CSCI 620 Phase 4: Income Prediction

Ketan Kokane
Rochester Institute of Technology
1 Lomb Memorial Dr,
Rochester, NY 14623
kk7471@rit.edu

Siddarth Sargunaraj
Rochester Institute of Technology
1 Lomb Memorial Dr,
Rochester, NY 14623
sxs2469@rit.edu

Ameya Nagnur
Rochester Institute of Technology
1 Lomb Memorial Dr,
Rochester, NY 14623
an4920@rit.edu

Kavya Kotian
Rochester Institute of Technology
1 Lomb Memorial Dr,
Rochester, NY 14623
kk2014@rit.edu

ABSTRACT

This report highlights the tasks done in preparing a data set to perform data mining activity on it, to discover the non obvious information from the data set. It describes how the data set was analyzed with respect to the set goal of predicting an attribute in the data set based off of remaining attributes. It also gives proper justification of why the data set was analyzed in particular way and what information was gained from it. The data set is selected from UCI machine learning repository `http://archive.ics.uci.edu/ml/datasets/Adult` which contains basic data points about every individual (like education, profession, age, etc). The task is to predict weather the individual earns more than 50K. This report provides graphical representation of what knowledge was gained from the data set regarding the set goal. It uses Data Classification method to predict the annual Income based on the few selected attributes in the data set. Also the report gives justification of why the particular data mining task was chosen and what is the accuracy and other measures of the generated model.

Keywords

Data Mining, Classification, Data Analysis, Data Description, Income Prediction

1. INTRODUCTION

The primary task is to predict the annual Income of an individual, based on his general information collected during the census. As the data set was not created with the sole purpose to determine the annual income of an individual, initial analysis involved finding the irrelevant attributes in the data set.


```
##  [1] "age"            "workclass"      "fnlwgt"         "education"
##  [5] "education.num"  "marital.status" "occupation"     "relationship"
##  [9] "race"           "sex"            "capital.gain"   "capital.loss"
## [13] "hours.per.week" "native.country" "prediction"
```

Figure 1: The data points about each individual in the data set with 32516 instances

The Task is to predict the *Prediction* Column, which consists of two class labels : $<= 50K$ and > 50.

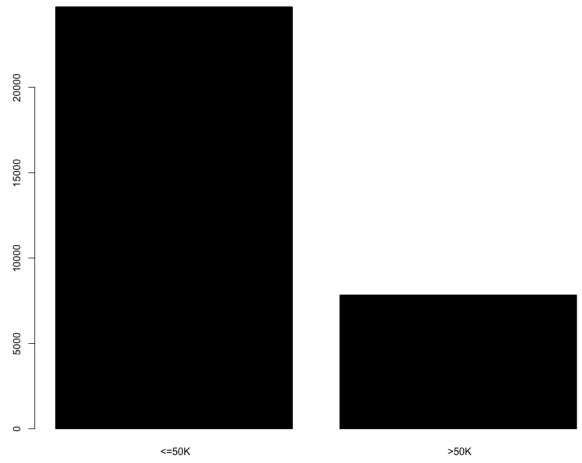

Figure 2: Distribution of Label in the data set

The distribution in *Figure*2 showed that the data set is skewed towards the individuals having income less than 50K. We assumed this to be the correct distribution and carried the further analysis with this assumption. Based on this distribution, we analyzed the other attributes and dropped

the attributes which had very skewed distribution (same values have very little information gain). E.g. we dropped the attribute of native Country from the data set as many of them had the same value so wont be affecting the prediction attribute.

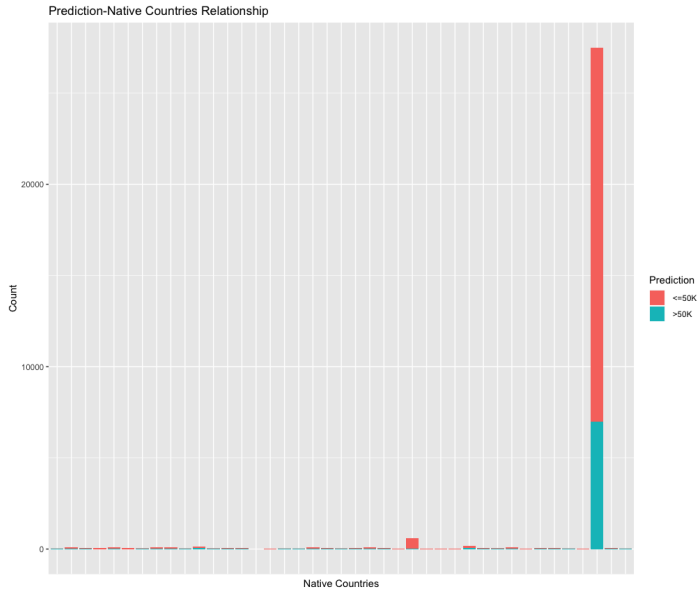

Figure 3: Effect of the Native Country attribute on the class labels

1.1 Missing Values

After dropping the columns, the next step in the data cleaning process was to find the number of missing values in each attribute. We found that only 2 columns had any missing values in the data set (The missing values in the data set are represented by " ?"). From the high correlation of null values between workclass and occupation, we deduced that the values assigned to them were simply missing from the list i.e all the missing values should have had the same values.

Attribute Name	Missing Values
workclass	1836
occupation	1843

Thus instead of omitting these instances, we replaced it with a random value "X" for both the attributes.
Replacing values with a placeholder, instead of omitting these rows was justified for two reasons:

1. The number of missing values was high and forms a good portion of values in the *workclass* attribute as deduced from *Figure*4
2. Using domain knowledge it suggests that as both of these attributes have almost same number of null values meaning they have a value which was not present in the list during the collection of data.

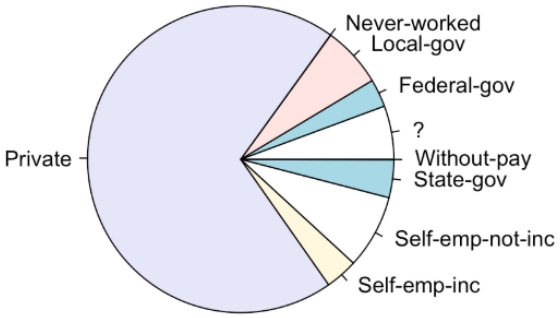

Figure 4: Distribution of workclass in the data set

At the end of Data Cleaning, we dropped 4 attributes(fnlwgt, race, native.country, capital.loss) and replaced the missing values with (assumed) appropriate value. Also based on the correlation, we dropped the column *education.num* as it had a high correlation with *education* attribute, hinting that both signify the same thing.

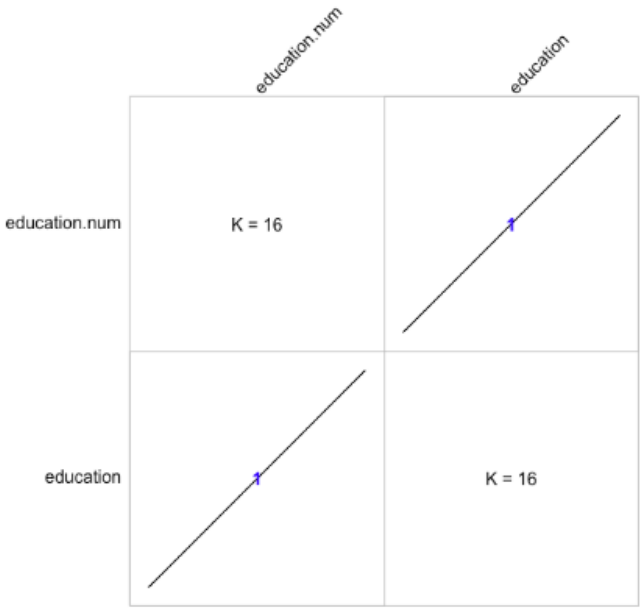

Figure 5: Correlation between eduction.num and education column

2. DATA MINING TASK SELECTION

Various techniques in data mining.

1. Classification
2. Regression

3. Clustering

After studying the data it was obvious to eliminate some of the data mining tasks, like regression, as the value to be predicted is not continuous. Similarly as the label were given and the task was not to see how many ways the data could be partitioned, we eliminated clustering. So we defined the task as to classify a given instance into one of the two classes, more specifically the data mining task was of binary classification.
Algorithms used in classifying the given data set.

1. Decision Tree `(rpart)`
2. Random Forest `(randomForest)`
3. Naive Bayes`(e1071)`

After selecting these algorithms; the need to normalize, convert categorical values to numeric was eliminated. Before running the stated algorithm, we performed correlation of attributes with the labels to check which attributes would be of higher importance. For example we assumed education, Capital Gain and age would be more important to predict the label.

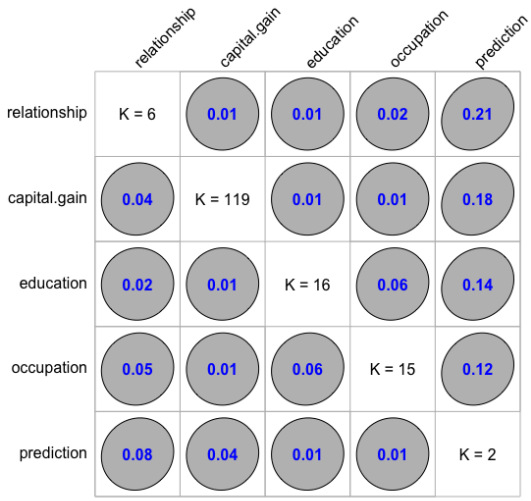

Figure 6: Correlation plot of relevant attributes in the data set

2.1 Decision Tree

On running the decision tree algorithm, the accuracy printed in *Figure*7 was observed. One interesting observation was that attributes which showed some significant correlation with class labels *prediction* played key role in the implementation phase. This can be seen in the decision tree for the data set (attached in the R script report).

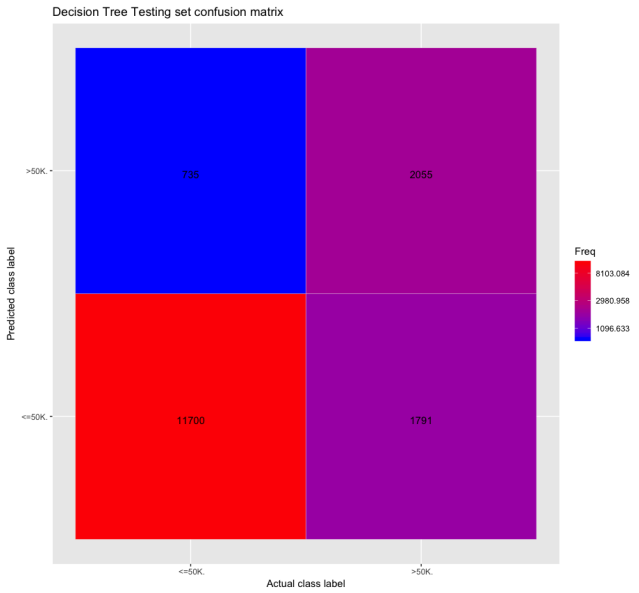

Figure 7: Confusion Matrix for Decision Tree Classifier

2.2 Naive Bayes model

The rationale of using Naive Bayes model for classification was that the data set contains information about every individual and tries to collect the data points which can define the annual income. This model is more of a probabilistic model than the hard rules i.e it makes more sense to say that given this information its more likely that the income of this individual is greater than 50K.

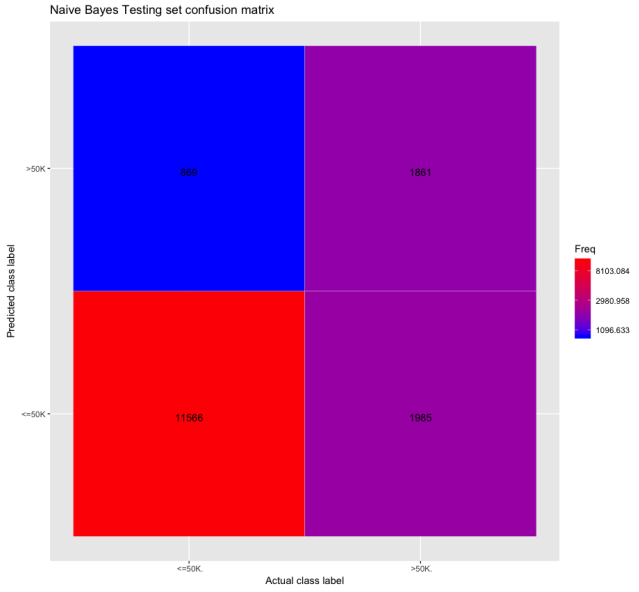

Figure 8: Confusion Matrix for Naive Bayes Classifier

2.3 Random Forest model

Random Forest is an ensemble method. It uses multiple decision trees created by adding weights to the training examples, and uses majority voting to decide the final output of the ensemble. As the ensemble algorithm gives different weight age to every instance, it its able to reduce of effect of noise in the data set.

By looking at the confusion matrix, the model predicts incorrectly mostly when the label is greater than 50 K (which the model predicts to be less than the 50K), this is expected because the data set is more biased towards the instances being having label less than 50K.

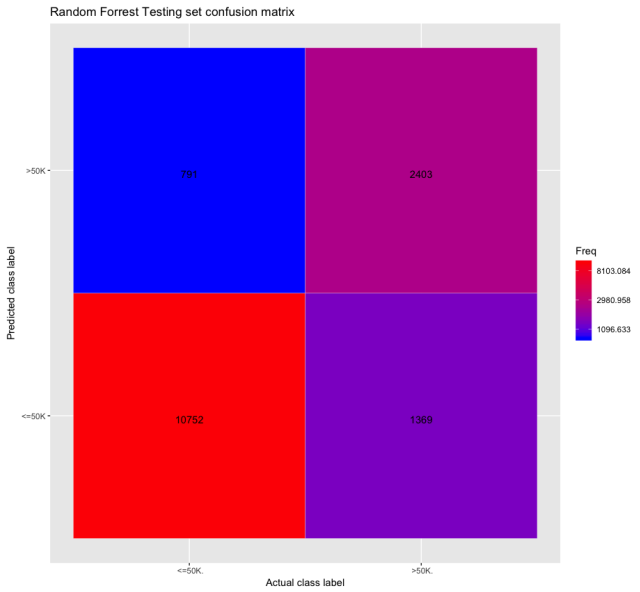

Figure 9: Confusion Matrix for Naive Bayes Classifier

3. WORK DISTRIBUTION

Task	Handled By
Missing Value analysis	Ketan
Atrribute selection	Kavya
Model Selection	Siddharth
Model Coding and comparison	Ameya
Documentation	Ketan
inference drawing	Ameya, Siddharth
Data visualization	Kavya

4. CONCLUSION

In this project, we built three different models i.e Decision Tree, Random Forest and Naive Bayes for the selected data set, and chose Random Forest as the appropriate model to classify the instance into required label. After analyzing the confusion matrix of all the three model it is evident that when the model predicts incorrectly it mostly predicts it as label $<= 50k$. This is due to the absence of concrete rules to define the annual income. The Naive Bayes performs the worst, looking at the confusion matrix, simply because Naive Bayes will have higher probability for instance having label $<= 50k$ given that the training data set is skewed towards $<= 50k$ class labels. The Random Forest does the best job of avoiding the noise and creating general enough rules to classify the data into the labels. Based on rules generated by the decision tree, the attribute *Capital.Gain*, *Education*, *Relation* are required to predict the annual income of an individual. Below are the accuracy of the three trained models on testing data.

model	Decision tree	Naive Bayes	Random Forest
accuracy	84.45%	82.78%	85.49%

The Random Forest performs better because it creates multiple trees which helps it to generalize the data better, Naive Bayes would have out performed the other two models hadn't the data set being so skewed.