Karim Naous
CS395: Artificial Intelligence
Dr. Joon Suk Park
5/2/2020

Final Project Report

1. Abstract

We traditionally think of income as related to the job you have. However, there are many other determinants of income which in combination result in our income. More accurately, there are certain characteristics which on average are related to higher or lower income. For example, did you know that if you are a husband in a family, you are very likely to make over US$50K per year? We seek to discover what characteristics correlate like this with income. The Adult Income Data-set (UCI Machine Learning Repository) provides a perfect opportunity for us. This data-set provides over 48,000 points of labelled data including information on people's Income, Age, Type of Work, Education, Marital-status, and more. By mining this information, we build a binary classifier using Gradient Boosting that predicts income. Our classifier predicts whether or not somebody makes over US$50,000 using their census data. We are able to achieve accuracy of over 85.45% with our classifier, which is in line with what others were able to as well. By experimenting with feature combinations, this accuracy is increased to 85.65%. Our experiment studies 14 different features and their effect on accuracy. We drop features one by one to see how much loss/gain in accuracy results and use that information as a proxy for correlation of that feature with income. Surprisingly we learn that neither Age, Race, Sex, Native-country, nor Marital-status helps much in predicting income. On the other hand, we learn that Work-class (ex. Private or Govt.), Education, Occupation (Industry), Relationship (ex. Husband, Wife, Child) and amount of Capital Gain/Loss per year to be strong determinants of income.

2. Introduction

A person's income determines much about their quality of life, overall happiness, and the opportunities available to them. We seek to learn which characteristics contribute to income the most. By isolating those characteristics, we can learn about what exactly determines our chance to make a high income in the US. Some things we believe should impact income, for example: level of education or type of work and how much skill it requires. Other things, we think best should not have an effect on income. For example, we prefer that somebody's Race is separate from how much money they make. Similarly, we would want Age or Sex or similar characteristics to have no impact on our chance of earning a high income. By mining this data, we hope to identify both consistencies and inconsistencies with these beliefs. On one hand, we can learn how to best raise our chance of making a high income, and on the other, we can learn about

potential unfairness in our economy. This can potentially help us make decisions about who in our economy needs a better chance to make high income.

We tackle this problem by building a binary classifier to predict whether or not someone makes $> \$50K$ based on their characteristics from the census data. We then experiment with which features increase or decrease our accuracy of prediction, and use that information as a proxy to the correlation of those characteristics with income.

3. **Survey of related work**
Please find full citations at the end of the document.

UCI Machine Learning Repository Census Income Data Set Description
This source tries different machine learning algorithms to solve the same task on the data set. This is similar to what we do, but we experiment with feature combinations and they do not. We learn from this source that most models fall within the 84 to 86% accuracy range. This provides us a ball-park to shoot for with our model.

Kohavi, R. Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid
This paper experiments with our data-set using Naive-Bayes and Decision Trees classifiers to do the same task. It highlights that Naive-Bayes does not work well with large data sets and that Decision Trees do. This paper works on a combination of the two (ensemble) which theoretically works well on the data-set. We try a similar ensemble model in our experiments. We will go more into depth on how feature combinations may help increase accuracy.

Hamers, B. and Suykens J. and Moor, B. Coupled Transductive Ensemble Learning of Kernel Models
This paper works with the same data-set as us and solves our task using different machine learning models. It highlights that ensemble models out-perform regular learning algorithms on it. It continues on to develop a new algorithm based on "coupling" models. We will continue on another path with analyzing feature combinations to increase accuracy. However, we learn from it that we should try using an ensemble classifier. In addition, we learn that we can drop data with missing values as they did since this is a large enough data-set.

Wang, K. and Zhou, S. and Ada Fu, C. and Xu Yu, J. Mining Changes of Classification by Correspondence Tracing
This paper works with census data as well, this time to predict veteran status. It is similar to our work since it uses similar data. Their experiments seeks to see how a classifier can change as new data becomes available. They trained a classifier on data from 1970 then on data from 1990. The old classifier was not very accurate when

tested with the 1990 data. This suggests that characteristics which determine class can change with time. This is important for us to keep in mind when analyzing results and drawing conclusions. We are working with data from 1993, so we may not be getting a perfect picture for 2020.

Brownlee, J. Imbalanced Classification with the Adult Income Dataset
This article solves our task and works with our data-set. They try to find the most accurate classifier by using cross-validation and hyper-parameter tuning. We will try increasing accuracy by changing feature combinations. Nonetheless, we learn important things from this source. Namely, it highlights that our data-set has imbalanced classes. This can cause some problems for us, so it is important to keep that in mind when analyzing results. However, the author assures us that the imbalance is light, so with the right classifier we can still have good accuracy.

Agarwal, A. Logistic Regression Classifier on Census Income Data
This article also solves our task on the same data-set. Though, it focuses on fitting a Logistic Regression classifier on the data and experiments with its accuracy using parameter tuning. We will be experimenting with feature combinations. We learn from this source that when preparing to fit our model, we need to process our data so that numerical data is scaled and categorical data is appropriately encoded.

4. **Formulation**

We seek to use the census data and the features that come with it to build a classifier. This classifier will predict whether or not someone makes more than US $50K based on their characteristics. We will try different machine learning models until we find one with accuracy in our target range (84% to 86%). When we do, we will then experiment with feature combinations. Our goal is determine how much each feature contributes to accuracy, so we will be doing exactly that by dropping each feature one-by-one and isolating the loss in accuracy that results from that. We do this by retraining our model with our data minus the feature we are isolating and calculating an accuracy score. This way, we can compare features and how much they contribute to classifying income.

The machine learning algorithms we try are the Dummy Classifier, Naive-Bayes, Decision Tree, and Gradient Boosting. We use the dummy classifier as a base-line to compare the others. We compare the accuracy of each of Naive-Bayes and Decision Tree to the ensemble model Gradient Boosting. We move forward with using Gradient Boosting as it is the most accurate for our data.

5. **Approaches**

We will use the Census Income Data Set provided in the UCI Machine Learning Repos-

itory (link: http://mlr.cs.umass.edu/ml/datasets/Census+Income). This data set is created from US Census data.

In this experiment, we create a Jupyter notebook and use Python programming language. We make use of the Sickit Learn library for machine learning algorithms and use numpy, pandas, seaborn, mathplotlib for data exploration and processing.

The data we are working with is split into a training and a test set already. We initially have 14 labelled columns to work with. Since we already have features extracted in this data-set, we will be experimenting with their individual efficacy.

Features: *Age, Workclass, Final Weight, Education, Education Number of Years, Marital-status, Occupation, Relationship, Race, Sex, Capital-gain, Capital-loss, Hours-per-week, Native-country.*

There are 48842 data points in this data-set and 2809 missing data points. Because of the large number of data here, we can entirely remove the data with missing values.

After dropping these missing values, we are left with 30,162 data-points in our training set and 16,281 points in our test data set for train:test split of 65%:35%. This is lower than usual (there is a lot of data in the test set), but we should still see good results given the size of our data-set.

We ensure the class distribution is similar across the train and test data-sets. The ratio of '$\leq 50K$ to $> 50K$ is 75.11% to 24.89% in the train data-set. In the test data-set it is a similar split of 76.38% to 23.62%. This is good because we don't want a big difference between the two as that can skew the model's accuracy.

We notice again the class imbalance in our data-set, which we expect to hinder accuracy. With the right learning algorithm though we can still see good accuracy.

We run through simple data exploration to get an idea of the data we are working with. We graph histograms of our numerical data, which can be found in the accompanying notebook. This shows that our data is on different scales and thus must be scaled before using it to fit the model. Our categorical values are graphed as well, showing as a breakdown between those in the '$\leq 50K$' class and those in the '$> 50K$' class. We see a promising correlation between many of them and income, especially Education, Marital-status, and Relationship. We recognize here that we must encode these categories into numerical data before fitting our model.

We create a custom pipeline so we can scale and encode our data. Our pipeline is structured so that it can handle both numerical and categorical data. It includes

a Numerical Pipeline which selects numerical columns and scales them using Sickit Learn's 'StandardScaler'. Our Categorical Pipeline selects categorical columns and uses pandas' 'getdummies' function to create category codes. Before we do this we temporarily join the train and test data-sets, so that we don't miss any categories. The Full Pipeline joins the Numerical and Categorical pipelines.

With that done, we are ready to process our data and experiment. We process both our train and test data-sets. Our experiment includes finding an accurate classifier for our data-set and subsequent experiments on that classifier. Those experiments, which include experimenting with feature combinations, reveal to us how much each feature/characteristic contributes to classifying income.

As mentioned above, the data-set comes with a description: this can help us determine what accuracy score to shoot for. It quotes error rates of 16 different machine learning models. The best performing ones have error rates of around 14 to 16%. Thus, a good accuracy score should be around 84 to 86%. We are shooting for something in this range. Below are the models we tested and their associated accuracy scores. This is with all features left in the data.

Accuracy scores of tested machine learning models

Model	Accuracy Score
Dummy Classifier	75.4316%
Decision Tree	79.1899%
Naive-Bayes	55.3851%
Gradient Boosting	85.4714%

We use the Dummy classifier here as a baseline. As you can see it is right 75.43% of the time. This is unsurprising though considering the class imbalance we have. Decision Tree showed disappointing results here. It had accuracy only about 4 points higher than the dummy classifier. Naive-Bayes showed very disappointing performance here, with accuracy as low as 55.39%. Gradient boosting is promising though, with accuracy score of about 85.47%, much better than the dummy classifier and the other two, and is in within our range of 84 to 86%. For this reason, we move forward with Gradient Boosting as the machine learning model for our experiment.

6. **Experiments & Analysis**

Now, we can run our experiment. Now that we have the accuracy score of our model with all the features included, we want insight into how much each feature contributes to accuracy. A feature which contributes a lot to accuracy must be correlated with income. A feature which does not, meaning the model is less accurate with it, suggests

it is not well correlated with income. Our experiment will remove features from the data and observe the resultant change in accuracy.

Accuracy scores after dropping one feature

Dropped Feature	**Accuracy Score**	**Accuracy Loss / Gain**
None	85.4714%	0.0000%
Age	85.3918%	-0.0796%
Work-class	85.1859%	-0.2855%
Final Weight	85.3851%	-0.0863%
Education	85.3984%	-0.0730%
Education Number of Years	85.3984%	-0.0730%
Marital-status	85.6507%	+0.1793%
Occupation	85.0000%	-0.4714%
Relationship	85.1859%	-0.2855%
Race	85.4714%	0.0000%
Sex	85.5578%	+0.0864%
Capital Gain	84.3426%	-1.1288%
Capital Loss	84.7543%	-0.7171%
Hours-per-week	85.4117%	-0.0597%
Native Country	85.4117%	-0.0597%
Average	**85.2678**%	**-0.2181**%

We begin with the control set of all features, and that yields us a 85.4714% accuracy. Dropping age results in a loss in accuracy of .0796%, less than average which suggests age is not the best determinant of income: it is not as correlated with income as others. Dropping Work-class lowers accuracy by a significant .2855%, higher than average, which suggests work-class is strongly correlated with income. Final-weight in the sample drops accuracy by .0863%. We did not expect this to have much effect, but it had a small one. Education drops accuracy .0730%, much lower than the average of 0.2181%. This suggests education is not a very good determinant of income. That is certainly surprising! Education in the number of years yields the same results as Education in categories. Dropping Marital-status increases accuracy by .1793%, which is significant. This suggests a different insight than the previous results. Since this model's accuracy increased without Marital-status, then this feature was "confusing" it before. It is not very correlated with income and the model is better off without it. Occupation on the other hand is very correlated with income and drops accuracy a high .4714%. This suggests one's occupation is the greatest determinant of income so far. Relationship does well here as well, dropping accuracy .2855%, higher than average. Interestingly then Relationship is correlated to income, but Marital-status is not very. This suggests marital-status such as married or divorced is not as effective in determining income as relationship within a family such as husband or wife or child. This could be because Relationship distinguishes between genders. Though, a quick

peak at Sex disproves this. Sex has a gain of 0.0864% in accuracy when dropped which as discussed above suggests no correlation. Dropping Race results in no loss in accuracy which is very surprising. Now Capital Gain seems to be the best predictor of income here, causing a loss in accuracy of 1.1288% when removed. This is not surprising, as a big enough capital gain can classify income above 50K by itself. Capital Loss has a strong correlation as well, causing a loss of .7171%. Hours-per-week is not very correlated, with a loss of .0597%. Finally, native country is also not very correlated, with a small loss of .0597%.

We also wanted to make sure that we weren't missing out on correlations by only taking out one feature at a time. Some features are heavily related. An example is Education and Education in the number of years, which are perfectly correlated since one is a categorization of the other. Below are pairs we dropped together. This is not an exhaustive test, and is only meant to see whether further testing is needed.

Accuracy scores after dropping two or more features

Dropped Features	**Accuracy Score**	**Accuracy Loss / Gain**
None	85.4714%	0.0000%
Education, Education Number of Year	84.8074%	-0.6640%
Marital-status, Relationship	82.8287%	-2.6427%
Occupation, Workclass	84.9137%	-0.5577%
Capital Gain, Capital Loss	83.7450%	-1.7264%
Average (single feature)	**85.2678**%	**-0.2181**%

This shows us that Education is in fact correlated with income, causing a loss of .6640% when removed. We have to remove both Education measures to see a difference because both capture the same information. Removing marital-status and relationship together gives a very large loss, much more than either alone. This suggests that one's relationship to family or marital-status has a high correlation to income, the highest in fact. Removing Occupation and Work-class together shows a significant loss, greater than each individually. And finally, capital gain and loss also cause a significant loss. We want to be careful when interpreting this data. Because dropping features as a general trend seems to decrease accuracy, on average by .02181%, we don't want these results to be over-interpreted. The higher loss seen when pairs are dropped could simply be caused by the loss of a feature rather than a special relationship between the two and income. The notable exception is education, which spurred these tests in the first place. Dropping both confirmed to us that education is correlated to income.

Now that we have analyzed which features contribute to or detract from accuracy, we want to try optimizing the classifier with the best feature combination. We drop the features which resulted in an increase in accuracy prior (when removed from the feature set) and we observe the results.

Accuracy scores after dropping features not correlated with income

Dropped Features	Accuracy Score	Accuracy Gain
None	85.4714%	0.0000%
Marital-status Only	85.6507%	+0.1793%
Race Only	85.4714%	0.0000%
Sex Only	85.5578%	+0.0864%
Marital-status, Race	85.6440 %	+0.1726%
Marital-status, Sex	85.5046%	+0.0332%
Race, Sex	85.5644%	+0.0930%
Marital-status, Race, Sex	85.5511%	+0.0797%

We want to have the best accuracy, and that is achieved with dropping Marital-status only. This is interesting because we expect that dropping all of Marital-status, race, and sex together would have the greatest gain in accuracy, but that does not seem to be true.

7. **Results**

Thus, the data suggests the following about each of the features in our data-set:

Highly correlated with income:
Work-class, Education, Occupation, Relationship, Capital Gain, Capital Loss.

Slightly correlated with income:
Age, Final Weight, Hours-per-week, Native Country.

Not correlated with income:
Marital-status, Race, Sex

We learn that Race and Sex are not correlated with income. Knowing about the gender and race-based income inequalities that exist in our society, we would expect otherwise. Nonetheless, this is a positive discovery. We also learn that marital-status is not very correlated either. This suggests that being married or not, or being divorced, are not by themselves good predictors of income. This is good, because we prefer that people of all marital-status can make good income.

This data suggests that Education, Occupation, and Work-class are highly correlated with income. That is good. We want education to be highly correlated because it is a measure of how qualified we are. In addition, if anything were to determine income we would want it to be where we work and what we do. Thankfully then, Occupation and Work-class are great predictors of income.

Relationship in a family is also highly correlated. This suggests that one's relationship with those he lives with greatly affects income. We are interested in why that is the case. Perhaps those with strong relationships are motivated to work more? Or perhaps

those with close relationships are afforded more opportunities, such as getting more education or higher paying jobs?

Finally, Capital Gain and Loss are also high predictors. Since amount of money earned on investments classifies as income by itself, this is no surprise. Though, in an ideal situation, income is earned by labor and not from investments.

Other features appeared to be weakly correlated with income, such as Age, number of hours worked per week, and native country. Since older people usually earn a bit more, we expect age to be at least slightly correlated. It is good that it is not highly correlated with income, because it is a biological factor and not under our control. Number of hours worked is also weakly correlated. Since this is a direct measure of how long we work, and is under our control, we would want this to be highly correlated with income. Unfortunately, it is only weakly correlated, which suggests other factors are more important.

Country of origin is also weakly correlated. A high correlation would suggest income disparity based on where you are from, which thankfully is not the case. Ideally though, we want this to not be correlated at all. Final weight in the sample seems to be slightly correlated as well. This suggests being a common data-point in the census could determine something about income. Most likely, being a very common data point means you are more likely to not earn over $50K since that is the most common class.

8. Conclusions

Our exploration shows that income can predicted by different characteristics we possess. Some of those characteristics are under our control, like how much we work, where work, and what we do, and other are not. For the most part we learn that we can increase our chance of earning a higher income.

Anybody with any combination of characteristics can earn a high income, however, some people are just more likely than others to. It seems that to make a high income, pay attention to how much education you get, where you work, what type of work you do, and your relationships.

Of course it isn't that easy. If it were easy to earn a high income everyone would. There are certain barriers that prevent people from doing so. For example, some people cannot afford to get a lot of education. Others, while educated, do not have the right networks to reach high-paying jobs.

As a society, we should work to give everyone a decent chance at earning a high income. Our research suggests that the best way to do that is by improving access to education, increasing job opportunities in high-paying industries and encouraging healthy and constructive relationships in our communities.

A natural next step for us is to research the disparities in these factors of income. A future project for our team may work on further mapping a relationship between education levels, types of available employment, and relationships to the level of income

in different communities. If we can use community-level data to predict income, then we are able to identify those communities that need our help.

9. Appendix

This was an independent project. I wrote the program which performed the data exploration, processing, model training, and evaluation. I also wrote this report.

10. References

Agarwal, A. "Logistic Regression Classifier on Census Income Data." Medium, Towards Data Science, 22 Oct. 2018, towardsdatascience.com/logistic-regression-classifier-on-census-income-data-e1dbef0b5738.

Brownlee, J. "Imbalanced Classification with the Adult Income Dataset." Machine Learning Mastery, 6 Mar. 2020, machinelearningmastery.com/imbalanced-classification-with-the-adult-income-dataset.

Dua, D. and Graff, C. 2019. UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science.

Hamers, B. and Suykens, J. and Moor, B. "Coupled Transductive Ensemble Learning of Kernel Models." Journal of Machine Learning Research, vol. 1, no. 10, 2003, pp. 1-48.

Kohavi, R. Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid. In KDD-96 Proceedings, 1996.

Wang, K. and Zhou, S. and Ada Fu, C. and Xu Yu, J. "Mining Changes of Classification by Correspondence Tracing." In Third SIAM International Conference on Data Mining Proceedings, 2003.