977	497
True 1	911	1384

value, which we can see in Figure 11. Looking into accuracy may lead to misleading results for such an imbalanced dataset. To obtain a better understanding of the models' performance on imbalanced data, F1-score can be a more useful metric. Comparing the F1 scores, we can see the Random Forest models have the highest probability of predicting the right value or income. SVM or support vector machine and Logistic Regression have comparatively bad results.

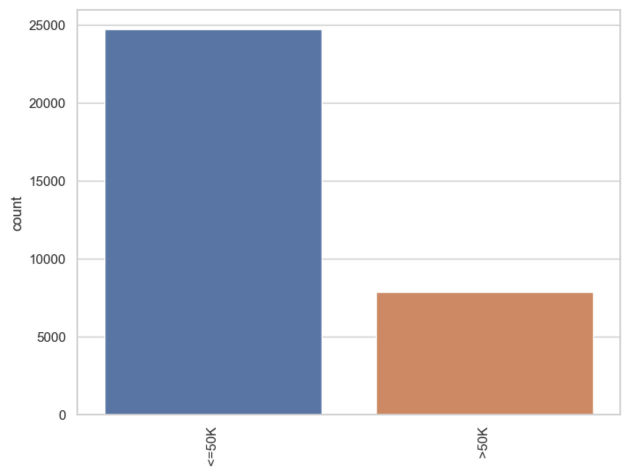


Fig. 11. Population Count of Target value

TABLE XVI. ACCURACY AND F-1 SCORE OF DIFFERENT MACHINE LEARNING MODELS

ML Algorithm Used	Accuracy	F1-Score
Decision Tree	85.60%	67.54%
Random Forest	85.59%	67.86%
SVM	79.71%	63.68%
ANN with ADAM	86.10%	66.60%
ANN with SGD	78.40%	28.80%
Naive Bayes	80.30%	42.8%
Logistic Regression	82.60%	55.52%
Ensemble Learning	85.98%	65.94%
Decision Tree with Feature Engineering	85.77	66.56%
Random Forest with Feature Engineering	85.73	67.84%

V. FUTURE DIRECTIONS

1) Imbalanced Data Handling: The Adult dataset is imbalanced, with a significantly larger number of individuals earning less than $50,000 per year. One possible future work is to explore various techniques to handle imbalanced data, such as oversampling, undersampling, or using a different evaluation metric that considers the class imbalance.

2) Explainable AI: Another future work is to explore methods to make machine learning models more interpretable and explainable. This can help to build trust in the model's predictions and facilitate decision-making in real-world applications.

3) Additional Data Sources: Finally, future work can involve incorporating additional data sources into the analysis, such as social media data or other demographic data, to improve the accuracy of the model's predictions.

VI. CONCLUSION

To summarize, this project aimed to predict the annual income of individuals using various classification techniques. We used the Adult dataset, which contains demographic and socio-economic features, and experimented with different classification algorithms, hyperparameter tuning, and feature engineering. The results show that the Random Forest and ANN model has the highest probability of predicting the correct income level, while SVM and Logistic Regression have comparatively poor performance. Our findings show the importance of considering imbalanced data and using appropriate metrics, such as the F1-score, when evaluating classification models.

REFERENCES

[1] S. Deepajothi and S. Selvarajan, "A comparative study of classification techniques on adult data set," *International Journal of Engineering Research & Technology (IJERT)*, vol. 1, no. 8, 2012.

[2] R. Sabitha and S. Karthik, "Performance assessment of feature selection methods using k-means on adult dataset," *CCIIT*, vol. 4, pp. 606–612, 2013.

[3] N. Chakrabarty and S. Biswas, "A statistical approach to adult census income level prediction," in *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. IEEE, 2018, pp. 207–212.