Predicting Personal Total Income of Individuals in the United States in 2023*

XGBoost Captures the Complex Pattern Better Than Random Forest and Linear Regression

Justin (Jiazhou) Bi and Weiyang Li

November 14, 2024

This project aims to build a predictive model for personal total income using various demographic attributes, such as marital status, occupation, and residing state. We explored three popular machine learning algorithms: Random Forest, Linear Regression, and Extreme Gradient Boosting. Ultimately, Extreme Gradient Boosting was selected due to its superior predictive performance. While the model achieved a high R-squared value, it also exhibited a high Mean Squared Error, indicating challenges in accurately predicting all data points. Future steps for improving the model may involve incorporating additional demographic features to enhance model accuracy and reduce errors. Additionally, fine-tuning hyperparameters, experimenting with feature engineering, and handling outliers could further improve performance.

1 Introduction

Income is a fundamental concept as it affects nearly every member of society, influencing access to resources, quality of life, and overall economic stability. At the same time, it is also a difficult concept to define, as it can be interpreted differently depending on various perspectives, such as economic, social, legal, and tax purposes. Therefore, we must examine various demographical factors that may impact income to understand its concept better. Our project defines income as "all forms of net financial resources generated or lost from work, investments, or other activities." This project aims to predict personal income using data from the 2023 American Community Survey (ACS) (Ruggles et al. 2024). The ACS provides detailed demographic information, including age, education, employment, marital status, mortgage status, veteran

*All project related files available at: https://github.com/Jiazhou-Bi/ACS-Income-Model

status. As a result, it provides a robust dataset for studying potential factors that may influence income.

Due to the timeframe limitations, the findings of this project are specifically applicable to the 2023 dataset. As such, the results may vary across different periods, and their generalizability has not been assessed within this project. The raw data used in this analysis was sourced from IPUMS using the IPUMS API (IPUMS 2024). We utilized pandas (McKinney 2010) for data processing, and data visualization was carried out using Seaborn (Waskom 2021). Additionally, all tables presented in this report were generated with Plotly (Plotly Technologies Inc. 2015).

Income inequality is a serious problem and can lead to significant consequences and it may affect social structure, economic stability, as well as policy-making processes. Researchers found that demographic factors, such as education level and occupation type, can significantly impact an individual's financial standing (Autor 2013). Similarly, social relationships, such as marital status and household characteristics, affect income dynamics (Waite and Gallagher 2000). Therefore, applying machine learning models to analyze the ACS dataset and using the information mentioned to predict personal income is reasonable.

We have chosen 2023 ACS data because it offers many demographical details, as well as self-reported total income of the individuals. The sample is also representative and covers all geographical locations in the USA. Furthermore, it provides easy access to their database through API. We use predictive models, including random forest, linear regression, and extreme gradient boosting machine to estimate personal income based on demographic factors to provide insights into which potential factors have the most significant impact on personal income. We hope the findings generated from our project can provide insights to policy-making process so that we can, hopefully, reduce income inequality and support further economic development in the future.

If we can successfully predict personal income using demographic information from the survey, this analysis will be able to contribute to a nuanced understanding of the American economy in 2023. It can help to reveal how demographic changess, educational attainment, and employment status/type can impact personal income. On the other hand, if the model cannot predict personal income with a high degree of confidence, this would suggest that the factors included in the analysiscannot not fully capture the intricacies of economic behavior and individual circumstances. As a result, additional variables, such as detailed education background like school of graduation, languages spoken, health status, etc, might be necessary in accurately predicting income.

In this analysis, we explored three different models—Linear Regression, Random Forest, and XGBoost—to predict income based on various demographic factors. Each model provides valuable insights into the relationships between income and predictors, such as education, age, industry, and gender. Among the models, XGBoost demonstrated the best performance in terms of predictive accuracy, as evidenced by its lower RMSE and superior handling of complex data interactions.

The remainder of this paper is structured as follows: Section 2 introduces the raw dataset and describes the cleaned datasets, along with a preliminary analysis through numerical summaries and visualizations. Section 3 describes the three machine learning models being used in our study. Section 4 explores key findings from our analysis. Lastly, Section 5 addresses the limitations of the analysis and offers recommendations for reducing delays and improving subway service reliability.

2 Data

The dataset used here is downloaded from the IPUMS USA website via (Ruggles et al. 2024). Our project's extracted dataset contains all the responses from the American Community Survey (ACS) from 2023. Our extraction included the following variables: total income, dwelling mortgage status, sex of the individual, marital status, education attainment, private or public school, industry or occupation, veteran status, and age. Some of the variables (such as educational level, occupation) we have chosen here have a more detailed version that contains more information. However, due to the nature of categorical variables, having detailed variables will create more classes of that variable, thus leading to more features for our model when encoding those variables. Although more features will improve the overall performance of our predictive model, it will also create issues such as overfitting, needing more computational resources, and lack of interpretability. We have chosen higher-level variables in this study for better computational efficiency. Furthermore, we have cleaned the data for better data quality. Details of our data cleaning processes are provided in Section 6.1. In the following subsections, we will review all the variables used in this report and provide some basic descriptive statistics.

2.1 Descriptive Data Analysis

In this subsection, we will explore every variable used in our study by examining some of the basic statistics. These preliminary analyses can help us better understand the data we are working with.

2.1.1 Mortgage

The Mortgage variable describes the mortgage status of dwelling. There are five different values, they are "N/A", "No, owned free and clear", "Check mark on manuscript (probably yes)", "Yes, mortgaged/ deed of trust or similar debt", and "Yes, contract to purchase". We have dropped the N/A values for better data quality. Details available in Section 6.1. The count of the remaining classes is shown below in (Figure 1).

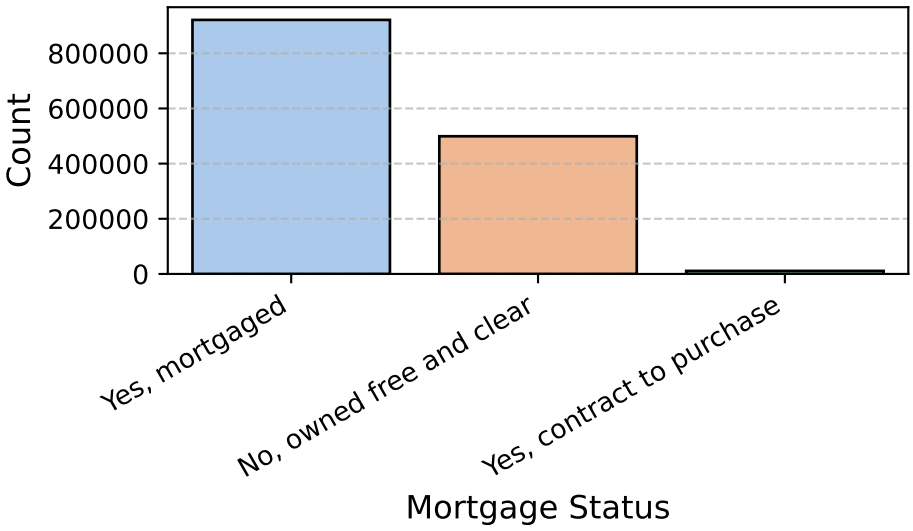

Figure 1: Most Homeowners Are on Mortgage

2.1.2 Sex

This variable is coded as a binary variable, and we will use it as-is in our project. This does not suggest that the authors agree with this type of sex classification, but rather that it aligns with the available data and serves the research purposes of this project. The binary coding is a practical choice to maintain consistency with the dataset, though it may not fully capture the complexity and diversity of gender identities. Table 1 shows the distribution of this variable.

2.1.3 Age

Age describes the age of the individual, and a age of 0 was coded for the individuals under 1 year old. For our dataset, the minimum age is 16, however. Age distribution of our dataset is shown below in Figure 2.

Table 1

(a)

Sex	Count
Male	734219
Female	697132

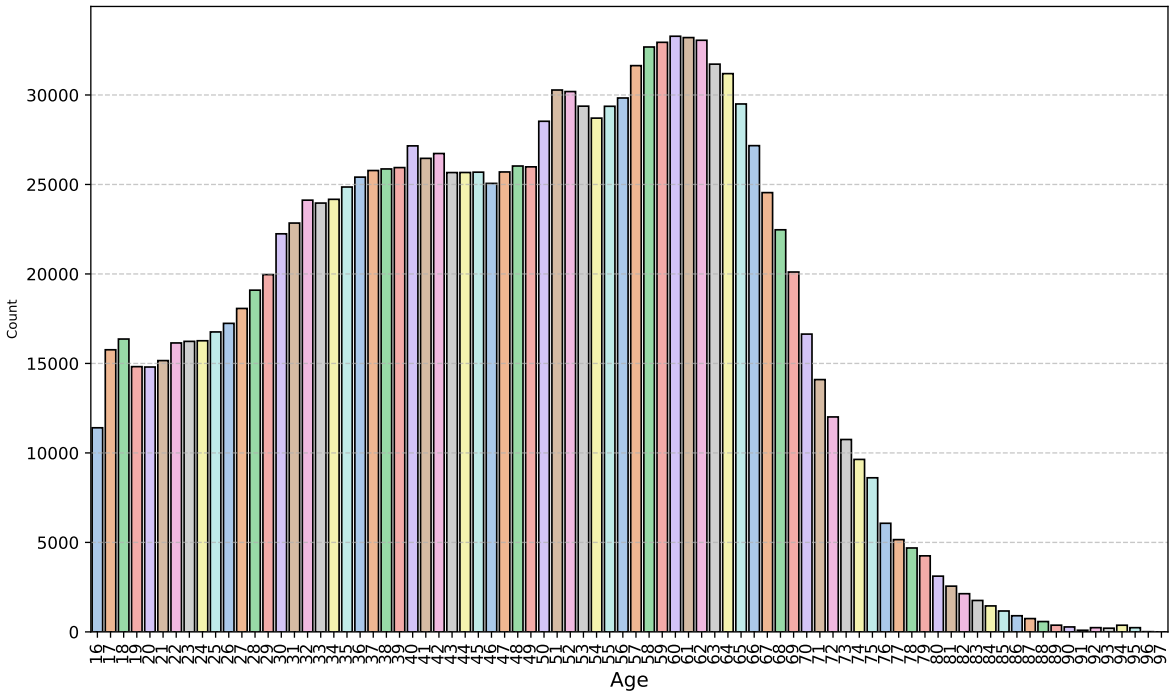

Figure 2: Age Distribution

2.1.4 Marital Status

Marital status describes the marital status of the individual. There are 7 different values, namely “Married, spouse present”, “Married, spouse absent”, “Separated”, “Divorced”, “Widowed”, “Never married/single”, and “Blank, missing”. We have dropped the N/A values for

better data quality. Details available in Section 6.1. The following Figure 3 illustrate the distribution of marital status of the participants in our dataset.

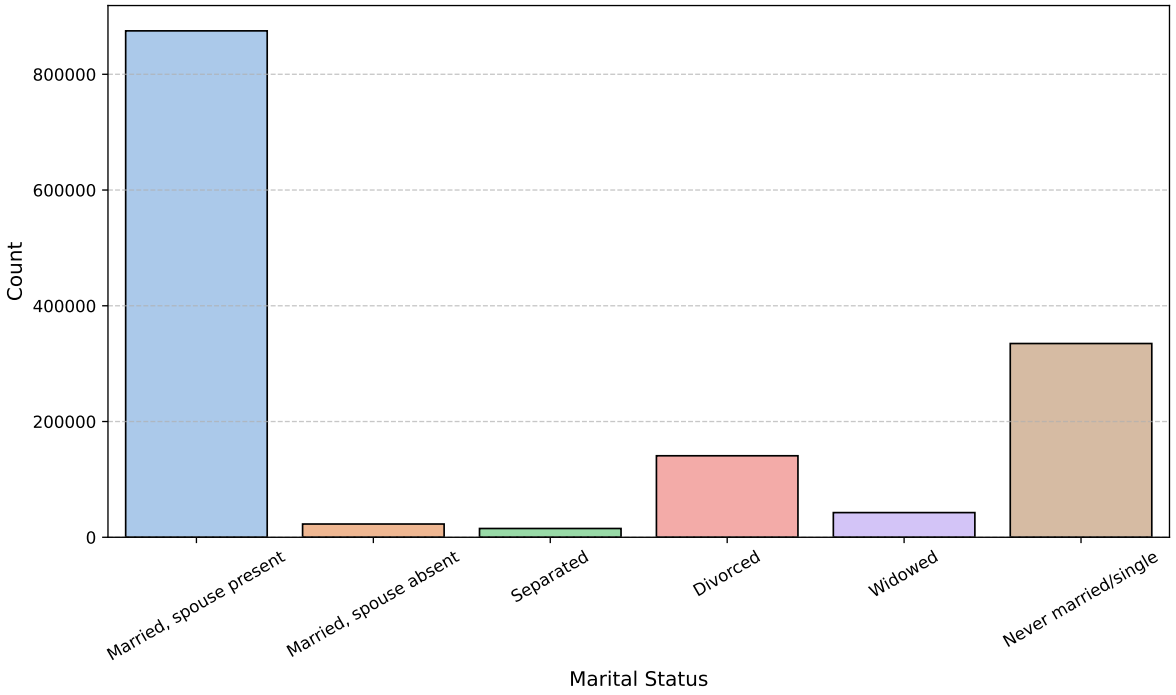

Figure 3: Counts of Marital Status, Most Participants Are Married with Spouse Present, or Never Married/Single

2.1.5 Educational Attainment

This is a general version (compared to the detailed version). The reason of choosing the general version is mentioned in Section 2 above. This column represents the highest educational attainment achieved by the individual. The distribution of the variable is shown in Figure 4.

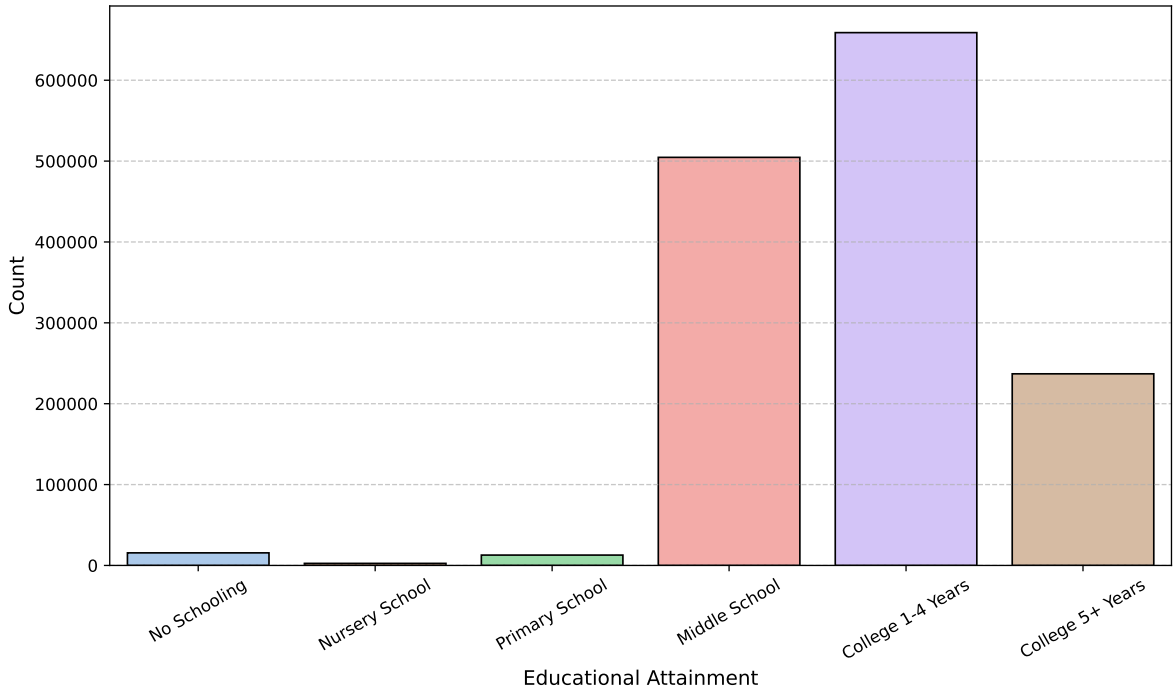

Figure 4: Most Participants Have Middel School Attainment or Above

2.1.6 School Type

School type indicates if the individual went to a public school or private school. Other than missing values, there are three different classes. They are "Not enrolled", "Public school", and "Private school". We are not sure how the participants would respond if they have been to more than one type of schools. Figure 5 shows the distribution of our sample.

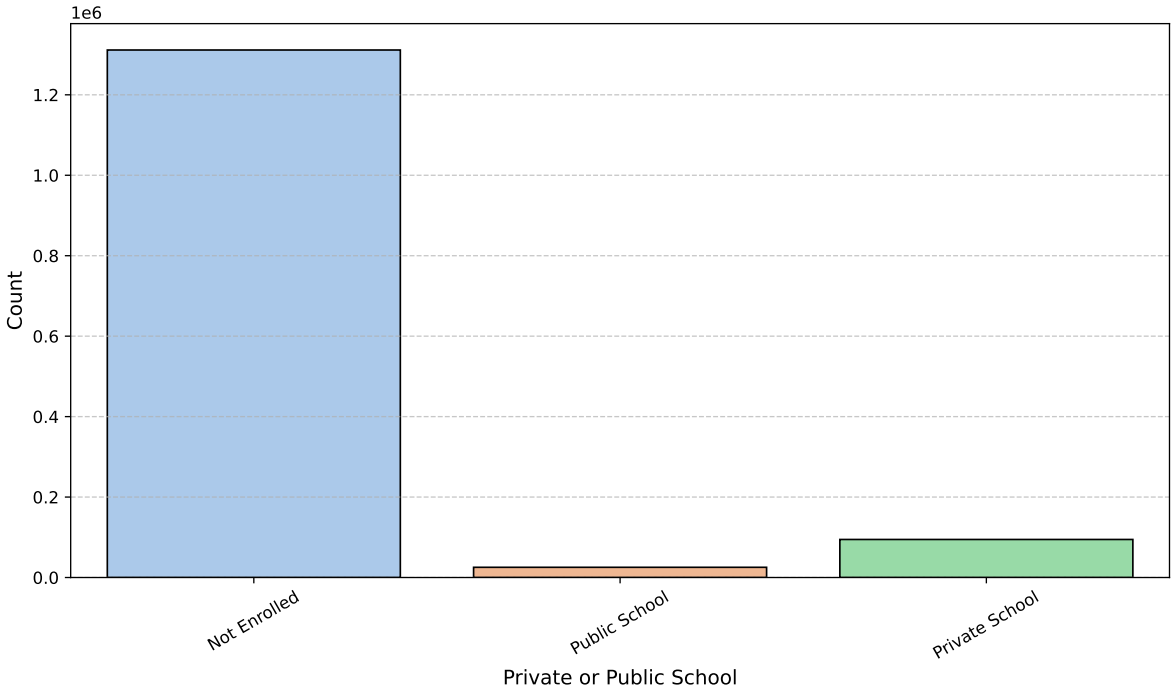

Figure 5: School Type

2.1.7 Industry

Industry describes the type of industry the individual works in. Same as the educational attainment variable, more detailed version of this variable is available (i.e., Occupation). There are more than 200 industries listed in our sample. A detailed description can be seen at IPUMS USA.

2.1.8 Veteran Status

Veteran status idicates whether the individual is a veteran or not. It can be either "Yes", "No", or "N/A". Please notice that "N/A" here is different from missing, suggesting this might be a different indicator, which is not explicitly explained from the ddl file provided. There is also a detailed version of this variable, which is not used in our project. The counts is displayed in Table 2.

Table 2

(a)

Veteran Status	Count
No	1334292
Yes	85654
N/A	11405

2.1.9 Total Income

Total income is the total income made by the individual for the year. The distribution of total income is depicted in Figure 6. We have removed the outliers from the graph to have a better understanding of the general pattern of the overall distribution. From the figure, it is clear that more income are reported near multiple of $10,000 as a unit. This may be caused by the "heaping and leaping" phenomonon found in self-reported survey data (Pudney 2008).

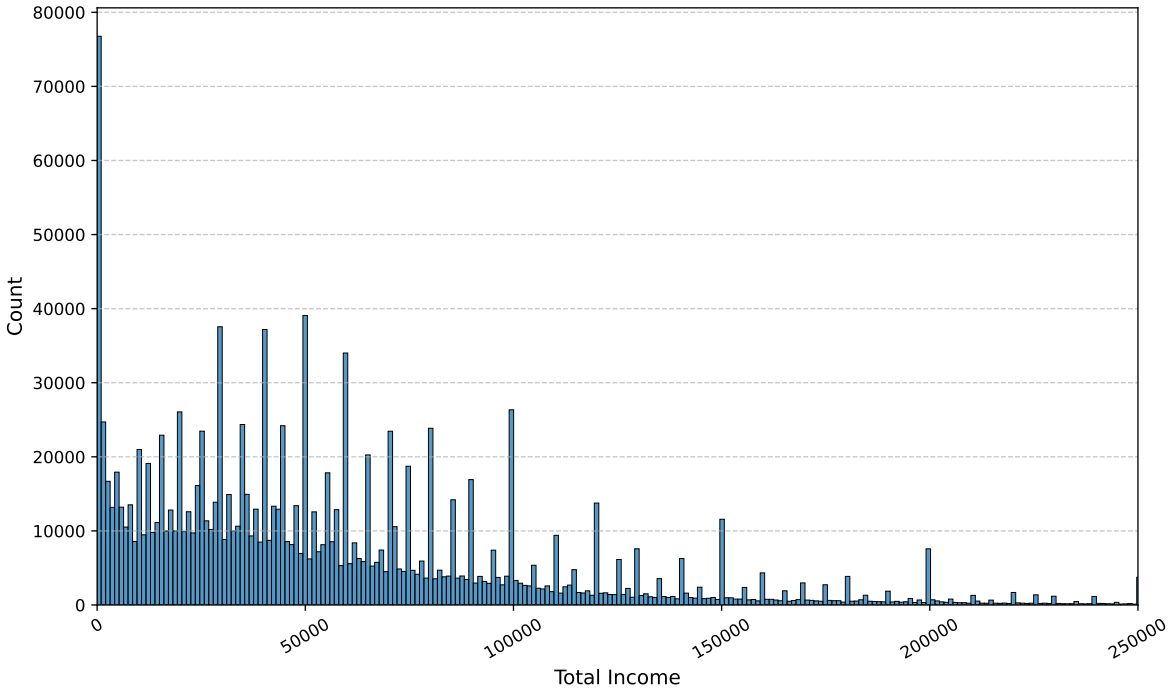

Figure 6: Total Income Distribution

2.2 Analysis of Feature Relationships

In this subsection, we will explore some of the potential relationships among the features chosen for our study. Understanding these relationships can prepare us better for our model construction in the next section.

2.2.1 Income Distribution by Mortgage Status

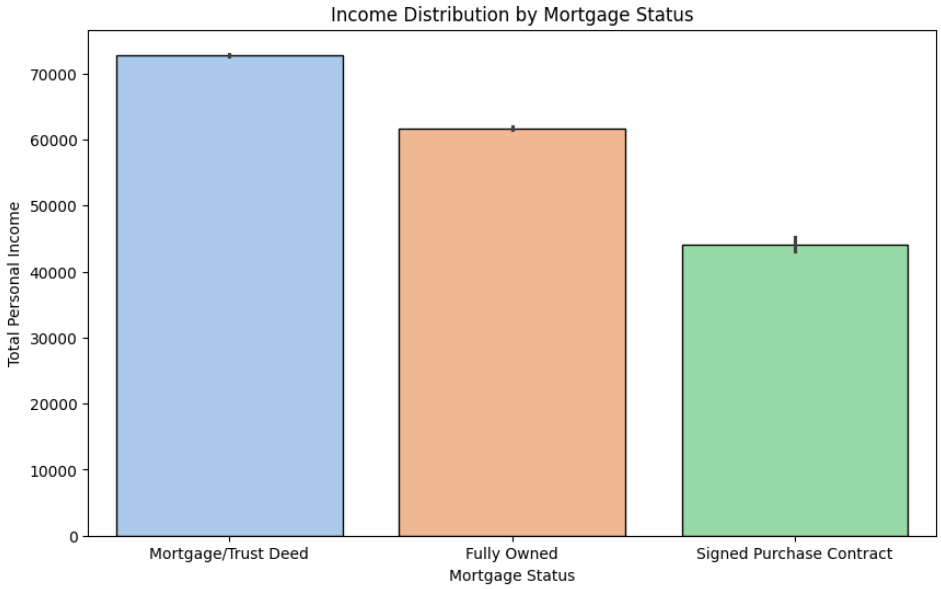

Figure 7: Income Distribution by Mortgage Status

Figure 7 illustrates the income distribution for individuals with different mortgage statuses. The highest average income is observed among individuals with a "Mortgage/Trust Deed," followed by those who own their homes outright ("Fully Owned"). Those with a "Signed Purchase Contract" have the lowest average income among the groups. This pattern may reflect a correlation between mortgage status and income level, where individuals with higher incomes are more likely to secure traditional mortgage loans, while those with lower incomes tend towards alternative purchasing arrangements. This could indicate the influence of financial stability and access to credit on mortgage status.

2.2.2 Correlation Analysis of Age, Education, and Income

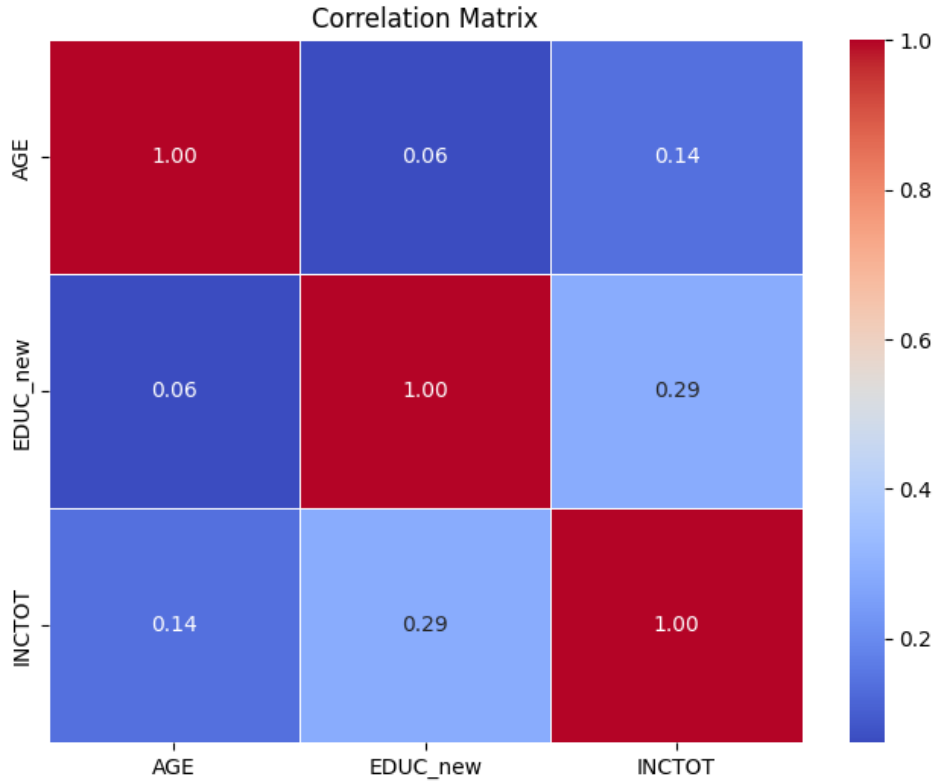

Figure 8: Correlation Matrix of Age, Education, and Income

Figure 8 shows the correlation matrix for three key variables: age, education level (EDUC_new), and total personal income (INCTOT). The matrix reveals that there is a moderate positive correlation between education level and income (0.29), suggesting that higher education levels are generally associated with higher income. Additionally, a smaller positive correlation exists between age and income (0.14), which might indicate that as individuals grow older, they tend to earn more due to accumulated experience. The correlation between age and education (0.06) is minimal, which is expected as these two variables are less directly related. Overall, this analysis supports the notion that education level plays a more significant role in income variation than age within this dataset.

2.2.3 Average Income by Education and Gender

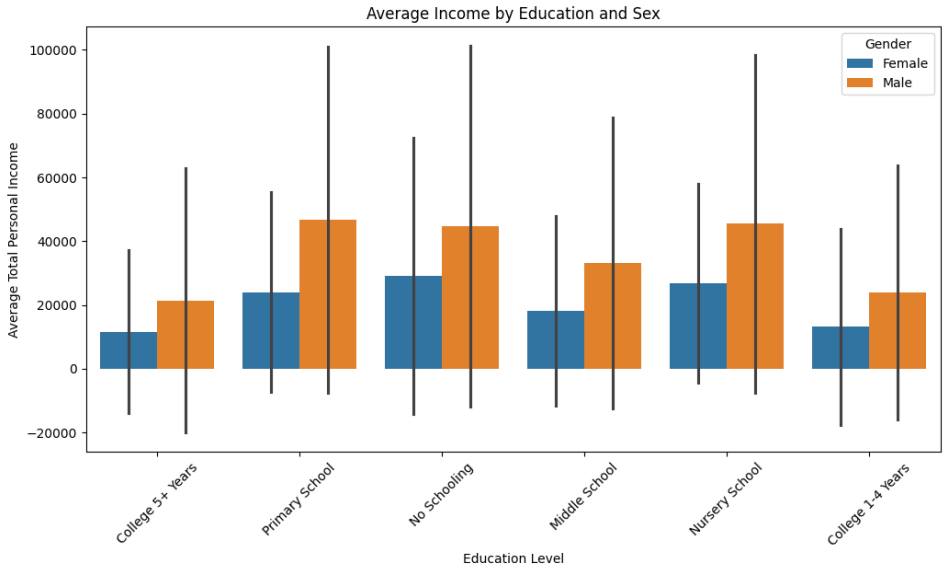

Figure 9: Average Income by Education and Sex

Figure 9 presents the average total personal income based on different levels of education for both male and female groups. From the plot, it is observed that individuals with higher levels of education, particularly those who attended "College 5+ Years," tend to earn more on average compared to other groups. However, there are some variations in income within each education level, as indicated by the error bars. Generally, males tend to have higher average incomes across most education levels. This difference suggests a possible gender-based income disparity, even after controlling for education levels. Additionally, the variability in income among education levels highlights that factors beyond educational attainment, such as occupation and industry, may further impact income. Thus, incorporating an interaction term between education and gender in the model is necessary to capture these nuanced effects more accurately.

3 Model

Our modeling strategy aims to explore the relationship between personal characteristics and income. We implemented three models: Linear Regression with Interaction Terms, Random Forest, and XGBoost, each chosen for their unique strengths in prediction and interpretation.

3.1 Train Test Split of Data

We have cleaned our data before we split it into training and testing datasets. We are aware of potential data leakage of our testing data, and we have mitigated this risk by not using overall statistics or normalization that requires the original data to calculate, not imputing any missing data, or engineering features that include the original dataset. By doing so, we can minimize the risk of data leakage, and ensure the perfect seventy and thirty percent train test split.

3.2 Linear Regression Model with Interaction Term

3.2.1 Model set-up

We implemented a linear regression model with an interaction term to explore the relationship between personal characteristics and income. The general form of the model is:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \beta_{\text{interaction}}(X_i \times X_j) + \epsilon$$

where:

- y represents the total income (`INCTOT`),
- $X_1, X_2, ..., X_n$ are the predictor variables, including both categorical and numerical features,
- $X_i \times X_j$ is the interaction term between education and sex (`EDUC_SEX_INTERACTION`)
- β_0 is the intercept,
- $\beta_1, \beta_2, ..., \beta_n, \beta_{\text{interaction}}$ are the coefficients for the respective predictors and interaction term.

To ensure the robustness of our linear regression model, we conducted a series of steps aimed at minimizing multicollinearity and improving model interpretability. Initially, we included all relevant predictor variables, both categorical and numerical, to explore the full range of relationships between personal characteristics and income. Next, we calculated the Variance Inflation Factor (VIF) for each predictor to identify potential multicollinearity issues. Typically, a VIF value greater than 10 indicates problematic multicollinearity, which can distort the model's estimates. In our analysis, variables such as `VESTAT` (Veteran Status) and `AGE` had VIF values exceeding 13, suggesting they were highly correlated with other predictors. To address this, we removed these high-VIF variables from the model. Afterward, we refit the linear regression model using the reduced set of predictors, which resulted in improved performance metrics, including a lower Mean Squared Error and a more interpretable set of coefficients.

Note that according to our previous analysis in Section 2.2.3, the interaction term, `EDUC_SEX_INTERACTION`, was included to capture the combined effect of education level and gender on income. This allowed us to model the non-linear relationship between these two variables and their joint impact on the outcome variable.

Table 2: Coefficients from the Linear Regression Model

Feature	Coefficient
cat___MORTGAGE	11267.90
num___AGE	415.91
cat___STATEICP	126.39
cat___IND1990	-43.18
cat___OCC2010	-139.55
num___EDUC_SEX_INTERACTION	-1832.32
cat___MARST	-5541.39
cat___GQ	-7956.89
cat___SCHLTYPE	-16320.74

Table 2: Linear Regression Coefficients Table.

We run the model in Python (Python Software Foundation 2024) using the `scikit-learn` package (Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. 2024). We use the default settings for the `LinearRegression` function from `scikit-learn`. We use the `statsmodels` package (Seabold and Perktold 2010) to calculate the VIF for each predictor.

3.2.2 Model Justification

In evaluating our model's performance, we selected key metrics to assess both its fit and predictive accuracy, including the Variance Inflation Factor (VIF), R-squared (R^2), and Mean Squared Error (MSE). These metrics were chosen to provide a comprehensive evaluation of the model's robustness and reliability in capturing the relationship between personal characteristics and income.

The VIF was used to diagnose multicollinearity among predictors, helping us ensure that no variables were highly correlated, which could distort the interpretation of the model's coefficients.

Table 3: Variance Inflation Factors (VIF) for Predictor Variables

Feature	VIF
cat__STATEICP	3.24
cat__GQ	1.00
cat__MORTGAGE	2.66
cat__MARST	1.74
cat__SCHLTYPE	1.24
cat__OCC2010	3.19
cat__IND1990	8.49
num__AGE	7.87
num__EDUC_SEX_INTERACTION	6.95

Table 3: This table displays the Variance Inflation Factors (VIF) for each of the predictor variables in the model. VIF values greater than 10 indicate potential multicollinearity issues.

We chose R^2 as it provides a measure of how well the model explains the variance in the dependent variable, making it an important indicator of the model's explanatory power. The MSE was selected to quantify the average squared difference between observed and predicted values, offering a clear view of the model's prediction error.

Table 4: Model Performance Metrics

Metric	Value
R-squared	0.0774
Adjusted R-squared	0.0774
Mean Squared Error (MSE)	7,448,800,507.01
Root Mean Squared Error (RMSE)	86,306.43

Table 4: This table presents the performance metrics of the linear regression model, including MSE, RMSE, R-squared, and Adjusted R-squared. These metrics are used to assess the model's fit and predictive accuracy.

Additionally, residual analysis was conducted by plotting residuals against fitted values to check for randomness around zero, ensuring that the model satisfied the assumptions of linearity and homoscedasticity.

We use the statsmodels package (Seabold and Perktold 2010) to calculate the VIF for each predictor. We use the matplotlib (Hunter and Droettboom 2024) and seaborn (Waskom 2021)

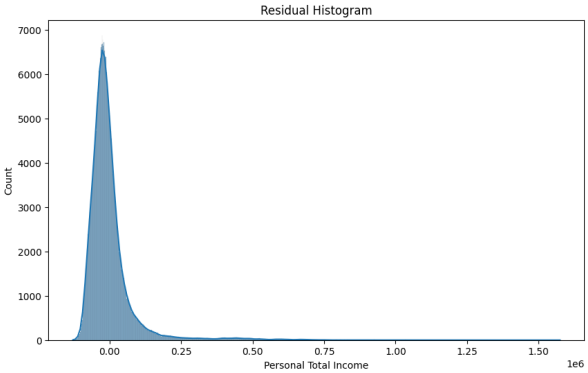

Figure 10: Linear Model Residuals Plot

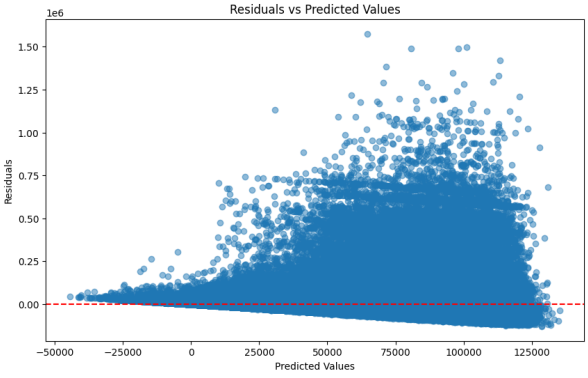

Figure 11: Linear Model Residuals vs. Predicted Values Plot

to visualize the residuals. We use the scikit-learn (Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. 2024) package to calculate the R^2 score.

3.3 Random Forest Model

3.3.1 Model set-up

To analyze the non-linear relationship between personal characteristics and income, we employed a Random Forest model. Random Forest is an ensemble learning method that builds multiple decision trees and averages their predictions, reducing overfitting and capturing complex, non-linear relationships in the data. We utilized Bagging (Bootstrap Aggregating) and random feature selection to enhance model performance and reduce overfitting. Bagging involves generating multiple bootstrap samples from the original training set, with each sample used to train an independent decision tree. By averaging the predictions of multiple trees, Bagging reduces the variance of the model, thereby improving its robustness. Additionally, Random Forest introduces further randomness by selecting a subset of features at each split, which reduces correlation among trees and improves the model's generalization ability.

To optimize the model, we conducted hyperparameter tuning, focusing on two key parameters: `n_estimators` (the number of trees) and `max_depth` (the maximum depth of each tree). Increasing `n_estimators` typically decreases model variance and improves stability, but too many trees can result in diminishing returns with increased computation time. Initially, we set `n_estimators` to 100, but through Grid Search, we tested values of 100, 200, and 300 and found that 300 trees yielded the best performance. Similarly, the `max_depth` parameter controls the complexity of the trees. While deeper trees can capture more intricate data patterns, they may also overfit. We initially set `max_depth` to 15 and test 10, 15, 20 but found that reducing it to 10 during hyperparameter tuning improved generalization by avoiding overfitting. Additionally, we tuned `min_samples_split` and `min_samples_leaf` to prevent overfitting by ensuring that nodes have a sufficient number of samples before splitting.

We run the model in Python (Python Software Foundation 2024) using the scikit-learn package (Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. 2024) and pandas (McKinney 2010). We used the GridSearchCV function from scikit-learn for hyperparameter tuning and subsampled the training data using pandas. The RandomForestRegressor function from scikit-learn was employed for model fitting.

3.3.2 Model Justification

To assess the performance of our Random Forest model, we evaluated several key metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), feature importance, and residual analysis. These evaluations provided insights into the model's predictive accuracy, interpretability, and overall fit.

First, we calculated the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) to assess the overall prediction accuracy of the model. These metrics provided insight into how well the model captured the variance in income based on the selected features.

Table 5: Best Random Forest Model Performance Metrics

Metric	Value
Best Model MSE	6,057,426,411.65
Best Model RMSE	77,829.47

Table 5: This table shows the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) for the best-tuned Random Forest model, demonstrating the model's performance after hyperparameter tuning.

Additionally, we performed a feature importance analysis, which allowed us to understand the relative significance of each predictor variable in determining income. This is particularly useful for verifying that the model identifies the key factors influencing income as expected.

Table 6: Feature Importance in the Best Random Forest Model

Feature	Importance
cat__EDUC_new	0.337
cat__IND1990	0.243
num__AGE	0.194
cat__SEX	0.149
cat__MARST	0.056
cat__MORTGAGE	0.013
cat__SCHLTYPE	0.004
cat__VETSTAT	0.003

Table 6: This table lists the feature importance values for the best-performing Random Forest model. The higher the importance value, the more significant the feature is in predicting the

target variable (income).

We also conducted a residual analysis by plotting the residuals (the difference between actual and predicted values) against the predicted values. This step helped us evaluate whether the model appropriately captured the underlying patterns in the data, ensuring that there were no significant biases or violations of model assumptions.

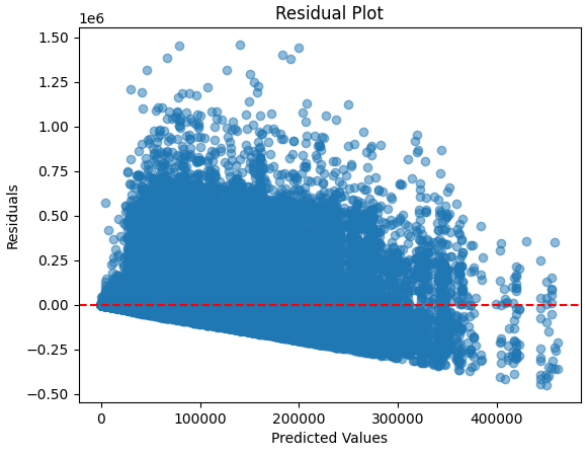

Figure 12: Random Forest Model Residuals vs. Predicted Values Plot

The feature importances were extracted from the best model using RandomForestRegressor from scikit-learn (Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. 2024) and displayed using pandas (McKinney 2010). We use the matplotlib (Hunter and Droettboom 2024) to visualize the residuals.

3.4 XGBoost

3.4.1 Model set-up

XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm that builds an ensemble of decision trees using gradient boosting techniques. In gradient boosting, each subsequent tree is trained to correct the residuals of the previous trees, iteratively improving the model's predictive accuracy. XGBoost enhances this process by incorporating regularization techniques to control overfitting, making it an ideal choice for capturing complex, non-linear relationships in data.

To optimize the performance of our XGBoost model, we employed GridSearchCV to conduct hyperparameter tuning. We explored a range of hyperparameters to find the optimal configuration for our dataset. The grid search included the learning rate (`learning_rata`),

max depth (`max_depth`), number of trees (`n_estimators`), and minimum child weight (`min_child_weight`). Specifically, we tested learning rates of 0.01, 0.05, and 0.1, tree depths of 4, 6, and 8, and `n_estimators` values of 100, 200, and 300. Using GridSearchCV with a cross-validation fold of 3, we systematically evaluated different combinations of these hyperparameters. The model was evaluated using negative Mean Squared Error (MSE) as the scoring metric. The optimal model used a learning rate of `0.1`, a maximum tree depth of `6`, a minimum child weight of `5`, and `300` boosting rounds. This configuration provided a good balance between model complexity and predictive accuracy, capturing important non-linear relationships while minimizing overfitting.

We implemented the XGBoost model in Python (Python Software Foundation 2024) using the scikit-learn package (Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. 2024), with pandas (McKinney 2010) for data processing and xgboost (Cho and Yuan 2024) for model fitting. The initial model was trained using default parameters of the XGBRegressor function, which was subsequently fine-tuned using GridSearchCV from scikit-learn (Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. 2024) for hyperparameter optimization.

3.4.2 Model Justification

Similar to the Random Forest model, the XGBoost model was trained on a 70-30 train-test split. We evaluated the model's performance using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) to assess its predictive accuracy.

Table 7: XGBoost Model Performance Metrics

Metric	Value
Mean Squared Error (MSE)	5,854,629,848.13
Root Mean Squared Error (RMSE)	76,515.55

Table 7: This table shows the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) for the final XGBoost model, reflecting its predictive accuracy.

We performed a residual analysis by plotting the residuals against the predicted values to assess the model's fit. This allowed us to check whether the XGBoost model accurately captured the patterns in the data and to verify that no major biases or violations of model assumptions were present.

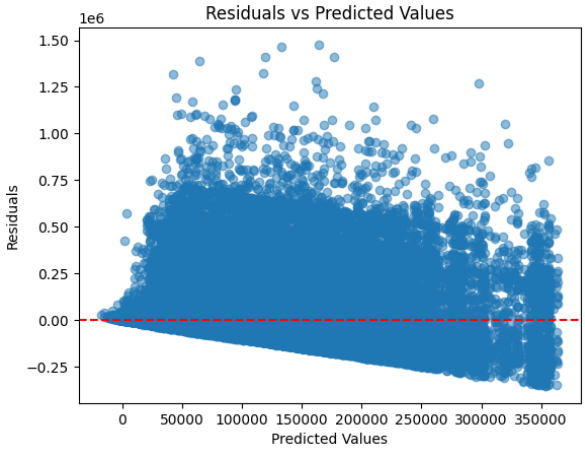

Figure 13: XGBoost Model Residuals vs. Predicted Values Plot

We also performed cross-validation to ensure the robustness of the XGBoost model. This technique allowed us to evaluate the model's performance across different subsets of the data, helping to confirm that the model generalizes well and is not overly sensitive to any particular portion of the training data.

Table 8: XGBoost Cross-Validation Performance

Fold	RMSE
1	76,693.86
2	75,692.41
3	76,193.42
4	77,365.82
5	76,552.10
Average	**76,499.52**

Table 8: This table shows the Root Mean Squared Error (RMSE) from cross-validation, demonstrating the model's performance across different subsets of the data.

We used matplotlib (Hunter and Droettboom 2024) to perform a residual analysis against the predicted values from the XGBoost model. We performed cross-validation using cross_val_score function from scikit-learn (Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. 2024) to evaluate the robustness of the model.

4 Results

4.1 Linear Regression Model with Interaction Term

4.1.1 Linear Regression Results

Table 3.2.1 provides the coefficients for each predictor in the model. The coefficients represent the estimated change in the target variable (income) for a one-unit change in the predictor, holding all other variables constant.

The largest positive effect on income is observed for `cat_MORTGAGE`, which has a coefficient of 11,267.90. This suggests that individuals with a mortgage tend to have significantly higher incomes compared to those without, when other factors are held constant. Additionally, the coefficient for `num__AGE` is 415.91, indicating that income tends to increase with age, though the effect size is relatively small.

On the other hand, some predictors have negative effects on income. The variable `cat__SCHLTYPE` (school type) has the largest negative coefficient of -16,320.74, suggesting that attending certain types of schools might be associated with lower income levels. Similarly, `cat__GQ` and `cat__MARST` (marital status) have negative coefficients of -7,956.89 and -5,541.39, respectively, implying that certain marital statuses or living arrangements are associated with lower income levels.

Interestingly, the interaction term `num__EDUC_SEX_INTERACTION` has a negative coefficient of -1,832.32, suggesting that the relationship between education and income may differ by gender. The negative value could indicate that the combined effect of education and gender has a suppressing effect on income for certain groups.

4.1.2 Model Performance

As shown in Table 3.2.2, the linear regression model explains a small portion of the variance in income, as evidenced by the R-squared value of 0.0775, which indicates that only 7.75% of the variability in income is explained by the predictors in the model. The Adjusted R-squared, which adjusts for the number of predictors, is slightly lower at 0.0775, confirming that the inclusion of additional predictors did not drastically improve the model's explanatory power.

Furthermore, the model's Mean Squared Error (MSE) is 7,448,800,507, and the Root Mean Squared Error (RMSE) is 86,306.43, suggesting that the model's predictions deviate significantly from the actual income values. These performance metrics indicate that the linear regression model has limited predictive accuracy for this dataset.

4.1.3 Residual Analysis

The residual analysis helps evaluate the linear regression model's performance. The residual histogram as shown in Figure 10 shows a strong right skew, indicating that while most predictions are close to the actual values, the model struggles with higher incomes. The long tail suggests significant under-prediction for some individuals.

In the residuals vs. predicted values plot as shown in Figure 11, the residuals exhibit a funnel shape, showing increasing variance as predicted income rises. This indicates heteroscedasticity, meaning the model's error increases for higher predicted values. Overall, the analysis suggests that the linear model fits well for lower incomes but underperforms for higher-income predictions, implying a need for more complex models to better capture these relationships.

4.2 Random Forest

We employed a Random Forest model to predict income, tuning several hyperparameters to improve model performance. The best model was achieved with the following parameters: **max_depth=10**, **min_samples_leaf=4**, **min_samples_split=10**, and **n_estimators=300**. As we shown in Table 3.3.2, the final model had a Mean Squared Error (MSE) of 6,057,426,411.65 and a Root Mean Squared Error (RMSE) of 77,829.47, indicating a substantial improvement in prediction accuracy compared to the linear regression model.

4.2.1 Feature Importance

The Random Forest model, Table 3.3.2, also allows for an examination of feature importance, which ranks the variables by their contribution to the model's predictions. The most important features in the model are:

- `cat__EDUC_new` (Education level) with an importance score of 0.337, indicating that education is the most significant factor affecting income in this model.
- `cat__IND1990` (Industry) has an importance score of 0.243, suggesting that the industry in which a person works is also a major determinant of income.
- `num__AGE` (Age) comes in third with a score of 0.194, implying that age plays a substantial role in income prediction.

Other variables, such as `cat__SEX` (Sex) and `cat__MARST` (Marital Status), show moderate importance, while factors like `cat__MORTGAGE` and `cat__SCHLTYPE` (School Type) have much smaller impacts.

4.2.2 Residual Analysis

The residuals from the Random Forest model as shown in Figure 12 were analyzed similarly to the linear model. The residuals plot, comparing predicted values with residuals, shows less heteroscedasticity compared to the linear model, suggesting that the Random Forest model handles the data more effectively, particularly for higher income levels. However, some degree of variability in the residuals still increases with predicted income, indicating that further tuning or model adjustments may be necessary for optimal performance.

4.3 XGBoost

The XGBoost model, a gradient boosting algorithm, was optimized using a hyperparameter grid search, yielding the best parameters: **learning_rate=0.1**, **max_depth=6**, **min_child_weight=5**, and **n_estimators=300**. These parameters were selected to balance model complexity and overfitting control. As we shown in Table 3.4.2, the final model achieved a Mean Squared Error (MSE) of 5,854,629,848.13 and a Root Mean Squared Error (RMSE) of 76,515.55, making it the best-performing model among the three evaluated.

4.3.1 Model Performance

The XGBoost model's Cross-Validation RMSE (Table 3.4.2) was 76,499.52, which aligns closely with the final RMSE, indicating that the model generalizes well across different data subsets. This consistency in performance suggests that XGBoost effectively captures the relationship between predictors and income without overfitting.

4.3.2 Residual Analysis

The residuals vs. predicted values plot as shown in Figure 13 shows a similar pattern to the Random Forest model, with residuals clustering around zero for lower predicted values but spreading out for higher predicted incomes. However, compared to the Random Forest model, the XGBoost residuals show a slightly narrower spread, particularly at the higher income levels, indicating that XGBoost handles high-income predictions somewhat better. Nonetheless, some degree of heteroscedasticity remains, as the spread increases with predicted income.

5 Discussion

In this analysis, we explored three different models—Linear Regression, Random Forest, and XGBoost—to predict income based on various demographic and categorical factors. Each model provided valuable insights into the relationship between income and predictors such as

education, age, industry, and gender. Among the models, XGBoost demonstrated the best performance in terms of predictive accuracy, as evidenced by its lower RMSE and superior handling of complex data interactions. However, this does not mean it is without limitations or that further improvements cannot be made.

5.1 Key Findings

5.1.1 Feature Importance

We have found that educational attainment and occupation industry are consistently the most important features in all three models, especially in XGBoost and Random Forest. This indicates that higher educational attainment and certain industries are correlated with higher personal income. This confirms our original hypothesis that certain demographic factors, such as education, may lead to specialized industry sectors where more income can be generated.

Age is also a useful predictor. However, despite the common belief that age is a determining factor when predicting personal income, its effectiveness is not as strong as education or industry. This may be caused by the fact that in some industries, your experience may hit a plateau, after which age may not contribute as much to experience as before certain age.

Marital status and sex do affect personal income, but not as significantly as the other features. It is great to prove that certain demographical features are not as informative when predicting personal income, suggesting a "fair" distribution of wealth.

5.1.2 Models

None of the three models used in our study provided an excellent prediction of personal income. For the residual analysis, it appears that the models' predictbility deterioate as the self-reported income gets higher. REsidual variance increased as the predicted income increased in the Random Forest and XGBoost model, suggesting potential heteroscedasticity. Our educated guess is that extra-high income may come from various sources, and simplifying them into a simple model like this is not a reasonable solution.

Among the three models, XGBoost has the highest accuracy overall. We believe that it can capture more complex relationships and interactions among the variables than the other two models. However, the high MSE for extra-high income individuals indicates that the model still needs to be refined more to achieve an accurate prediction.

Linear Regression has underperformed relative to the other two models. We speculate that simplicity in modeling linear relationships did not provide sufficient information to capture the non-linear patterns underlying our data, especially where multiple features interact to influence income, as suggested in Section 2.2.

5.2 Weaknesses and Limitations

One major limitation encountered throughout the analysis was the skewness of the income distribution. The income data is heavily left-skewed, with most individuals earning relatively low amounts and a few individuals earning substantially higher incomes. This skewness likely contributed to the residual patterns we observed, particularly in the linear and Random Forest models.

We attempted to address this by applying a logarithmic transformation to income in an effort to normalize the data. However, this transformation did not lead to substantial improvements in model performance. Even after the transformation, the residual patterns remained, indicating that the underlying relationships between predictors and income might require more sophisticated modeling approaches than simple transformations can provide.

Another potential solution—though not ideal—would be to exclude extreme income values from the analysis. Removing high-income outliers might reduce the heteroscedasticity issue and result in a better fit for the majority of the population. However, excluding high-income individuals would also mean losing valuable information about the upper end of the income distribution, which is crucial for a complete analysis. Before taking this step, it would be necessary to justify it based on domain-specific knowledge or further data exploration.

5.3 Next Steps

Looking forward, there are several potential avenues for further research and model improvement:

- While we focused on standard predictors like age, education, and industry, additional variables or interaction terms might improve model performance. For example, interactions between age and education, or industry and gender, could capture more nuanced relationships.
- Instead of removing high-income individuals, more sophisticated outlier treatment techniques could be applied. For instance, applying quantile regression or using robust regression models might help better model extreme values without distorting the broader data patterns.
- The skewness in income distribution poses a significant challenge for modeling. Exploring techniques like synthetic data generation for high-income individuals or applying weighting schemes to account for data imbalance might help the models better capture the variance across all income levels.

In conclusion, while XGBoost provided the most accurate results, there is room for improvement in handling skewed data and better modeling high-income individuals. Future work will focus on addressing these issues while exploring more advanced modeling techniques and feature engineering to enhance predictive performance further.

6 Appendix

6.1 Data Cleaning

In the data cleaning process, we first examined all required variables and removed any observations with missing values (N/A). Notably, the variable `CITY`, which we initially intended to include in the analysis, was excluded due to its high proportion of missing values, which compromised data completeness.

For the response variable, personal total income (INCTOT), we identified several non-informative coded values, such as `0000000 = None`, `9999998 = Unknown`, and `-009995 = -$9,900 (1980)` and so on. These values lack meaningful information for analysis, so we removed any records containing such entries to maintain the integrity of the income data.

In terms of predictor variables, we found that several included codes like "not applicable" or "does not exist," which represent cases without relevant data for our analysis. These values were also removed to ensure only relevant data points were used in modeling.

Furthermore, we simplified the `EDUC` variable (education level), which originally included highly granular categories (e.g., fifth grade, sixth grade, second year of college, fifth year of college). Such detailed classifications were unnecessary for our purposes, so we re-coded `EDUC` into a new variable, `EDUC_new`, with broader categories:

- 'No Schooling': 0,
- 'Nursery School': 1,
- 'Primary School': 2,
- 'Middle School': 3,
- 'College 1-4 Years': 4,
- 'College 5+ Years': 5

This simplified categorization enhances interpretability and usability in the analysis. By conducting these data cleaning steps, we produced a refined and consistent dataset, ready for accurate and reliable modeling.

References

Autor, David H. 2013. "The "Task Approach" to Labor Markets: An Overview." *Journal for Labor Market Research* 46 (3): 185–99. https://doi.org/10.1007/s12651-013-0125-7.

Cho, Hyunsu, and Jiaming Yuan. 2024. *XGBoost Python Package*. XGBoost Developers. https://xgboost.ai/.

Hunter, John D., and Michael Droettboom. 2024. *Matplotlib: Python Plotting Package*. Python Software Foundation. https://matplotlib.org.

IPUMS. 2024. "IPUMS API." IPUMS; Data retrieved via IPUMS API.

McKinney, Wes. 2010. "Data Structures for Statistical Computing in Python." In *Proceedings of the 9th Python in Science Conference*, edited by Stéfan van der Walt and Jarrod Millman, 56–61. https://doi.org/ 10.25080/Majora-92bf1922-00a .

Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. 2024. *Scikit-Learn: Machine Learning in Python*. Python Software Foundation. http://scikit-learn.org.

Plotly Technologies Inc. 2015. "Collaborative Data Science." Montreal, QC: Plotly Technologies Inc. 2015. https://plot.ly.

Pudney, Stephen. 2008. "Heaping and Leaping: Survey Response Behaviour and the Dynamics of Self-Reported Consumption Expenditure." ISER Working Paper Series.

Python Software Foundation. 2024. *Python: A Dynamic, Open Source Programming Language*. Python Software Foundation. https://www.python.org/.

Ruggles, Steven, Sarah Flood, Matthew Sobek, Daniel Backman, Annie Chen, Grace Cooper, Stephanie Richards, Renae Rodgers, and Megan Schouweiler. 2024. "IPUMS USA: Version 15.0 [dataset]." Minneapolis, MN: IPUMS. https://doi.org/10.18128/D010.V15.0.

Seabold, Skipper, and Josef Perktold. 2010. *Statsmodels: Econometric and Statistical Modeling with Python*.

Waite, Linda J., and Maggie Gallagher. 2000. *The Case for Marriage: Why Married People Are Happier, Healthier, and Better Off Financially*. New York, NY: Broadway Books.

Waskom, Michael L. 2021. "Seaborn: Statistical Data Visualization." *Journal of Open Source Software* 6 (60): 3021. https://doi.org/10.21105/joss.03021.