Creating a Model And Pattern Classification for Adult Census Data

Data Set(s) *(specify which)*: Adult

Amirhossein Forouzani, amirhosf@usc.edu

April 25, 2019

**Please organize your report along the lines of this template**; you may use any word processing software you like, as long as you submit your report as the required pdf file described below.

**Your report must be typewritten and submitted as a pdf document**, in machine readable form (no scans or screen shots).

**The report should not exceed 15 pages.** Please note that this is **not** the recommended length **nor is it a hard limit**. You should favor readability (proper font and image sizes) over possible page issues. However, be aware that part of your grade depends on the written quality of your report, and this includes your capabilities of summarizing your work and your results.

**You must properly cite your references.** If you are following ideas from papers or from online discussion forums (such as Kaggle), be sure to clearly distinguish between what is someone else’s work and what is yours and remember to cite the source. Wherever you include material that came from elsewhere, include a citation at that location (e.g., [1], or [*author’s last name(s)*]). At the end of the report, include the reference. We suggest the IEEE format (easily found in Work or Latex) like this [1] or this [2]. Failure to comply with this rule may lead to plagiarism penalties.

**For your code, if you use code from elsewhere, you must cite the source** in the pdf file of your code. This can be done informally, by stating in a comment where it came from (e.g., “thanks to Kaggle kernels for the following code [www.kaggle.com/…/…]”).

**Your code must be submitted in two different formats: a pdf file**, all code in the one file, also required to be machine readable (no scans or screenshots), **and all the original code files in a zip file**, so we can run your code.

# Abstract

As part of the final project in Mathematical Pattern Classification course, this research would explore different classification techniques throughout the course to find the best classification model for Adult Data set. The goal of the adult data set is to design the best model that would predict if the income of a certain person is more than $50k based on many factors including education, country etc. The data examination showed that adult data is a categorical data set with distributions. This indicates that the data should be encoded using 0s and 1s instead just their numbers. After feature reduction using select from model algorithm the data set (Small Data Set) was then subjected to different classification techniques including, Perceptron using one versus rest classifier, Logistic regression, SVM(Support vector machines) using radial basis Kernel, Multinomial Naïve Bayes Classifier and finally Artificial Neural Network (ANN) with 2D hidden layer of 15 by 2. The most accurate classification (ANN) has achieved an f1 score of 65% and accuracy score of 84.1%. In further search for better accuracy the Random Forest classifier was used which helped me achieve the classification accuracy of 64% which is very close to the ANN approach.

# Introduction

# Problem Statement and Goals

The Adult Data set is a two-class classifier problem. The goal is to tell if a person makes more than or less than $50k annually. This indicates that we can use binary classifier techniques to tackle this problem. Preprocessing is the first and most important step of analyzing the data. The right type of encoding has to be used for categorical features. The classifier has to be carefully picked according to number of classes.

# Approach and Implementation

## 3.0. Importing Data

To Import the file, I used pandas readcsv function. This function allows parsing through data easily and saving file to data frames. The data was already divided to data frames f1-f14, after reading the original file on the UC Irvine dataset, I decided to change the header names to respectable names. This is merely done to help with reading the data information for future preprocessing.

## Analyzing the Distribution and the correlation of the data

Firstly, I drew a complete table of feature’s distribution with respect to the classifier, which is whether someone makes more than $50k or not.

The resulting distribution table is as follows.

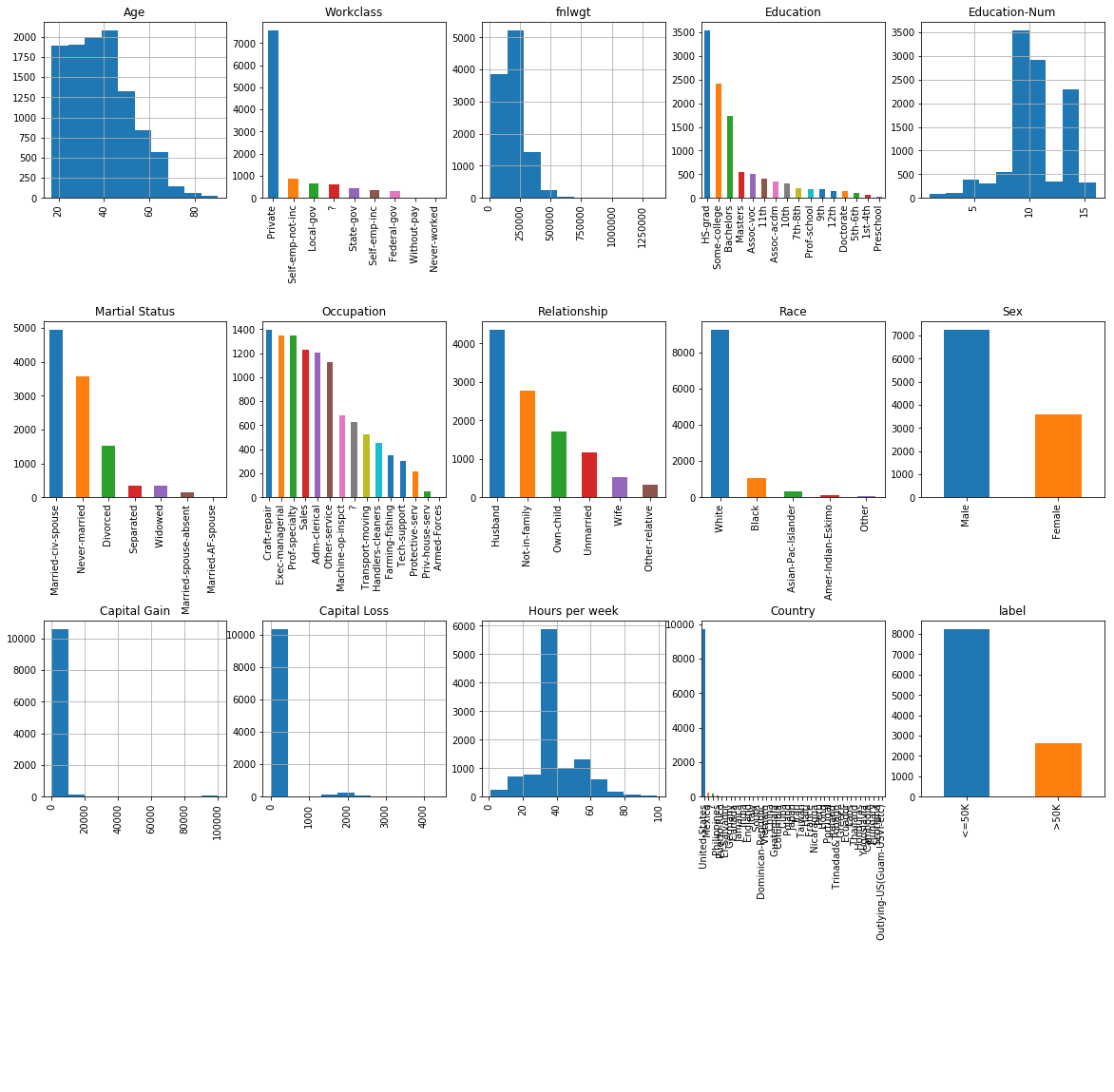


Figure 1. Showing the Distribution of the data

As can be seen in figure 1 the number analyzed people are mostly residing in America. We will use this later in preprocessing the data.

Next we need to find the correlation between features on the data



Figure 2. The correlation heatmap of the Data

As can be seen in the data there is a high correlation between the Education and Education Number. There is also high correlation between sex and marriage columns this can help us in the feature reduction later on.

## Preprocessing

In this section various techniques are done to prepare and preprocess data.

### 3.2.1 Encoding

The variables come in categorical names and string. I needed to come up way to only represent all the categories with numbers so it will be easier for the program to analyze them. The first approach is to use the encode function of the sklearn. This would only change the categorical features to numbers. While this might be a good way to encode the data, we would later see that this is not the best method to encode the categorical features with correlation. The following figure shows the distribution of the data after encoding.

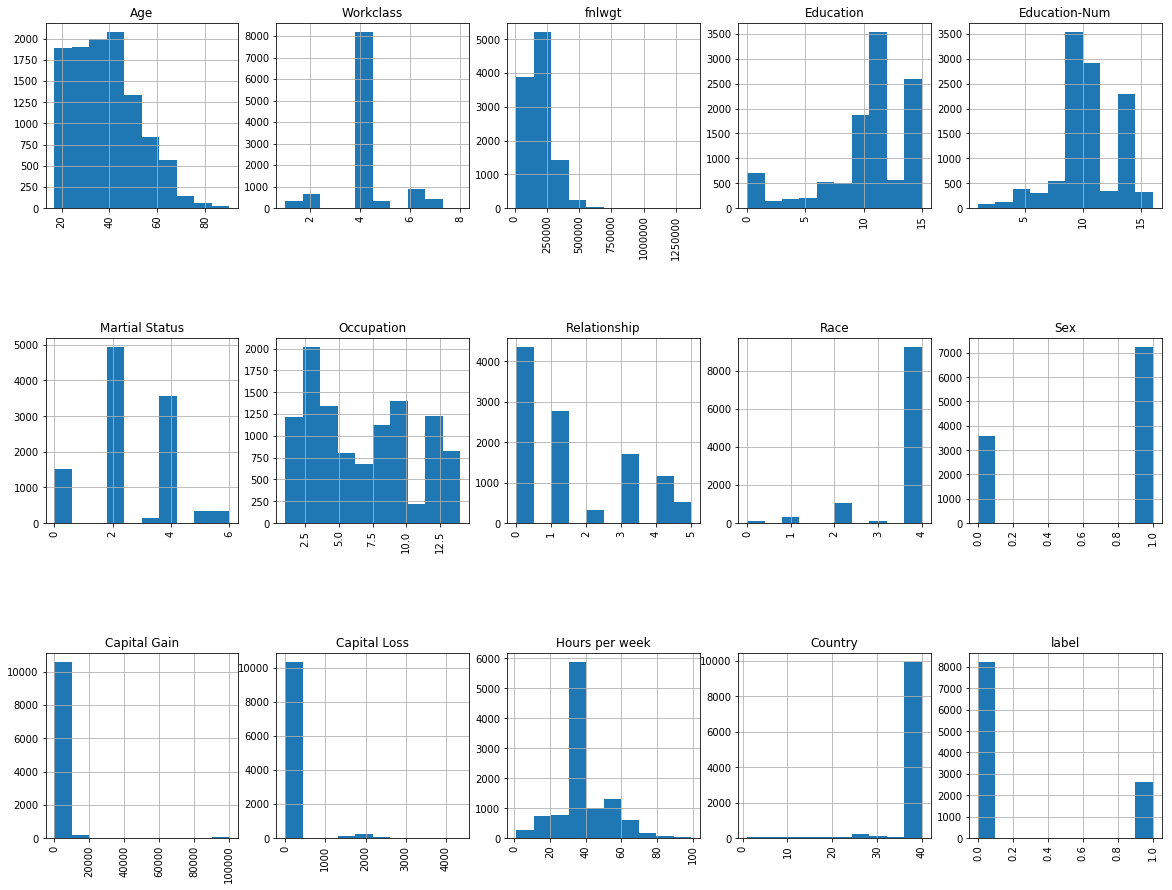


Figure 3. The Picture shows both Imputed and Encoded data Set.

Looking at the Label table we can see that we are dealing with an unbalanced dataset meaning that we have too much of a first class compare to another one. We would later fix this by resampling the data.

### 3.2.2 Imputing

Some of the features in the adult data are indicated as “?“. To fix this the data needs to be imputed so the feature would be filled. To do this I defined a class called impute categorical that uses the in-built sklearn sklearn function impute. I set the rules so the data would be imputed using the most frequent approach.

### 3.2.3 Scaling

Finally, to make the data more processable I scaled the data between [-1,1] using sklearn scale function.

## Compensation for unbalanced data

As I explained in the previous section the data is imbalanced and we have to much of a one class. Firstly, to analyze the accuracy of our classifiers we need to use f1 score instead of the accuracy score.

Moreover, to tackle imbalanced data I used under sampling technique. This would resample the training data and remove any data that would make one class more than the other. This method would, for every observation of class 0, randomly sample from class 1 without replacement and then join together class 0’s target vector with the down sampled class 1 targets. In scklearn it is defined as ﻿resamplingdata\_downsample(X,y).

## Feature extraction and dimensionality adjustment

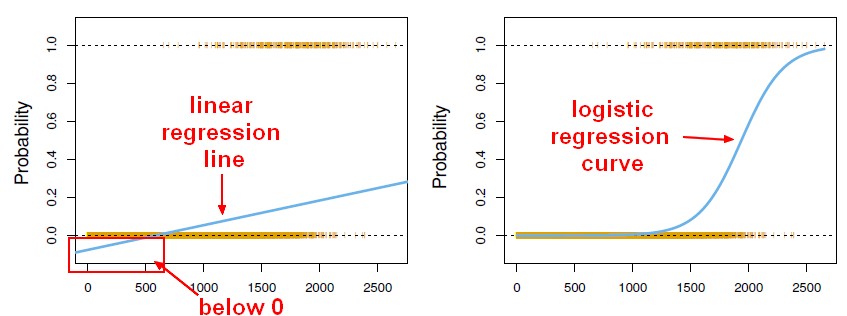
As explained in the first section, the data has two features that are high correlated. We can remove these to reduce the dimensionality of the feature space. I also used ‘select feature from classification‘model to reduce the number of the features in the data. This feature selector would reduce feature space based on the linear SVC modeling. As a result, I got rid of 2 features and was working with 12 out of 14.

## Dataset Usage

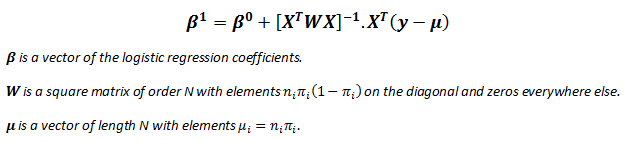
Using the training set and test set as is, is not going to be helpful because we are handling unfairly distributed data. As I will show in the result sections, just using the test and train set are not going to provide a model with good ROC curve. Train and test\_split functions were used alongside with tree expansion functions like “random forest” to increase the dimensionality of the data first then running it through a pipeline for classification to achieve a better model and classifier. I will Later explain this in the appropriate section.

## Training and Classification

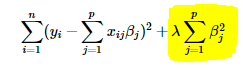
1. The first, suitable classifier I used for this data was logistic linear regression. There are some differences between linear regression and logistic linear regression. In linear regression, the outcome (dependent variable) is continuous. It can have any one of an infinite number of possible values. In logistic regression, the outcome (dependent variable) has only a limited number of possible values. This will widely help us classify the distributed features.



The formula is similar to linear regression with different parameters.



We are still defining a coefficient by which the regressor is going to converge. In this method I used the lbfg( Ridge regression) method. In Ridge Regressor the data might lead to under fitting but because we are using many iterations of the data set, we can avoid under fitting while not facing overfitting. The criterion of the Logistic regressor is as follows:



We have to choose lambda carefully to avoid under fitting. The max iteration allowed is 1000

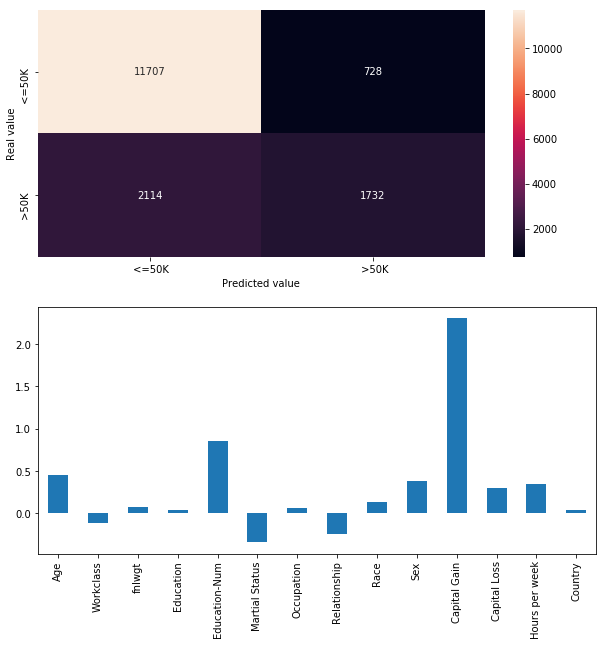


Figure 1. Showing the confusion matrix and coefficients

As can be seen in the figure 1. The heatmap shows the confusion matrix of the classification and the bar graph shows the coefficient of each feature.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Logistic Regression | | | | | Interpretation |
| Feature select & balance  performance | ACC | 82.20% | TP | 49 | In Logistic Regression method reduce feature decrease the predict accuracy, and balance is also unnecessary,  Overall, this method is good, it has a high accuracy in negative data and perform ok in positive site, and the ROC is high (0.8). |
| AUC | 0.69 | TN | 834 |
| F1 score | 0.54 | FP | 43 |
|  |  | FN | 65 |
| Feature Split and Dimension Expansion Using Random Tree Embedding | ACC | 87.50% | TP | 67 |
| AUC | 0.8131 | TN | 808 |
| F1 score | 0.5174 | FP | 78 |
|  |  | FN | 47 |

* State the parameters of the model and how they were chosen. If a parameter is chosen by heuristics, state so. If a parameter is chosen by some model selection, optimization, or validation process, state so and describe the method.
* If you compared the number of learning variables with the number of constraints or data points you have, describe that here.
* If you have sets of results to show for this pattern recognition method, include them here. (For a comparison of results from different pattern recognition methods, use the next subsection.)

# Comparison, Results, and Interpretation

Present performance comparison of your different models and methods here. Include a comparison with the given baseline system(s), and with any results you found in the literature or internet. For each result, be sure to clearly state whether it is from training, cross-validation, or test set. Use table(s) and/or plots. **Do not paste print screen images.** If you used more than one dataset, you can either have different subsections for different datasets or you can directly compare how each classifier performed on each dataset.

Include your interpretation of these results. Can you explain what you observe? If not, any conjectures? Did you observe anything particularly unexpected?

# Contributions of each team member

If a team project, state here what the contribution of each team member was (i.e., who did what).

# Summary and conclusions

Briefly summarize key findings, and optionally state what would be interesting or useful to do as follow-on work. Optionally, summarize some of the key things you learned while doing the project.

# References

For each source cited above, include the reference here. Examples:

|  |  |
| --- | --- |
| [1] | "Add citations in a Word document," 10 April 2019. [Online]. Available: https://support.office.com/en-ie/article/add-citations-in-a-word-document-ab9322bb-a8d3-47f4-80c8-63c06779f127. |
| [2] | F. P. C. S. a. C. S. Vitor Cerqueira, "Combining Boosted Trees with Metafeature," in *Advances in Intelligent Data Analysis XV: 15th International Symposium*, Stockholm, 2016. |