Adult Income Classification Prediction

Ryan Morganti  
Dakota State University  
*Beacom College of Computer and Cyber Sciences*Madison, SD  
ryan.morganti@trojans.dsu.edu

Jessica Tarr  
Dakota State University  
*Beacom College of Computer and Cyber Sciences*Madison, SD  
jessica.tarr@trojans.dsu.edu

*Abstract*— This project seeks to determine if it is possible to classify a person’s income based on a variety of collected features using supervised machine learning techniques. The data used for modeling is based on the 1994 US Census dataset donated by Barry Becker to the open-source UCI machine learning repository. Each row of this dataset is divided into fourteen features and properly labeled by the data set collector. With these samples, the project seeks to determine which, if any, machine learning models can accurately predict whether a person’s income is greater than or less than fifty thousand dollars annually. Four total models were analyzed and applied, these being Naïve Bayes, K-Nearest Neighbors, Decision-Tree, and Logistic Regression. In order to properly apply these models, the dataset was cleaned by dropping duplicate entries and missing data. Also, some features were dropped due to their repetitive nature or data skew. After analyzing, and experimenting with different parameters, we determined that the Logistic Regression and Decision Tree models performed the best with an 84% accuracy rate, while the Naïve Bayes model performed the worst with a 62% accuracy rate.

# Introduction

Jobs and income play an integral role in the function of everyday life, creating the vast world economy that exists today. But income is not evenly distributed to all people, and major discrepancies arise due to various factors. Our project seeks to determine if there are quantifiable differences and what features play a part in this. We also seek to determine if there are any ways to predict a person’s income, or if there is any connection. Since the United States collects census data and tax records of all citizens, the dataset chosen brought these features together. Based on the 1994 Census data, our chosen dataset is a collection of fourteen various features that can possibly be used to determine a person’s income. Due to sensitivity of income, the labels are broken into either greater than 50 thousand or less than 50 thousand earned income annually. Some of the features include age, education, gender, race, marital status, capital gain, and more.

Four different machine learning approaches were tried to validate whether the hypothesis that income can be predicted holds true or not. The models used were Logistic Regression, Naïve Bayes, Decision Tree Classifier, and K-Nearest Neighbors. These models were chosen as three were discussed in class and we wished to put that knowledge to use. The other Decision Tree Classifier was chosen since we wanted to gain more familiarity with a model that we had to learn outside of class. In Sec. II of the paper, we discuss the various machine learning models used. In Sec III we discuss the results of our models, and end with Sec IV detailing our conclusions and future work.

# Methods

Initially the dataset had to be cleaned due to missing data and duplicate values. This was achieved by using built in pandas functions. We also converted the mixed numeric and text dataset given into a strictly numeric data set by using one-hot encoding and using the pandas get\_dummies function. We then normalized all of the data to fit between 0 and 1 so that large numbers would not get overweighted, and all data is more comparable. This normalization was done using minmaxscaler imported from the sklearn python library. This resulted in a dataset of 30,000 entries with 61 columns and binary values.

## Logistic Regression

A binary logistic regression model was used from the sklearn python library. A logistic regression model predicts the probability of a certain event occurring and maps that onto a logit transformation, returning a zero or one result.

## Naïve Bayes

We implemented a Gaussian Navie Bayes classification model. A Naïve Bayes model is based on conditional probability between independent features. We assume that the presented features are all independent, which may not necessarily be true, thus impacting the model.

## Decision Tree Classifier

Similar to a Naïve Bayes approach, the decision tree classifier algorithm predicts the class of a new data point based on a set of learned rules. This splits the data point down a tree of conditions and will ultimately classify the data. In our project we specified a decision tree based on entropy using a min sample split of 8 and max depth of 10. These numbers were chosen after experimenting with different values, with these resulting in about an 85% accurate model.

## K-Nearest Neighbors

A K-Nearest Neighbors algorithm was used to identify the k closest data points to any new entry. Then the sample is labeled the same as the majority of data points in the neighborhood. To identify the best number of neighbors to compare with, we ran multiples tests and concluded that 27 yielded the highest accuracy.

When applying the aforementioned models, we used an 80-20 training to testing split of the dataset. This resulted in a training set of size 24574 entries and a testing set of size 6144.

# Results

After properly training the various models, the testing data set was applied to the models and analyzed for accuracy. Since the testing data was kept hidden from the model, we could determine how well the model performed on unknown data.

For each model a confusion matrix was generated. These matrices are divided into four parts: true positive, true negative, false positive, and false negative. Based on the setup of the program the descriptions of each of the sections of the confusion matrix are as follows.

The top left of each confusion matrix is the true negative, meaning those that were predicted to have an annual income of less than or equal to fifty thousand dollars and were correctly classified as such.

The top right of each confusion matrix is the false positive, that is, people who have an annual income of greater than fifty thousand dollars but were incorrectly predicted as making less than or equal to fifty thousand dollars.

The bottom left of each confusion matrix is the false negative. This category is people who were predicted to have an annual income of less than or equal to fifty thousand but were incorrectly classified as making greater than fifty thousand dollars annually.

The bottom right of each confusion matrix is the true positive, or people who were predicted to have an annual income of greater than fifty thousand dollars and who were correctly classified.

All models used the same testing and training split generated using a random state of 1. Using a different random state could impact the performance of the models, but all results were collected using this specific state.

## Logistic Regression

Error Rate: 15.5%

Accuracy Rate: 84.5%

Chart, treemap chart

Description automatically generated

Fig 1. Confusion Matrix for Logistic Regression

As demonstrated by the confusion matrix, 5,194 samples were properly classified by the model and 950 samples were improperly classified.

After obtaining the accuracy of the model, we looked at the generated weights that the model assigned to the features. Of the 61 features that the model trained on, most were under the 1.0 range, meaning that their importance to the final categorization is low. However, there were three top features that stood out as being highly weighted, those being:

Capital-gain: 10.613

Hours-per-week: 2.794

Capital-loss: 2.583

By looking at these weights, the most important predictor of annual income is capital gain. This is followed by the number of working hours in a week and capital loss. Logically speaking these results follow what might be expected. Below is a graph demonstrating the weight distribution of all features

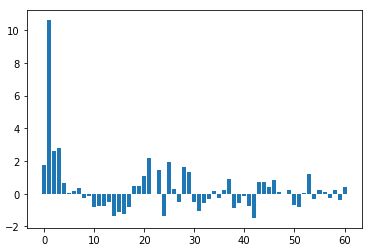


Fig 2. Feature weight distribution for Logistic Regression

## Naïve Bayes

Error Rate: 37.2%

Accuracy Rate: 62.8%

Chart, treemap chart

Description automatically generated

Fig 3. Confusion Matrix for Naïve Bayes

Of all the models, Naïve Bayes performed the worst. It correctly predicted 3,861 samples and incorrectly predicted 2,283 samples. Of the incorrectly predicted data, 2,171 data points fell into the false positive category meaning that they were predicted to make more than fifty thousand a year, while they actually do not.

Due to the naïve assumption of this model, all features are weighted the same. This also assumes independence of all possible features. Within the selected features, it is possible there were some dependent features that were unknown to have correlation at the start of this project.

## Decision Tree Classifier

Error Rate: 15.2%

Accuracy Rate: 84.8%

**Chart, treemap chart

Description automatically generated**

Fig 4. Confusion Matrix for Decision Tree

The Decision Tree Classifier model was the best classification model as it produced the accuracy rate of 84.8%. It correctly predicted 5,396 samples with only 748 incorrect predictions.

After generating the results, and properly creating the model, we analyzed the generated tree of the model. The graph has a maximum depth of 10 before terminating in leaf nodes. The colors in the graph represent the final categorization class. The orange and blue boxes are the classification of greater than, or less than fifty thousand a year. The rest are a gradient between the values representing a leaning one direction or the other in the chain.

Diagram

Description automatically generated

Fig 5. Decision Tree Graph

## K-Nearest Neighbors

Error Rate: 17.1%

Accuracy Rate: 82.9%

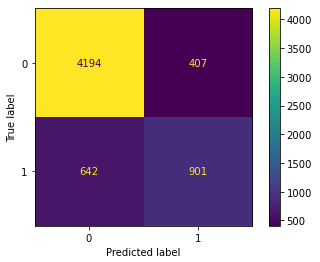


Fig 6. Confusion Matrix for KNN

In the K-Nearest Neighbors model, a total of 5,095 samples were correctly predicted, while only 1,049 were incorrectly predicted. These results are on the same level as the Decision Tree and Logistic Regression models.

Since there are many different ways to properly choose the K value, we iterated over multiple values to determine the final chosen K value. In general, choosing different K values resulted in the accuracy of the model fluctuating between 82.5% to 83%. The value of 27 tended to yield the best results with an accuracy of 82.92% with the value of 17 following closely behind at 82.91%. The below chart is a plot of various K values and the resulting accuracy of that model. Values between 2 and 30 are shown in the graph.

Chart, line chart

Description automatically generated

Fig 7. KNN K-value accuracy plot

# Conclusion

The project sought to classify people into one of two income brackets, that of making greater than fifty thousand dollars annually and that of making less than or equal to fifty thousand dollars annually. To attempt this classification, a collection of data taken from the 1994 US census was used. Inside of this data was fourteen different features that might possibly be integral to determining someone’s income.

Using this data and the four chosen machine learning models, we were able to determine that it is possible to determine, with a high likelihood of success, someone’s income level. Although the Naïve Bayes model performed poorly with a 62% accuracy rate, the other three models all performed well with accuracy scores in the low 80s. Although we would hope to see the models perform better with higher accuracy scores, at this stage the scores are acceptable. With further cleaning of the data and analysis of feature importance, the accuracy ratings would improve. From the Logistic Regression model, we were able to determine that capital gain and loss were important indicators to income level as well as hours worked each day. Other features like race or relationship play less of a role in the final category determination. Further study and research into economics and census records would need to be done to verify the preliminary feature results of the machine learning models. But these results give a glimpse into possible indicators of economic prosperity. Further, with some of the lesser important fields being removed, along with using dimensionality reduction in large categorical samples like education, the different models performances would be likely to increase.

Further study can be done using this same dataset by further processing the dataset and fine tuning the parameters of the model. Also, studies into whether the same models could accurately predict more modern income or if factors have changed in the modern economy. Due to inflation and economic changes, the study would have to consider the different income levels of the modern US economy compared to what it was in 1994.

##### References

1. Bajwa, M. (2019, April 30). *How to interpret the weights in logistic regression*. Medium. Retrieved December 7, 2022, from https://medium.com/@mubarakb/how-to-interpret-the-weights-in-logistic-regression-89bb03249f27
2. Bento, C. (2021, July 18). *Decision tree classifier explained in real-life: Picking a vacation destination*. Medium. Retrieved December 7, 2022, from https://towardsdatascience.com/decision-tree-classifier-explained-in-real-life-picking-a-vacation-destination-6226b2b60575
3. Géron, A. (2020). *Hands-on machine learning with scikit-learn, keras and tensorflow: Concepts, tools, and techniques to build Intelligent Systems*. O'Reilly.
4. Itdxer, I. (n.d.). *Adult-dataset-analysis*. GitHub. Retrieved December 4, 2022, from https://github.com/itdxer/adult-dataset-analysis/blob/master/Data%20analysis.ipynb
5. Johnson, A. (2020, October 1). *Adult Census Income Dataset: Using Multiple Machine Learning Models*. Medium. Retrieved December 4, 2022, from https://medium.com/analytics-vidhya/adult-census-income-dataset-using-multiple-machine-learning-models-f289c960005d
6. *Learn*. scikit. (n.d.). Retrieved December 7, 2022, from https://scikit-learn.org/stable/index.html
7. Ninjas, C. (2020, September 4). *Classification: Decision trees, naive Bayes & Gaussian Bayes classifier*. Coding Ninjas Blog. Retrieved December 7, 2022, from https://www.codingninjas.com/blog/2020/09/04/classification-decision-trees-naive-bayes-gaussian-bayes-classifier/
8. Ron Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid", Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996
9. *The world's most popular data science platform*. Anaconda. (n.d.). Retrieved December 7, 2022, from https://www.anaconda.com/
10. UCI Machine Learning Repository: Census Income Data Set. (n.d.). Retrieved December 4, 2022, from https://archive.ics.uci.edu/ml/datasets/Census+Income