Predicting Income Classification Using Machine Learning

Author: Jathin Banala

Introduction

Predicting whether an individual's income exceeds $50,000 annually based on demographic and employment data is a significant challenge in binary classification. This project utilizes the "Adult Income" dataset from the UCI Machine Learning Repository (Kohavi) to explore the efficacy of machine learning algorithms in solving this problem. We will compare two models using the Logistic Regression and K-Nearest Neighbors algorithms to gain further insights into predicting income. These two models are relatively computationally efficient and easy to comprehend. By tackling this problem, I aim to uncover what demographics are highly correlated with one another and how each demographic affects the models' estimated income classification. In other words, my goal is to expose some of the underlying factors behind income disparities in the United States.

Data

I used the "Adult Income" dataset from the UCI Machine Learning Repository, which contains 14 attributes including age, education, occupation, and hours worked per week. The target variable indicates whether an individual's income is less than or equal to $50,000 or greater than $50,000. The dataset was extracted by Ronny Kohavi and Barry Becker in 1994 from the Census database, which explains why the nominal median income is significantly lower than it is today. According to the Federal Reserve Economic Data (2024), the median personal income has more than doubled since 1994. If we were to perform a similar analysis on up-to-date data from a more recent census, we would likely wish to increase the target threshold by a factor of 2.

Attributes

- **Continuous variables:** Age, final weight (fnlwgt), education number, capital gain, capital loss, and hours worked per week.
- **Categorical variables:** Workclass, education, marital status, occupation, relationship, race, sex, and native country.

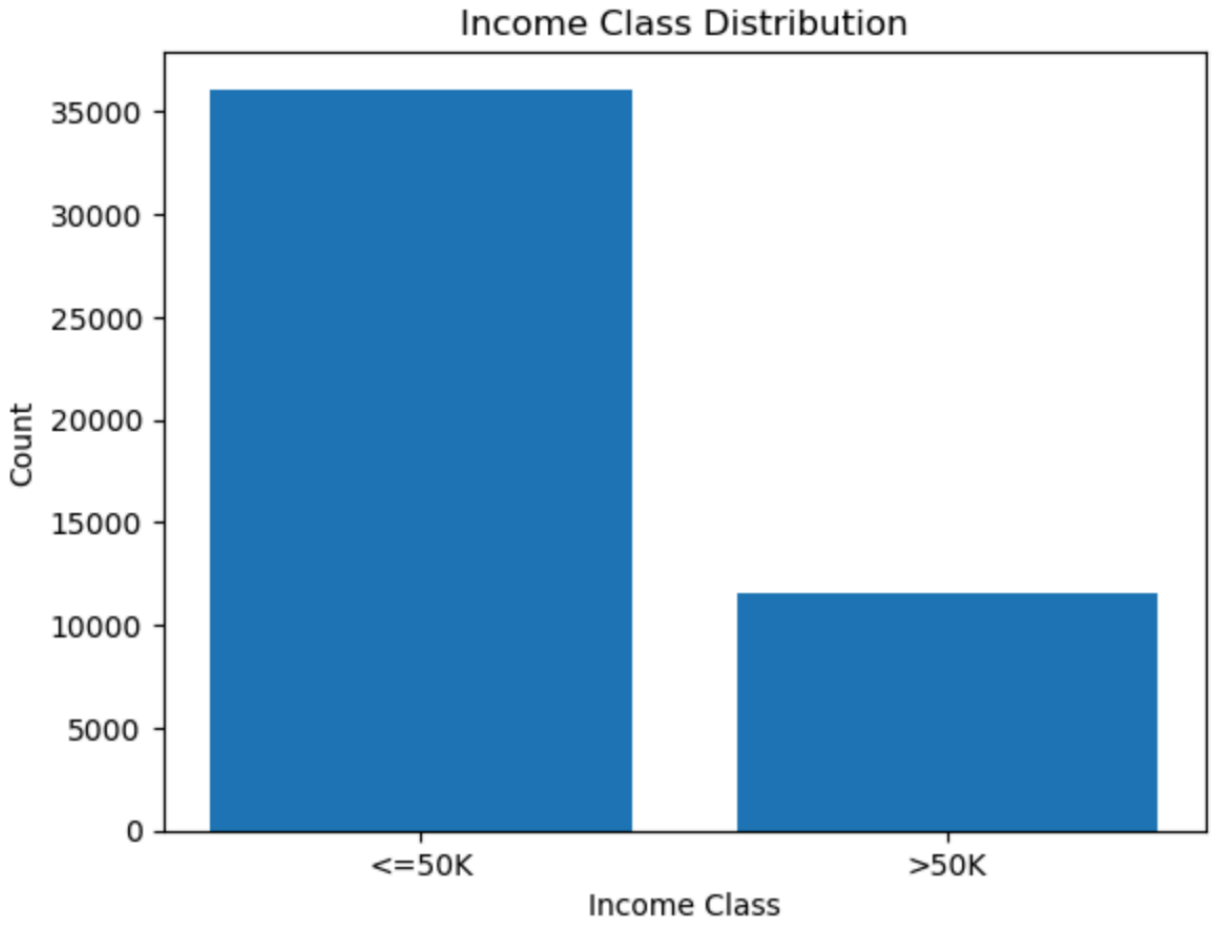
Dataset Summary

- **Size:** The dataset contains 48,842 data points and 15 features.
- **Target Distribution:** The income classes are not equal, with the vast proportion of individuals earning less than $50,000 per year.
- **Missing Values:** Several features, such as workclass, occupation, and native-country, contain missing values that need to be addressed during the preprocessing stage.

Visual Analysis

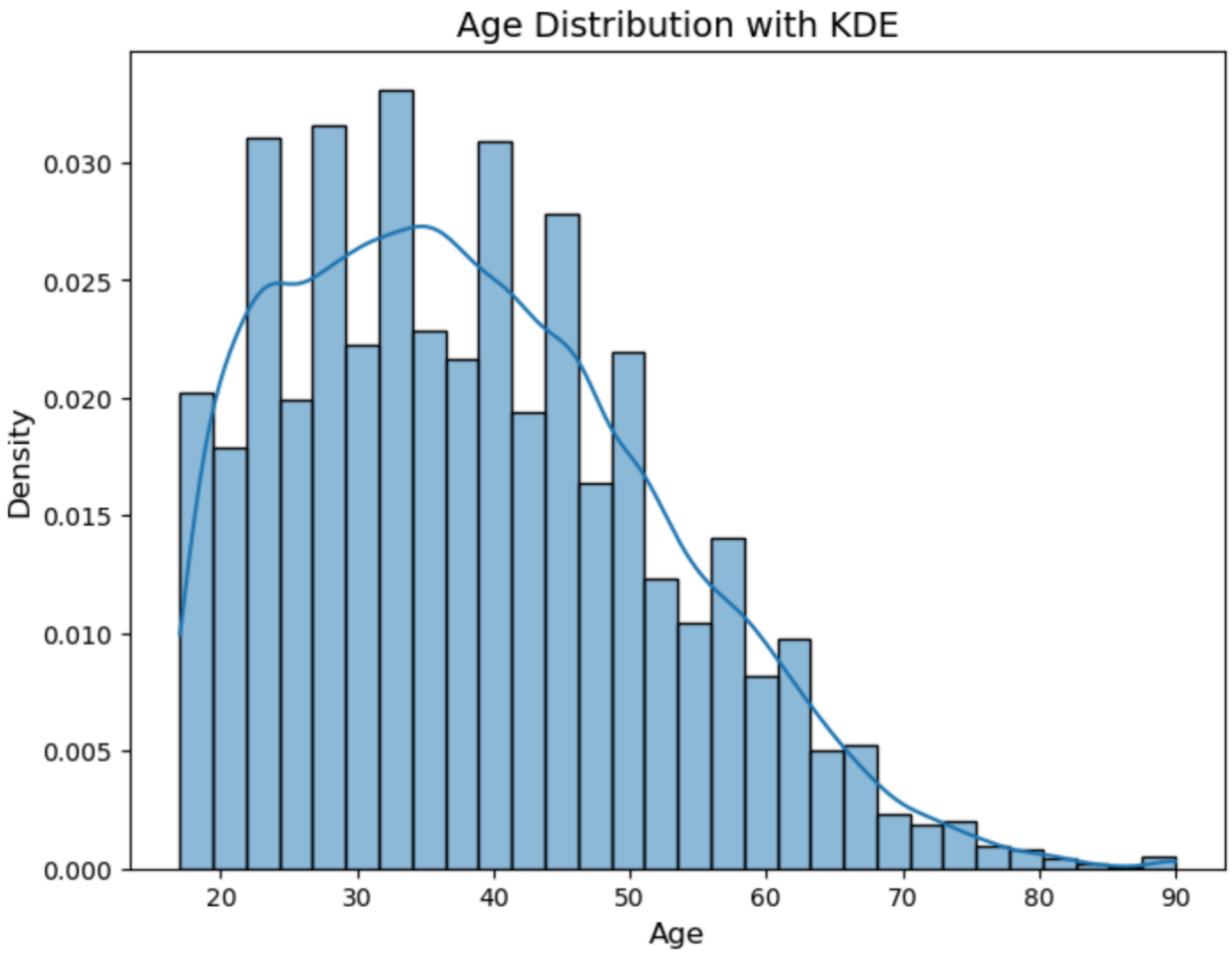
Class Distribution

I created a bar chart using the Matplotlib visualization library in Python to show the distribution of the target variable (<=$50,000 and >$50,000). Approximately 75% of the individuals in the dataset were shown to have earned an annual income less than or equal to $50,000, which indicates a striking disparity in income.

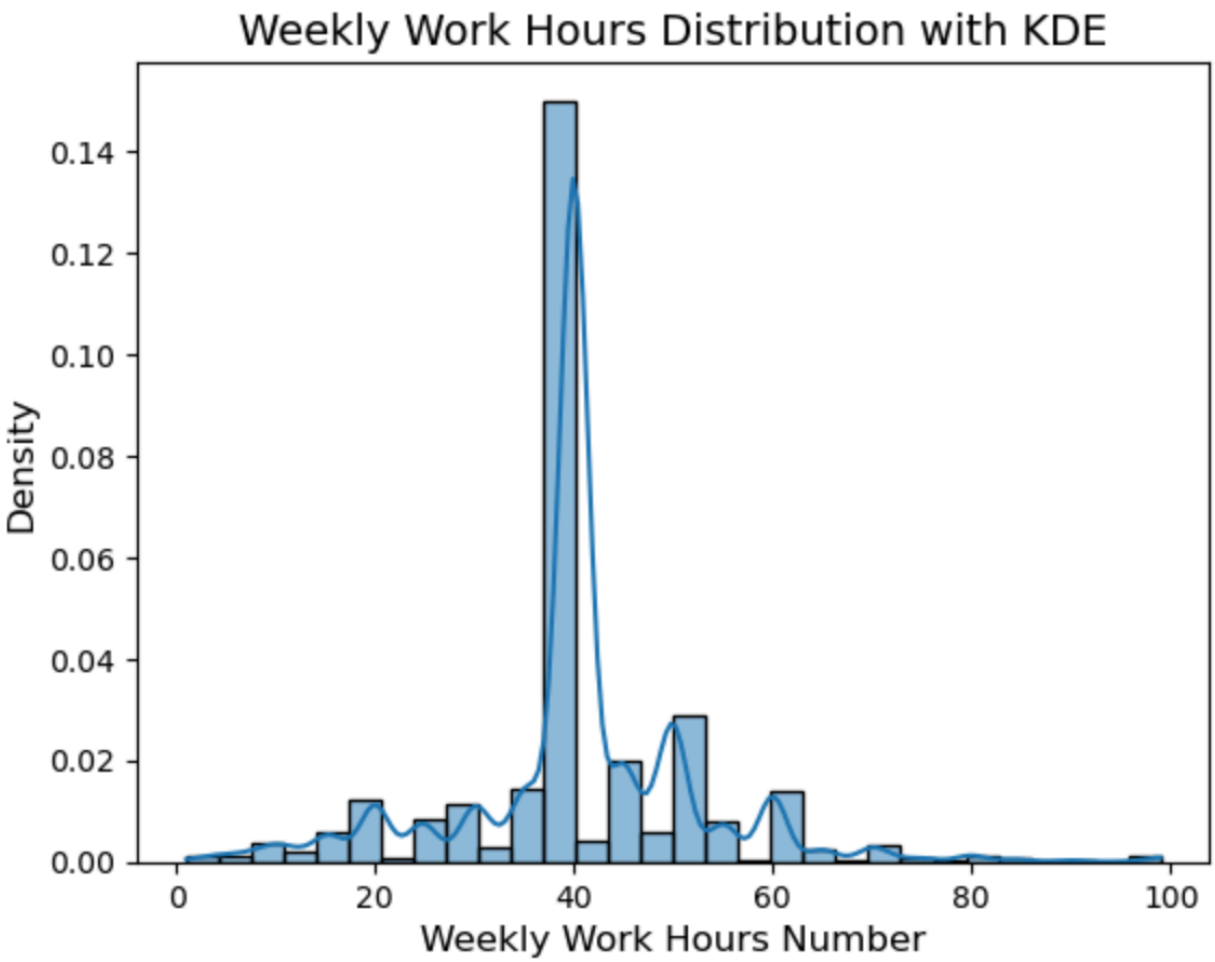


Feature Exploration

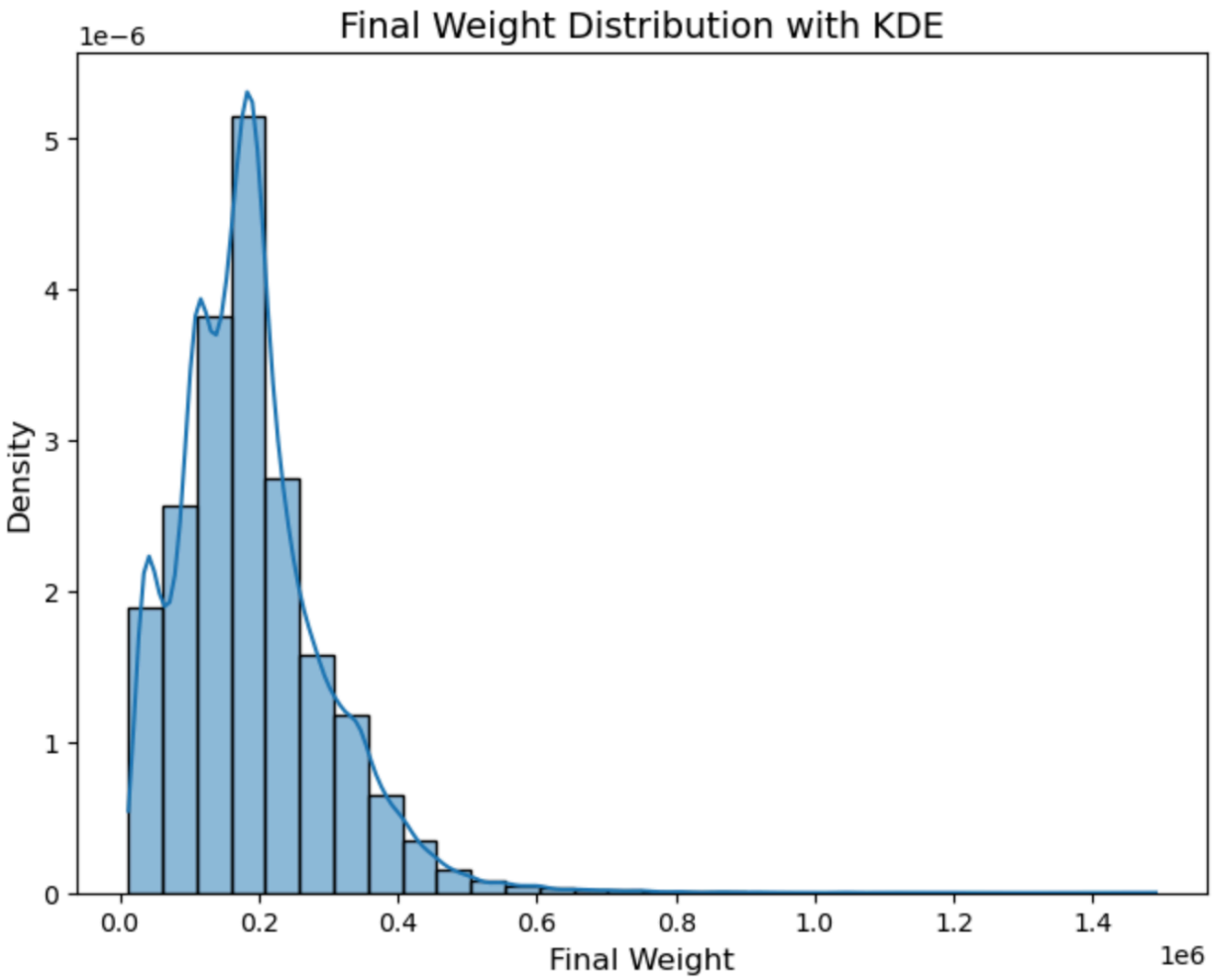
- **Continuous Variables:**
    - *Age:* The majority of individuals are between 25 and 50 years old.



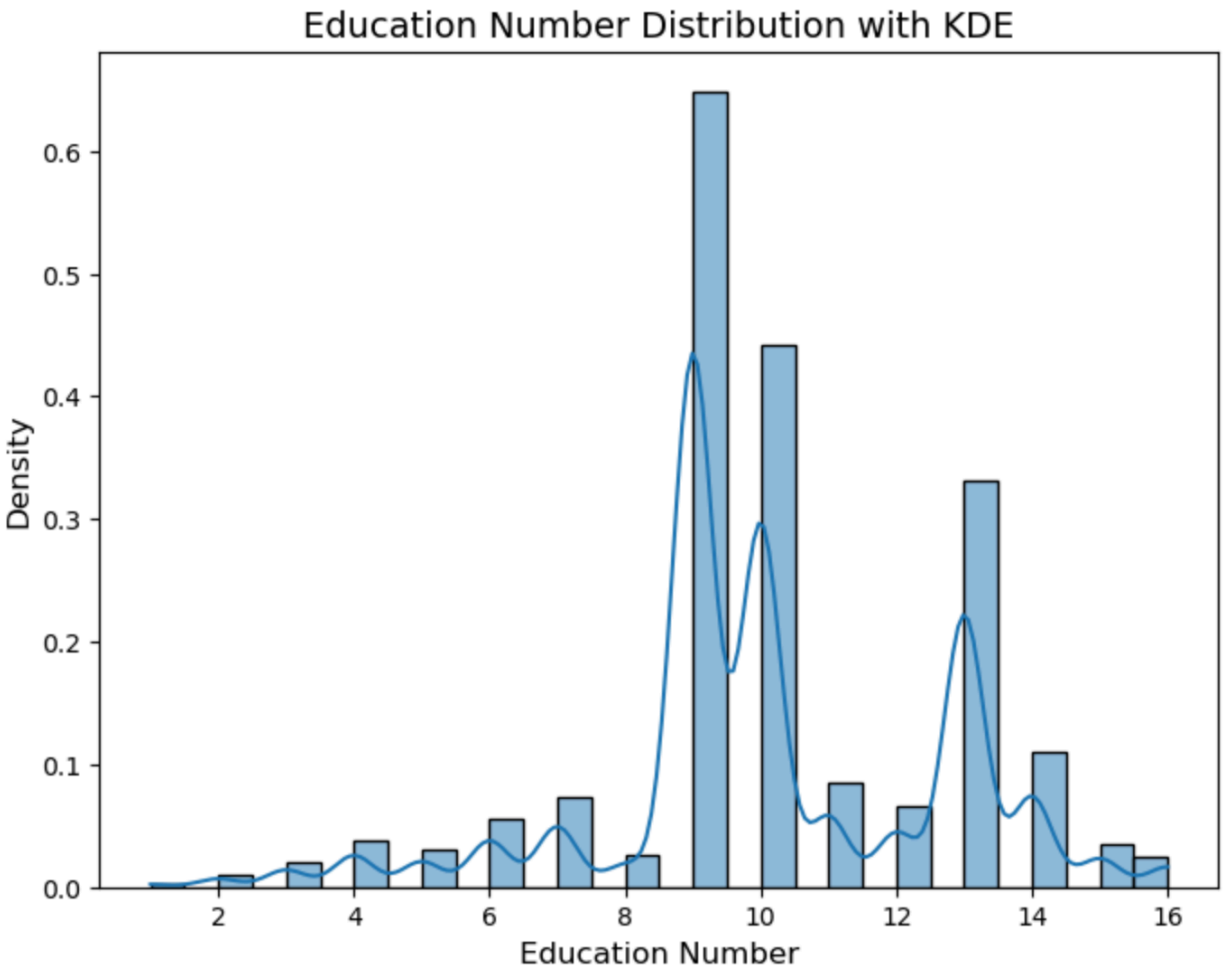
    - *Hours per week:* Most people worked between 35 and 45 hours per week, with a significant cluster around 40 hours.



- *Final weight*: This value represents the total number of people that the census believes this data point represents. The distribution was heavily skewed right with most entries being near 0.2.



- *Education-num*: The median level of education was around 11 with a peak around 8.5.



- **Categorical Variables:**
    - *Education:* A significant proportion of individuals have completed high school or some college.
    - *Occupation:* The dataset contains a rich, diverse distribution of occupations, with the highest counts in roles such as “Craft-repair” and “Exec-managerial”.

Preprocessing

Handling Missing Data

The dataset contains missing values in workclass, occupation, and native-country, which I handled by simply dropping the rows in which they occurred as these rows only accounted for a small percentage of the dataset, meaning their exclusion would likely have minimal performance impact on our models while greatly simplifying the code and streamlining my workflow.

Encoding Categorical Variables

Categorical variables were encoded using one-hot encoding to convert them into binary classes suitable for machine learning algorithms. I considered encoding the age variable into 9 different classes that matched the age groups in the U.S. Census, but I ultimately abandoned this plan as this method made it difficult to distinguish between the various age groups when attempting to classify income, meaning that this approach would muddle the predictive capabilities of the program.

Train-Test Split

The dataset was split into training and testing subsets which comprised 80% and 20% of the data, respectively, to evaluate the models' performance on unseen data.

Algorithms

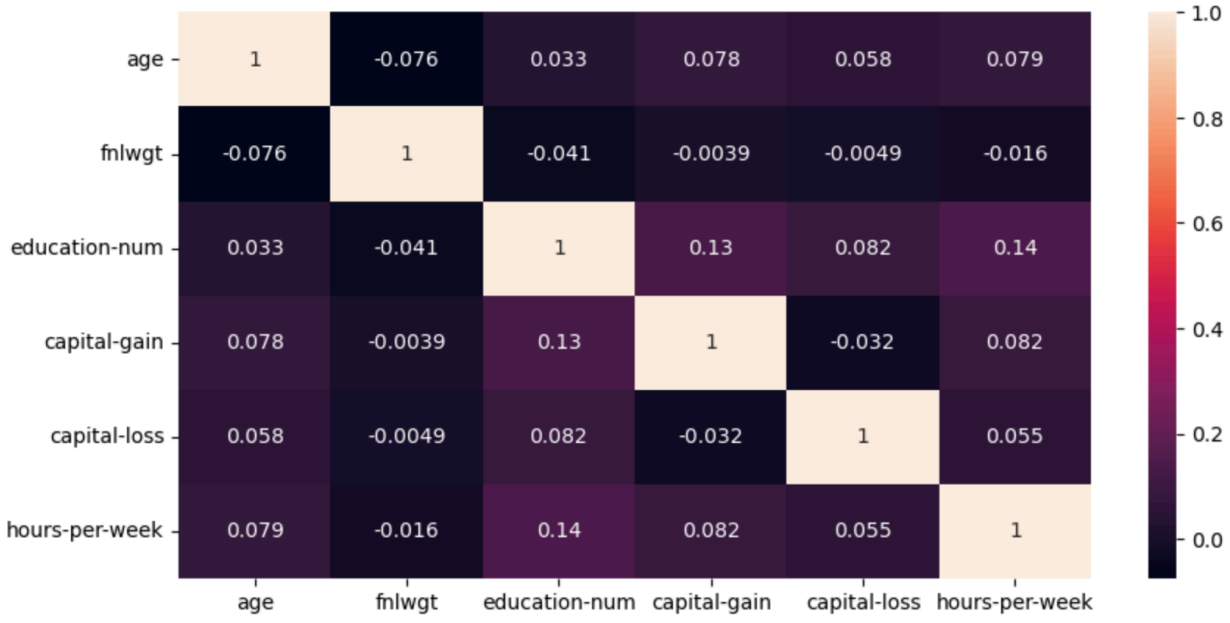
Logistic regression is a supervised machine learning algorithm that estimates the log-odds of the target variable. Logistic regression minimizes the negative log-likelihood (NLL) loss, which estimates how closely the predicted probabilities match the actual class labels. This method encourages the model to output probabilities closer to 1 for the correct class and 0 for the

incorrect class. We use the sigmoid function to convert the linear model's output into probabilities between 0 and 1 where the threshold of 0.5 is used to assign the class labels.

K-Nearest neighbors is a classification algorithm that predicts the class by comparing the data point in question to its K nearest neighbors. In this project, I used the Euclidean distance metric to compare data points and performed the classification using a simple majority voting system.

Results

I checked for potential multicollinearity in order to avoid the problem of potentially biased weight estimates. Fortunately, the correlation coefficients were relatively low between the relevant features, indicating that multicollinearity was not a significant enough problem to seriously affect our models.



Conclusion

The data preprocessing steps, including handling missing values, encoding categorical features, and standardizing numerical variables, helped ensure that my dataset was ready to begin the training stage of the machine learning pipeline. These foundational steps set the stage for building and evaluating the necessary machine learning models in order to accurately predict income classification.

While working on this project, I encountered numerous problems even when attempting to perform basic tasks such as importing the dataset as my program did not recognize that the uciml library was installed on my computer. I resorted to using pip to manually verify the installation of this library near the start of my program. I also had trouble with applying basic visual analysis techniques as a significant portion of my features were categorical variables composed of strings, which many popular plotting and mathematical functions do not have the capability to handle. I had to carefully encode my data into binary classes of 0s and 1s in order to proceed with my project. Overall, working on this problem greatly improved my understanding of the machine learning pipeline and enhanced my knowledge of essential data visualization and transformation techniques.

Acknowledgements

I used Google Gemini to help me import my dataset and debug my code as well as help me fix grammatical errors in my project report. It also enabled me to better format my paper to be more in-line with the rubric and resemble professional research papers. I also reused code from my previous homework assignments in order to accelerate my progress and implement the necessary algorithms properly. I referenced data from the Federal Reserve Bank of St. Louis in

order to explain the income distribution as a similar distribution would not make sense in the context of the United States' economy in 2024.

References

Kohavi, Ron. "Census Income." UCI Machine Learning Repository, 1996, https://doi.org/10.24432/C5GP7S.

Federal Reserve Bank of St. Louis. (2024, September 10). Median personal income in the United States. FRED. https://fred.stlouisfed.org/series/MEPAINUSA646N.

Python Notebook: https://github.com/JReddy-123/Census-Income.