Weak Pattern Theory

**Project report**

**COSC 6342: Machine Learning**

**Submitted by**

**S M Salah Uddin Kadir (1800503)**

**Rubayat Jinnah (1891217)**

Let’s classify any dataset with perfect accuracy! In this paper, we are going to propose a theory that can be used to classify a dataset to get the very close of perfect accuracy. We know, strong always ***dominates*** the weak by nature. When we build a classifier from a dataset, the classifier represents the maximum examples, ***strong pattern***, by dominating the other minor groups, ***weak patterns***, of examples. So, the idea is that if we can separate the weak pattern examples from the dataset and build a different classifier using them then it is possible to achieve a very good accuracy close to perfect.

To verify the idea, we have used the ***Census Income Data Set*** [1] where the accuracy of the dataset using a single classifier is not very good. The dataset is very famous, and lot of researchers have already worked on it. At first, we will have a survey on their work.

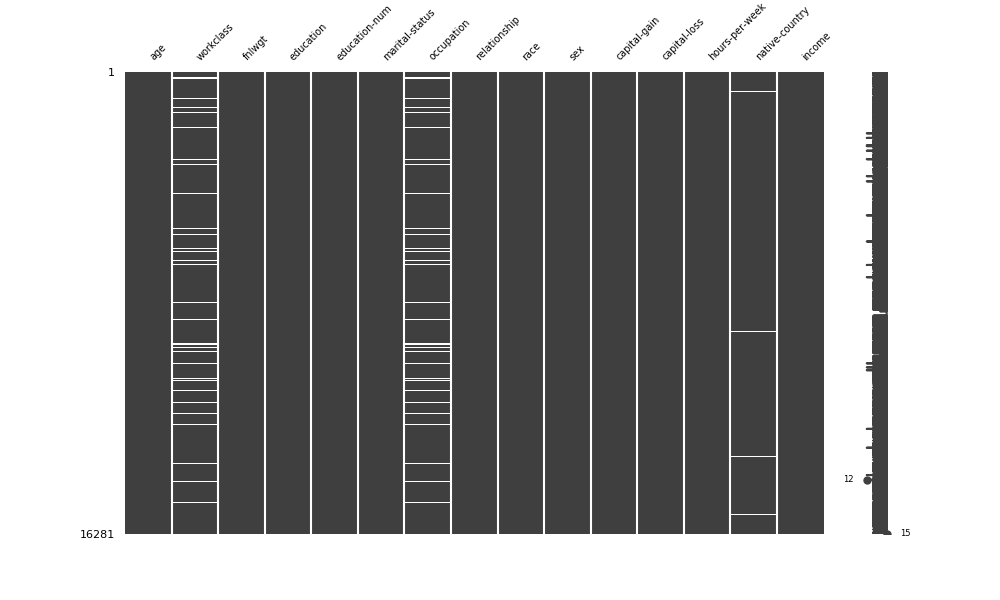
**Relevant Work**

The dataset has a “Data Set Description” link that contains a list of 17 algorithms ran on the dataset and the results. One of those algorithms is naive bayes with 16.12% error rate. Another study used naïve bayes algorithm but omitted the entries with unknown values and had a higher error rate of 20.43%. [2]. In Ron Kohavi’s paper “Scaling Up the Accuracy of Naïve Bayes Classifier: a Decision Tree Hybrid”, the NBTree is described. Kohavi described NBTree as a hybrid of naïve bayes and decision tree. In this approach the error rate was 14.01% and this is the best among 17 algorithms.[3]

Another approach was taken in the paper “Ensemble Selection from Libraries of Models” by Rich Caruana. Here he concluded that ensemble selection consistently finds ensembles that outperform all other models, including models trained with bagging, boosting, and Bayesian model averaging.[4] In a study it was concluded that **a Log Transformed Linear Regression is best model for predicting Income Per Capita.[5]**

**Data set preprocessing**

The dataset has in total 48,842 (16281 test examples, 32561 training examples) instances with 14 different types of categorical and continuous attributes to predict the result that whether the income is greater than 50K or not. There are lot of examples in both training and testing dataset with missing values. Here, I have a plot on testing dataset where the white line represents the missing value of that example.



From the plot, we see that all the missing values are categorical type of attributes. We treated them in two ways,

* Removing the entire example, or
* Filling with the most frequent value of that attribute

We found that filling the missing value with the most frequent value gives the better result than removing the entire example. So, we used the later preprocessing option for the rest of the analysis.

**The Analysis on the dataset using different estimators**

Our used algorithms to test the dataset,

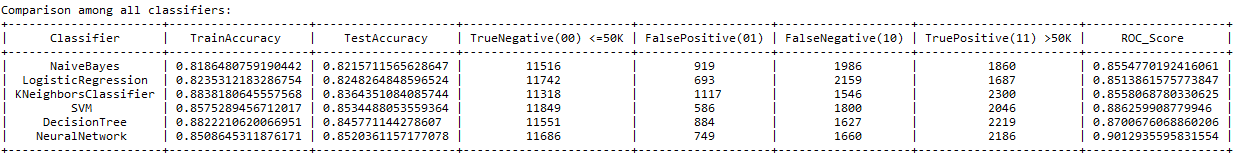
* Naïve Bayes,
* Logistic Regression
* k-nearest Neighbors
* Support Vector Machine
* Decision Tree and
* Neural Network

We also used,

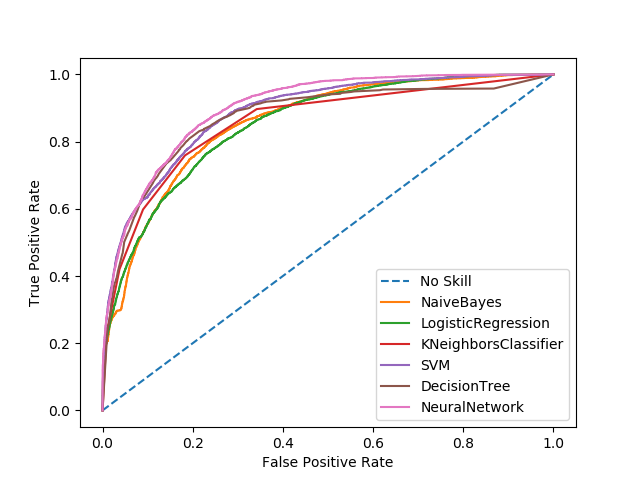
* Bagging
* Boosting

First, we will have a comparison table of performance applying those algorithms,

(*Please zoom in to read the table, sorry for the inconvenience. We made this to keep all things together for the easy comparison.*)



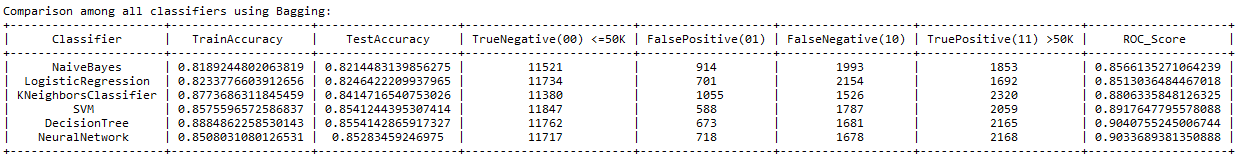
And the corresponding ROC curve comparison,



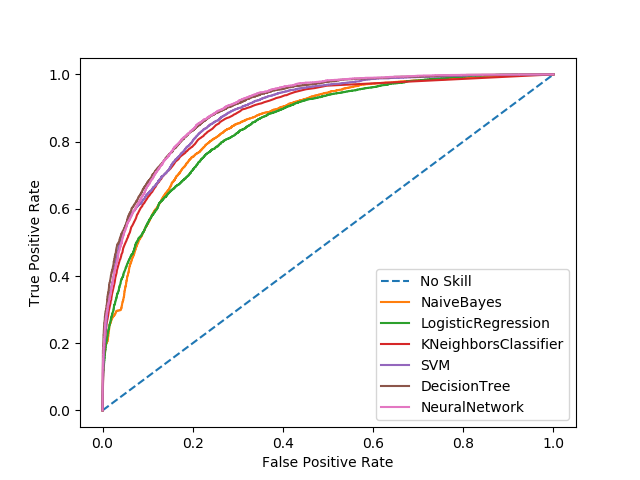
From the table, we see that we have obtained the maximum test accuracy from SVM (85.34%) and the maximum ROC score from Neural Network that covers 90.13% of the plot.

We wanted to check whether we can get a better result using Ensemble algorithms. We used Bagging and Boosting techniques,

***The same comparison table using Bagging***,

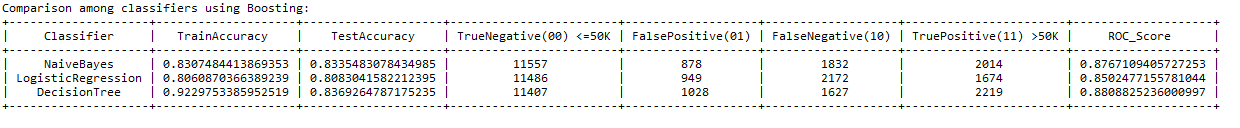


And the ROC plot comparison,

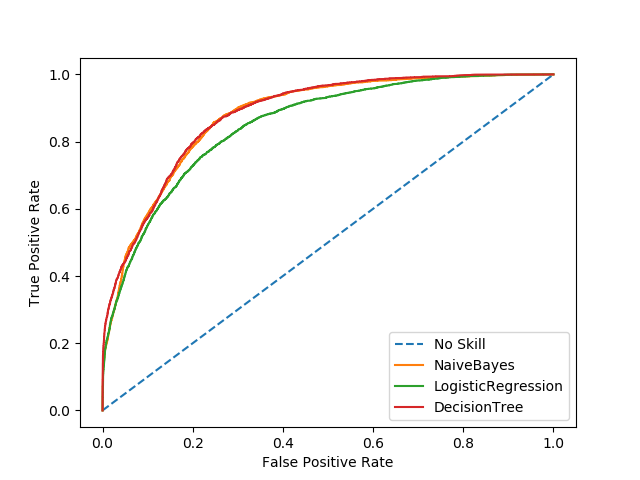


Here, we have got the best result from the decision tree with 85.54% accuracy on the test dataset where the ROC score is 90.49%.

And using **Boosting**,



With the corresponding ROC curve,

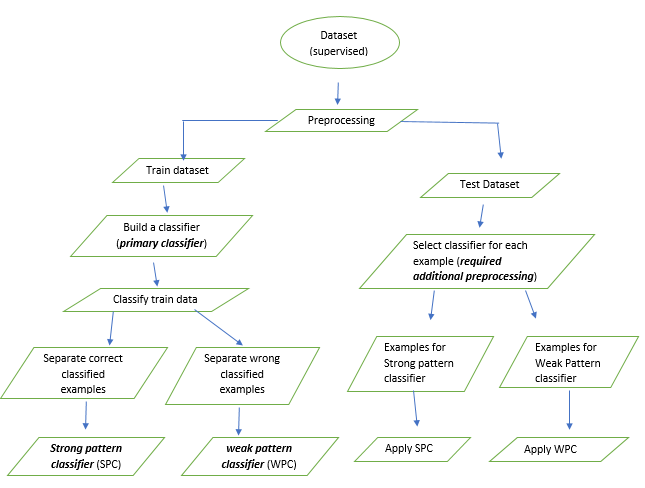


Using Boosting, we have got the best test accuracy (83.69%) and the ROC (88.89%) score from the decision tree, no better than the previous scores.

**A different approach**

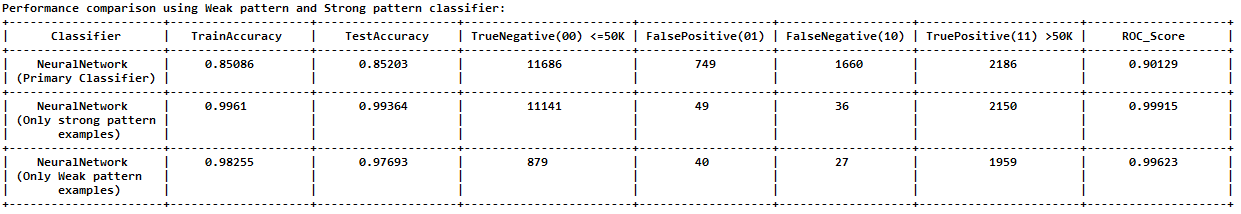
So, we can conclude here that ***the obtained result is not good enough***. To find out the reason, I analyzed separately the correctly classified examples and the examples that are not. We discovered a pattern using the wrong classified examples that is ***very different*** from the pattern using correctly classified examples. So, the pattern from the wrongly classified examples is ***the weak pattern***, and the pattern from the correctly classified examples is ***the strong pattern***. Now, if we can add one more ***preprocessing step*** to separate the weak pattern examples from the strong pattern examples and classify them using different classifier, then it is ***possible to achieve a very high accuracy*** close to perfect.

The ***flow chart*** for the weak pattern theory,

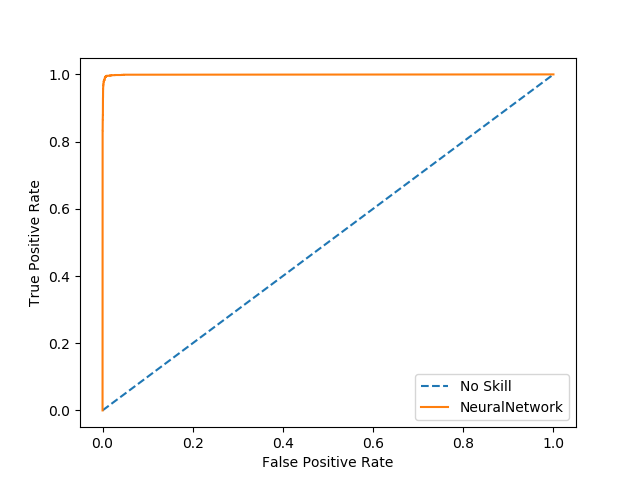


Let’s assume, ***somehow***, we can separate the weak pattern examples from the strong pattern examples in the test dataset so that we can ***verify our weak pattern idea***,

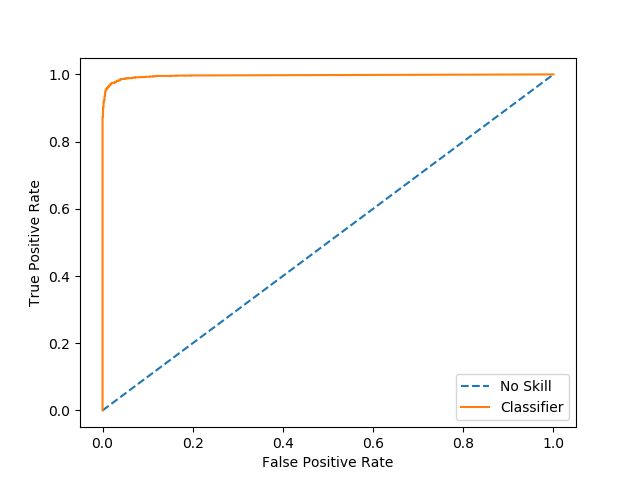
Now, let’s check the ***test result using weak pattern and strong pattern classifier***,



Corresponding ROC plot for ***strong pattern classifier***,



And the ROC plot for the ***weak pattern classifier***,



From the table, we see that we have **obtained a very high accuracy** on the test data for both strong pattern classifier (99.36%) and the weak pattern classifier (97.69%). Originally, there are in total 152 (85 using strong pattern classifier, and 67 using weak pattern classifier) misclassified examples. So, the ***original acquired accuracy*** is 99.07% which is pretty good compared to the previous results. The ROC score is also very high, for strong pattern classifier is 99.92%, and the weak pattern classifier is 99.62%.

**Conclusion**

We want to ensure every example the equal opportunity to ***show their own identity*** so that the dominant classifier (primary classifier) cannot misclassify weak pattern examples. There may have several weak patterns, so we may need several weak pattern classifiers to classify them correctly. I belief it may require a huge research to select the right classifier, the additional preprocessing step, during the prediction of an example. Initially, I thought it can be possible to select the right classifier of an example based on the ***prediction probability*** of primary classifier but did not show the promising result. I have another thought using ***clustering algorithms*** to separate the weak pattern examples from the strong pattern examples but could not verify yet. Finally, we can say that if we can overcome the additional preprocessing step then it is ***possible to obtain a very high accuracy close to perfect accuracy*** for any dataset.

**References:**

1. The Dataset (<http://archive.ics.uci.edu/ml/datasets/Adult>)