Name: Shuvrima Alam | Team Number: 3 | CSE 5334-001 | Date: 09/24/2019

Project1 Analysis

Basic Information

Sample Size: 2000	**Train/Test:** 70/30	**Train Data Size:** 1388
Test Data Size: 612	**Population Size (before cleaning):** 32561	**Population Size after cleaning:** 30162
census.adult: The dataset imported in Rstudio environment and used in code.	**census.adult_clean**: The cleaned dataset in Rstudio environment and used in code.	**Seed Value :**10

The required sample size is 1034 at confidence level of 95% and margin error of 3% for a given population size of 32561. So, our sample size satisfies that much.

Overall Status

For the final project, classification of the dataset using information gain, naïve bayes and GINI index were completed. For information gain and GINI index one column was withheld for each cases and the given data was classified again. The process was started by cleaning the data set by omitting all the NA values in R. Using the import button in R studio the data was set in the environment and was named “census.adult”. After the data was imported, all “?” values were replaced with “NA”. Second step to clean the data involved omitting those NA values. For that “!complete.cases” was used in R to count the number for rows with missing values. That count was stored in a vector called “na_vec”. A census.adult_clean variable was created where all the rows without NA values were stored.

Once the data was processed, using the given seed value of 10 and a sample size of 2000 as per the requirement was generated. Now the sample had to be split in ratio of 70/30 for train/test. Again, using the sample function and setting the “prob” value in it the data was partitioned and stored separately in train and test variables. As mentioned in the project description rpart library was used to generate the decision trees. By default rpart uses gini index to split so to use information gain “parms = list(split = 'information')” was used and when classification using GINI index was done the default was kept. Using predict function in R the test data was ran against the train data in order to classify each time. For naïve bayes R has a library “naivesbayes” which enabled modelling the training data. Before the huge data set was tested small datasets were used to understand the techniques used for this analysis each time.

The column to withhold for information gain and GINI index was found by using “varImp” which ranked the variables in order of importance. The third column(fnlwgt) had varImp value of 0.0000000 so that was omitted for those classifications. To analyze my classification processes, I have calculated confusion matrix, accuracy, F1 score and related information below.

File Descriptions

These files are in sub-folder **census** of primary folder **Proj1Fall19_team_3**

File Name	Description
census_ig.R	Contains the R code written to classify the dataset using information gain contained in subfolder **decision_trees**.

census_gini.R	Contains the R code written to classify the dataset using GINI index contained in subfolder **decision_trees**.
census_nb.R	Contains the R code written to classify the dataset using Naïve Bayes contained in subfolder **naive_bayes.**
census-adult.txt	Contains the dataset given to us.
TableOfTechniques	This file contains a list of all the procedures and algorithms used to complete this project.

Division of Labor

Task	**Time Taken (in hours)**
To get familiar with R for these classifications.	1
To learn the concepts discussed for these classification techniques.	3
To do classification.	2
To analyze and write the report and all documents.	4
To do the bonus analysis.	2
Overall Time Taken: 12	

This project was done ***individually*** except for bonus section.

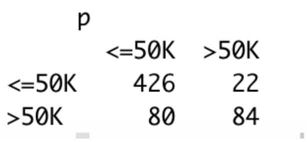
Problems encountered and handled

Problem Description	**Method of Handling**
The concepts were not fully understood just by briefly reading about it from slides. Time was spent in figuring out the basics required to do the project.	After going through the textbook chapters and doing research online the doubts were cleared.
R has many libraries and many algorithms regarding one approach. Too many options caused some confusion.	To clear this confusion, I read the documentation and researched online to find best and simplest way that would save my time.
Logical confusion regarding which information to keep in the report and which to discard.	To solve this confusion I tried to restrict my analysis report to 7 pages and only mentioned the very key points regarding my analysis.

Detailed Analysis

1)I. Classification using Information Gain

Confusion matrix after applying model to test data:



```
          p
           <=50K  >50K
  <=50K     426     22
  >50K       80     84
```

Accuracy: TP+TN/Total =510/612=0.833

Miscalculate Rate: (FP+FN)/Total = 80+22/612 = 0.167

Recall: TP/TP+FN = 426/(426+80)= 0.842

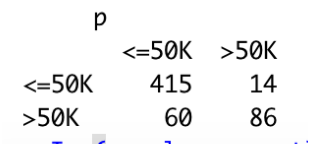
Precision: TP/TP+FP = 426/(426+22)= 0.951

F1 score =2*((precision*recall)/(precision+recall)) = 1.6015/1.793 = 0.893

1)II. Classification with Information Gain after withholding one column(third one)

V3 (fnlwgt column) is one of the **least important** columns which has been found by **varImp function** and so it was chosen to be withheld.

Confusion matrix after applying model to test data:



```
         p
          <=50K  >50K
  <=50K     415    14
  >50K       60    86
```

Accuracy: TP+TN/Total = 501/575= 0.87 **Higher than above method!**

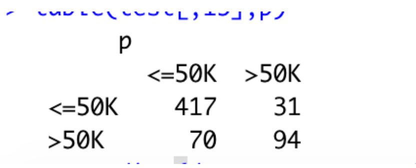
Miscalculation Rate: FP+FN/Total= 60 + 14/ 575 = 0.129 **Lower than above!**

Recall: TP/TP+FN = 415/(415+60)= 0.874 **Higher than above method!**

Precision: TP/TP+FP =415/(415+14) = 0.967 **Higher than above method!**

F1 score=2*((precision*recall)/(precision+recall)) = 1.6903/1.841 = 0.918 **Better score!**

1)III. Classification using GINI index



```
> table(test[,15],p)
         p
          <=50K  >50K
  <=50K     417    31
  >50K       70    94
```

Accuracy: TP+TN/Total = 511/612= 0.835

Miscalculation Rate: FP+FN/Total= 70+ 31/ 612 = 0.165

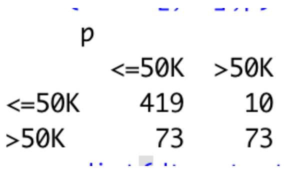
Recall: TP/TP+FN = 417/(417+70)= 0.856

Precision: TP/TP+FP =417/(417+31) = 0.931

F1 score =2*((precision*recall)/(precision+recall)) = 1.59387/1.787= 0.892

1)IV. Classification using GINI index after withholding one column(third one)

V3(fnlwgt column) is one of the **least important** columns which has been found by **varImp function** and so it was chosen to be withheld.



```
          p
            <=50K  >50K
  <=50K       419    10
  >50K         73    73
```

Accuracy: TP+TN/Total = 419+73/575= 0.856 **Better than above GINI method!**

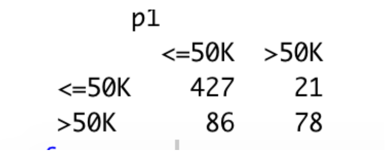
Miscalculation Rate: FP+FN/Total= 73+ 10/ 575 = 0.144 **Better than above GINI method!**

Recall: TP/TP+FN = 419/(419+73)= 0.852 **Slightly less than above GINI method!**

Precision: TP/TP+FP =419/(419+10) = 0.977 **Better than above GINI method!**

F1score =2*((precision*recall)/(precision+recall)) = 0.910 **Better than above GINI method!**

2) Classification using Naïve Bayes



```
          p1
            <=50K  >50K
  <=50K       427    21
  >50K         86    78
```

Accuracy: TP+TN/Total = 427+78/612= 0.825

Miscalculation Rate: FP+FN/Total= 21+ 86/ 612 = 0.175

Recall: TP/TP+FN = 427/(427+21)= 0.832

Precision: TP/TP+FP =427/(427+86) = 0.953

F1 score=2*((precision*recall)/(precision+recall)) = 0.889

Comparision between Classification Methods

a. Information Gain VS. GINI Index

Metric Used	**Information Gain**		**GINI Index**	
	Without Withholding any column	**With Withholding column**	**Without Withholding any column**	**With Withholding column**
Accuracy	**0.833**	**0.87(best)**	**0.835(better)>0.833**	**0.856**
Miscalculation Rate	**0.167**	**0.129(best)**	**0.165(better)<0.167**	**0.144**
Recall	**0.842**	**0.874(best)**	**0.856(better)>0.842**	**0.852**
Precision	**0.951(better)>0.931**	**0.967**	**0.931**	**0.977(best)**
F1 Score	**0.893(better)>0.892**	**0.918(best)**	**0.892**	**0.910**
Overall Conclusion: Information Gain was **better** than GINI according to majority of metrics used specially after withholding V3.				

As we can see from above table of description Information Gain clearly performed better than GINI Index. Although without withholding column GINI index performs slightly better than Information Gain, when it comes to withholding columns Information Gain is a clear winner. Overall, if we count the number of times information Gain has performed better than GINI Index then it is greater which demonstrates that it did a better job at classification.

b. Information Gain VS. Naïve Bayes

Metric Used	Information Gain		Naïve Bayes
	Without Withholding any column	With Withholding 1 column	
Accuracy	0.833(better)>0.825	0.87(best)	0.825
Miscalculation Rate	0.167(better)<0.175	0.129(best)	0.175
Recall	0.842(better)>0.832	0.874(best)	0.832
Precision	0.951	0.967(best)	0.953(better)>0.951
F1 Score	0.893(better)>0.889	0.918(best)	0.889
Overall Conclusion: Information Gain was **better** than Naïve Bayes specially after withholding V3 and also without withholding any column.			

In this above case, information gain is a clear winner and specially after withholding column. Since, no column was withheld for Naïve Bayes even if we just compare it with Information Gain without withholding any column it performs worse for our given data set.

c. GINI Index vs Naïve Bayes

Metric Used	GINI Index		Naïve Bayes
	Without Withholding any column	With Withholding column	
Accuracy	0.835(better)>0.825	0.856(best)	0.825
Miscalculation Rate	0.165(better)<0.175	0.144(best)	0.175
Recall	0.856(best)	0.852(better)>0.832	0.832
Precision	0.931	0.977(best)	0.953(better)>0.931
F1 Score	0.892(better)>0.889	0.910(best)	0.889
Overall Conclusion: GINI Index is **better** than Naïve Bayes and specially after withholding V3 and also without withholding any column.			

In this above case, GINI Index is a clear winner and specially after withholding column. Since, no column was withheld for Naïve Bayes even if we just compare it with GINI Index without withholding any column it performs worse for our given data set. Naïve Bayes only has better precision, but all the other metrics' values were better for GINI Index.

d. Information Gain without withholding column(I) VS. Information Gain with withholding column 3(II)

Metric Used	Information Gain	
	Without Withholding any column	With Withholding column

Accuracy	**0.833**	**0.87(better)**
Miscalculation Rate	**0.167**	**0.129(better)**
Recall	**0.842**	**0.874(better)**
Precision	**0.951**	**0.967(better)**
F1 Score	**0.893**	**0.918(better)**
Overall Conclusion: Information Gain after withholding one column yields better results.		

e. GINI Index without withholding column(III) VS. GINI Index with withholding column 3 V3(IV)

Metric Used	**GINI Index**	
	Without Withholding any column	**With Withholding column**
Accuracy	**0.835**	**0.856(better)**
Miscalculation Rate	**0.165**	**0.144(better)**
Recall	**0.856(better)**	**0.852**
Precision	**0.931**	**0.972(better)**
F1 Score	**0.892**	**0.910(better)**
Overall Conclusion: Overall GINI Index yielded better results after withholding column.		

Feature selection yields better results. GINI Index performs better in every way after column V3 was withheld except in case of recall value as observed from the above table.

Summary of Analysis and Details

- **V3 (fnlwgt column)** is one of the **least important** columns which has been found by **varImp function** and so it was chosen to be withheld for necessary parts. As we know that feature selection and ignoring unnecessary variable largely improves accuracy and training time. Both GINI Index and Information Gain gave better results after withholding column.
- The green cells under the "Comparision between Classification Methods" indicate values that are better comparatively. Some of the sub-column names are shaded to demonstrate the comparison between values. For instance, "Without withholding any column" is colored yellow in part c and the values that come from that column for comparison purposes are colored the same.
- The confusion matrices printed using R code shows the true positive values, true negatives, false positives and false negatives using which the metrics used for comparison were calculated. The importance of each metric varies in different scenarios. For instance, **Accuracy** is used when the True Positives and True negatives are more important while **F1-score** is used when the False Negatives and False Positives are crucial. For my analysis, I used each metric and compared them and judged keeping all of them under consideration.
- Overall, Information Gain yields best results followed by GINI Index and then Naïve Bayes for this sample size. **Naive Bayes** does quite well when the training data doesn't contain all possibilities so it can be very **good** with low amounts of data. **Decision trees** work **better** with lots of data **compared** to **Naive Bayes.**

Bonus Analysis (Team 2,3,4)

1. Analysis with different splits with the same seed(10)

a. Split 80:20

	Naïve Bayes	Information Gain	GINI Index
Accuracy	0.825	0.86	0.8675
Precision	0.85	0.875	0.875
Recall	0.934	0.96	0.9639
F1	0.89	0.917	0.9173

b. Split 70:30

	Naïve Bayes	Information Gain	GINI Index
Accuracy	0.825	0.87	0.856
Precision	0.953	0.967	0.977
Recall	0.832	0.874	0.852
F1	0.889	0.918	0.910

c. Split 60:40

	Naïve Bayes	Information Gain	GINI Index
Accuracy	0.8088	0.84	0.835
Precision	0.822	0.864	0.8721311
Recall	0.9399657	0.9262	0.9125214
F1	0.877502	0.8940	0. 8909091

- The **Naïve Bayes** model performs the worst for all the splits when compared to the Decision tree classifiers. Results for splits a and b are the same. However, split c has the lowest performance for the model indicating **under-fitting**.
- The **information gain** model shows the best results for all the splits when compared to the Naïve Bayes model and GINI index. Results for split b has the highest accuracy which **indicates over fitting** for split a and **under-fitting** for split c.
- Decision tree classifier using GINI index has a moderate performance for all the splits. Split a shows best results indicating probable **over fitting** for the other 2 splits.
- After comparing these values with the values obtained for the entire dataset, we observed that the values are quite close. This indicates that the sample obtained is a **good representative** of the original data.