Predicting Individual Income

ECS 171 Machine Learning

Group 12 Project Report

Group Members:
Zachary Van Vorst, Yudi Lai, Saunak Singh, Yuval Danino, and Ryan Vakhshoori

Github repository:

https://github.com/Zack1243/171-Project-12

Introduction and Background

In contemporary society, an individual's reported income holds profound implications, influencing tax obligations, access to government assistance, and the extension of credit by financial institutions. However, the daunting task of meticulously tracking and verifying each person's income places an immense burden on manpower and resources. Complicating matters further, those exploiting the system often remain hidden within the masses, potentially compromising the integrity of tax systems and equitable distribution of assistance.

The best approach to efficiently processing this data, while remaining accurate, does not lie in devoting more resources and double-checkers but in the very method of processing the data itself. Replacing both the rigid system and people who rigidly interpret data on a surface level with a specialized system that can, with just a little more data, produce a more accurate, in-depth interpretation, results in the optimal solution; minimizing errors, catching malicious discrepancies in data, and freeing up manpower for other tasks.

In response to these challenges, machine learning emerges as a transformative force, offering sophisticated models capable of predicting an individual's income based on a myriad of factors beyond reported earnings. This innovation not only enables the identification of those circumventing the system with minimal resource expenditure but also facilitates a more accurate and targeted allocation of assistance to those genuinely in need. The only catch is that machine learning relies on additional data - a small cost that the current rigid system of data interpretation does not have. However, this is heavily outweighed by the benefits.

The implementation of machine learning models introduces the following benefits:

- Utilizing diverse factors beyond reported income, these models uncover individuals taking advantage of the system
  - Fosters fairness and compliance
- The integration of more accurate data into statistical studies reduces biases
  - Enhances the overall effectiveness of machine learning models
- This positive feedback loop of improvement contributes to a more equitable system for the masses
  - Individuals will obtain tangible benefits because of accurate data interpretation
- Will evolve the more it's utilized
  - Increases accuracy and reduces bias

As machine learning continues to evolve, there is a growing consensus that these technologies hold significant promise in enhancing various aspects of income prediction and financial assessments. Fintech companies, for instance, have embraced machine learning to specialize in advanced analytics for credit scoring and risk assessment. Operating at the forefront of innovation, these firms cater to individuals and businesses with limited traditional credit histories, thereby expanding financial inclusivity.

In essence, machine learning not only addresses the challenges of accurate income prediction but also fosters a more inclusive and fair financial landscape by optimizing credit assessments, risk evaluations, and the allocation of resources, contributing to a more just and efficient societal framework.

Literature Review

In the dynamic landscape of financial assessment, machine learning proves particularly advantageous when dealing with individuals needing more extensive credit history or familiarity with financial institutions. Traditional statistical models may exhibit susceptibility to errors in such scenarios, prompting the need for machine-learning techniques to bridge gaps in historical data.

These techniques excel in predicting an individual's future income and assessing their likelihood of financial responsibility. The ongoing evolution of research in this domain reflects the constant quest for more accurate methods to predict an individual's financial future.

Examining notable contributions to this field, Lazar's seminal work in 2004 stands out. Leveraging demographic data from the CPS spanning five decades and encompassing individuals aged 16 and older, Lazar applied machine learning techniques, specifically Support Vector Machines (SVM).

Notably, Principal Component Analysis (PCA) was employed to condense the data, reducing the number of independent variables while still encapsulating the dataset comprehensively. This innovative approach significantly minimized computing requirements. It is noteworthy that Lazar's study utilized the same dataset from UCI as our investigation, albeit employing distinct techniques, which will be elucidated later in this report.

Another noteworthy study by Matz SC, Menges JI, Stillwell DJ, and Schwartz HA (2019) endeavors to measure and predict an individual's income based on demographic features. Diverging from Lazar's methodology, they employ a different dataset and incorporate techniques such as Singular Value Decomposition (SVD) in conjunction with ridge regression and cross-validation.

Interestingly, while Matz et al.'s study yields a significantly lower r-value compared to Lazar's paper, it underscores the impact of diverse machine-learning techniques on model accuracy. The variation in predictive accuracy indicates that the choice of machine learning methodologies can lead to substantial increases or decreases in model precision. Notably, Lazar's paper also demonstrated near-statistically significant predictions based on an individual's demographic, further emphasizing the potential predictive power of these models.

These studies collectively contribute to the growing body of literature on machine learning applications in predicting income, highlighting the importance of methodology and dataset choice in shaping the accuracy and reliability of predictive models.

Dataset Description and exploratory data analysis of the dataset

The income dataset serves as a comprehensive repository of information on approximately 44,000 individuals, encompassing both their demographic details and earnings. This dataset forms the bedrock for developing and refining our predictive model, conveniently segmented into two distinct files: test.csv and train.csv, which would facilitate the training and testing phases of our machine learning models the test.csv file lacks any indication of individuals earning above or below the 50k, forcing us to disregard the file. We thus divided the train.csv file into training and test datasets with a ratio of 90:10.

Within each dataset, a rich array of records provides insight into crucial factors, including age, gender, race, hours-per-week, income classification (above or below 50k), education, marital status, and work class. This diverse set of variables ensures a holistic consideration of individual characteristics, contributing to a nuanced and accurate predictive model. For non-numerical terms, one-hot encoding was utilized via the ohe.pkl file in order to translate them into numerical values for the models to utilize.

To enhance the reliability of our analysis, meticulous steps were taken to address gaps in the dataset, particularly concerning certain individuals' statuses. In instances where data was incomplete, adjustments were made to mitigate potential biases. Additionally, individuals with incomplete data were either accounted for through careful adjustments or, when necessary, omitted from the dataset to maintain data integrity.

In addition, the overwhelming majority of data came from individuals with less than 50k income. The ratio was 76:24. To balance this, samples of the majority were compared against copies of the minority. We utilized the RandomOverSampler function in Python, which takes a random sample of the minority data, with replacement, and clones the samples until the minority data is balanced with the majority data.

We found this method of handling the data to be more accurate than the similarly named RandomUnderSampler, which takes random samples from the majority data to compare against the minority data. We believe this is because, put simply, more data is better than less data. Although both functions balance or scale the majority and minority data, RandomUnderSampler culls some potentially valuable data while RandomOverSampler utilizes all data present.

From then on, we attempted to standardize and rescale the data, however, strange trends appeared. While they are both commonly accepted and encouraged ways of cleaning up the raw data, our model barely reflected any changes. In some cases, the accuracy even decreased. For example, our random forest classifier's accuracy fell from 0.93779 to 0.93766. Therefore, we purposely omitted to standardize or rescale the data. After many attempts at processing the data, the only significant way of manipulating the raw data proved to be, as previously mentioned, RandomOverSampler. This should give a more accurate representation of the data to our model.

Lastly, we chose to omit the country of origin. This was because the feature proved the least correlated (a value of 0.014), but had an overwhelming bias since the overwhelming majority of individuals were from the United States. Therefore we omitted individuals not from the United States.

Maintaining a clear distinction between the training and testing phases, the final dataset, train.csv, is the culmination of a meticulous process of handling missing data. The result is a refined and robust dataset, free of N/A values, rescaled, and least biased, poised for effective model training and evaluation. This carefully curated dataset forms the foundation for our exploration into machine learning applications in predicting income based on demographic features.

Proposed Methodology

We implemented three machine learning models to predict individuals' income (below or above 50k per year) based on demographics. This task involves binary classification, and we aimed to build on previous analyses by employing random forest classifiers, logistic regression, and a neural network model.

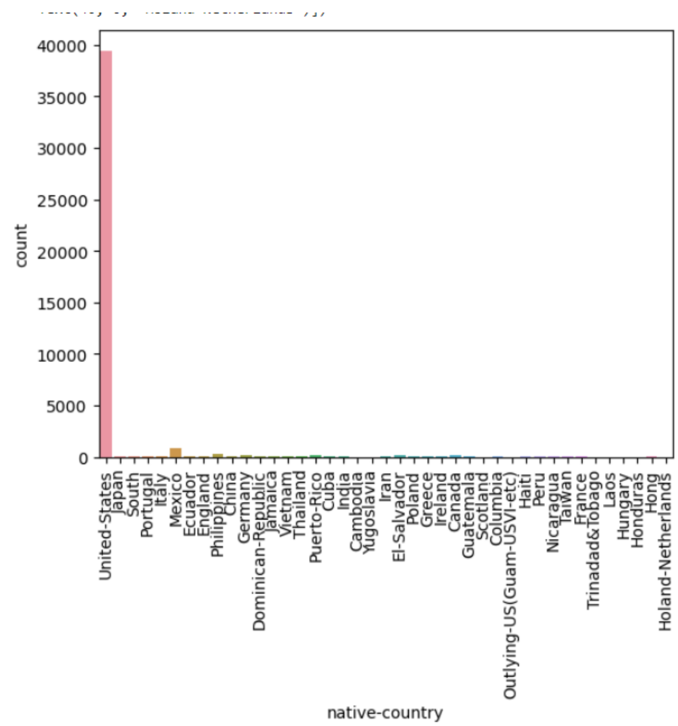

Figure 1: Disparity in data types. 0 is below and 1 is above 50k

While the initial model selection was straightforward, our focus shifted to a more comprehensive evaluation of each model's performance beyond relying solely on accuracy metrics. Additionally, we sought to ensure fair treatment of each model, assigning equal importance to hyperparameter tuning for all. Nevertheless, we acknowledged the inherent differences in the models, such as logistic regression's feature independence and random forest's unique treatment of hidden layers.

We strived to maintain consistency in certain variables, such as epochs/iterations, across models, but encountered challenges that compromised model accuracy. Consequently, we adjusted hyperparameters uniformly when possible, without negatively impacting model performance. However, adjustments were made selectively when significant differences warranted such modifications.

As previously mentioned in the exploratory analysis of the dataset, we tried to utilize various forms of scaling and normalizing the data, but ultimately ended up using RandomOverSampler to resize the minority data to match with the majority data.

For the web application, we simply allowed the

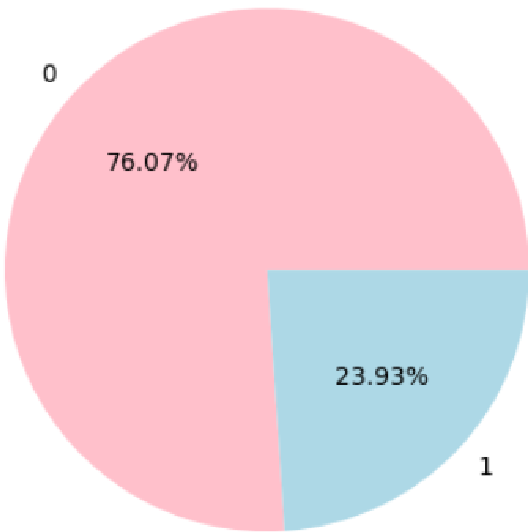

Figure 2: Disparity in data types. 0 is below and 1 is above 50k

user to fill information in for each feature such as age, race, education, etc. After filling in the information, the user can then click the button at the bottom to receive a prediction of whether they earn above or below 50k. We used our random forest classification model to form this prediction - the reason behind which will become clear later.

Experimental results

For logistic regression, we used an iter value of 1000 to get an accuracy of 0.82 and an MSE of 0.18. Other values are shown in the classification report and confusion matrix.

For the Random Forest Classifier, we utilized hyper-parameter tuning to choose both the most optimal estimator of 100 (chosen among values of 100, 150, or 300) and the most optimal number of features, 14 (chosen among values of 3, 5, 7, 14, or 20). With our two hyper-parameters tuned, we then implemented a 10-fold cross-validation to confirm our results. The result of the final model was an optimal accuracy of 0.93 and an MSE of 0.064.

For the Neural Network, we similarly used hyperparameter tuning, except on parameters such as our hidden layers and iterations. From hidden layers of (6, 8), (9, 13), or (13, 11), our program determined hidden layer (13, 11) to be the most optimal. From iterations of 500, 800, or 1000, our program determined 500 iterations to be the most optimal. We combined these two hyper-tuned parameters with a batch size of 100 and a learning rate of 0.3 to get our most optimal model, which reported an accuracy of 0.83 and MSE of 0.17.

```
Confusion Matrix for each label :
[[[2548  498]
  [ 584 2328]]

 [[2328  584]
  [ 498 2548]]]
Classification Report :
              precision    recall  f1-score   support

           0       0.82      0.80      0.81      2912
           1       0.81      0.84      0.82      3046

    accuracy                           0.82      5958
   macro avg       0.82      0.82      0.82      5958
weighted avg       0.82      0.82      0.82      5958
```

Figure 3: Logistic Regression Results

```
Confusion Matrix for each label :
[[[2980   66]
  [ 313 2599]]

 [[2599  313]
  [  66 2980]]]
Classification Report :
              precision    recall  f1-score   support

           0       0.98      0.89      0.93      2912
           1       0.90      0.98      0.94      3046

    accuracy                           0.94      5958
   macro avg       0.94      0.94      0.94      5958
weighted avg       0.94      0.94      0.94      5958
```

Figure 4: random forest classifier results

By far, the most optimal model for predicting the income of an individual proved to be the Random Forest Classifier - in terms of both its greater accuracy and lower MSE values. Neural Network was the second most optimal both in terms of MSE and accuracy, which were both barely more optimal than the logistic regression model.

One may argue, however, that a model's performance is not determined by MSE and accuracy alone. Precision, recall, and, most importantly, F-1 score provide more in-depth analysis

```
Confusion Matrix for each label :
[[[2394  302]
  [ 620 2046]]

 [[2046  620]
  [ 302 2394]]]
Classification Report :
              precision    recall  f1-score   support

           0       0.87      0.77      0.82      2666
           1       0.79      0.89      0.84      2696

   micro avg       0.83      0.83      0.83      5362
   macro avg       0.83      0.83      0.83      5362
weighted avg       0.83      0.83      0.83      5362
 samples avg       0.83      0.83      0.83      5362
```

Figure 5: Neural network results

of the model(s). Maybe logistic regression is better than Random Forest Classifier in predicting whether an individual is below 50k...except Random Forest classifier outperformed the other models in literally every metric, destroying any hope of an intellectual discussion of precision, recall, and F-1 score.

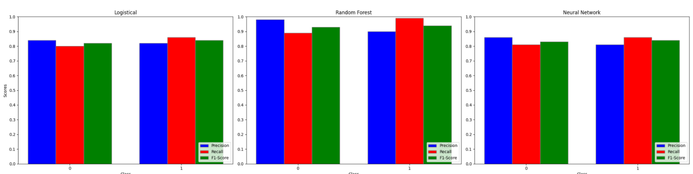

Figure 6: logistic vs random forest vs neural network

However, we must not forget our original objective of determining a more optimal model for the dataset than in past studies, which may resurrect this unborn discussion of precision, recall, and F-1 score. Our Random Forest model, once again, denied any possibility of debate, outperforming both Lazar's and Matz et al.'s in terms of performance in a statistically significant manner. In addition, the other studies didn't have these three metrics. We therefore focus our basis of finding a more optimal model on accuracy and MSE score.

Lazar formed models with SVM and PCA, with the best accuracy score reaching 42. Moving on to Matz et al.'s...

Matz et al.'s study employed two models, Singular Value Decomposition (SVD) and ridge regression, and combined them with cross-validation to eventually achieve their highest accuracy of 84 percent, doubling Lazar's most optimal model's accuracy.

Our more optimal model clocked in with 93 percent accuracy - with an extremely low MSE in comparison. This difference in MSE means that, if our model were to be a line on a graph, the datapoints would have very close to this line.

We have found that our random forest classifier is overall the most optimal model!

Conclusion and discussion

In conclusion, the demographics of individuals can indeed provide enough data for a machine learning model to accurately predict their income. Accuracy can be improved by better preparing the data before feeding it into a model and by improving the model itself. Governments can use this innovation not only to enable the identification of those circumventing the system with minimal resource expenditure but also to facilitate a more accurate and targeted allocation of assistance to those genuinely in need.

However, there is room for improvement. Just as our study assumed that there existed a better model and way of handling the data than in previous studies, it would be sheer arrogance to assume future studies won't do the same. Future studies may find ways and or models to compensate for biases and null values in the data.

As of now, by limiting the countries of origin to the United States, for instance, we have limited the scope of our model in real-world situations, which means our model/dataset handling is innately flawed even with all its improvements. Noteably, this may not always be a bad thing as more specialized models mean more accurate predictions. However, following this logic, one may claim that our model could have been more specialized by narrowing into states or cities. In addition, better scaling methods could be found in the future. We could also use more hyper-parameter tuning on our random forest classifier model to find even more optimal parameters and thus increase its accuracy.

Timeline:

Deadline: Task

10/22: Describing the problem scientifically

10/29: Background study (literature review or related work)

11/5: Dataset Understanding and Exploratory Data Analysis

11/12: Developing Accurate Prediction Model(s)

11/19: Evaluation of the model(s) and Testing the performance

11/26: Developing a basic web-based front-end to invoke and run the model(s) on input data and display the prediction output

12/01: Final report

References

Dataset: https://www.kaggle.com/datasets/mastmustu/income/data

Literature Review

Matz, Sandra C., Jochen I. Menges, David J. Stillwell, and Andrew H. Schwartz. "Predicting Individual-Level Income From Facebook Profiles." Plos One 14, no. 3 (March 28, 2019): 1–13. https://doi.org/10.1371/journal.pone.0214369.

Lazar, Alina. "Income Prediction via Support Vector Machine." 2004 International Conference on Machine Learning and Applications, December 16, 2004. https://doi.org/10.1109/icmla.2004.1383506.