# Using Classification Techniques to Predict the Adult Income

Ajay Bisht, Kuan-Chih Lee, Hao Li, Junmei Luo

***Abstract*—An individual's annual income is often governed by various factors. In the United States, there is diversity in work class and educational background among individuals, so the variability in annual income among individuals is expected to be more. In this project, the aim is to study the patterns and analyze the attributes that govern an individual's annual income through machine learning and data mining methods. Moreover, the purpose of this paper is to classify whether an individual will have an annual salary more than $50,000 or less than $50,000, based on certain given attributes. Dataset has class imbalance and the SMOTE technique was used to handle it. We used various tree-based models and neural network to achieve our goal. Based on True Positive Rate, accuracy and computational cost, model evaluation and model selection is done. Finally, we discuss the observations made from analysis and how better performance can be achieved for predictions.**

***Keywords — Machine Learning, Data Mining, Modeling, and Data Analytics***

## I. INTRODUCTION

Annual income in a society is an indicator of how prosperous the community as a whole is. As the world is becoming more global, people from diverse background converge and contribute in the society. Hence, it becomes important that improvements are made in areas which have big impact on the annual salary of an individual. The government can use the analytic results to focus on developing areas that will have major effects. For example, if higher education results in better annual income then more awareness programs can be initiated to encourage individuals to go for higher studies. Similarly, individuals can make decisions based on analysis results, such as, the focus on job profiles that are most rewarding in terms of annual salary or the amount of impact taking a house loan will have on their annual salaries.

This project aims to highlight key features which govern any individual's annual income. The goal of this project is to analyze any pattern, if exists, that governs an individual's annual income. Moreover, this project also uses machine learning algorithms to build a classifier that can accurately predict whether an individual with certain attributes will have annual salary more than $50,000 or less than $50,000.

The project explains methods that are followed to analyze, pre-process and engineer data. Visualizations are created to get insight into the data. The project also aims to explain the machine learning models used to build a classifier and their performances. Comparison of these machine learning models can be used to determine the best model that provides the best overall performance. Finally, the analysis report is discussed to form a conclusion and scope of improvement.

## II. DATA MINING TECHNIQUES

### *A. Data Collection*

The dataset is extracted from University of California Irvine (UCI) Machine Learning Repository [1]. The author of this dataset Barry Becker extracted it from the 1994 Census database. A set of reasonably clean records was extracted using the following conditions: Ages between 16 and 100 with work hours per week more than 0. The dataset consists of 14 features (input) and 1 response (output) variable. The output represents the salary that is either less than or equal to $50K or more than $50K. The number of records in the dataset is 32,562. The 14 features are 'Age', 'Workclass', 'Fnlgwt', 'Education', 'Education-number', 'Marital-status', 'Occupation', 'Relationship', 'Race', 'Sex', 'Capital-gain', 'Capital-loss', 'Hours-per-week', and 'Native-country'.

An important characteristic of this dataset is that it is an imbalanced dataset, i.e. the number of records with salary less than or equal to $50K is 76% while the number of records with salary more than $50K is 24%. To tackle this issue, we use the SMOTE technique to oversample the data.

### *B. Data Preprocessing*

#### *i. Missing Data*

For the categorical data, there are 1836 missing data in 'Workclass', 1843 missing data in 'Occupation', 583 missing data in ‘Native-country’. No missing data among numerical features were found. Missing data is replaced with ‘unknown’ to avoid losing some information from original dataset.

### *ii. Categorical Data*

Categorical attributes are encoded into numeric sequences to analyze the correlation between variables. ‘Workclass’ attribute was grouped into 5 different classes: ‘Govt’, ‘Self’, ‘Private’, ‘No-salary’ and ‘Others’. For ‘Education’, we grouped them into 6 classes: ‘below HS’, ‘HS’, ‘below bachelor and above HS’, ‘Grad’, ‘Master’ and ‘above masters’. For ‘Occupation’, we grouped them into 3 classes: ‘white collar’, ‘blue collar’ and ‘others’. For ‘Marital Status’, the data is grouped into binary form to indicate whether a person is currently living with a spouse or not. ‘Native County’ was grouped into binary form to indicate whether a person is from United States or not. ‘Race’, was grouped into binary form to indicate whether a person is white or not. This transformation gave us 15 features for further analysis.

Additionally, for the features with number of classes more than 1, one hot encoding is performed when data is fed into a neural network.

### *iii. Imbalanced Observations*

In classification, ‘class imbalance’ problem exists whenever there is a classification problem where one class has abnormally more samples than its counterpart. In this dataset, the ratio of size of data with output 0 to that with output 1 is almost 1:3. This will lead to inaccurate predictions. For instance, in the current scenario, it is more likely to predict that a data will have an output of 0. Even a naive approach on a model will result in high accuracy, but not high recall. Hence focus is put on sensitivity to see how much data with an output of 0 in the sample are successfully predicted as 0. Sensitivity is used as a parameter to improve models, rather than using accuracy.

Another technique to tackle ‘class imbalance’ problem is called SMOTE. It is an oversampling method that increases the size of samples in a minority class. SMOTE [2] One sample is selected from the minority class. Next, one neighbor amongst its nearest k neighbors is selected. Then, difference between its feature vector and the selected neighbor is taken and multiplied with the difference by a random number between 0 and 1. Finally, the result is added to the feature vector under consideration to generate one new data. By repeating this process N times, N new samples for the minority class are obtained.

### *C. Exploratory Data Analysis*

For exploratory data analysis, univariate, bivariate and multivariate analyses are performed on each attribute to observe any pattern.

Using correlation matrix for numerical variables, it is observed that ‘Age’, ‘Education’, ‘Capital-gain’, ‘Capital-loss’, and ‘Hours-per-week’ are related to target variable ‘Income’, while ‘Fnlgwt’ (weight) is not. To explore further, more plots are created, such as ‘Age vs. Income’, ‘Education vs. Income’, and so on.

Fig.1 and Fig. 2 shows that ‘Age’ and ‘Education Number’ are strong signals. And then these plots are generated together for comparison.

Furthermore, Fig. 3 shows that individuals who have income above $50k are usually more than 25 years old with ‘Education Number’ more than 8.5 (representing higher education). Although ‘Age’ and ‘Education Number’ cannot give us a strong prediction, they can tell us higher income associates with higher age

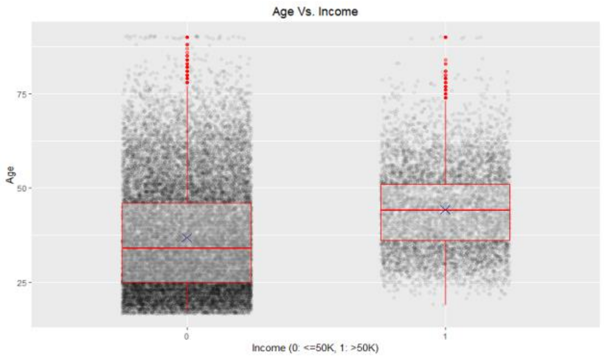

Fig. 1 Distribution of Attribute `Age` by Class

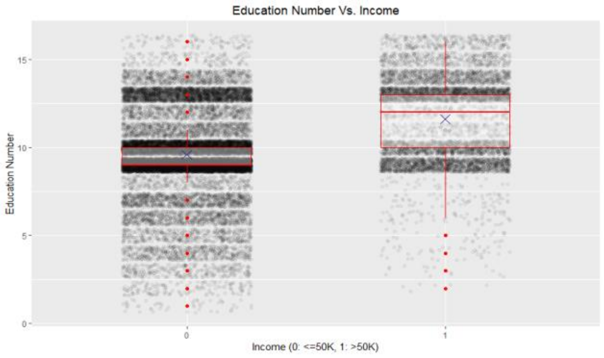

Fig. 2 Distribution of Attribute `Education Number` by Class

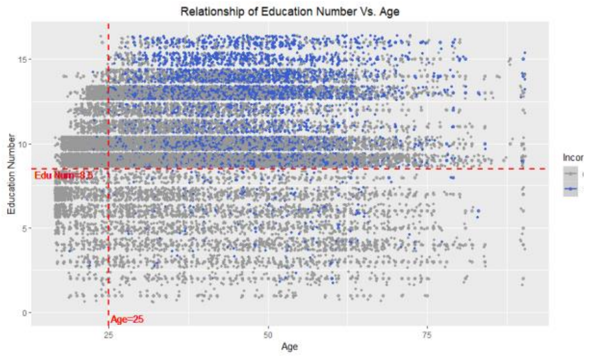

Fig. 3 Scatter Plot for Bivariate Analysis: 'Age vs. Education Number'

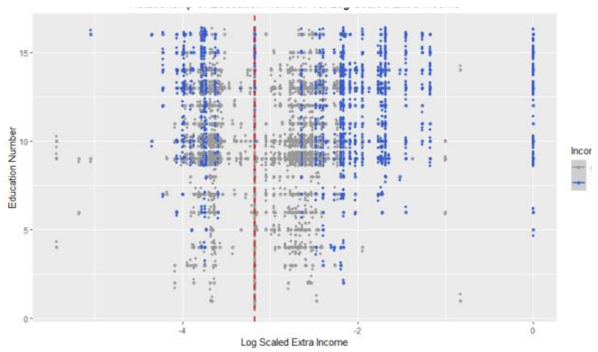

Fig. 4 Scatter Plot for Bivariate Analysis: `Extra Income vs. Education Number`

and education level.

On the contrary, some variables don't work well. Here, the 'Fnlgwt' and 'Extra.Income' (explained more in detail later) are not good signals for predicting income.

However, Fig. 4 shows that for some observations, such as individuals who have no matter capital gain or loss, they will be likely to have higher income. In other words, the blue points (label 1) are observed when capital gain and loss happened. It's also helpful for us to understand the insights from given dataset

### D. Modeling

#### i. Classification Tree

A single classification tree is obtained as the baseline model to compare it with other models in this report. Here, the best pruned subtree is selected by cost-complexity. The best parameter, cp=0.01, in R *library(rpart)* is selected by grid search.

#### ii. Bagging and Random Forest

The purpose of random forest is to establish plenty of maximal trees and vote for the best result from these constructed trees. The tuning parameter, m=f(p), is known as the number of predictors with replacement engaging in forest construction. With the given processed data, grid search is used to find the best performance in validation dataset with m=3 and the average tree size. The number of nodes in trees was kept at 1260.

#### iii. Classification Tree

Compared to random forest, tree boosting is created by iterating a tree stump (single split tree) to find the model with the smallest residual errors. In this paper, the tree stump experienced an update of 3000 times with shrinkage parameter 0.08.

#### iv. Neural Network

One-hot encoding is used to transform the data. This transformation gave us a total of 28 predictors.

A general structure of neural network is known. One important criteria of hidden layer design, is that a deeper network is better than a shallow one. Also, the first hidden layer reduces feature dimension. Because of this reason, the units in first hidden layer are chosen to be 20, which is less than the input layer units. Here, the neural network is designed with 3 hidden layers, each having 20 units. Learning rate was kept at 0.001 with momentum equal to 0.5.

## III. Results & Comparison

### A. Measurement

Two metrics are chosen to evaluate the model, accuracy and sensitivity. Accuracy measures the overall percentage of times the classifier is correct. It is calculated as follows:

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + FN}$$

Sensitivity, also known as true positive rate (TP), measures the percentage of predicting 'True' when the actual value is 'True'. It is measured as follows:

$$Sensitivity = \frac{\mathrm{TP}}{TP + \mathrm{FN}}$$

Since our data is imbalanced, it is a better approach to evaluate sensitivity along with accuracy, rather than accuracy alone.

## *B. Results*

### *i. Performance on Validation Set*

- *Classification Tree*

For best punned tree, the validation accuracy rate is 84.55%, but sensitivity is 52.2%. Table 1 shows the confusion matrix of classification tree model. Fig. 5 shows the best pruned subtree with 5 terminal nodes.

- *Bagging and Random Forest*

Based on grid search, the validation accuracy rate is 85.83%, but sensitivity is 59.22%. Table 2 shows the confusion matrix of Random Forest Model with m=3. Fig. 6 shows variable importance for predictors, which gives us insight of importance of attributes.

- *Boosting Tree*

For tree stump model, the validation accuracy rate is 85.25%, but sensitivity is only 48.05%, which is lower than 50%. Table 3 shows the confusion matrix of boosting tree model. Fig 7 shows the importance of each variable in the model. The top four important variables are 'Relationship' (25.115%), 'LogScaledExtra_Income' (24.789%), 'LogFnlwgt' (10.10%), and 'LogEdNum' (9.64%). The percentage in the brackets are the importance of each variable among all variables.

An interesting observation is made in 'Variable Importance' for Random Forest and Tree Boosting. Both of them use 'LogScaledExtra_Income' and 'Relationship' as top important variables. For random forest, the tree is split by 3 random chosen predictors, repeated 500 times. This indicates that at least, one predictor can be chosen 100 times in Random Forest. However, in Tree boosting, two dominant predictors play an important split for each iteration. This might cause the reason why sensitivity of tree boosting is lower than the one of random forest.

Table 1. Confusion Matrix for Classification Tree

| | *Actual 1* | *Actual 0* |
|---|---|---|
| *Pred 1* | *1257* | *355* |
| *Pred 0* | *1151* | *6987* |

Table 2. Confusion Matrix for Random Forest

| | *Actual 1* | *Actual 0* |
|---|---|---|
| *Pred 1* | *1426* | *399* |
| *Pred 0* | *982* | *6943* |

Table 3. Confusion Matrix for Tree Boosting

| | *Actual 1* | *Actual 0* |
|---|---|---|
| *Pred 1* | *1157* | *187* |
| *Pred 0* | *1251* | *7155* |

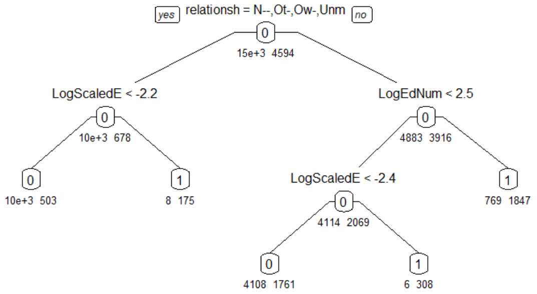

Fig. 5 Best Pruned Tree with 5 Terminal nodes

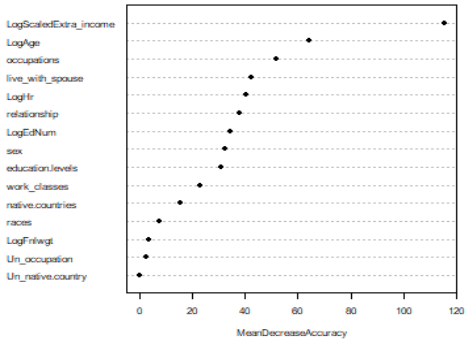

Fig. 6 Variable Importance for Random Forest

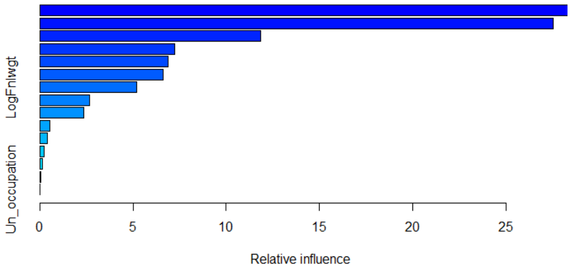

Fig. 7 Variable Importance for Tree Boosting

- *Neural Network*

The performance of (20, 20, 20) structured neural network is shown. The validation accuracy rate is 83.73%, and sensitivity is 70.52%. Table 4 shows the confusion matrix of this network. Fig. 8 shows the tradeoff plot for training and validation dataset to justify the convergence of the network. Furthermore, Fig. 4

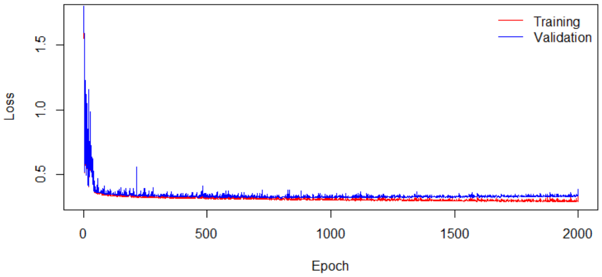

Fig. 8 Training and Validation Error Tradeoff

Table 4. Confusion Matrix for Neural Network

| | *Actual 1* | *Actual 0* |
|---|---|---|
| *Pred 1* | *1698* | *876* |
| *Pred 0* | *710* | *6466* |

indicate that the model is experiencing overfitting or not. As a result, the designed network structure will not cause overfitting, which is helpful as we design a more complicated model.

*ii. Model Selection*

Sensitivity from above models are chosen as a metric to evaluate and select the best model. Neural Network is found to perform best for classifying labels correctly. Moreover, neural network has good performance on accuracy too. Fig. 9 shows the overview of accuracy and sensitivity of different models.

### *C. Model Development & Comparison*

As the given dataset is imbalanced, SMOTE is used to deal with this problem. Fig. 10 shows accuracy and sensitivity of all four models with smote and without smote. Fig. 10 and Fig. 11 show that after using smote all the four models saw improvement.

The data was fitted into given models after implementing SMOTE technique on the dataset. SMOTE resulted into more balanced labels.

At this stage, the best model is evaluated based on accuracy, true positive rate, model complexity and computation time cost. Using above mentioned metrics for model selection, Random Forest is chosen as the best model, because it gives high accuracy and true positive rate and also low computation time cost.

The reason for better performance of Random Forest than a Boosting tree can be observed through a hypothesis. The hypothesis is that a more complicated model is more likely result into higher accuracy and true positive rate. The reason for proposing this hypothesis is that a boost tree is trained with tree stump. However, it also involves some dominant predictors contributing to lower variation of each iteration in boosted tree building. Rather than boost tree, random forest establishes maximal trees which might lead to higher prediction performance because of detailed category in the terminal node.

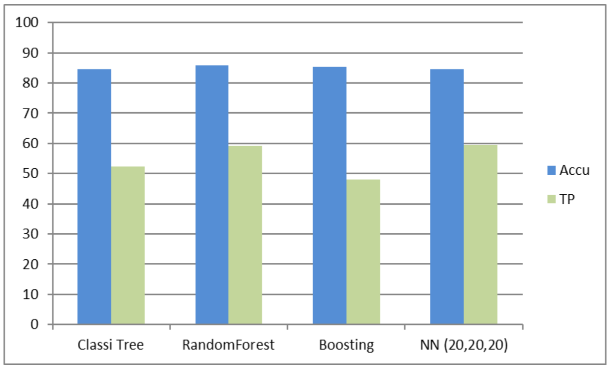

Fig. 9 Accuracy and Sensitivity over Given Models

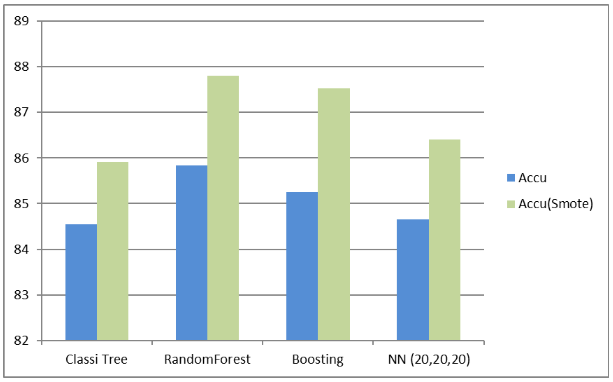

Fig. 10 Accuracy of Different Models with and without SMOTE

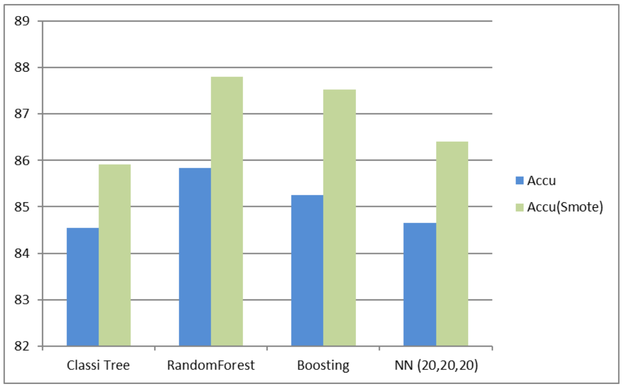

Fig. 11 Sensitivity of Different Models with and without SMOTE

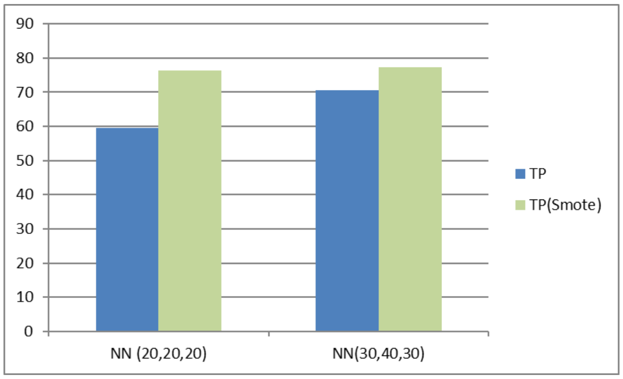

Fig. 12 Comparison between Simpler and Complicated Neural Networks

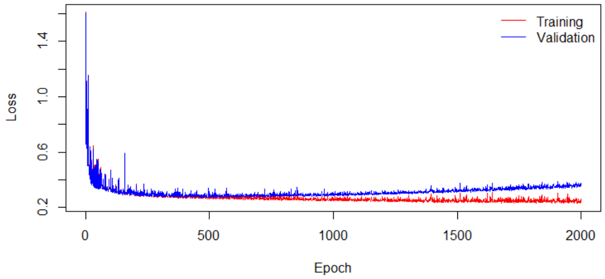

Fig. 13 Training and Validation Error Tradeoff for (30, 40, 30) Neural Network with SMOTE

To verify this hypothesis, another neural network was constructed with more units in each hidden layer, 30, 40 and 30 units respectively. The performance is shown in Fig. 12. It's obvious that complicated neural structure is better than simpler one. However, based on the training and validation error tradeoff plot in Fig. 13, it's likely that highly complicated model will lead to overfitting. In overfitting case, actually, the complicated model is helpless.

### D. *Influence of SMOTE*

SMOTE leads to more complicated model output. For example, the original number of terminal nodes of trained Classification Tree were 5, however, after SMOTE preprocessing, the model output is an approximately maximal classification tree. Also, the average tree size of Random Forest rises from 1270 to 1470. Maybe, it's also one of the reasons that original given parameters in Boost Tree can't fit this data.

### E. *Testing*

Although the dataset has the highest sensitivity on neural network model, regarding the cost of time, the random forest model with smote is the best choice in this report. Because the sensitivity is close in these two models, but neural network cost so much time to train, and random forest model needs much less time. Then the random forest model is tested on the test dataset. The accuracy for it is 87.39%, and the sensitivity is 76.25%, which is similar as it performs on the validation dataset, so it is a solid result, showing the random forest model with smote would be a good selector.

## IV. DISCUSSION

### A. *Analysis and interpretations*

- It is observed that `Education` is an important feature where higher education indicates higher earnings.
- It is also observed that individuals who worked more hours per week had better annual income.
- Another inference is that people tend to earn better as they get older.
- Being married also seems to increases the chance of having better income than not being married.

### B. *Predictions*

- "Accuracy" is almost same in all models, and this might have been because of an imbalanced dataset.
- There is no substantial difference found between "Training" and "Testing" accuracy. This indicates that the feature space was good enough to avoid overfitting.
- As there is no substantial difference between the "Accuracy" among models, "True positive rate" is used as the metric and Neural Network with 3 hidden layers and nodes 30, 40, 30 in each layer gave the best classification with 77.21% true positive rate. However, Random Forest is computationally much faster than Neural Network model and has similar true positive rate. So regarding true positive rate and the cost of computation time, the Random Forest model is the best model for this report.
- It is also observed that using SMOTE to tackle class imbalance problem improved the performance of all models.

### C. *Improvements:*

Improvements that can be done to achieve better "True positive rate" are:

- Using Stratified K-fold Cross validation. This approach can be tried as another technique to solve the imbalanced class problem
- Adding polynomial degrees to the feature space may result in some improvement in the performance
- There is potential to achieve better performance if more data is added into this dataset.

## References

[1] 1994 Adult Census dataset, University of California Irvine (UCI) Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets/Adult

[2] slade_sal. (2017, July 7). Python：SMOTE Algorithms [Web log post]. Retrieved from https://www.jianshu.com/p/ecbc924860af.