MARKET BASKET ANALYSIS

Submitted by

Ritika	18BLC1027
S. Harshavardhini	18BLC1053
Shreni Agrawal	18BLC1012

J Component - Report

ECM2002 – Machine Learning Algorithms

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMPUTER ENGINEERING

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

June 2020

TABLE OF CONTENTS

ABSTRACT

In this project, we experiment with a real world dataset, and to explore how machine learning algorithms can be used to find the patterns in data.

Income is the consumption and saving opportunity gained by an entity within a specified time frame. Household income is the combined gross income of all members of a household who are 15 years or older. Information about income can be used in many parts crucial parts of our daily life like in how different patterns of income distribution influence household well-being and people's ability to get the goods and services to meet their needs and wants, for tax determination, loan and other banking purposes.

Market Basket Analysis deals with a data set which uses various factors like education level, occupation, marital status and ethnicity to determine the annual income of a person. We use these social facets to find those which among them plays a major role in deciding the annual income of a person and which are irrelevant.

SECTION I

DATA SET DESCRIPTION

The dataset is an extract from this survey. It consists of 14 demographic attributes. The dataset is a good mixture of categorical and continuous variables with a lot of missing data. This is characteristic for data mining applications.

A data frame with 8993 observations on the following 14 variables.

Income

Annual Income of Household

1. Less than $10,000 2. $10,000 to $14,999 3. $15,000 to $19,999 4. $20,000 to $24,999 5. $25,000 to $29,999 6. $30,000 to $39,999 7. $40,000 to $49,999 8. $50,000 to $74,999 9. $75,000 or more

Sex

1. Male 2. Female

Marital Status

1. Married 2. Living together, not married 3. Divorced or separated 4. Widowed 5. Single, never married

Age

1. 14 thru 17 2. 18 thru 24 3. 25 thru 34 4. 35 thru 44 5. 45 thru 54 6. 55 thru 64 7. 65 and Over

Edu

1. Grade 8 or less 2. Grades 9 to 11 3. Graduated high school 4. 1 to 3 years of college 5. College graduate 6. Grad Study

Occupation

1. Professional/Managerial 2. Sales Worker 3. Factory Worker/Labourer/Driver 4. Clerical/Service Worker 5. Homemaker 6. Student, HS or College 7. Military 8. Retired 9. Unemployed

Lived

How long the person lived in San Fransisco/ Oakland /San Jose area?

1. Less than one year 2. One to three years 3. Four to six years 4. Seven to ten years 5. More than ten years

Dual_Income

Dual Incomes (If married)

1. Not Married 2. Yes 3. No

Household

No. of people in the household

1. One 2. Two 3. Three 4. Four 5. Five 6. Six 7. Seven 8. Eight 9. Nine or more

Householdu18

People in the household under 18

0. None 1. One 2. Two 3. Three 4. Four 5. Five 6. Six 7. Seven 8. Eight 9. Nine or more

SECTION II

BASIC MODELS

1. LINEAR DISCRIMINANT ANALYSIS

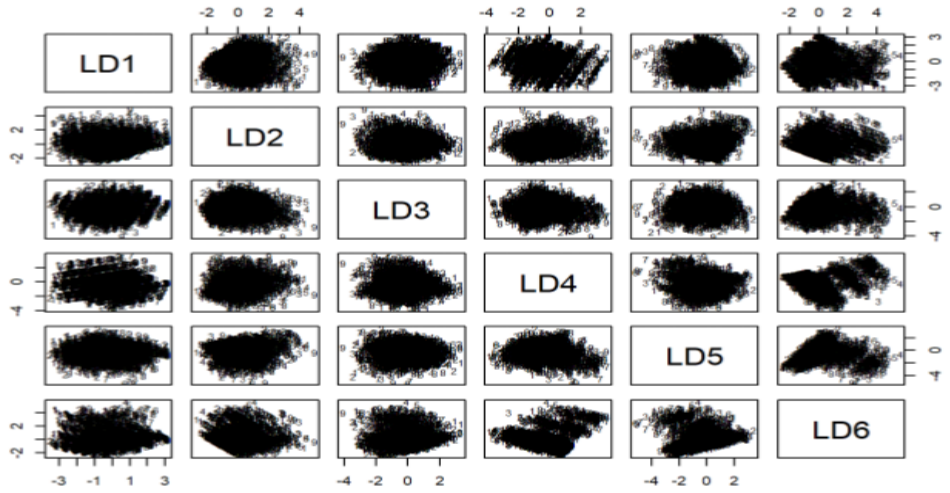
In our dataset (market basket analysis) LDA is performed since it is a classification problem with more than two attributes.

It is done on the basis of taking AGE less than or equal to 64 years for training data set and rest for testing.

The predict () function returns a list with three elements. The first element, class, contains LDA's predictions about the Annual Income

Applying a 50% threshold to the posterior probabilities allows us to recreate the predictions contained in lda.pred$class.

The LDA fit for the following dataset is given below:




```
## [1] 0.1679389
```

Here we got accuracy as **16.79%** which is not a good prediction.

So, we will use further extension of linear model like KNN and Bootstrap.

2. K NEAREST NEIGHBOUR (KNN)

While proceeding further to increase our accuracy, we applied KNN using k=83.

Applying KNN to our dataset, we got the following confusion matrix:

```
##                     m1
## market_test_target   1   2   3   4   5   6
##                  1   0  15   3   6   0   0
##                  2   0  59   9  39   1   0
##                  3   0   3  21 143   3   0
##                  4   0   0  17 249  14   5
##                  5   0   0  10 131  23  13
##                  6   0   0   6  72  12  22
```

The percentage accuracy from this matrix is **42.69%.** This means that KNN gives better prediction than simple LDA. But still we will use bootstrap to see whether it improves our prediction or not.

3. BOOTSTRAP (LINEAR MODEL)

Here we used Occupation as X variable to predict Y (Annual Income).

```
## [1] 0.5368792
```

So, we got accuracy of **53.69%** by using Bootstrap. It means that we got better accuracy but still it is not a good prediction. Which proves that it is not a linear model. Then to improve on this we will use Nonlinear models like Trees, Neural network, SVM, etc.

Meanwhile, we also tried Best subset but it didn't fit our dataset and even proved worse than KNN and Bootstrap.

Here we can say that among the Linear models Bootstrap fits our dataset well in comparison to other models.

SECTION III

Alternate/Advanced Models

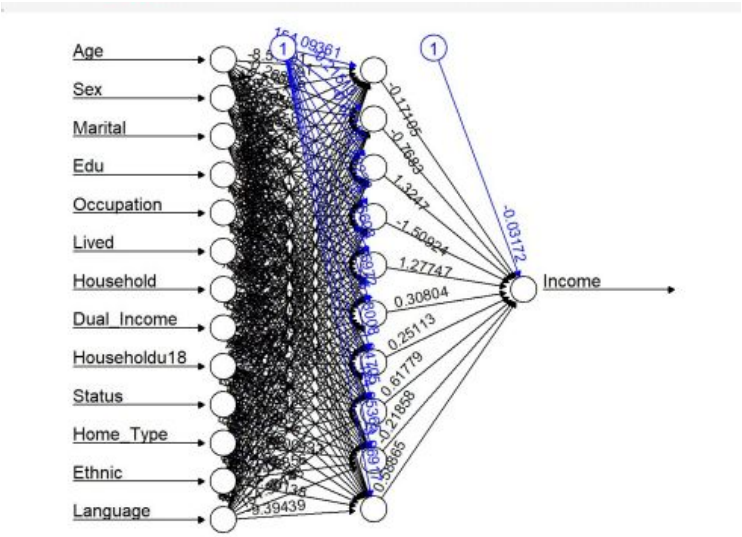
Since our dataset was nonlinear and the predictions from linear models were not accurate. So, here we will use advanced nonlinear models to improve our accuracy.

1. NEURAL NETWORK

First we will try neural network on our Dataset.
An an artificial neural network, composed of artificial neurons or nodes is used for solving artificial intelligence (AI) problems. It is a highly intelligent model which can catch patterns which may be difficult for humans to find.

Here in this project, we use the data to predict the income level using the other 13 attributes. We use the first 1500 instances for training and the rest for testing.

The function neuralnet() is used to construct the neural network with 1 hidden layer with 10 nodes.



```
sum(diag(tab2))/sum(tab2)
```

```
## [1] 0.1605283
```

On training the ANN with training data, the accuracy obtained for testing was **16.05%.** The accuracy was not as good as expected. This is because many of the factors were not relevant in the predicting the income level. So the ANN was finding unnecessary pattern in the dataset.

```
## [1] 0.2328333
```

So we will use SVM to see if we will get a better result with it.

2. SUPPORT VECTOR MACHINES (SVM)

In SVM also we divided our dataset for training and testing like before and predicted Income by using Education as attributes.
The confusion matrix is shown below for SVM:

```
##                   truth
## predict            1 2 3 6 7 8 9
##   1.17644058475668 1 0 0 0 0 0 0
##   1.27153900451675 0 1 0 0 0 0 0
##   2.80871367276248 0 0 1 0 0 0 0
##   5.61626628893989 0 0 0 0 0 0 0
##   5.80972194327446 0 0 0 1 0 0 0
##   6.34970524633368 0 0 0 0 1 1 0
##   6.90008389724613 0 0 0 0 0 0 1
```

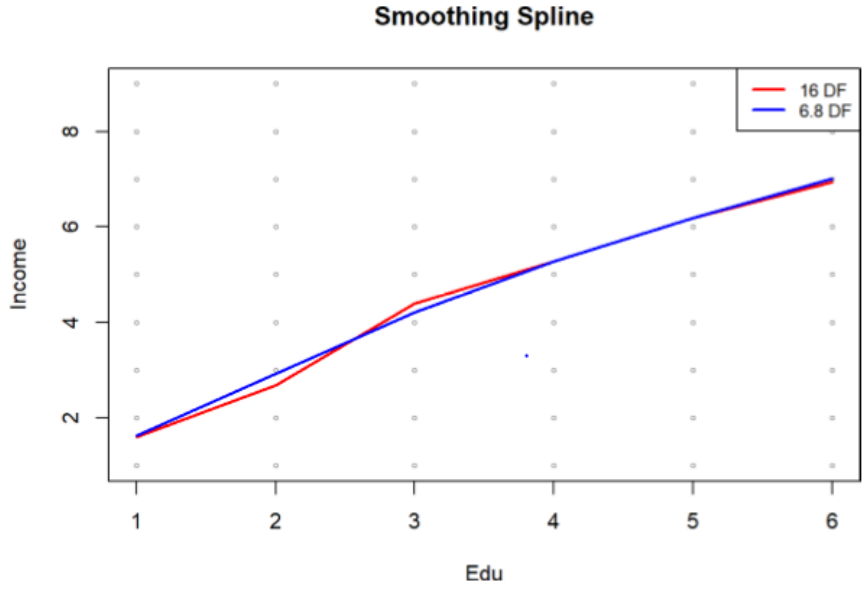
```
(8000+14000+18000+73000+80000)/(8000+14000+30000+40000+18000+73000+80000)
```

```
## [1] 0.7338403
```

So, here we got **73.38%** accuracy from this confusion matrix. Which is the best result till now. But still we will further proceed with trees, random forest, bagging etc. to see whether we can get more accurate results or not.

3. SPLINES

Here in splines we tried to predict Income from education and we got the following smoothing splines curve.




```
## [1] 2.997699
```

The M.S.E of splines is 2.997. So we can say that splines also fit our dataset well. And we will further try to reduce our MSE in next models.

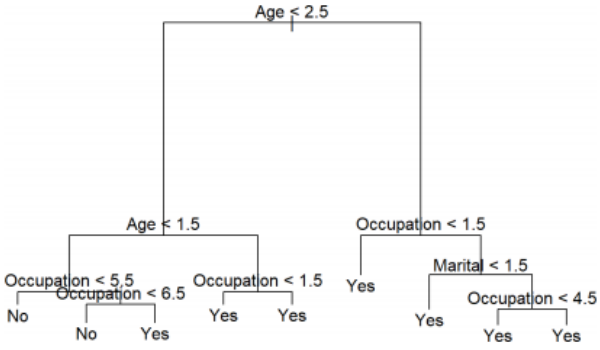
4. Trees

4.1. Decision Tree

A decision tree is a flowchart-like structure in which each internal node represents a test on a feature, each leaf node represents a class label and branches represent conjunctions of features that lead to those class labels.

Here we are going to classify the instances if they have an income status 2 (Below 15000 USD) and below or more than it. Trees is a really good method for classification because it only uses the relevant attributes.

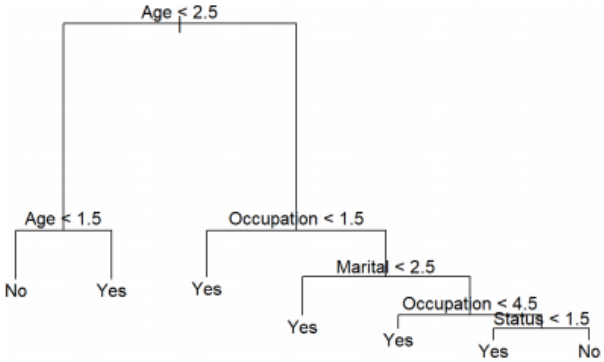
The tree here considers only Education, Occupation and Marital status as the main attributes which contributes to the determination. It has 9 nodes and an accuracy of **81.4%.** Now let us check if pruned tree provides a better output.




```
##           High.test
## tree.pred   No Yes
##       No  1076  680
##       Yes  319 3301
```

4.2. Pruned Decision Tree

The pruned tree has 7 nodes and **82.1%** accuracy. This is because the shaved part of the tree aren't important part for classifying.
So, we got better accuracy after pruning and this prediction is the most accurate one till now.
But still we will see regression trees, Bagging and Boosting to see whether we can get even better fit for our dataset.

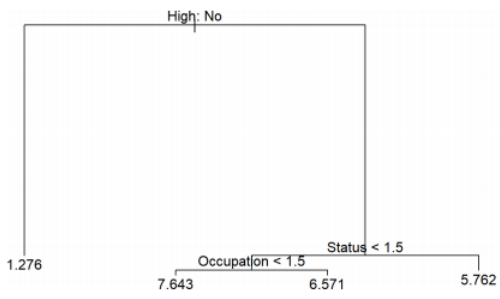


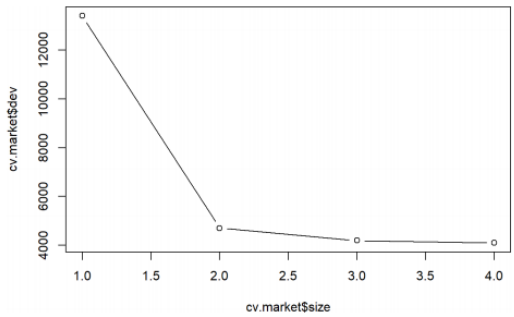

```
##          High.test
## tree.pred   No  Yes
##       No   582  149
##       Yes  813 3832
```

4.3. Regression Tree

A regression tree is built through a process known as binary recursive partitioning, which is an iterative process that splits the data into partitions or branches, and then continues splitting each partition into smaller groups as the method moves up each branch.

Here, the regression tree uses the high (attribute made in trees), status and occupation for deciding.
We use cross validation to find if it needs pruning. From the graph we see that the best tree is the most complex one so no need to prune it.





We use this information to predict values of testing data. And we get MSE of 2.2568. This shows that the error is small in this.

```
mean((yhat-market.test)^2)
```

```
## [1] 2.256684
```

4.4 Bagging and Random Forest

Bagging or (Bootstrap Aggregation) creates several subsets of data from training sample chosen randomly with replacement. Each collection of subset data is used to train their decision trees. Average of all the predictions from different trees are used which is more robust than a single decision tree classifier.

Random forest similar to Bagging except it uses only selected features of the dataset unlike the latter which uses all of them at the start.

Function randomforest() can be used for both the methods. The summary of the function says 5 of the variables can be considered for each split in the tree. The MSE of bagged regression tree is 1.856 and that of random forest is 1.939.

```
mean((yhat.bag-market.test)^2)
```

```
## [1] 1.856304
```

```
yhat.rf = predict(rf.market,newdata=market[-train,])
mean((yhat.rf-market.test)^2)
```

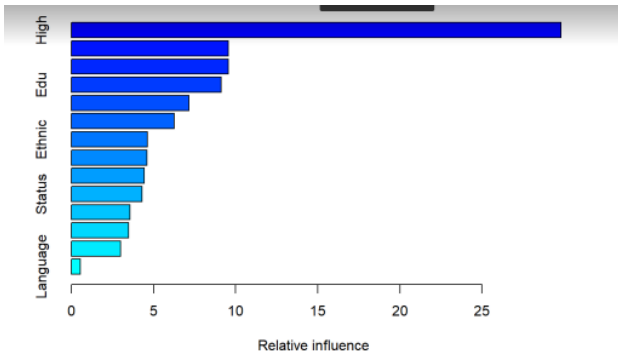
```
## [1] 1.939694
```

The importance of each feature can be viewed using function importance() and to plot it we use the function varImpPlot ( ). Therefore in this case the accuracy in Bagging is better than that of random forest and regression tree.

5. Boosting

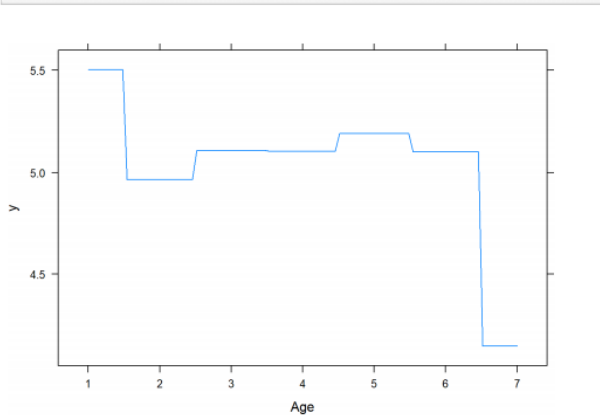
Boosting works in a similar to bagging except that the trees are grown sequentially i.e., each tree is grown using information from previously grown trees. Boosting does not involve bootstrap sampling instead each tree is fit on a modified version of the original data set.

The function gbm() produces boosted regression trees and summary plots and displays the relative influence of the attributes.




```
##                          var    rel.inf
## High                    High 29.7997106
## Age                      Age  9.5540772
## Occupation        Occupation  9.5524459
## Edu                      Edu  9.1215449
## Household          Household  7.1557530
## Marital              Marital  6.2572653
## Ethnic                Ethnic  4.6346858
## Lived                  Lived  4.6012460
## Home_Type        Home_Type    4.4309989
## Status                Status  4.2904512
## Sex                      Sex  3.5765837
## Dual_Income      Dual_Income  3.4734714
## Householdul8  Householdul8    2.9908868
## Language          Language    0.5608793
```

From the partial dependence plot of age we can say that with increase in age the income of a person goes down.



The MSE of boosting regression tree with shrinkage parameter of 0.001 is 2.322 and that with shrinkage parameter of 0.2 is 2.714.

```
mean((yhat.boost-market.test)^2)
```

```
## [1] 2.3222
```

```
mean((yhat.boost-market.test)^2)
```

```
## [1] 2.714409
```

Thus from this we can conclude that the error obtained in bagging regression tree is the least among singly pruned, boosting and random forest regression trees.

SECTION IV

Comparison

S.No.	Linear Models	Accuracy(%)
1.	Linear Discriminant Analysis (LDA)	16.79
2.	K Nearest Neighbour (KNN)	42.69
3.	Bootstrap (linear)	53.69
	Non Linear Models	**Accuracy(%)**
1.	Neural Network	16.05
2.	Support Vector Machine(SVM)	73.38
3.	Tree	81.4
4.	Pruned Tree	82.1

CONCLUSION

After fitting our dataset with a wide varieties of linear and not linear models we can conclude that :
Among the Linear models Bootstrap fitted our dataset best with an accuracy of 53.69%. With this it revealed that our dataset is not a linear model so then we tried various nonlinear models.
Among the non linear models in tree we got an accuracy of **81.4%** with 9 nodes. But then we pruned the tree , now with only 7 nodes and our accuracy got improved by **82.1%.**
So, finally we can conclude that **Tree (pruned)** with is the best model for our dataset Market Basket Analysis.