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

"A Comparative Study of Classification Techniques On Adult Data Set" by S. Deepajothi and S. Selvarajan (2012). This paper aims to provide a comparative study of the classification accuracy provided by different classification algorithms, including Naïve Bayesian, Random Forest, Zero R, and K-Star on the census dataset. The authors present a comprehensive review of these algorithms on the dataset and discuss their strengths and weaknesses in terms of classification accuracy. The paper also provides an introduction to data mining and classification techniques.

"Performance Assessment Of Feature Selection Methods Using K-Means On Adult Dataset" by R. Sabitha and S. Karthik (2013). This paper 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. The 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.

"A Statistical Approach to Adult Census Income Level Prediction." by Navoneel Chakrabarty and Sanket Biswas (2018). This paper 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. The study 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

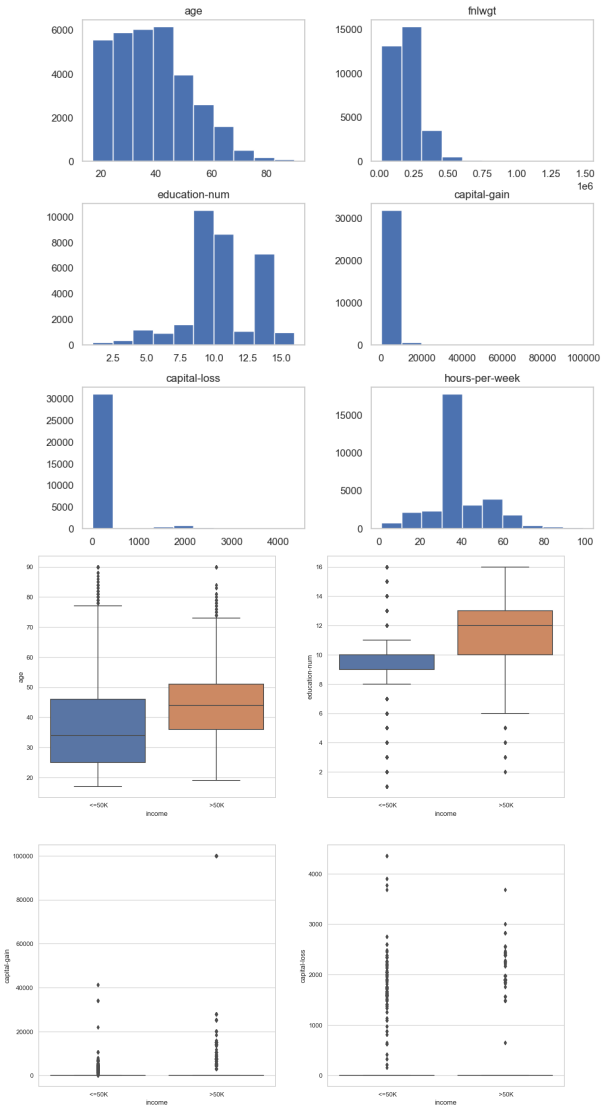
In this section we introduced some well known Machine Learning Algorithms to our dataset and observed the accuracy of those algorithms.

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, with the following descriptions for each column:

- **age**: the age of the individual (numeric)

- **workclass**: the type of work the individual is engaged in (categorical: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked)
- **fnlwgt**: the final weight, which represents the number of individuals the census believes the entry represents (numeric)
- **education**: the highest level of education attained by the individual (categorical: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool)
- **education-num**: the numerical representation of education, corresponding to the number of years of education completed (numeric)
- **marital-status**: the marital status of the individual (categorical: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse)
- **occupation**: the occupation of the individual (categorical: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces)
- **relationship**: the relationship status of the individual (categorical: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried)
- **race**: the race of the individual (categorical: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black)
- **sex**: the sex of the individual (categorical: Female, Male)
- **capital-gain**: the capital gains of the individual (numeric)
- **capital-loss**: the capital losses of the individual (numeric)
- **hours-per-week**: the number of hours worked per week by the individual (numeric)
- **native-country**: the country of origin of the individual (categorical)
- **income**: the income level of the individual (categorical: $\leq 50K, \geq 50K$)



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.

B. Machine Learning Algorithms

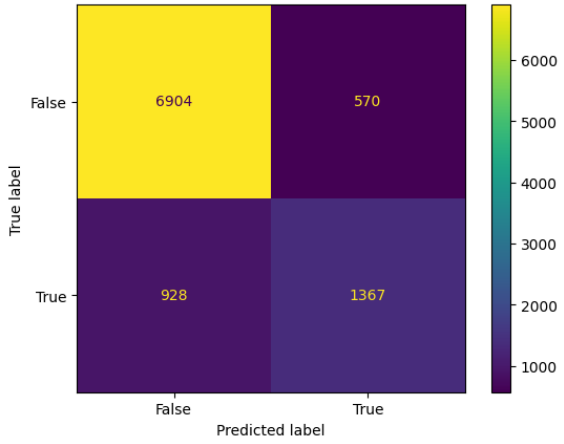
1) Decision Tree: The decision tree algorithm is a non-parametric method used for both classification and regression tasks. It builds a model in the form of a tree structure by recursively splitting the dataset into smaller subsets based on the most significant variables. We used the scikit-learn library 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 also applied cross-validation to ensure the robustness of our model.

We experimented with different hyperparameters of the decision 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 8, 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 1.

Our decision tree model achieved an accuracy of 85.6% 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.

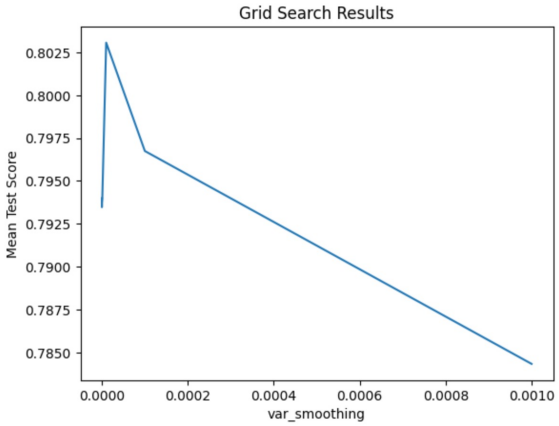
Class	Precision	Recall	F1-score	Support
0	0.81	0.95	0.87	4942
1	0.64	0.30	0.41	1571
Accuracy			0.79	6513
Macro avg	0.72	0.62	0.64	6513
Weighted avg	0.77	0.79	0.76	6513



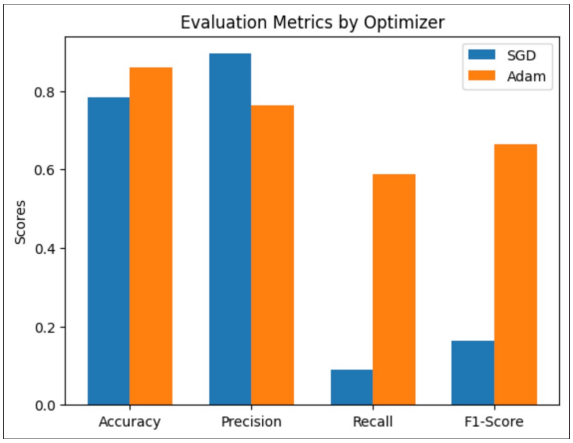
2) Naive Bayes: The Naive Bayes algorithm is a probabilistic algorithm used for classification tasks. It assumes that the features are independent of each other and calculates the probabilities of the target variable based on these features. We used the scikit-learn library to implement the Gaussian Naive Bayes algorithm.

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.

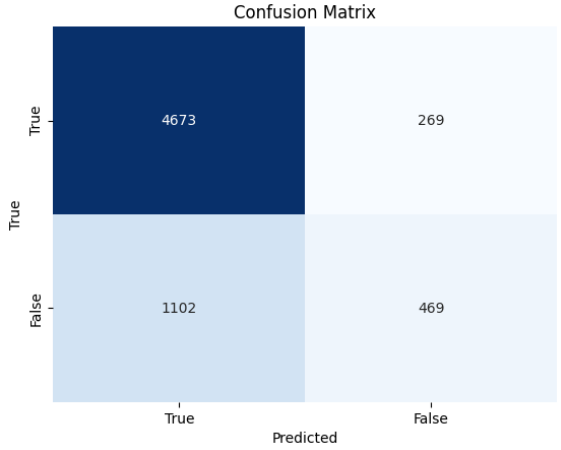


3) ANN: We also experimented with deep learning model(ANN) using the Keras library with TensorFlow backend. We built a neural network model with two hidden layers of 64 and 32 nodes and an output layer with a sigmoid activation function for binary classification. We trained the model with two different 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%, a precision of 76.3%, recall of 58.8%, and an F1-score of 0.664. On the other hand, the model trained with SGD optimizer achieved an accuracy of 78.4%, a precision of 89.5%, recall of 9.0%, and an F1-score of 0.163. Further hyperparameter tuning can be done to improve the performance of the model. We can experiment with different activation functions, increase the number of hidden layers, or adjust the learning rate of the optimizer. We can also consider using more advanced techniques such as dropout or regularization to prevent overfitting of the model. It is also important to continue evaluating the performance of the model using different metrics and cross-validation to ensure its robustness.



4) 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. 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.



	Precision	Recall	F1-score	Support
0	0.81	0.95	0.87	4942
1	0.64	0.30	0.41	1571
Accuracy			0.79	6513
Macro avg	0.72	0.62	0.64	6513
Weighted avg	0.77	0.79	0.76	6513

We are still on experimenting with hyperparameter tuning further to get better accuracy and improved results.

C. Implementation Details

IV. Comparison

V. Future Directions

VI. Conclusion

References