**PROJECT 1 MILESTONE REPORT**

**Team No. 19**

(tuple size: 4k, seed-val:35, train:test = 60:40)

Submitted By:

1. Sonika Sarangi Annaiah – 1001733497
2. Vaishakh Deepak Ghati - 1001738753

**PROJECT OVERVIEW**

* The dataset containing the census data consisting of 32,000+ rows was downloaded and imported to R studio as a txt file for analysis.
* The dataset contained over 2000+ missing or unknown(?) values.
* The tuples with missing/unknown values(?) were omitted using the na.omit() function. Over 2,000 rows were omitted and a clean dataset was obtained for analysis.
* The names of variables from the attributes list was taken from the given project description file and a dataframe called ‘df’ was created.
* The seed value was set to 35 as instructed. About 4000 rows were sampled and was split into training data and test data.
* Schema was imposed on the dataset.
* The classification was done using Information gain, GINI index and Naïve Bayes on the dataset.

**INFORMATION GAIN**

* Using the rpart API in R, decision tree was built for the training set. By giving the parameter information gain, the API built the decision tree using information-gain instead of the default gini index.
* The decision tree helped us in understanding the attributes which are used in split criteria.
* The Information.gain API ranked all the attributes in the relative order of Information gain as given below:

attr\_importance

|  |  |
| --- | --- |
| age | 0.086344158 |
| workclass | 0.021142475 |
| fnlwgt | 0.000000000 |
| education | 0.117544148 |
| education\_num | 0.111162798 |
| marital\_status | 0.157030142 |
| occupation | 0.107065152 |

|  |  |
| --- | --- |
| relationship | 0.172465800 |
| race | 0.007671101 |
| sex | 0.037873947 |
| capital\_gain | 0.097742991 |
| capital\_loss | 0.040897648 |
| hours\_per\_week | 0.059180305 |
| native\_country | 0.026166586 |

* The priority attributes involved in construction of decision tree were relationship, education and capital\_gain. They were the highest ranked in order of information gain.
* The other attributes that have higher information gain other than the priority attributes was found to be marital\_status, occupation and education\_num.
* After the first branch in rpart API, cp=-1 was used and it developed itself into a complex tree. In a complex tree, occupation and marital\_status were used as attributes other than the priority attributes to perform decision making. (Thereby justifying the utility of the unused priority attributes)
* The predict method was used to predict the test dataset and a confusion matrix was obtained with the predicted values of the test dataset and actual values of the test dataset.

Predicted Values

|  |  |  |
| --- | --- | --- |
| Actual Values | <= 50k | >50k |
| <=50k | 1163 | 171 |
| >50k | 61 | 205 |

The results of Accuracy, Precision, Recall and F1 score was found to be:

**Accuracy : 0.8550000 Recall : 0.9501634 Precision : 0.8718141 F1 score : 0.9093041**

If we dropped any attribute that isn’t used in decision making of the tree, we would get the same result. But as a principle, we drop the attribute that is of the least importance by information gain, i.e fnlwgt.

Furthermore, education\_num and hours\_per\_week are attributes that are never used in the creation of the complex decision tree or the normal decision tree as well. So these attributes can also be dropped.

Upon withholding these attributes, (since these attributes weren’t used in the building of the decision tree) we get the same result as ones without withholding.

Here the choice of the attributes was based on low ranking of information-gain, and the fact that these attributes were never used in decision making both the simple and complex decision trees.

The confusion matrix obtained for complex-decision tree is

Predicted Values

|  |  |  |
| --- | --- | --- |
| Actual Values | <= 50k | >50k |
| <=50k | 1093 | 157 |
| >50k | 131 | 219 |

The results of Accuracy, Precision, Recall and F1 score was found to be:

**Accuracy : 0.820000 Precision : 0.8744000 Recall : 0.829739 F1 score : 0.8835893**

Accuracy, Recall, F-score are lesser than the ones for a non-complex decision tree perhaps due to overfitting.

**GINI INDEX**

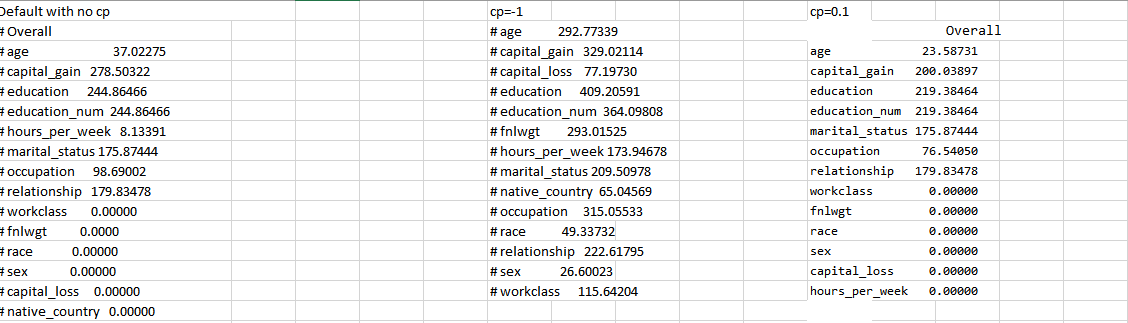
* For Gini index we built the decision tree before removing the attributes for a variety of **complexity values(cp). decision\_tree <- rpart(income ~ ., data = train, method = "class",parms = list(split = 'gini'), minsplit = 2,minbucket = 1, cp =-1)[cp indicates complexity]**
* **printcp(mytree\_gini**) gave the result as below:

CP nsplit rel error xerror xstd

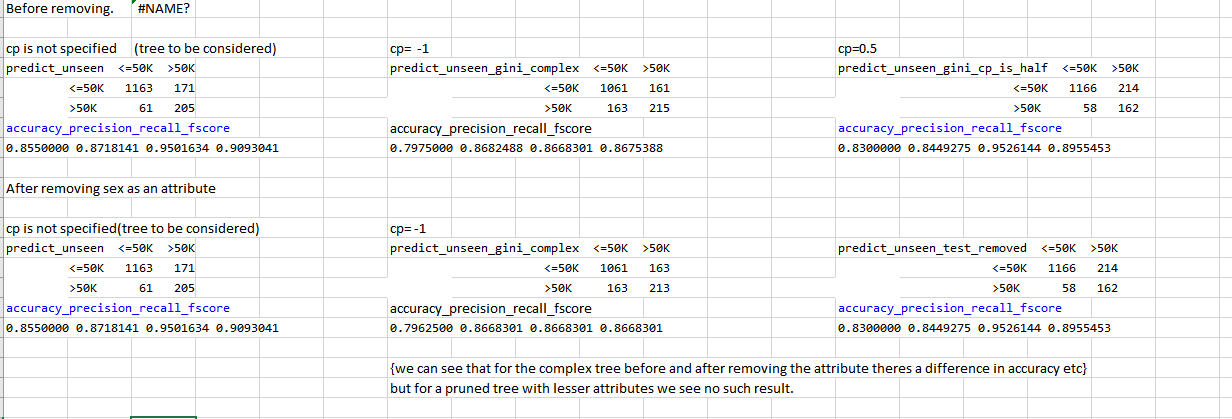
1 0.154174 0 1.00000 1.00000 0.035874

2 0.073254 2 0.69165 0.71039 0.031622

* We can build decision trees with different complexities. Ideally, we must **minimize the root-mean-square xerror and ensure a greater confidence interval of xstd. Such a tree is the best pruned tree with least overfitting.**
* Not specifying any cp parameter will obtain the best pruned tree which we have used to interpret results(However it’s best to manually calculate least xerror and choose such a cp value)
* Upon experimenting with trees of different complexities(different cp values), and running the **varImp(mytree\_gini,scale = FALSE)** API to get the different importance of various attributes, we get **sex as an attribute with least importance** in all the trees, a sample of which is provided below.



* The confusion matrix obtained when sex attribute is withheld for different values of cp is shown below:



We can see that for the complex tree(cp= -1) before and after removing the attribute there is a difference in accuracy, precision, recall and f1-score. But for a pruned tree with lesser attributes we see no such difference.

**In the best pruned tree, relationship, capital-gain, education are the only attributes used to build the tree. Therefore, it doesn’t make much of difference if we remove sex as an attribute**. **However, for the complex tree that uses all attributes we can notice there is a difference by removing ‘sex’ as an attribute as accuracy etc varies as shown in the figure above. However it doesn’t vary much(very less fluctuation in Accuracy and other indicators) indicating that we’ve chosen the attribute to be removed that impacts the least.**

The confusion matrix obtained for Gini Index is shown below:

Predicted Values

|  |  |  |
| --- | --- | --- |
| Actual Values | <= 50k | >50k |
| <=50k | 1163 | 171 |
| >50k | 61 | 205 |

The results of Accuracy, Precision, Recall and F1 score was found to be:

**Accuracy : 0.8550000 Recall : 0.9501634 Precision : 0.8718141 F1 score : 0.9093041**

**NAIVE BAYES**

* **naiveBayes(income ~., data=train)** API in **library(e1071)** was used to perform Naïve Bayes classification and the confusion matrix was obtained as shown below:

Predicted Values

|  |  |  |
| --- | --- | --- |
| Actual Values | <= 50k | >50k |
| <=50k | 1140 | 183 |
| >50k | 84 | 193 |

The results of Accuracy, Precision, Recall and F1 score was found to be:

**Accuracy : 0.8331250 Precision : 0.8616780 Recall : 0.9313725 F1 score : 0.8951708**

**COMPARISON**

1. **Compare Information Gain and GINI index**

A: From the analysis performed, we can infer that the confusion matrix obtained for both Information Gain and Gini Index is the same as below:

Predicted Values

|  |  |  |
| --- | --- | --- |
| Actual Values | <= 50k | >50k |
| <=50k | 1163 | 171 |
| >50k | 61 | 205 |

The results of Accuracy, Precision, Recall and F1 score was found to be:

**Accuracy : 0.8550000 Precision : 0.8718141 Recall : 0.9501634 F1 score : 0.9093041**

For **complex tree**, the confusion matrix obtained after performing classification using Information gain and Gini Index is shown below:

1. Information Gain 2) Gini Index

|  |  |  |
| --- | --- | --- |
| Actual values | <=50k | >50k |
| <=50k | 1061 | 161 |
| >50k | 163 | 215 |

Predicted Values Predicted Values

|  |  |  |
| --- | --- | --- |
| Actual values | <= 50k | >50k |
| <=50k | 1093 | 157 |
| >50k | 131 | 219 |

**Accuracy :** **0.820000 Precision :** **0.8744000**

**Recall : 0.829739 F1 score : 0.8835893**

**Accuracy**: **0.7975000** **Precision**: **0.8682488**

**Recall** : **0.8668301**  **F1-score** : **0.8675388**

**We can see that Information gain classification performs slightly better than Gini Index on all parameters for the complex tree. This is because as the number of attributes increase as the tree gets increasingly complex. If there are 2-3 attributes we can either use Gini or information gain. But if attribute complexity increases we need better splitting ways as difference in values between 2 attributes get smaller and smaller as splitting is happening at micro level. (coz the number of rows of a dataset in consideration after a split is lesser.)**

**So, on large splits information gain (uses log summation) is better.**

**2.Compare Information Gain and Naïve Bayes**

1. A: Information Gain 2) Naive Bayes

|  |  |  |
| --- | --- | --- |
| Actual values | <=50k | >50k |
| <=50k | 1140 | 183 |
| >50k | 84 | 193 |

Predicted Values Predicted Values

|  |  |  |
| --- | --- | --- |
| Actual values | <= 50k | >50k |
| <=50k | 1163 | 171 |
| >50k | 61 | 205 |

**Accuracy : 0.8550000 Precision : 0.8718141**

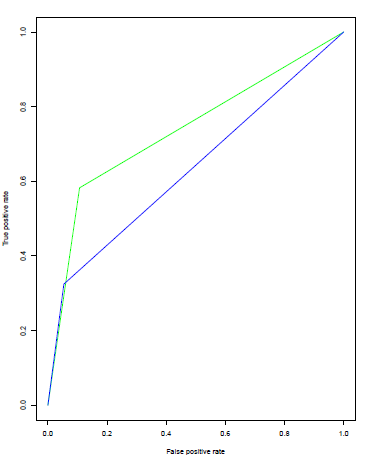
**Recall : 0.9501634 F1 score : 0.9093041**

**Accuracy**: **0.8331250 Precision**: **0.8616780**

**Recall** : **0.9313725**  **F1-score** : **0.8951708**

The **Receiver Operating Characteristic (ROC)** curve was plotted for Information Gain and Naive Bayes and the graph shown below was obtained.

Green Color curve depicts **Information Gain** Blue Color curve depicts **Naïve Bayes**



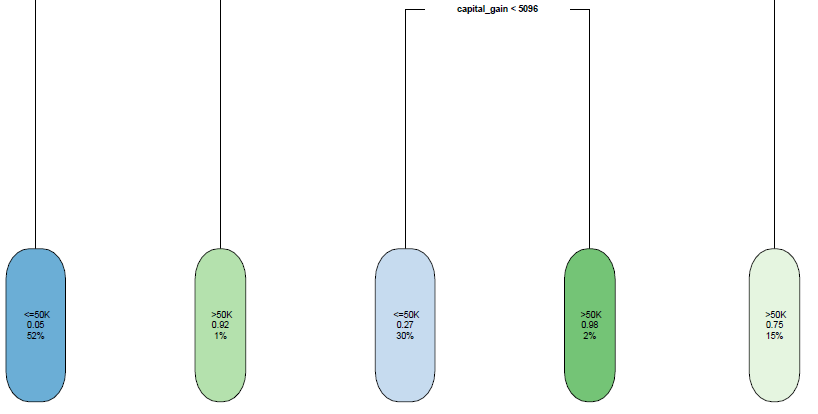
**We can see that the area under ROC curve is more for Information Gain than Naïve Bayes. Hence, information gain is a better classification model than Naïve Bayes.**

**Also, we can see that,**

**Accuracy (Info\_gain)>accuracy (Naïve bayes)**

**(precision, fscore,recall)[info\_gain] > (precision, fscore,recall)[Naïve\_Bayes]**

**The reason why Information gain is better than Naïve Bayes is the rpart API builds a most optimized tree by internally splitting the train input data to train and test and and building the tree such that all of training data isn’t classified rightly which may result in overfitting.(That’s why even inside the tree there is a percentage of classification/misclassification.)**



Misclassification percentage is printed in the image above

What Naïve Bayes does is that – it just computes P(<50K/A1=val1,A2=val2….) using P(A1=val1,A2=val2,…/<50K.)

**Basically it takes into account all the Training examples where A1=val1 etc for all the <50K attributes.**

**Therefore, there are chances of overfitting. Furthermore, if the number of training examples are more, the chances of P(A1=val1/<50K) where val1 for A1 maybe misclassified/Outlier/ is more. Here we have 4000 sample size. Hence, a well pruned decision tree fits better than Naïve Bayes for a larger training and test set.**

**3. Compare Gini Index and Naïve Bayes**

1. Gini Index 2) Naive Bayes

|  |  |  |
| --- | --- | --- |
| Actual values | <=50k | >50k |
| <=50k | 1140 | 183 |
| >50k | 84 | 193 |

Predicted Values Predicted Values

|  |  |  |
| --- | --- | --- |
| Actual values | <= 50k | >50k |
| <=50k | 1163 | 171 |
| >50k | 61 | 205 |

**Accuracy : 0.8550000 Precision : 0.8718141**

**Recall : 0.9501634 F1 score : 0.9093041**

**Accuracy**: **0.8331250 Precision**: **0.8616780**

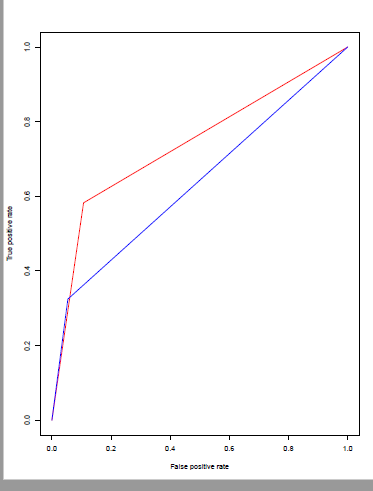
**Recall** : **0.9313725**  **F1-score** : **0.8951708**

**Also, we can see that, Accuracy (Info\_gain) > accuracy (Naïve bayes) (precision, fscore,recall)[info\_gain] > (precision, fscore,recall)[Naïve\_Bayes]**

The **Receiver Operating Characteristic (ROC)** curve was plotted for Gini Index and Naive Bayes

and the graph shown below was obtained.

Red Color curve depicts **Gini Index** Blue color depicts **Naïve Bayes**



**We can see that the area under ROC curve is more for Gini Index than Naïve Bayes. Hence, Gini Index is a better classification model than Naïve Bayes.**

**In fact the gini index values for the most pruned tree is the same as the info gain tree( it’s the same tree with same confusion matrix)**

**Therefore for the comparison of Gini vs Naïve Bayes, the interpretation is the same as that of Infogain vs NaiveBayes for the same reason Gini Tree performs better than Naïve Bayes. Essentially decision tree performs better than Naïve Bayes for a larger dataset.**

**CHALLENGES**

* To compute decision trees of various complexities to understand which attribute to remove, and to arrive at the conclusion on which cp value to remove (after observing similar xerror values)
* To derive insights from confusion matrix and understand comparisons between various classification models(gini index, naïve bayes, Info gain)
* Understanding various APis in R and how to use them to arrive at right insights. Printing the decision tree and understanding what API does, how it's able to arrive at this result, and coz it's arriving at a given result, to know how and why it must be computing it, and thereby appreciate the effectiveness of the model.

**FILE NAMES**

1. **data.txt** – Text file containing the raw data
2. **R Files** folder contains the following files: **gini\_index.R** contains analysis for Gini Index, **infogain\_analysis.R** contains analysis for Information Gain, **naive\_bayes.R** contains analysis for Naive Bayes and **roc.R** containing the code for plotting ROCs between different classification models.
3. **treeplotpdfs** folder contains the documents having decision trees with and without complexity and complextree text file which is a representation of complex tree to understand which attributes are involved.