Wan Chen
Zhe Li
Gabriel Roth
Tiffany Yang
Horace Zhen

Census Income Level Prediction

Introduction

As citizens of a large country, it is important to understand the demographics of the areas surrounding us. The US census data contains the official recording of individuals across the nation; each region comprises the data, hence studying and understanding this data can help businesses and the government make decisions that help society thrive. The proposed project is to classify if the income level is below or above a threshold of 50,000 dollars. During this process, several data mining techniques would be applied to identify the important factors contributing to the classification. When the important factors are pinpointed, government leaders can focus on those areas to improve the income level.

The dataset used in the project is the census income dataset from UCI, which grouped the total personal incomes at the $50k level to present a binary classification problem [1]. Due to the inherent constraint of the dataset, we are only provided labels for below or above a threshold of $50k, which separates the upper class back to the year 1994. Besides, due to privacy reasons, the actual personal income level is not given per row. In fact, each row is an aggregation of thousands of individuals. We are not able to and it is not useful to predict the income of each individual based on this dataset because the point of the census is to gather statistics about a sample. Thus, our current dataset is more suitable for a binary classification problem rather than a regression problem.

Individual income data points are widely spread, so it is unrealistic and impossible to analyze the entire population. By classifying the income level into binary categories, it helps us see the big picture of certain regions in an organized and easily digestible format. A few applications of this analysis are: narrowing down which regions require more public funding based on the proportion of low-income individuals, targeting specific regions for job creation and construction of facilities for higher education, and helping companies establish branches or purchase land for manufacturing or processing facilities in regions with perceived cheap land or housing.

Literature Review

Our literature review covers five papers, which are related to the analysis of census data using machine learning techniques. The first two papers from 2016 and 2018 focus on applying models such as Naive Bayes and Decision Trees to predict the income levels of individuals. The final three papers from 2010 and 2021 address issues that arise in census data, such as the bias and imbalance of gender, age, and race.

The massive US census dataset which contains detailed information on 3.5 million households enables policymakers to make wiser decisions to support economic growth by drawing insights for each economic class from the income domain via data mining. Work conducted by Sharath et al. [2] includes 5 modules of data mining including gender distribution in occupation, education-salary relationship, economic hierarchy using k-means and prediction of economic classes, Benford's law of US income, and the income distribution across all states.

In the education-salary relationship module, the authors discovered that professional degree holders earn more than doctorate holders. This paper also includes a classification module that predicts which economic class people belong to based on selected attributes using the Naive-Bayes, C4.5, and Boosted C5.0 classifiers. The authors first used k-mean clustering to generate six classes based on income as the economic hierarchy. They trained and compared three classifiers: Naive Bayes, C4.5 DT, and C5.0 DT. The accuracy scores were 48.13%, 51.3%, and 53.07% respectively.

Chakrabarty & Biswas [3] utilized a Gradient Boosting Classifier (GBC) on the 1994 UCI Adult Dataset to predict whether individuals earned less than or greater than $50,000. The Adult Dataset consists of 48,842 rows and 14 features, which is a subset of the updated Census Income dataset, which has 299,285 rows and 40 features. The researchers used an Extra Trees Classifier model for feature selection; extremely randomized trees are similar to random forests in that each tree learns from a bootstrapped sample and selects features from a random subset. The model also adds a condition where the threshold for feature selection is randomized. The two least useful features were dropped based on the results of the model's feature importances. For preprocessing, missing values were treated as an additional category. Before training, categorical variables were one-hot encoded, then the data was split randomly into a training and testing set using an 80/20 proportion. The Gradient Boosting Classifier is an ensemble model that follows the principles of boosting, which constructs weaker models sequentially. These individual models are slightly better than random guessing but learn iteratively by focusing on data samples with high weights. Samples that are misclassified are adjusted to have higher weight and correctly classified samples are adjusted to have lower weights. Thus, the combined output of the weak learners will improve over time. The prediction of a boosting model is typically a weighted average of the trained trees. Gradient boosting extends boosting to situations where the loss function is differentiable and gradient descent can be used to update the weights of the learners. Using grid search, the researchers found the optimal depth and number of estimators. The evaluation metrics used were accuracy, recall, precision, F1-score, and AUC. The test accuracy was 88.17%, which was slightly better than prior works that used XGBoost, and PCA with SVM.

The census dataset classification contains biases towards gender and race historically. Calders & Verwer [4] suggested that when using a naive Bayes classifier to classify high and low income for the census-income dataset, the male and female counts under high income are highly imbalanced. The number of male individuals was more than five times the number of females. When learning from this model, the prediction outcome created a bigger distinction in the dataset. Even when the gender feature was dropped from the dataset, the results were similar due to other correlated features which made indirect discrimination in the classifier. This was called the "red-lining effect". The proposed three naive Bayes classifiers tested the dataset to best remove the gender discrimination as well as the red-lining effect. The first approach is to modify the naive Bayes by adding the probability of the negative bias sensitive value to the positive class and removing the positive bias sensitive value from the positive class. The approach removed discrimination but not the red-lining effect. The second approach was the Two Naive Bayes models which trained two different models, one for males and one for females. The third approach was to add an unbiased latent variable by maximizing the likelihood function. After applying the three models to the census data, the decrease in bias was a lot more than the decrease in accuracy score. Of the three models, the Two Naive Bayes models were less dependent on the bias-sensitive value and gained a higher accuracy score without discrimination.

Data analysis that exploits personal information is crucial for enhancing public welfare in preventing infectious diseases, boosting social equity, and improving economic development. However, some results from data mining without a fairness-aware concept can damage society. Kamishima et al. [5] introduce three causes of unfairness: prejudice, underestimation, and negative legacy. Prejudice refers to the direct or indirect dependence between features. Underestimation is the case where classifiers produce unfair results because the training was incomplete or the models did not converge. Negative legacy is the issue of data imbalance. The authors proposed a new technique by introducing a regularizer to make the classifier independent from sensitive information and reduce prejudice. By adding the regularization term to the loss function and minimizing the resulting objective function, the model will make sensitive features less influential in the final determination. The authors compared their methods with Calders and Verwer's method (CV2NB) [4] on the same census dataset from the UCI repository. Compared with CV2NB, the prejudice regularization technique performs better only with specific parameter settings. In general, their PR technique is inferior to CV2NB with respect to the efficiency of prejudice removal, but it can learn how to account for the influence of different features on sensitive information. In addition, the prejudice regularization concept can be applied to any probabilistic discriminative classifier including the Naive Bayes classifier.

The census income dataset, also called the UCI Adult dataset, is extracted from the 1994 Current Population Surveys by the U.S. Census Bureau and has been used in several research papers to compare the fairness of different machine learning algorithms, but the dataset itself has some limitations. Ding et al. [6] examined the origin and limitations of the UCI Adult dataset, reconstructed a US new census dataset based on the existing data ecosystem, and provided access to the reconstructed datasets. The most significant problem of the UCI Adult dataset is that the target label for a person's income is binary; it only shows whether the income is less or greater than $50,000. The researchers found that the $50,000 threshold misrepresents the general population as this threshold is equal to the 76th quantile of US individual income in 1994. In addition, the UCI Adult dataset contains sensitive attributes such as race, age, and sex, which demonstrates demographic parity and influences algorithmic fairness. Therefore, in the reconstruction process, Ding et al. matched fifteen input features in the UCI Adult dataset to the current population survey (CPS) data from the IPUMS website. Besides, the researchers tested different individual income thresholds since the original threshold of the UCI dataset limits the external validity for algorithmic fairness. They generated a new collection of datasets with different income thresholds varying from $6,000 to $70,000 and trained a gradient boosted decision tree to evaluate the performance based on equality of true positive rates and equality of positive rates, or demographic parity. After the data reconstruction, the researchers also proposed multiple prediction tasks.

Data Description

The file size of the UCI census income dataset is 156MB and the number of rows is 299,285. There are 199,523 rows in the training set and 99,762 rows in the testing set. In the training set, 187,141 rows (93.79%) belong to the class whose income level is below $50,000, and 12,382 rows belong to the class whose income level is above $50,000. In the testing set, 93,576 rows (93.80%) belong to the class whose income level is below $50,000, and 6,186 rows belong to the class whose income level is above $50,000. There are a total of 41 features, which contain 8 numeric variables and 33 categorical variables. Please refer to Appendix A for each variable name and its type.

The numeric variables are age, wage_per_hour, capital_gains, capital_losses, dividends_from_stocks, instance_weight, num_persons_worked_for_employer, and weeks_worked_in_year. The wage, capital gains and losses, and dividends are related to an individual's finances. The number of people working for an employer refers to the number of people in a household that were working and the weeks worked in a year refers to how consistently a person worked.

The categorical variables are further divided into several categories: work, education, demographics, employment, residence, family, and veteran status. The work-related features are class_of_worker, industry_code, occupation_code, major_industry_code, and major_occupation_code. The class of worker refers to the type of business (federal, private, etc.) and the occupation codes are specific numbers used to identify the sector and title of jobs. The education variables are called education and enrolled_in_edu_inst_last_wk. The first variable specifies the highest degree awarded and the second specifies if the person is currently enrolled in any program. The demographic variables are marital_status, race, hispanic_origin, and sex, which cover basic self-explanatory attributes. The employment-related features are called member_of_a_labor_union, reason_for_unemployment, full_or_part_time_employment_stat, tax_filer_status, and own_business_or_self_employed. These variables are used to track if workers are in a union, if they are unemployed and the reason for unemployment, whether someone is working full or part-time, how taxes are filed (as an individual, jointly, etc.), and whether someone works as a contractor or owns their own business. The residence-related variables are called region_of_previous_residence, state_of_previous_residence, detailed_household_summary_in_household, migration_code_change_in_msa, migration_code_change_in_reg, migration_code_move_within_reg, live_in_this_house_one_year_ago, and migration_prev_res_in_sunbelt. These variables are used to track the region and state of the immediately prior residence, the legal status of an individual (relative, spouse, etc.), and migration information which explains if the movement is within the same area or not. The family variables are called family_members_under_18, country_of_birth_father, country_of_birth_mother, country_of_birth_self, and citizenship. These variables are used to account for the number of minors in a household, the birthplace of parents and children, and citizenship status. The veteran variables are called fill_inc_questionnaire_for_veterans_admin and veterans_benefits. These indicate whether individuals are veterans and if they are eligible for benefits. The last categorical variable is the year, which is either 1994 or 1995.

There is a reason that the UCI census income dataset is used for binary classification with a cut-off of $50,000. To differentiate between the middle working class and upper class which is considered "rich", a specific cut-off is required to understand the factors that distinguish the rich and middle working class. In 1994, the US median household income was $33,178 [7], and the US median personal income was $26,000 [6]. The $50,000 cut-off is approximately 150% of the median household income and 200% of the US per capita income which is considered to be high income in 1994. To break into the top 20% of the population, one has to make at least 150% of the median income [8]. According to the census data, the $50,000 cut-off corresponds to the 76th quartile of the income distribution for 1994 [6]. The $50,000 mark separates the top 24% of the population and the rest. Some researchers define rich to be 7 to 10 times higher than the poverty line which is one-third of the US median household income [8]. The US poverty line for a one-person household is $6,970, a four-person household is $14,350 [9]. The average income per person in a household is within the range of $3,587 to $6,970. The $50,000 cut-off is also right in

between the 10 times the poverty line range in 1994 which is considered to be high income in 1994. This cut-off is relatively accurate considering that there are 6.6% of the population with income over $50,000 for all ages and 10.8% of the population with income over $50,000 for ages between 18 to 65. To put inflation into perspective, the buying power of $50,000 in 1994 is equivalent to an average of $97,000 in 2022. In some metropolitan cities, such as San Francisco or New York City, the buying power can reach over $100,000 [10].

Methodology

We selected four interpretable classification methods from the scikit-learn library and mixed-naive-bayes library, which are Random Forest, Logistic Regression, Bernoulli Naive Bayes, and Mixed Naive Bayes. Basic preprocessing was performed to fill in missing values, drop columns, encode the categorical variables and the label, and standardize the features. First, a Random Forest model was fit to extract the feature importances. Next, a second dataset with a subset of important features was constructed. All of the models were trained on both the full dataset and the reduced dataset, then evaluated using a confusion matrix, precision-recall curve, and ROC curve.

Data Cleaning, Preprocessing, and Feature Selection

A small number of features contain missing values, which are indicated by the "?" symbol. In Table 1, the first column is the feature name, the second column is the number of missing values in the training dataset, and the last column is the number of missing values in the test dataset. Instead of removing these columns, the missing values were filled with "Unspecified" resulting in an additional category.

From the documentation of the dataset, the instance weight variable is necessary for interpretation, but should not be used as a feature in the classification models. Therefore, this column was separated and then removed from the DataFrame. Additionally, the variable detailed_household_and_family_stat contained a category that was present in the training dataset, but not in the test dataset, so this was removed to prevent a shape mismatch during model evaluation. Next, the label was encoded as a binary variable; originally, the values were " - 50000" and " + 50000." Afterward, the incomes below 50,000 are encoded as 0 and the incomes above 50,000 are encoded as 1. After using the get_dummies function to encode the categorical variables, there are a total of 472 columns.

For feature selection, a RandomForestClassifier was fitted to the training set with all columns. Features with importance greater than 0.005 were retained, then the classification models were retrained.

Feature Name	Train	Test
state_of_previous_residence	708	330
migration_code_change_in_msa	99696	49946
migration_code_change_in_reg	99696	49946
migration_code_move_within_reg	99696	49946

migration_prev_res_in_sunbelt	99696	49946
country_of_birth_father	6713	3429
country_of_birth_mother	6119	3072
country_of_birth_self	3393	1764

Results

The three classification methods selected are Naive Bayes, Random Forest, and Logistic Regression. On both the full and reduced datasets, the Naive Bayes variants (Bernoulli, Mixed) do not perform as well as Random Forest and Logistic Regression on all metrics besides recall. When trained and evaluated on the full dataset, the Random Forest model has identical performance to the Logistic Regression model according to specificity, precision, recall, F1-score, and accuracy. On the reduced dataset, the Random Forest model has higher recall and f1-score but lower specificity and precision compared to Logistic Regression. On both datasets, the Bernoulli Naive Bayes model has the highest recall by almost 0.5 but performs worse on all other metrics.

When evaluating the accuracy of the models, Random Forest performs the best on the full dataset with 95.36% test accuracy whereas Logistic Regression performs the best on the reduced dataset with 95.16% test accuracy. For the training accuracy, Random Forest achieves 99% on both the full and reduced datasets. Clearly, Random Forest is overfitting the training data in both situations since the training accuracy is nearly 100% but the test accuracy drops to 95%. The Mixed Naive Bayes model obtains 89.82% test accuracy on the full dataset and 90.87% test accuracy on the reduced dataset. The Bernoulli Naive Bayes model obtains 74.20% test accuracy on the full dataset and 83.35% accuracy on the reduced dataset. Thus, both Naive Bayes models are far behind the Logistic Regression and Random Forest models. We believe that the high increase in accuracy for Naive Bayes on the reduced dataset is that the unimportant features were a source of noise. After the noise was removed from the data, the likelihood probabilities were more accurate, allowing the classifier to make better predictions.

When interpreting the classification reports from scikit-learn, the LR and RF models are nearly identical for the full dataset. The Bernoulli Naive Bayes model has significantly more false positives and true negatives compared to the others. Although BNB and MNB achieve 74% and 86% overall accuracy respectively compared to the 95% accuracy of LR and RF, they have 90% and 83% recall whereas the others have 40% recall. However, the specificity of the Naive Bayes models are 73% and 87% while the others have 99% specificity. Furthermore, the precision for LR and RF is 73% while BNB and MNB have 18% and 29% precision.

Overall, LR and RF are good at detecting and correctly classifying samples belonging to the negative class but are worse at detecting samples from the positive class (higher precision, less recall). The Naive Bayes models have a higher recall, but less precision and specificity. For the reduced dataset, the NB model improves specificity by 10%, precision by 7%, and accuracy by 9%, but decreases recall by 8%. For MixedNB, the specificity improves by 3%, the precision improves by 6%, and the accuracy improves by 4% but the recall falls by 6%. For LR, the precision decreases by 1% and the recall decreases by 4%. For RF, the specificity decreases by 1%, the precision decreases by 7%, and the recall increases by 5%. Overall, the NB model improves when using the reduced dataset, while LR and RF lose some predictive power.

The ROC curves only include the BernoulliNB, Logistic Regression, and Random Forest models because the mixed-naive-bayes library is not compatible with the scikit-learn plotting functions. The ROC curves confirm that LR and RF have similar performance with 0.95 AUC and 0.94 AUC respectively, while NB has 0.92 AUC. The AUC decreases to 0.94, 0.92, and 0.91 for LR, RF, and NB respectively for the reduced dataset. These results are aligned with the classification reports since LR and RF have higher precision but lower recall and their overall performance is better than Naive Bayes. The tables below compare the models based on different metrics; the bold entries indicate the best model or models for that column.

Train and test accuracy on full dataset

Model	Train Accuracy	Test Accuracy
Bernoulli Naive Bayes	0.7418392866987766	0.7419959503618613
Mixed Naive Bayes	0.8639655578554853	0.8649185060443857
Logistic Regression	0.9533337008765906	0.9533589944066879
Random Forest	**0.9995288763701428**	**0.953589543112608**

Train and test accuracy with important features (impurity > 0.005)

Model	Train Accuracy	Test Accuracy
Bernoulli Naive Bayes	0.8318790314901039	0.8334636434714621
Mixed Naive Bayes	0.8944683069119851	0.8963132254766344
Logistic Regression	0.9515544573808533	**0.9515647240432229**
Random Forest	**0.9955894809119751**	0.9511036266313827

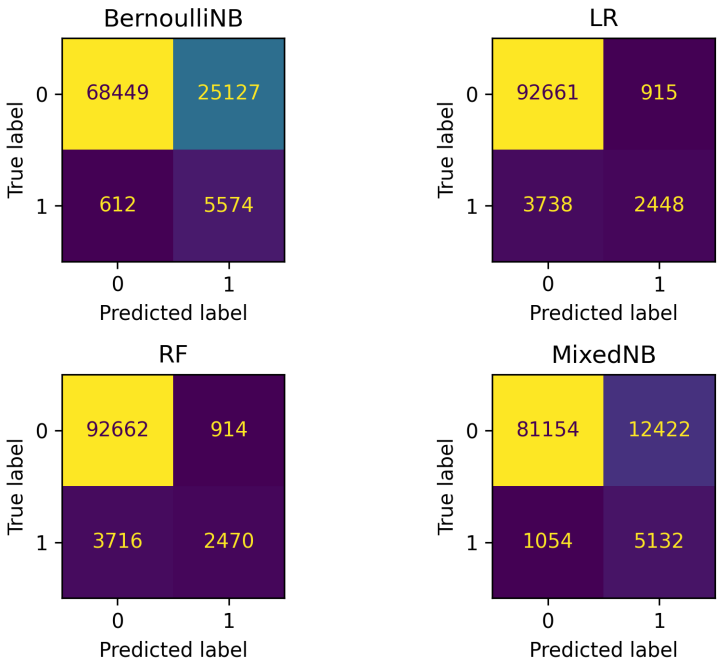
Metrics for full dataset

Model	Specificity	Precision	Recall	F1-Score	Accuracy
BernoulliNB	0.73	0.18	**0.90**	0.30	0.74

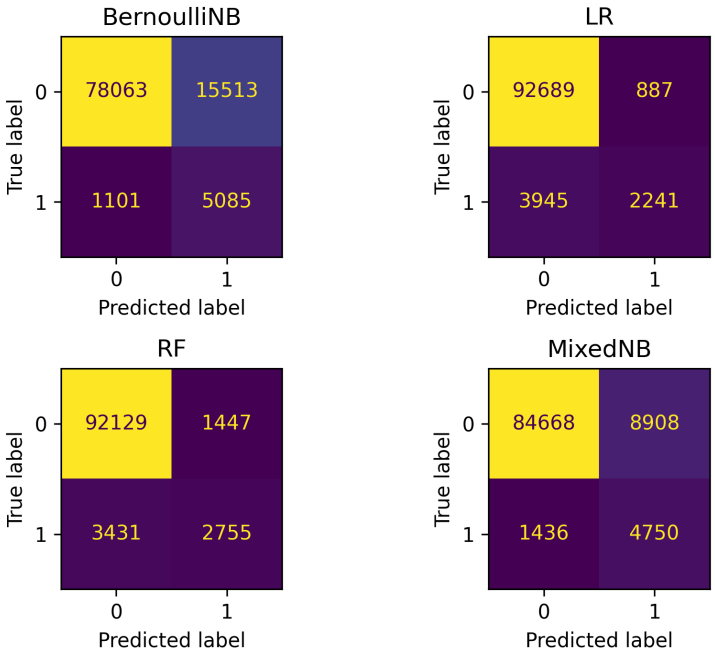
MixedNB	0.87	0.29	0.83	0.43	0.86
LR	**0.99**	**0.73**	0.40	0.51	**0.95**
RF	**0.99**	**0.73**	0.40	**0.52**	**0.95**

Metrics for reduced dataset

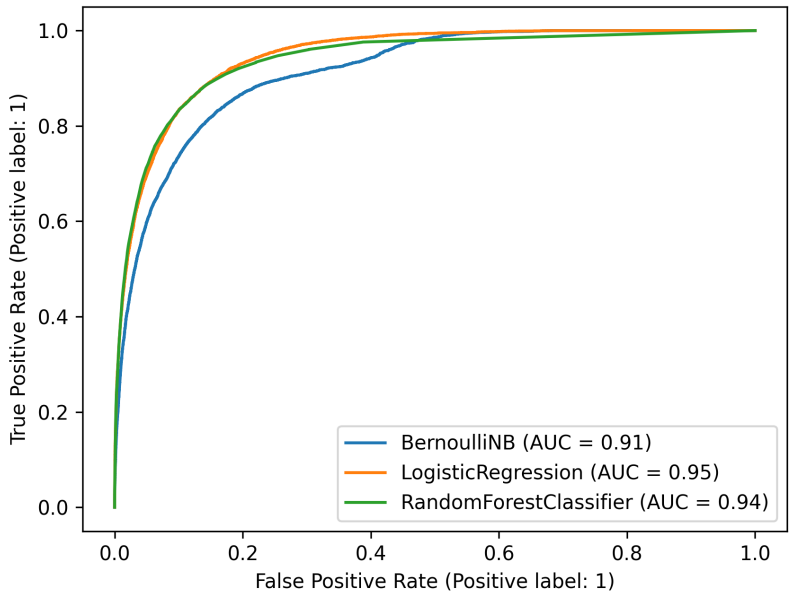
Model	Specificity	Precision	Recall	F1-Score	Accuracy
BernoulliNB	0.83	0.25	**0.82**	0.38	0.83
MixedNB	0.90	0.35	0.77	0.48	0.90
LR	**0.99**	**0.72**	0.36	0.48	**0.95**
RF	0.98	0.66	0.45	**0.53**	**0.95**



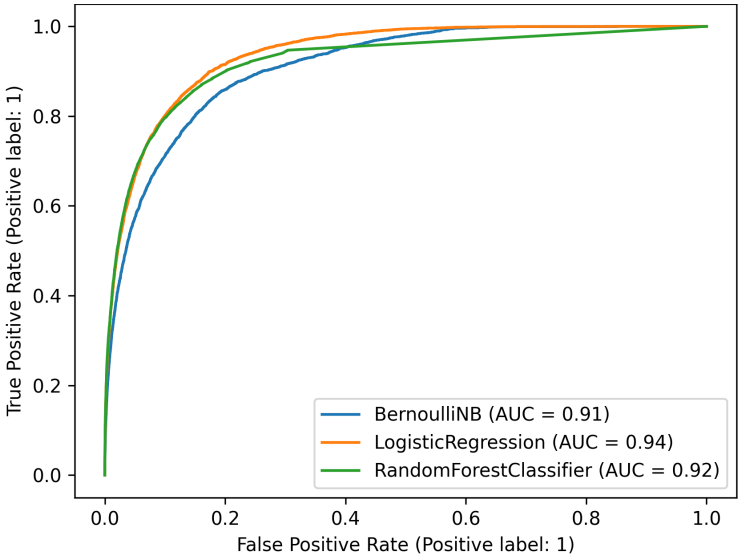
Confusion matrix for models trained on full dataset



Confusion matrix for models trained on reduced dataset



ROC curve for models trained on full dataset



ROC curve for models trained on reduced dataset

Discussion and Conclusion

Model Comparison

Excluding recall, the Random Forest classifier was quite good when trained on the full dataset compared to the reduced dataset. Because Random Forest is an ensemble model, it is able to combine the predictions of many decision trees that are trained on bootstrapped samples and random subsets of features. The diversity of decision trees and strength as a group allowed it to outperform the other models. On the other hand, when we reduce the number of features, the decision trees become less diverse and are unable to generalize as well. This is what caused a slight drop in performance on the feature subset. In general, tree-based models are capable of good performance on classification tasks but are prone to overfitting the training data. This is because trees can be grown to arbitrary depth and split variables until every data point is correctly classified. These rules or boundaries learned during training are not guaranteed to generalize to test data. As observed in our model evaluation, the training accuracy of the Random Forest classifier was almost 100%, but the test accuracy was 5% lower. Regardless, it is a strong model.

Although Logistic Regression is a simpler model compared to Random Forest, it was the most consistent model on both the full dataset and reduced dataset, meaning it was able to learn the differences between the two classes quite well. Since Logistic Regression is a probabilistic model, it returns a probability after applying the sigmoid activation function to the input (weighted sum). Since the sigmoid function is nonlinear, we believe the model was able to capture the complex relationship between the input features. Furthermore, the performance before and after feature selection implies the model was less sensitive to noise and less important features.

The Naive Bayes assumption is that the features are conditionally independent given the class. There are a large number of variables in the dataset, which Naive Bayes can easily handle, but there are also many categories within the variables. Therefore, this Bayesian approach needs a significant amount of data to get accurate likelihood probabilities. Furthermore, the assumption of conditional independence of features is not realistic for census data. An individual's income is

tied to many factors such as their education, their job, and their parents' education. Ignoring the relationship between these variables naturally results in worse performance.

Scikit-learn supports several types of Naive Bayes algorithms, but these cannot be mixed together. For instance, GaussianNB assumes that features are continuous, BernoulliNB assumes that features are boolean, and MultinomialNB assumes that features are counts. However, if the input data consists of both continuous and boolean features, then it would not be possible to use the native algorithms to process the data. Furthermore, converting all features to be continuous or boolean will result in lower accuracy because the models cannot learn the relationship between them. The mixed-naive-bayes library [11] is capable of mixing CategoricalNB and GaussianNB from scikit-learn. After specifying which columns are categorical and which columns are continuous, the MixedNB model is trained on the data. The performance of MixedNB is better than BernoulliNB since it is able to leverage information from both numerical and categorical features, but it is still weaker than the RF and LR models due to the Naive Bayes assumption.

In conclusion, the Logistic Regression model is the best model of all. It performs consistently on both full and reduced datasets. It gets the highest score on the metrics precision, specificity and accuracy. Compared to Naive Bayes, the LR model makes no assumptions about the distribution of classes. Compared to the Random Forest, the LR model is less inclined to overfit the training data. In addition, the reason why logistic regression performs well is that our data can naturally be represented as a combination of weighted factors such as weeks worked, capital gains/losses, gender, and education, which all contribute to higher income levels.

Features Interpretation

The full dataset after pre-processing contains 472 columns including the numerical features and dummy categorical features. Both Random Forest and Logistic Regression are on par in terms of model performance with full features at 95.35% and 95.33% respectively. While the model performance is achieved at a satisfactory level with full features, it is almost impossible to interpret features and investigate the relationship between features and outcome. Thus, the impurity-based feature selection method was adopted for narrowing down the number of features for training, making the result much more interpretable. The importance of a feature is calculated by the total amount of performance increase by incorporating the feature in training [12]. The importance score for all features sums to 1. We selected features with an importance score larger than 0.005 and resulted in 37 features. The 0.005 feature importance cut-off is the sweet point between the number of features and selected feature importance. At this cut-off, 37 features contribute 61.35% of the total importance of the full feature set. With the selected features based on the 0.005 cut-offs, the accuracy of the Random Forest and the Logistic Regression model only decreased by 0.2% while 435 features were disregarded. Feature selection resulted in training models with much fewer features which makes inference faster. The trade-off of 0.2% in accuracy for 435 features in reduction is worthwhile and it enables feature interpretation as well.

The lasso-based feature selection method adds a regularizer to the loss function of the linear model [13]. This forces the model to balance between reducing classification mistakes and making the model simpler. In this case, simpler means making the weights sparse (force non-important features' weights to be zeros) hence facilitating feature selection. We set the alpha parameter of the lasso model to 0.005 and selected features with non-zero weights, resulting in 16 features. The accuracy of the Random Forest and the Logistic Regression model only decreased by 0.5% while 456 features were disregarded.

To summarize, we used impurity-based feature selection methods for model comparison but features from both selection methods can be interpreted together. All 16 features selected by lasso regularization are included in the top features selected by the impurity-based feature selection method. Unlike the impurity-based method, the Lasso regularization method provides the direction and the magnitude of the impact of a feature. Features with positive values have a positive impact on classifying a person to the positive class which is the upper-income level, and features with negative values tend to have more weight towards the negative class which is the non-upper class. The additional information from the Lasso regularization helps to understand the way a feature impacts the outcome on top of the impurity score from the Random Forest model. In the impurity-based feature selection method, those features are important since they are the most used to split the data at nodes in the tree model. In the lasso-based feature selection method, those features are important since the weights of those features are used to calculate the propensity that data belong to certain classes.

Importance of Selected Features

According to the feature importance score from Random Forest feature selection:

- Age is one of the most important factors. The income of an average taxpayer rises dramatically as he or she ages and gains education and experience. Research shows that young college students who are fresh out of campus in particular contribute to a large number of low-income taxpayers [14].
- Education level (high school, master or doctorate) takes an important role in classification. Clearly, people with higher education have opportunities and access to better jobs that pay more.
- The numerical variables such as wage per hour, weeks in a year, dividends from stocks, and capital gains/loss contribute heavily to the income level for a straightforward reason since income is defined as money received from working or capital investment.
- Career choice is another important factor. The jobs like managers, administrators, and professional workers are inclined to have better income levels. Considering the number of people who work for him/her is a top factor as well, it indicates that human capital like managerial skills and professional skills are vital.
- The self-employed people who own their own business and people who are house owners are also a vital split for classification. In reality, only people with a high income afford to buy a house or run a business.
- Demographic factors like country of birth and race can also take a role in classification. Those whose parents were born in the USA may have connections to find a job easily. Furthermore, there are privileged individuals that have systemic advantages over minority groups and immigrants. This finding raises awareness of social unfairness.

According to the feature importance scores from Lasso Regularization, the sex_female feature has the lowest negative importance score, which impacts the outcome negatively. The $50,000 cut-off corresponds to the 76th quantile of the overall income distribution, and the 88th and 89th quantiles of the income distribution for the black and female groups respectively [6]. How does the $50,000 income cut-off affect algorithmic fairness? The objective of classifying the top 24% and the result provides evidence of gender inequality and racial inequality in 1994. Models trained with the $50,000 threshold add unfairness towards minority groups in 1994. The feature importance resulting from both the Random Forest and Lasso regularization method

resembles real-life scenarios. The sexuality factor in lasso feature selection confirms the assumption. The male feature importance is positive whereas the female feature importance is negative. The impact magnitude of the female factor is also larger than the male factor in the opposite direction. For inference, gender inequality in the workplace, including unequal pay and disparity in promotions are likely the root causes of the resulting data. This is also true for racial features, where race_white is selected from the RF method but not the race_black feature. Thus, the selected features are important and they can help us to narrow our focus to uncover the social issues with the $50,000 threshold.

What is the meaning of your result? How to explain your result (interpretability) based on your domain knowledge and references.

Having multiple income streams, especially passive income, is the key to being in the top 24% of the population. We observed that the impact of wage per hour on income is almost natural with an extremely slight negative impact. This means as the wage per hour increases, the likelihood of someone being classified as the upper class remains unchanged or even decreases slightly. On the contrary, capital losses and capital gains both have a high importance score and impact the outcome positively. Dividend from stocks is also the second most important feature in Random Forest. This implies that wage is not the sole stream of income for people who make over $50,000 a year. Passive income like capital gains, stock dividends, and other investment incomes are also important for those who want to break into the upper class. Capital losses are not always a bad thing depending on an individual's tax situation. For multi-business owners or those who have a large number of tax withholdings, having capital losses means that the losses can be deducted from their total income which might decrease the tax bill or even have a tax return. In these situations, capital losses can lead to an increase in disposable income for the current year. Age is also the most important factor that impacts the outcome positively. When we add the age factor to this context, the formula of making it to the top 24% becomes working hard, exploring multiple income streams, investing in capital, and starting to invest early. People tend to underestimate the power of investing when they are young. The value of one dollar is greater in the present than at any given time in the future because of inflation. And spending earnings immediately provides the greatest satisfaction. However, if one starts investing early in their life, with compound interest, the initial investment will grow significantly within an appropriate time horizon. For example, if a 25-year-old individual starts investing 600 dollars per month with an expected rate of return of 8% for only 10 years and lets it grow for another 30 years, the individual will have a million dollars at age 65. Assuming that the individual retires at the age of 65 and starts taking distributions from the million-dollar, with the same expected rate of return of 8%, the individual can withdraw $7,143.98 monthly for 30 years. In this case, the individual is high on age without earning wages, the expected annual income of $85,000 is much above the $50,000 cut-off. In other words, to be able to break into a higher social status or to increase your income significantly, one must create multiple income streams and start investing as early as possible.

Future Work

Although our model of choice is the logistic regression model because it has the highest accuracy, specificity, and precision on the reduced dataset, it is the worst model in terms of recall. However, we notice the Naive Bayes variants are good in terms of recall, which means they are more capable of finding data belonging to class 1. We conclude that it is a good idea to

explore an ensemble model or stacked model to combine both Logistic Regression and Naive Bayes to mitigate the weaknesses of the individual models.

A different approach to the classification task could be to remove the numerical variables wage per hour, capital gains, capital losses, and dividends from stocks. These features are directly tied to an individual's financial gain, which is why the models considered them important and were able to achieve high accuracy. It would be beneficial to perform an ablation study and analyze the feature importances and model performance using information from categorical variables only.

Over 90% of the data in both the train and test datasets belonged to class 0, or the groups where the income was below $50,000. Clearly, the data is imbalanced. To deal with this issue, we could over or undersampling algorithms such as Synthetic Minority Oversampling Technique (SMOTE) in future work. SMOTE is a data augmentation method that generates synthetic data to oversample the group with fewer samples (minority group) in order to even out the proportion of data belonging to each group. To properly evaluate the classification models, additional metrics such as Cohen's Kappa and Matthew's Correlation Coefficient (MCC) would be used. When data mostly belong to one class, it is important to consider metrics besides accuracy because it is not a balanced metric. As observed in our experiment, the GaussianNB model had high test accuracy because it classified every sample as the majority class. Since it never predicted the minority class, the model was useless.

As discussed by Ding et al., the UCI Adult dataset does not take algorithmic fairness into account [6]. Following the researchers' suggestion, we would use their reconstructed dataset and extend the binary classification task to a multi-classification task. Setting $50k as the threshold is more likely to result in bias as the income levels of the black population, women, and other minority groups are overrepresented. We would match the input features in the UCI dataset to the current population survey data from the IPUMS website and bin the personal incomes at multiple levels such as $10k to $60k. By doing this, we improve the fairness and validity of the data. Additionally, we can combine data from a longer period to analyze the trend of how important features affect the income threshold. This allows us to understand the geographic variation, temporal shift, and demographic parity throughout the years.

Appendix A: Data Description Table

Feature name	Data type (categorical w/# categories, int, or float)
age	int
class_of_worker	categorical (9)
industry_code	categorical (52)
occupation_code	categorical (47)
education	categorical (17)
wage_per_hour	int
enrolled_in_edu_inst_last_wk	categorical (3)
marital_status	categorical (7)
major_industry_code	categorical (24)
major_occupation_code	categorical (15)
race	categorical (5)
hispanic_origin	categorical (10)
sex	categorical (2)
member_of_a_labor_union	categorical (3)
reason_for_unemployment	categorical (6)
full_or_part_time_employment_stat	categorical (8)
capital_gains	int
capital_losses	int
dividends_from_stocks	int
tax_filer_status	categorical (6)
region_of_previous_residence	categorical (6)
state_of_previous_residence	categorical (51)
detailed_household_and_family_stat	categorical (38)

detailed_household_summary_in_household	categorical (8)
instance_weight	float
migration_code_change_in_msa	categorical (8)
migration_code_change_in_reg	categorical (10)
migration_code_move_within_reg	categorical (9)
live_in_this_house_one_year_ago	categorical (3)
migration_prev_res_in_sunbelt	categorical (4)
num_persons_worked_for_employer	int
family_members_under_18	categorical (5)
country_of_birth_father	categorical (43)
country_of_birth_mother	categorical (43)
country_of_birth_self	categorical (43)
citizenship	categorical (5)
own_business_or_self_employed	categorical (3)
fill_inc_questionnaire_for_veterans_admin	categorical (3)
veterans_benefits	categorical (3)
weeks_worked_in_year	int
year	categorical (2)
Binary label (<$50K or >$50K)	categorical (2)

References

[1] “Census-Income (KDD) Data Set,” *UCI Machine Learning Repository: Census-income (KDD) data set*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Census-Income+%28KDD%29.
[2] Sharath R., Krishna Chaitanya S., Nirupam K. N., Sowmya B. J., and K. G. Srinivasa, "Data analytics to predict the income and economic hierarchy on Census data," *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, pp. 249-254, 2016, doi: 10.1109/CSITSS.2016.7779366.
[3] Chakrabarty, N., & Biswas, S., “A statistical approach to adult census income level prediction,” *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pp. 207-212, 2018. doi: 10.1109/ICACCCN.2018.8748528.
[4] T. Calders and S. Verwer, “Three naive Bayes approaches for discrimination-free classification,” Data Mining and Knowledge Discovery, vol. 21, no. 2, pp. 277–292, 2010. doi: 10.1007/s10618-010-0190-x.
[5] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, “Fairness-aware classifier with prejudice remover regularizer,” *Machine Learning and Knowledge Discovery in Databases*, pp. 35–50, 2012. doi: 10.1007/978-3-642-33486-3_3.
[6] Ding, F., Hardt, M., Miller, J., & Schmidt, L., “Retiring adult: New datasets for fair machine learning, *ArXiv, abs/2108.04884*. doi: 10.48550/arXiv.2108.04884.
[7] "US Income Per Capita by Year," [Online]. Available: https://www.multpl.com/us-income-per-capita/table/by-year.
[8] A. B. Atkinson and A. Brandolini, “On the identification of the middle class,” Income Inequality, pp. 77–100, 2013.
[9] “Family poverty status and family poverty level variables,” *Family Poverty Status and Family Poverty Level Variables | National Longitudinal Surveys*. [Online]. Available: https://www.nlsinfo.org/content/cohorts/nlsy79/other-documentation/codebook-supplement/nlsy79-appendix-2-total-net-family-3.
[10] “Inflation rate between 1994-2022: Inflation calculator,” *$50,000 in 1994 → 2022 | Inflation Calculator*. [Online]. Available: https://www.in2013dollars.com/us/inflation/1994?amount=50000.
[11] “Mixed-naive-bayes,” *PyPI*. [Online]. Available: https://pypi.org/project/mixed-naive-bayes/.
[12]“Sklearn.ensemble.randomtreesembedding,” *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomTreesEmbedding.html#sklearn.ensemble.RandomTreesEmbedding.
[13] “Sklearn.linear_model.lasso,” *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html.
[14] E. York, “Average income tends to rise with age: Measuring income inequality,” *Tax Foundation*, 31-Jul-2020. [Online]. Available: https://taxfoundation.org/average-income-age/.