Analysis Of Black Friday

Ajinkya Ambike

April 16, 2019

Contents

Summary :

In this project I am trying to do some analytics on the Black friday dataset taken from kaggle website.The dataset has 537577 observations and 12 variables.My aim is to do some analytics using Product Categories and showcase some probabilities of purchasing a product from a particular category.

Objective :

To predict the probalities of users who purchased items from among which categories using Multinomial logistic regression.Also i am using aproiri algorithm to show the market basket analysis of the Product categories.I have also show some of the EDA to better understand the data and derive some conclusions from it and provide some suggestions for the seller.

Background:

For centuries, the adjective “black” has been applied to days upon which calamities occurred. Many events have been described as “Black Friday”, although the most significant such event in American History was the Panic of 1869, which occurred when financiers Jay Gould and James Fisk took advantage of their connections with the Grant Administration in an attempt to corner the gold market. When President Grant learned of this manipulation, he ordered the Treasury to release a large supply of gold, which halted the run and caused prices to drop by eighteen percent. Fortunes were made and lost in a single day, and the president’s own brother-in-law, Abel Corbin, was ruined. The earliest known use of “Black Friday” to refer to the day after Thanksgiving occurs in the journal, Factory Management and Maintenance, for November 1951, and again in 1952. Here it referred to the practice of workers calling in sick on the day after Thanksgiving, in order to have a four-day weekend. Black Friday is a shopping day for a combination of reasons. As the first day after the last major holiday before Christmas, it marks the unofficial beginning of the Christmas shopping season. Additionally, many employers give their employees the day off as part of the Thanksgiving holiday weekend. In order to take advantage of this, virtually all retailers in the country, big and small, offer various sales including limited amounts of doorbuster/doorcrasher/doorsmasher items to entice traffic. The early 2010s have seen retailers extend beyond normal hours in order to maintain an edge or to simply keep up with the competition. Such hours may include opening as early as 12:00 am or remaining open overnight on Thanksgiving Day and beginning sale prices at midnight.

Data Description:

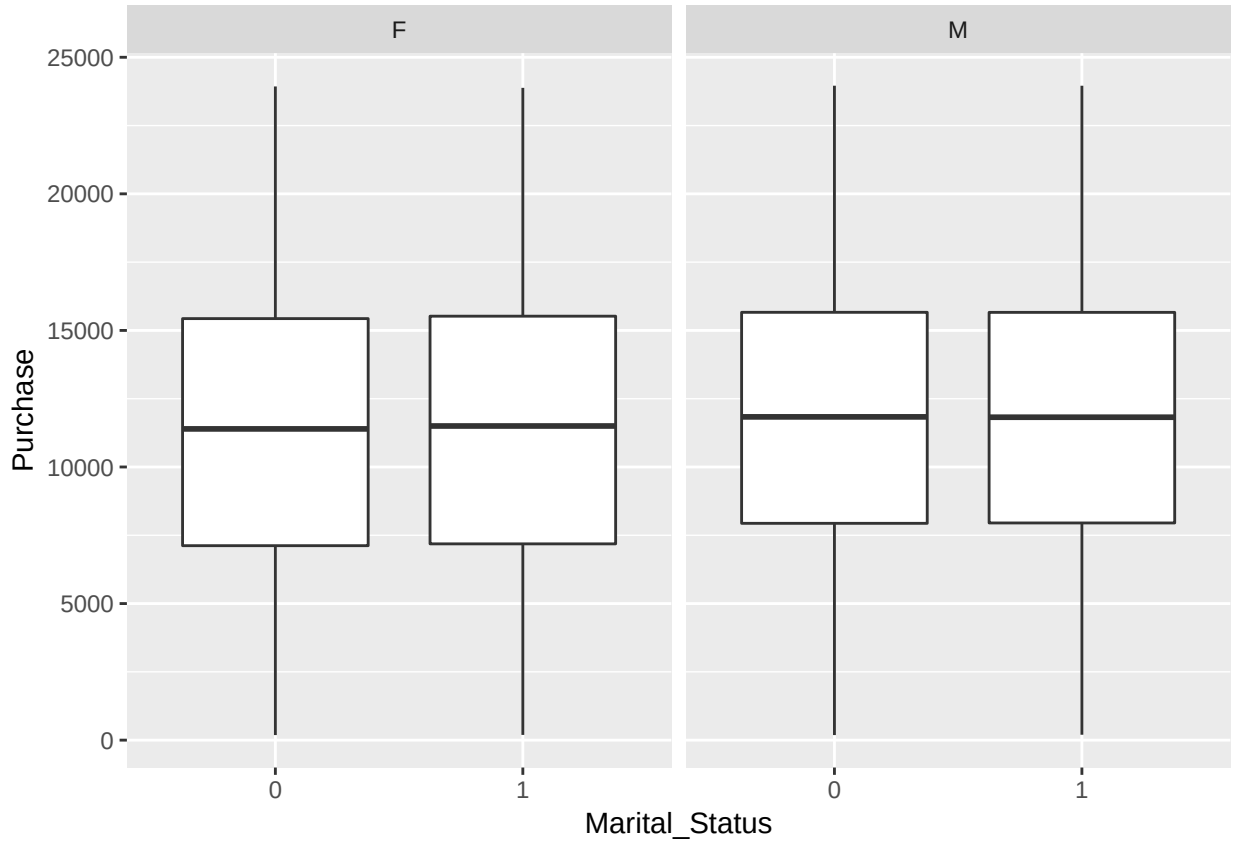
We have the following columns in the description. User_ID : Integer variable with the id of each user. Product_ID : Factor variable as the idea of each product. Gender : Factor variable which contains the gender of all the users. Age : Factor variable which contains the age interval of all the users. Occupation : Integer Variable which contains the integer value of all the occupation. City_Category : Factor variable which contains the Factor levels of 3 city categories that is A,B,C. Stay_In_Current_City_Years: Categorical variable which contains the number of years a person lives in the city. Marital_Status : Categorical variable which contains the information of all the users whether he is married or not. Product_Category_1 : Categorical variable Product category 1 in which we have the various product categories. Product_Category_2 : Categorical variable Product category 2 in which we have the various product categories. Product_Category_3 : Categorical variable Product category 3 in which we have the various product categories. Purchase : Integer variable which gives total Purchase by each user.

Taking input in R:

I have installed few libraries here in this chunk.I have also imported the data file using read.csv.After viewing the data I realized that there are multiple null values in this datafile.So using na.omit I removed all the null values and finally we are left with 164278 observations.If we find the structure of the dataset we find that there are 3 product categories which are factor variables along with Gender,Age and City_Category are also factor variables.Only occupation,Stay_in_current_years and purchase are integer variables.

- Purchase by males and Females.

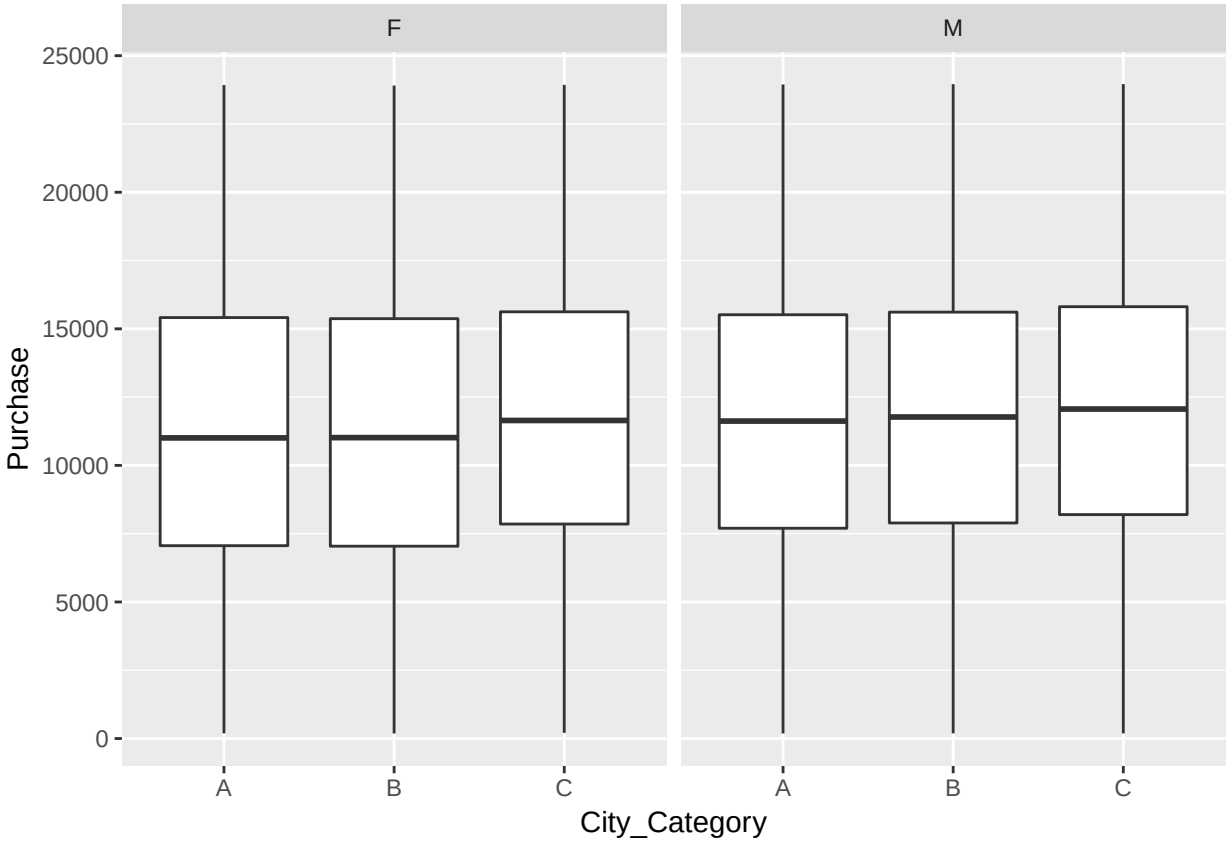
```
ggplot(blackwithoutna, aes(x=Marital_Status, y=Purchase))+ geom_boxplot()+facet_grid(~Gender)
```



We find that Males who are married has higher sales than married females.The purchase for the males is around 13000 for both married and unmarried males.

- City Category Versus Purchase.

```
ggplot(blackwithoutna, aes(x=City_Category, y=Purchase))+ geom_boxplot()+facet_grid(~Gender)
```



Males and females in city category C has the highest sales compared to other city categories.The sale is more than 11000.

Exploratory Data Analysis.

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypothesis and to check assumptions with the help of summary statistics and graphical representations. It is the first step in your data analysis process. Here, we make sense of the data you have and then figure out what questions we want to ask and how to frame them, as well as how best to manipulate our available data sources to get the answers we need.

- City Category compared with different variables.

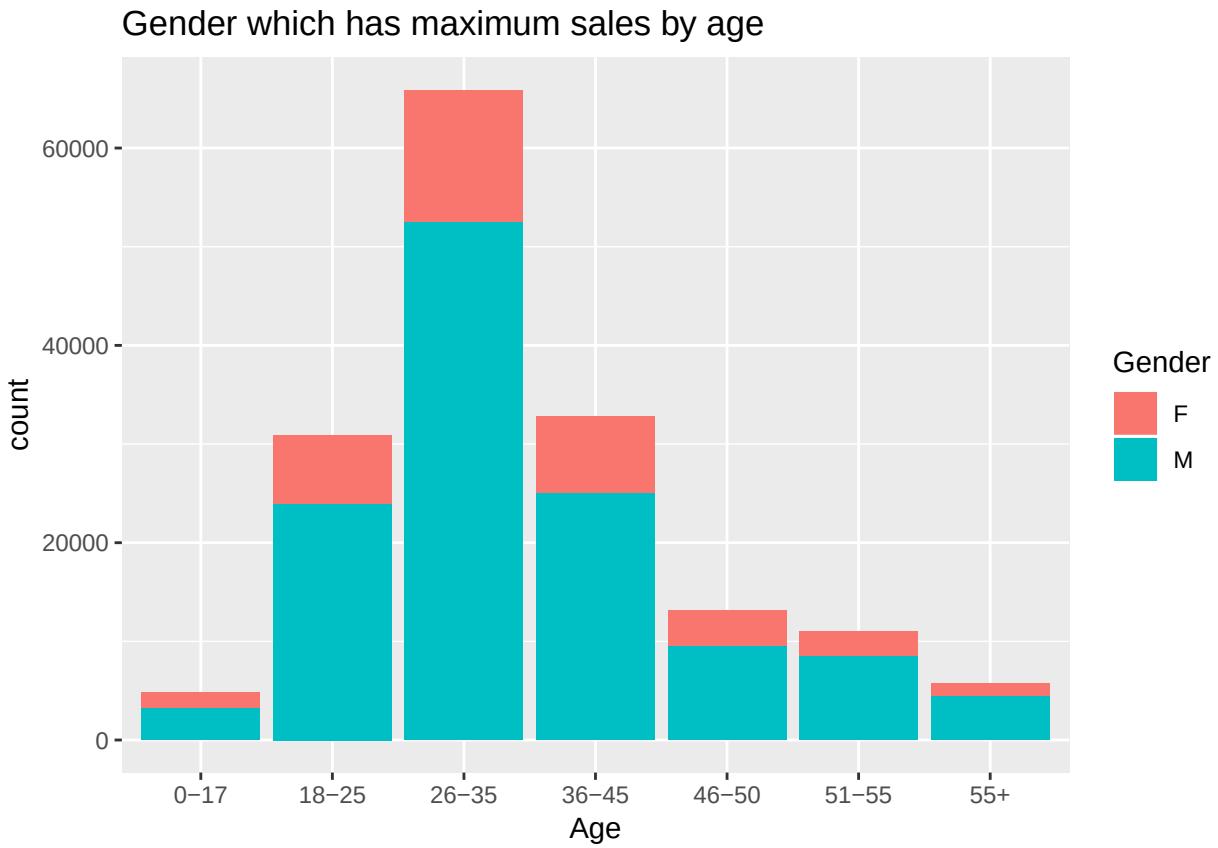
```
customer_count_per_city <- blackwithoutna %>%
  group_by(User_ID) %>%
  filter(row_number(User_ID) == 1) %>%
  group_by(City_Category, Age) %>%
  summarise(customers_count = n()) %>%
  arrange(City_Category, Age)

citySales2 <- blackwithoutna %>%
  group_by(City_Category)%>%
  summarise(sold_by_city = n(),
            percentage_by_city = round(n()/nrow(blackwithoutna)*100,0))
```

```r
ggarrange(
  customer_count_per_city %>%
    ggplot(aes(x = City_Category, y = customers_count, fill = Age)) +
    geom_bar(stat = "identity", position = position_dodge()) +
    ggtitle("Number of customers across cities"),

  blackwithoutna %>%
    group_by(User_ID, City_Category, Age) %>%
    summarise(bought_items = n()) %>%
    group_by(City_Category, Age) %>%
    summarise(items_bought_per_city = sum(bought_items)) %>%
    arrange(City_Category, Age) %>%
    cbind(customers_count = customer_count_per_city$customers_count) %>%
    mutate(average_items_per_person_in_city = items_bought_per_city/customers_count) %>%
    ggplot(aes(x = City_Category, y = average_items_per_person_in_city, fill = Age)) +
    geom_bar(stat = "identity", position = position_dodge()) +
    ggtitle("Average number of bought items across cities"),

  blackwithoutna %>%
    group_by(City_Category, Age) %>%
    summarise(sales = n()) %>%
    left_join(citySales2, by = "City_Category") %>%
    mutate(percentage = round(sales/sold_by_city*100, 0),
           percentage = ifelse(percentage < 3, "", as.character(paste0(percentage, "%")))) %>%
    #select(-percentage_by_city) %>%
    ggplot(aes(x = City_Category, y = sales, fill = Age)) +
    geom_bar(stat="identity", position = position_dodge()) +
    ggtitle("Number of sales across age groups in the three cities"),

  blackwithoutna %>%
    group_by(City_Category) %>%
    summarise(sales = n(),
              percentage_by_city = paste0(round(n()/nrow(blackwithoutna)*100,0), "%")) %>%
    ggplot(aes(x = City_Category, y = sales)) +
    geom_bar(stat="identity", fill = "royalblue") +
    geom_text(aes(label = percentage_by_city),
              vjust=1.6,
              color="white",
              size=8) +
    ggtitle("Number of sales in the three cities"),
  nrow = 2, ncol = 2, common.legend = TRUE)
```

In the above graph we have shown four different quadrants.In 1st quadrant on top right,I have shown the items brought by different age groups accross city categories.In city categoryA,B and C we find that age group between 26-35 has the highest items bought.In 2nd quadrant top right we find that customers with 26-35 has highest customers count for all the 3 city categories.In 3rd quadrant we find the maximum sales for the age groups 26-35 for all the city categories.And finally in 4th Quandrant we have maximum sales by city category B.

- Now we will see the age distribution by males and females.

```
g <- ggplot(blackwithoutna, aes(Age))
g + geom_bar(aes(fill = Gender)) + ggtitle("Gender which has maximum sales by age")
```



Again here in the above graph we find that the people between age groups 26-35 has maximum sales.And we have more males than females who purchased in that age group.

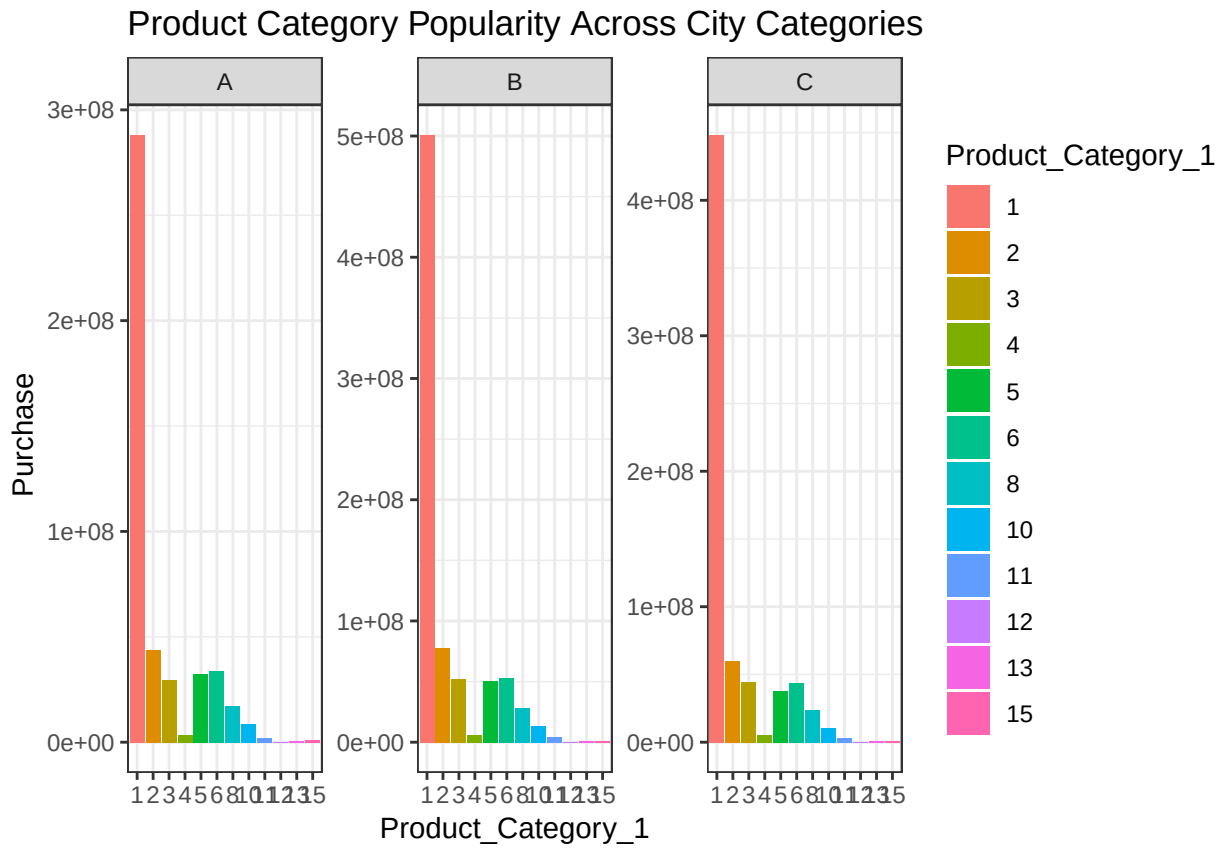
- Lets See Gender across city categories.

```
ggplot(blackwithoutna, aes(x=Gender, y=Purchase,fill=Gender))+geom_boxplot()+stat_summary(fun.y = "mean
  ggtitle("Purchase Distribution by Gender across City Category")+facet_wrap(~City_Category ,ncol=3)
```

In the above boxplot,we find males from city category C has maximum purchase which again approximately 13000.

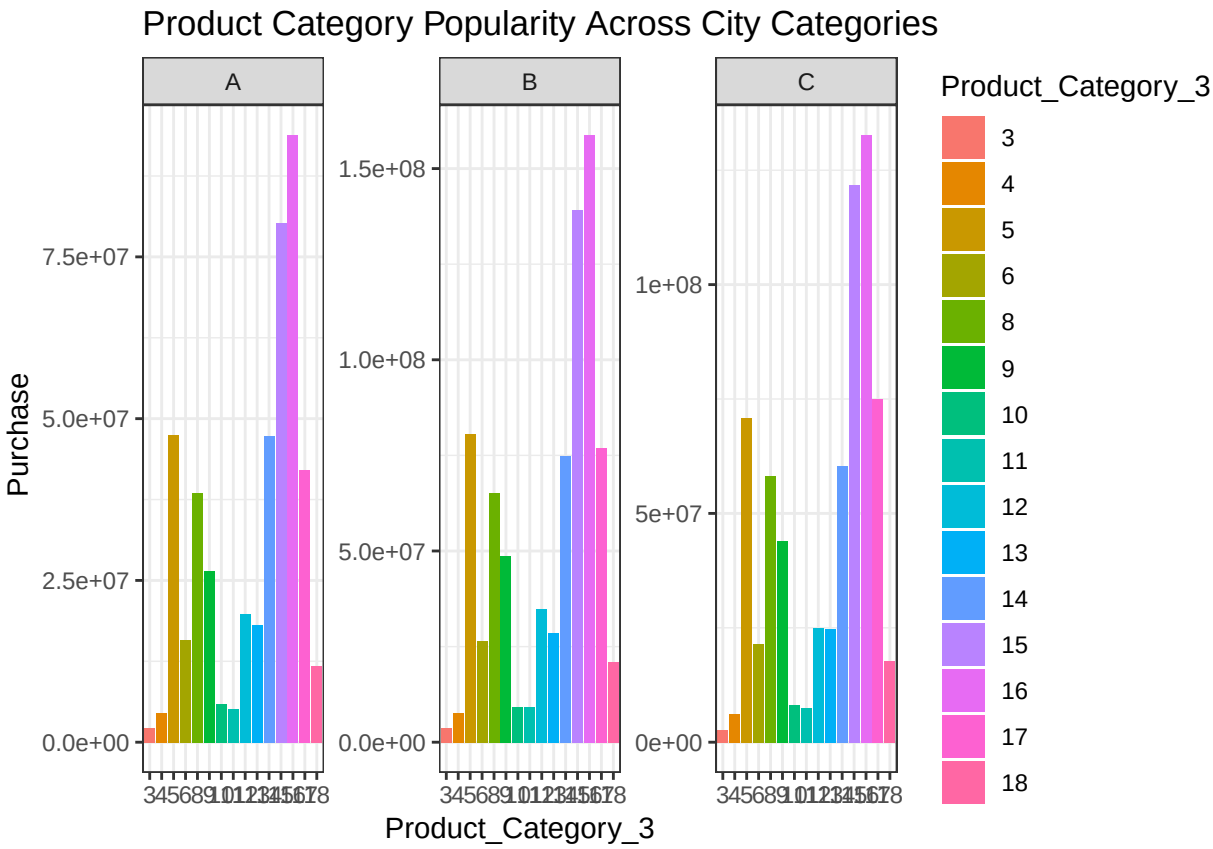
- Now lets see the product categories which has maximum purchase.

```
plot <- ggplot(data = blackwithoutna, aes(x=Product_Category_1, y=Purchase, fill=Product_Category_1)) +
  ggtitle("Product Category Popularity Across City Categories") +
  theme_bw()
     plot + geom_bar(stat="identity") + facet_wrap(~City_Category,scales="free_y")
```




```
plot2 <- ggplot(data = blackwithoutna, aes(x=Product_Category_2, y=Purchase, fill=Product_Category_2)) +
  ggtitle("Product Category Popularity Across City Categories") +
  theme_bw()
     plot2 + geom_bar(stat="identity") + facet_wrap(~City_Category,scales="free_y")
```


```
plot3 <- ggplot(data = blackwithoutna, aes(x=Product_Category_3, y=Purchase, fill=Product_Category_3)) +
  ggtitle("Product Category Popularity Across City Categories") +
  theme_bw()
     plot3 + geom_bar(stat="identity") + facet_wrap(~City_Category,scales="free_y")
```



In the above 3 plots,we find the purchase by each product categories in each city categories.In city category A,B and C we find for product category 1 that category 1 has the highest purchase over all the 3 city categories.

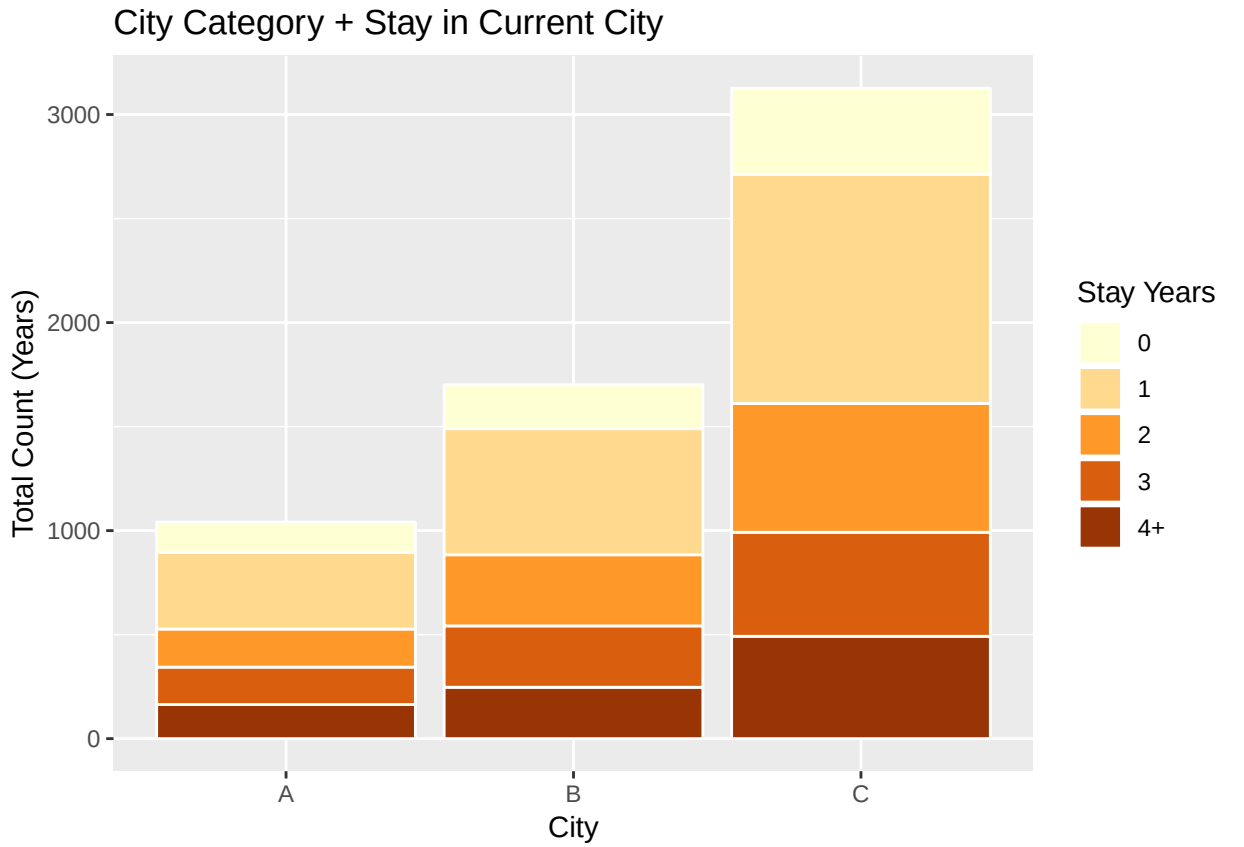
In product category 2 ,we find that category 2 has maximum purchase in all the 3 city categories followed by category 8.

In product category 3 ,we can see that the category number 17 has maximum purchase in all the 3 city categories followed by product category number 5.

- Now let us execute the stay in current city of each user by city category.

```
customers_stay <- blackwithoutna %>%
                    dplyr::select(User_ID, City_Category, Stay_In_Current_City_Years) %>%
                    group_by(User_ID) %>%
                    distinct()


residence <- customers_stay %>%
                group_by(City_Category) %>%
                tally()


customers_stay_vis <- ggplot(data = customers_stay, aes(x = Stay_In_Current_City_Years, y = ..count.., f
                              geom_bar(stat = 'count') +
                              scale_fill_brewer(palette = 15) +
                              labs(title = 'Customers Stay in Current City', y = 'Count', x = 'Stay in C
```

```
stay_cities <- customers_stay %>%
                group_by(City_Category, Stay_In_Current_City_Years) %>%
                tally() %>%
                mutate(Percentage = (n/sum(n))*100)

ggplot(data = stay_cities, aes(x = City_Category, y = n, fill = Stay_In_Current_City_Years)) +
    geom_bar(stat = "identity", color = 'white') +
    scale_fill_brewer(palette = 17) +
    labs(title = "City Category + Stay in Current City",
            y = "Total Count (Years)",
            x = "City",
            fill = "Stay Years")
```

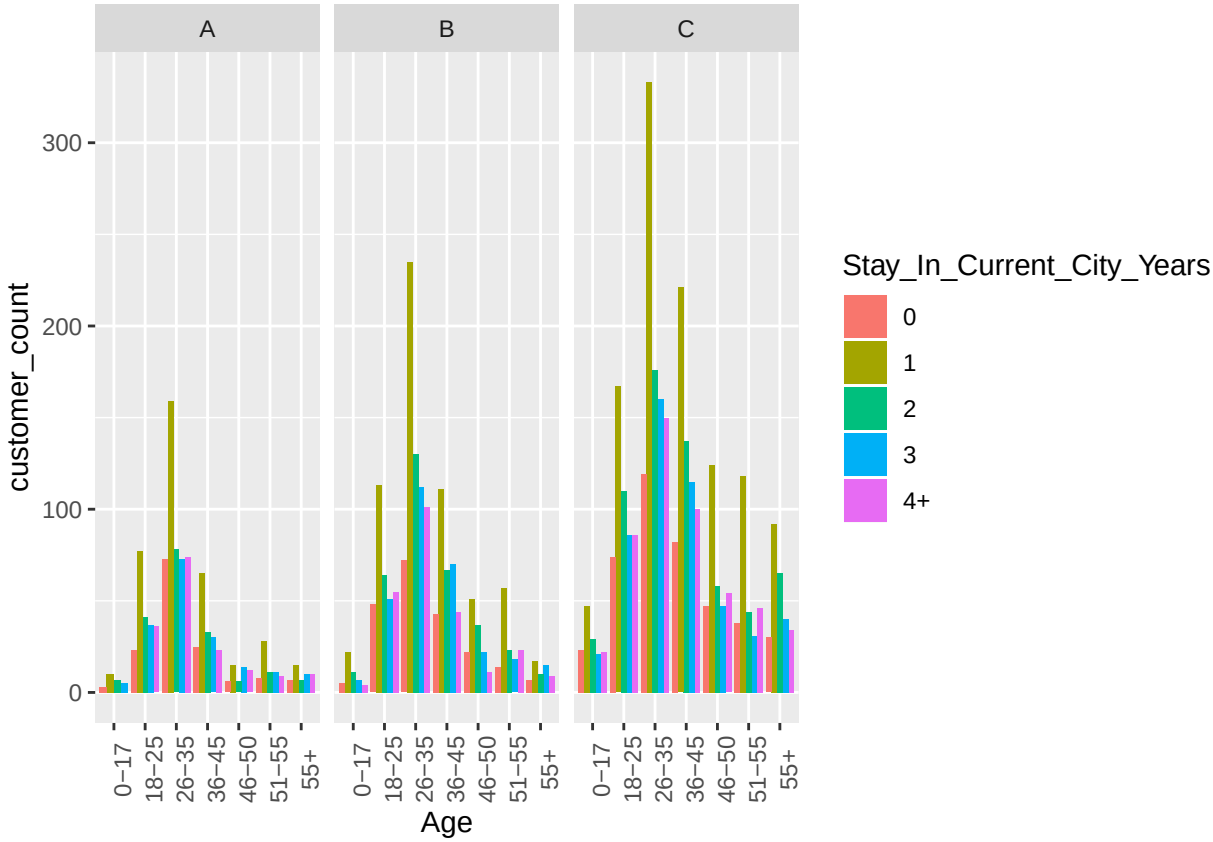


In this city category we find that maximum users stay just for 1 year in city A,B and C and it is followed by the duration of stay for 2 years.

- Stay in The city by each age group

```
blackwithoutna %>%
  group_by(User_ID) %>%
  filter(row_number(User_ID) == 1) %>%
  group_by(Age, Stay_In_Current_City_Years, City_Category) %>%
  summarise(customer_count = n()) %>%
  ggplot(aes(Age, customer_count, fill = Stay_In_Current_City_Years)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  facet_wrap(City_Category~.) +
```

```
  theme(axis.text.x.bottom = element_text(angle = 90))
```



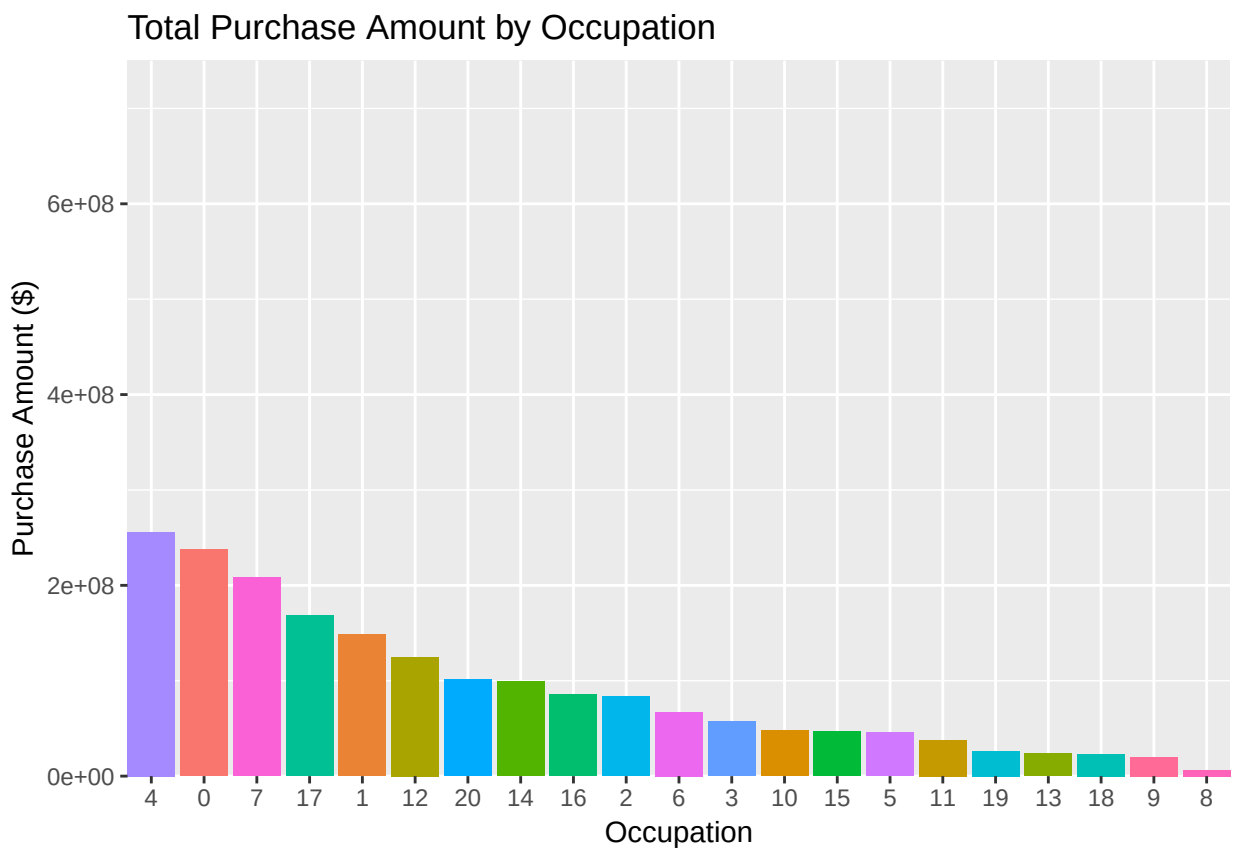
We can see that within each age group for all the 3 city categories maximum number of people stay in a particular city is 1 year.Majority of the people prefer to leave the city and transfer to new city after an year.

- Now let us see the purchase amount by the occupation.

```
customers_total_purchase_amount <- blackwithoutna %>%
                                    group_by(User_ID) %>%
                                    summarise(Purchase_Amount = sum(Purchase))


customers_total_purchase_amount <- arrange(customers_total_purchase_amount, desc((Purchase_Amount)))


customers_Occupation <- blackwithoutna %>% dplyr::select(User_ID,Occupation) %>%
                            group_by(User_ID) %>%
                            distinct() %>%
                            left_join(customers_total_purchase_amount, Occupation, by = 'User_ID')

totalPurchases_Occupation <- customers_Occupation %>%
                                group_by(Occupation) %>%
                                summarise(Purchase_Amount = sum(Purchase_Amount)) %>%
                                arrange(desc(Purchase_Amount))

totalPurchases_Occupation$Occupation = as.character(totalPurchases_Occupation$Occupation)
typeof(totalPurchases_Occupation$Occupation)
```

```
## [1] "character"
```

```
occupation = ggplot(data = totalPurchases_Occupation) +
                  geom_bar(mapping = aes(x = reorder(Occupation, -Purchase_Amount), y = Purchase_Amount
                  scale_x_discrete(name="Occupation", breaks = seq(0,20, by = 1), expand = c(0,0)) +
                  scale_y_continuous(name="Purchase Amount ($)", expand = c(0,0), limits = c(0, 7500000
                  labs(title = 'Total Purchase Amount by Occupation') +
                  theme(legend.position="none")

print(occupation)
```



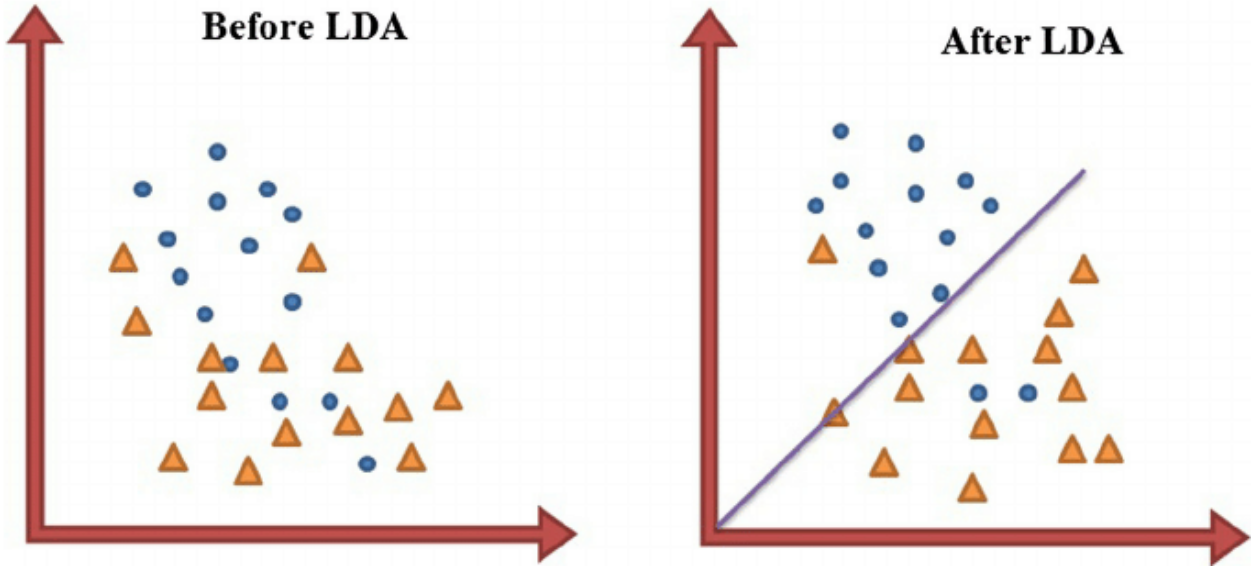
We can see from the barplot above that the occupation with id 4 has maximum purchase of the products followed by the occupation with id 0.

Linear Discriminant Analysis(LDA)

As far as logistic regression is concerned,when classes are separated,parameters estimate from logistic regression tend to be unstable.Also it is not the best model if we have more than 2 classes.And therefore we use linear Discriminant Analysis.We can use LDA if we have more than 2 classes.Logistic regression does not create plots thats where the LDA comes to our rescue.It generates helpful plots,especially territorial maps for our data analysis.LDA is closely related to PCA and factor analysis,in both they look for linear combination of variables which best explains the data.LDA explicitly attemts to model the difference between the classes of data which is not done by PCA.

How does Linear Discriminant Analysis work? The goal of Linear Discriminant Analysis is to project the features in higher dimension space onto a lower dimensional space. This can be achieved in three steps: The first step is to calculate the separability between different classes (i.e. the distance between the mean of different classes) also called as between-class variance Second Step is to calculate the distance

between the mean and sample of each class, which is called the within class variance. The third step is to construct the lower dimensional space which maximizes the between class variance and minimizes the within class variance. Let P be the lower dimensional space projection, which is called Fisher's criterion.



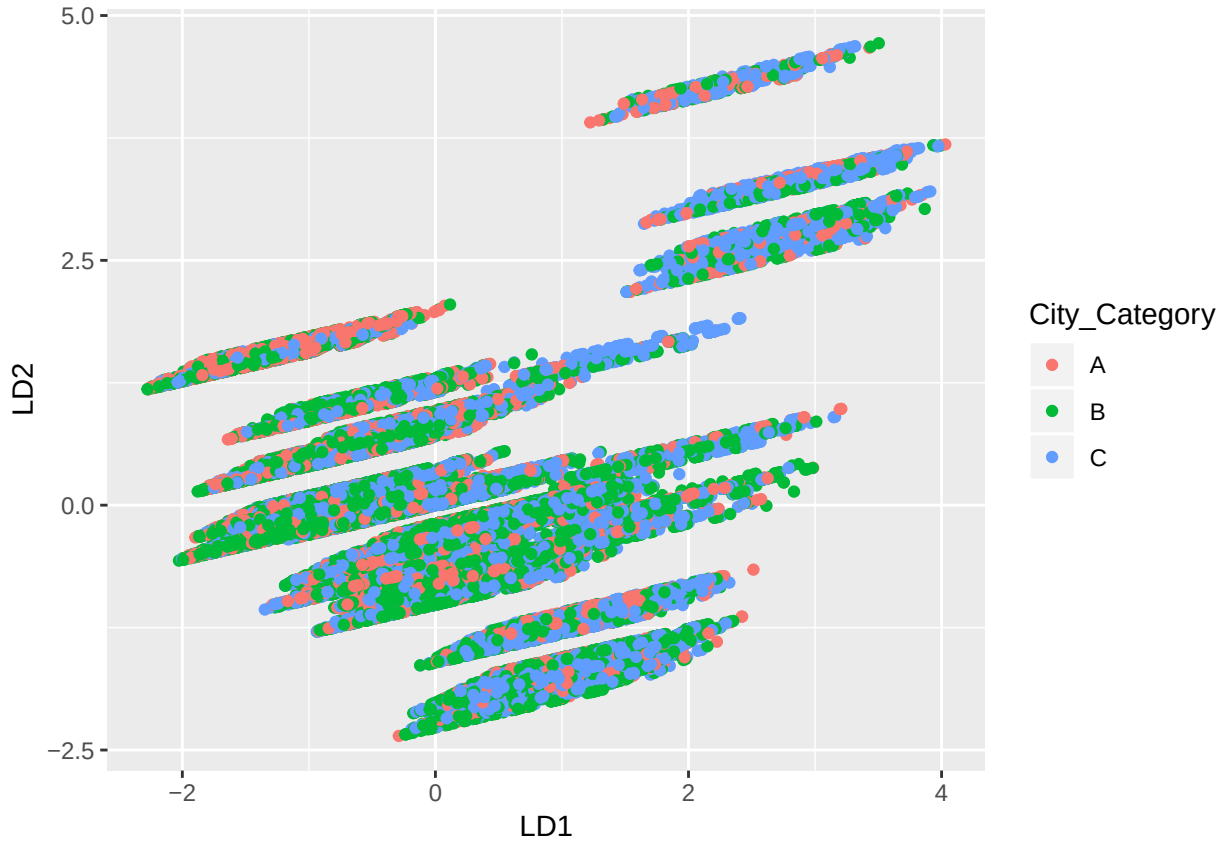
In the code below we perform LDA on the City_Category variable.

```
training.index <- createDataPartition(blackwithoutna$City_Category , p = 0.8 , list = FALSE)
train.df <- blackwithoutna[training.index, ]
valid.df <- blackwithoutna[-training.index, ]

norm.values  <- preProcess(train.df, method = c("center", "scale"))
train.norm <- predict(norm.values, train.df)
valid.norm <- predict(norm.values, valid.df)

lda3 <- lda(City_Category ~  Purchase + Occupation + Age + Stay_In_Current_City_Years , data = train.no
lda3
```

```
## Call:
## lda(City_Category ~ Purchase + Occupation + Age + Stay_In_Current_City_Years,
##     data = train.norm)
##
## Prior probabilities of groups:
##         A         B         C
## 0.2486551 0.4150567 0.3362882
##
## Group means:
##      Purchase  Occupation  Age18-25  Age26-35  Age36-45   Age46-50
## A -0.08885478 -0.04395412 0.1940084 0.4912941 0.1799321 0.04997093
## B -0.03426692 -0.01989830 0.1907128 0.4056611 0.2026289 0.08398108
## C  0.10799342  0.05705922 0.1785003 0.3318400 0.2105168 0.09607204
##     Age51-55     Age55+ Stay_In_Current_City_Years1
## A 0.04293277 0.02371554                  0.3316809
## B 0.07197331 0.01974408                  0.3591516
## C 0.07821975 0.06238121                  0.3492624
##   Stay_In_Current_City_Years2 Stay_In_Current_City_Years3
## A                   0.1893571                   0.1713333
## B                   0.1855430                   0.1854697
```

```
## C                   0.1940221                   0.1675491
##   Stay_In_Current_City_Years4+
## A                    0.1478931
## B                    0.1477048
## C                    0.1600371
##
## Coefficients of linear discriminants:
##                                      LD1         LD2
## Purchase                        0.4229037  0.12454949
## Occupation                      0.1509966  0.09665509
## Age18-25                       -2.0160841 -0.48622368
## Age26-35                       -2.6780605  0.01865026
## Age36-45                       -1.6035076 -0.71137280
## Age46-50                       -0.7926432 -1.66725437
## Age51-55                       -0.9730520 -1.78097496
## Age55+                          0.8023051  2.73779568
## Stay_In_Current_City_Years1     0.4212209 -1.50091871
## Stay_In_Current_City_Years2     0.4390703 -1.03117541
## Stay_In_Current_City_Years3     0.2761779 -1.74178160
## Stay_In_Current_City_Years4+    0.5911421 -0.98180501
##
## Proportion of trace:
##    LD1    LD2
## 0.8474 0.1526
```

```r
pred <- predict(lda3,train.norm)

lda3.plot <- cbind(train.norm, predict(lda3)$x)
ggplot(lda3.plot, aes(LD1,LD2)) +
  geom_point(aes(color = City_Category))
```




```
pred3 <- predict(lda3, valid.norm)
names(pred3)
```

```
## [1] "class"     "posterior" "x"
```

```
# check model accuracy
table(pred3$class, valid.norm$City_Category)
```

```
##
##         A     B     C
##   A   227   172   149
##   B  7185 11849  8681
##   C   757  1616  2219
```

```
mean(pred3$class == valid.norm$City_Category)
```

```
## [1] 0.4350936
```

First of all,we generally dont run any algorithm on the entire dataset.Therefore we divide the dataset into two parts:training and validation.We do this by using createDataPartition funtion in caret package.Then we normalize the dataset using preprocess function. When I execute LDA we get information about the prior probabilities of each category.For example we find that for city category A the prior probability is 0.2486 which means that 24.86% of the data can be categorized in city category A.41.5% of the data can be categorized in city category B and Similarly last 33.6% of the data can be categorized to be in city category C. We get the information of LD1 and LD2.These coefficients of linear discriminants can help us to predict the classification boundary.The boundary values can be calculated by multiplying the LD values with the variables.For example we can multiply 0.4253 with Purchase ,0.1670 with Occupation and similarly with all the other variables we find the boundary values and we can classify it based on city categories. We also get

to know the proortion of trace.This information help us to know that 87.8% of the data is stored in LD1 while the remaining 12.2% of the data is stored in LD2. We also determine the accuracy of the model.The model is only 43.13% accurate and therefore we can see that this model has not that good accuracy and which shows that we cannot make much accurate prediction using this model.

Multinomial Logistic Regression.

First let us see what is Multinomial Logistic regression. Multinomial logistic regression is a classification method that generalises logistic regressions to multiclass problems that is with more than two discrete possible outcomes.Multinomial logistic regression is chosen when the dependent variable is nominal and for which there are more than two categories.For example,the major a student will choose,blood type of a person,the name of the countries etc.

```
blackwithoutna$Product_Category_1factor <- as.factor(blackwithoutna$Product_Category_1)
blackwithoutna$out <- relevel(blackwithoutna$Product_Category_1factor , ref = "1")
model_multinom <- multinom(out ~ Occupation+Marital_Status+Purchase,data = blackwithoutna)
```

```
## # weights:  60 (44 variable)
## initial  value 408215.494615
## iter  10 value 249411.401581
## iter  20 value 236094.172976
## iter  30 value 229997.693102
## iter  40 value 213938.241829
## iter  50 value 202480.936524
## iter  60 value 198029.101926
## iter  70 value 196150.529366
## iter  80 value 195598.720065
## iter  90 value 195440.720621
## final  value 195440.522682
## converged
```

```
summary(model_multinom)
```

```
## Call:
## multinom(formula = out ~ Occupation + Marital_Status + Purchase,
##     data = blackwithoutna)
##
## Coefficients:
##    (Intercept)    Occupation Marital_Status1      Purchase
## 2    0.2170983 -5.525046e-05    0.0018720716 -1.552008e-04
## 3    0.7188146 -8.155168e-03    0.0006325785 -2.212207e-04
## 4    8.5771365  3.740305e-03    0.2901041872 -2.156384e-03
## 5    3.9890798 -2.620224e-03    0.1663532344 -5.879384e-04
## 6   -4.2780695 -9.538814e-04    0.0565858865  1.274002e-04
## 8    2.2009346 -3.308158e-04    0.2088109370 -4.421291e-04
## 10 -11.2872417  5.050610e-04    0.1201159196  4.383065e-04
## 11   4.1620036 -2.846391e-03    0.1211349243 -1.056655e-03
## 12   8.9067194 -5.696538e-04    0.6308170138 -4.391671e-03
## 13  15.6370424 -2.633575e-02    0.4300739303 -9.707201e-03
## 15  -8.0200530  1.566417e-02    0.2190208457  9.496202e-05
##
## Std. Errors:
##     (Intercept)   Occupation Marital_Status1     Purchase
## 2  2.142034e-05 0.0012505198    1.018920e-05 1.069771e-06
## 3  2.624725e-05 0.0014288570    1.197839e-05 1.291847e-06
## 4  4.083775e-05 0.0029347331    1.951683e-05 9.106124e-06
```

```
## 5  2.428775e-05 0.0014108252    1.018811e-05 1.802333e-06
## 6  1.687177e-05 0.0017260959    1.101459e-05 1.136397e-06
## 8  3.290104e-05 0.0016892220    1.500520e-05 2.015570e-06
## 10 1.345202e-05 0.0038158994    1.850348e-05 2.155785e-06
## 11 7.042437e-05 0.0034817306    2.880464e-05 7.388429e-06
## 12 5.191897e-06 0.0001666816    2.143874e-06 6.007512e-05
## 13 4.187276e-06 0.0003423863    1.592895e-06 7.929111e-05
## 15 1.300574e-07 0.0000141440    9.252884e-08 5.254616e-06
##
## Residual Deviance: 390881
## AIC: 390969
```

```
predictmodel <- predict(model_multinom,blackwithoutna,type="prob")
head(predictmodel, 10)
```

```
##            1          2          3            4            5           6
## 2  0.7648685 0.08976524 0.05011962 2.454401e-11 0.0052905272 0.072862810
## 7  0.7520676 0.04742866 0.02078916 5.541945e-15 0.0005843129 0.126806190
## 14 0.7675239 0.08391587 0.04184999 1.253747e-11 0.0046466348 0.081315141
## 15 0.1824917 0.09836017 0.10588970 9.211651e-03 0.4075932509 0.004978897
## 17 0.7141845 0.11693242 0.07583224 2.329967e-09 0.0174808036 0.051815857
## 19 0.6482670 0.12950390 0.09730452 4.215427e-08 0.0403058939 0.042676931
## 20 0.7447181 0.04413287 0.01809379 2.365121e-15 0.0004516591 0.131486306
## 25 0.6615118 0.12925090 0.08802440 3.321385e-08 0.0368249999 0.043864248
## 29 0.1864773 0.10026142 0.10107729 9.430633e-03 0.4047499385 0.005057328
## 30 0.7672105 0.07547739 0.04195731 2.653320e-12 0.0032596030 0.090339200
##              8           10           11           12           13
## 2  0.008305913 7.536603e-03 5.052271e-06 5.736120e-26 3.022591e-58
## 7  0.001707013 4.848697e-02 8.128149e-08 2.334896e-33 5.858259e-75
## 14 0.008334021 1.050883e-02 3.402906e-06 1.395485e-26 3.925989e-60
## 15 0.152458349 2.426357e-05 3.887683e-02 7.409888e-08 1.891916e-17
## 17 0.020027429 2.747074e-03 4.563374e-05 6.610003e-22 3.198135e-49
## 19 0.039326526 1.607112e-03 1.824538e-04 2.955334e-19 1.209252e-43
## 20 0.001414620 5.733345e-02 5.205290e-08 3.997251e-34 1.057349e-76
## 25 0.037600956 1.752677e-03 1.557943e-04 1.627636e-19 2.396042e-44
## 29 0.154485760 2.503611e-05 3.830174e-02 7.119027e-08 1.380319e-17
## 30 0.006187186 1.405992e-02 1.737384e-06 6.901572e-28 8.219400e-63
##              15
## 2  0.0012457221
## 7  0.0021300351
## 14 0.0019021808
## 15 0.0001151333
## 17 0.0009340690
## 19 0.0008255933
## 20 0.0023691441
## 25 0.0010141773
## 29 0.0001335189
## 30 0.0015071376
```

```
cm <- table(predict(model_multinom),blackwithoutna$Product_Category_1factor)
print(cm)
```

```
##
##          1     2     3     4     5     6     8    10    11    12    13
##   1  84453 13353 10944     0  9563  8056  5850  1686   147     0     0
```

```
##   2      0     0     0     0     0     0     0     0     0     0     0
##   3      0     0     0     0     0     0     0     0     0     0     0
##   4      0     0   284  3694   865     0   264     0   421   144   145
##   5   5369  2692  1014  1228  8269   224  2823    34  1369     0     0
##   6      0     0     0     0     0     0     0     0     0     0     0
##   8      0     0     0     0     0     0     0     0     0     0     0
##   10     0     0     0     0     0     0     0     0     0     0     0
##   11     0     0     0     0     0     0     0     0     0     0     0
##   12     0     0     0     0     0     0     0     0     0     0     0
##   13     0     0     0   335     0     0     0     0     0    13   890
##   15     0     0     0     0     0     0     0     0     0     0     0
##
##         15
##   1    141
##   2      0
##   3      0
##   4      0
##   5      8
##   6      0
##   8      0
##   10     0
##   11     0
##   12     0
##   13     0
##   15     0
```

```
sum(diag(cm))/sum(cm)
```

```
## [1] 0.5923252
```

```
z <- summary(model_multinom)$coefficients/summary(model_multinom)$standard.errors
p <- (1 - pnorm(abs(z),0,1)) * 2
p
```

```
##    (Intercept)   Occupation Marital_Status1 Purchase
## 2            0 9.647593e-01               0        0
## 3            0 1.146631e-08               0        0
## 4            0 2.024878e-01               0        0
## 5            0 6.327874e-02               0        0
## 6            0 5.805212e-01               0        0
## 8            0 8.447361e-01               0        0
## 10           0 8.947019e-01               0        0
## 11           0 4.136303e-01               0        0
## 12           0 6.317178e-04               0        0
## 13           0 0.000000e+00               0        0
## 15           0 0.000000e+00               0        0
```

So,in this code we have used product_category_1 column and we are keeping 1 as the reference category.We are running this model with occupation,Marital_status and purchase as independent variables. Ones I run the multinomial logit,we find that it generates the errors and with each iteration it reduces the value of the error and finally it converged with the value 195440.522682

When I execute the summary of the model,we get how each independent variable has an effect on each categories.For example for category 2 we see that the occupation negatively affects the category which means people from one particular occupation are less likely to purchase from product category 2.Similarly Marital_status has positive influence on category 2.We can give similar explanation for all the other categories.

When I execute the predictmodel we get the probabilities for all the users purchase from any of the category.For example for user 2 the probability of his purchase from category 1 is 0.7648,for category 2 its 0.0897 and for category 3 its 0.05011.We find that out of all the probability of categories for user2 the maximum probability is for purchase of category 1 which is 0.7648 which shows that there is highest chance of user2 to purchase from category 1.Similarly we can find for all the users.

When I find the accuracy it is around 60%,which means that 60% of the data has been predicted correctly by the model.

When I run the z test,we see which independent variables are significant.Marital_Status and Purchase are significant for all the categories since its pvalues are 0.Occupation is significant only for categories 12,13 and 15 as the pvalue is very small for these categories.

Apriori Algorithm.

What is Apriori algorithm? Apriori algorithm is useful in mining frequent itemsets and relevant association rules.We operate this on a database containing large number of transactions.It has got this name because it uses "prior" knowledge of frequent itemsets. There are 3 components in apriori algorithm: Support. Lift. Confidence.

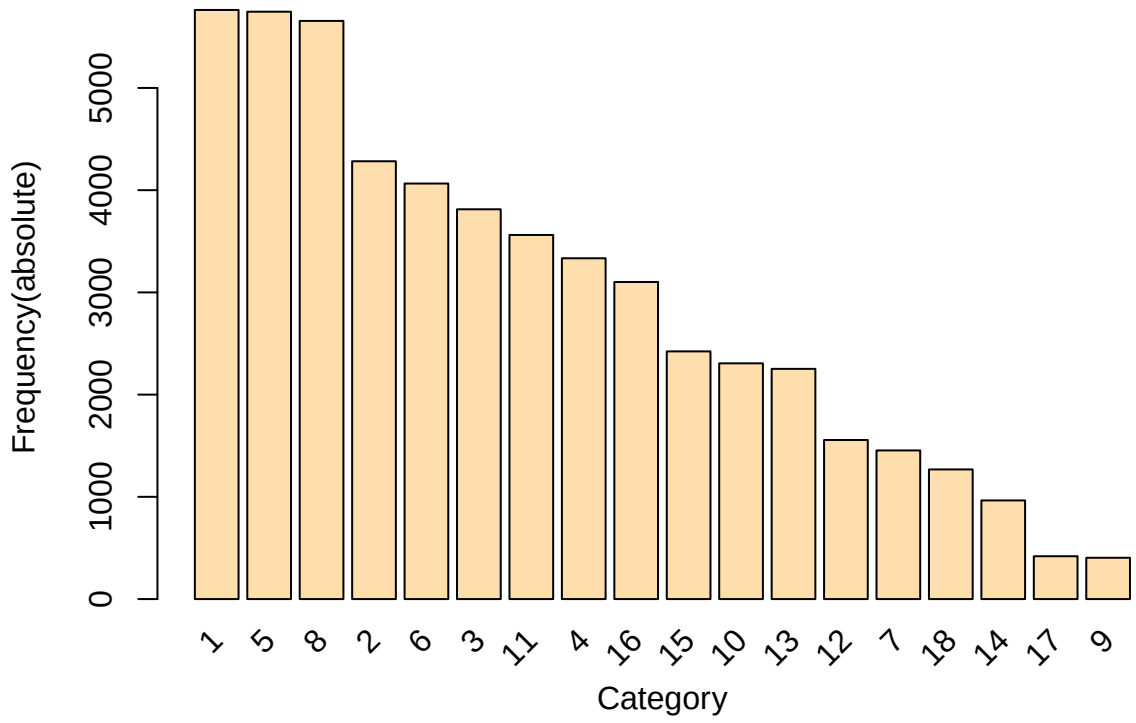
Support:Number of transactions that include antecedent and consequent itemsets. Confidence:the % of antecedent transactions that also have the consequent itemset. Lift: [confidence / (benchmark confidence)] Benchmark Confidence:Transactions with consequent as % of all transactions.

Apriori Algorithm for product category 1:I have run the Apriori algorithm on category 1 and userid.

```
Blackfriday_arules <- read.transactions("BlackFriday.csv",format = "single",cols = c(1,9),sep =",",rm.du
summary(Blackfriday_arules)
```

```
## transactions as itemMatrix in sparse format with
##  5892 rows (elements/itemsets/transactions) and
##  19 columns (items) and a density of 0.4678243
##
## most frequent items:
##       1       5       8       2       6 (Other)
##    5763    5746    5656    4283    4065   26859
##
## element (itemset/transaction) length distribution:
## sizes
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   8  41 164 355 552 637 632 576 553 483 415 388 351 248 239 159  74  17
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   6.000   8.000   8.889  12.000  18.000
##
## includes extended item information - examples:
##   labels
## 1      1
## 2     10
## 3     11
##
## includes extended transaction information - examples:
##   transactionID
## 1       1000001
## 2       1000002
## 3       1000003
```

```
itemFrequencyPlot(Blackfriday_arules,topN=18,type = "absolute",col = "#FFDEAD",xlab = "Category",ylab =
```




```
rules <- apriori(Blackfriday_arules, parameter = list(supp = 0.5, conf = 0.5, target = "rules"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5     0.5      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 2946
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[19 item(s), 5892 transaction(s)] done [0.00s].
## sorting and recoding items ... [9 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [188 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
#inspect(head(sort(rules, by = "lift")))

rules.tbl <- inspect(tail(rules,20))
```

```
##      lhs          rhs support   confidence lift     count
## [1]  {1,5,6}   => {2} 0.5395451 0.8023725  1.103801 3179
## [2]  {1,2,5}   => {6} 0.5395451 0.7605263  1.102342 3179
## [3]  {5,6,8}   => {1} 0.6631025 0.9891139  1.011254 3907
## [4]  {1,6,8}   => {5} 0.6631025 0.9871147  1.012196 3907
## [5]  {1,5,6}   => {8} 0.6631025 0.9861181  1.027264 3907
## [6]  {1,5,8}   => {6} 0.6631025 0.7208487  1.044832 3907
## [7]  {2,5,8}   => {1} 0.6914460 0.9905179  1.012690 4074
## [8]  {1,2,8}   => {5} 0.6914460 0.9881154  1.013222 4074
## [9]  {1,2,5}   => {8} 0.6914460 0.9746411  1.015309 4074
## [10] {1,5,8}   => {2} 0.6914460 0.7516605  1.034038 4074
## [11] {2,3,5,8} => {1} 0.5186694 0.9938211  1.016067 3056
## [12] {1,2,3,8} => {5} 0.5186694 0.9967384  1.022065 3056
## [13] {1,2,3,5} => {8} 0.5186694 0.9788597  1.019703 3056
## [14] {1,3,5,8} => {2} 0.5186694 0.8428020  1.159418 3056
## [15] {1,2,5,8} => {3} 0.5186694 0.7501227  1.159120 3056
## [16] {2,5,6,8} => {1} 0.5347929 0.9952622  1.017540 3151
## [17] {1,2,6,8} => {5} 0.5347929 0.9933796  1.018620 3151
## [18] {1,2,5,6} => {8} 0.5347929 0.9911922  1.032550 3151
## [19] {1,5,6,8} => {2} 0.5347929 0.8065012  1.109480 3151
## [20] {1,2,5,8} => {6} 0.5347929 0.7734413  1.121062 3151
```

To run this algorithm,I have used read.transactions which converts the two given columns into transactions.

In itemFrequency plot we find that categories 1,5 and 8 has maximum purchase.The graph is right skewed. When we actually execute apriori algorithm with support and confidence equal to 0.5 ,in the output we find that for 1st transaction if the user purchase from category 1,5 and 6 then the probability that he also purchases from category 2 is higher.The confidence is 0.80 which shows that 80% of the times users purchase from categories 1,5 and 6 also purchase from category 2.Similarly for 2nd transaction we find that if the user purchases from categories 1,2 and 5 then there is a high probability that he will also purchase from category 6.The confidence is 0.76 which shows that the user who purchase from categories 1,2 and 5 also purchased from category 6 76% of the times.We can give similar explanations for all the other transactions.

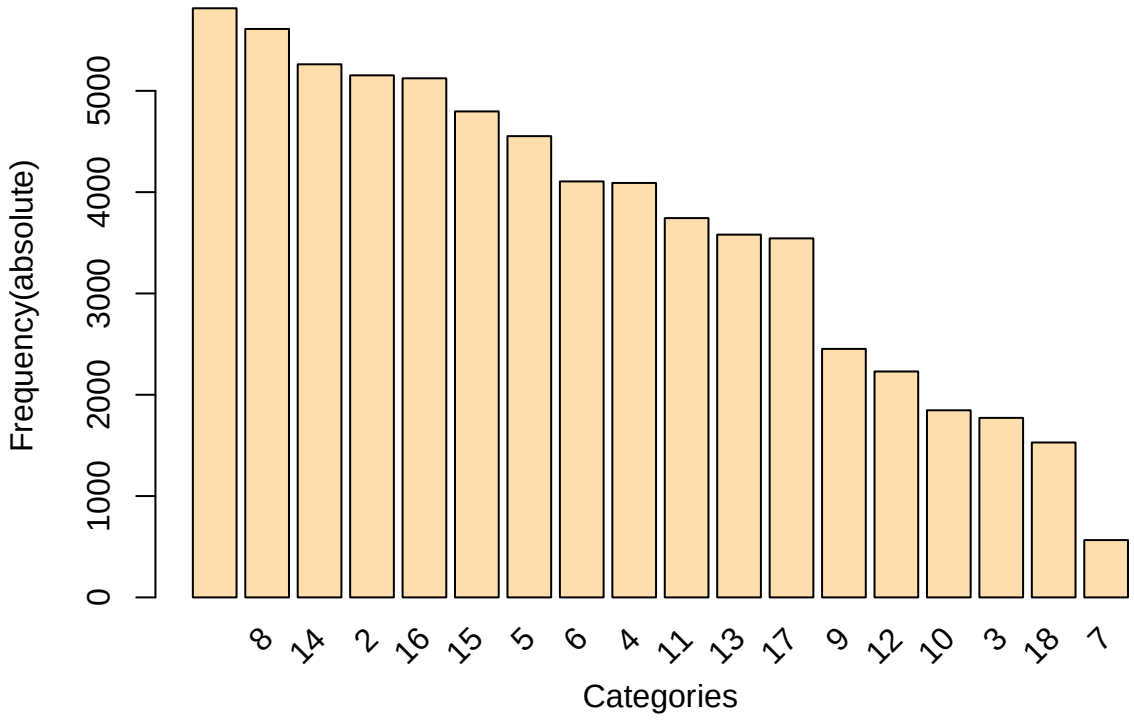
Let us see the apriori algorithm for product category 2. I have run the Apriori algorithm on category 2 and userid.

```
Blackfriday_arules2 <- read.transactions("BlackFriday.csv",format = "single",cols = c(1,10),sep =",",rm
summary(Blackfriday_arules)
```

```
## transactions as itemMatrix in sparse format with
##  5892 rows (elements/itemsets/transactions) and
##  19 columns (items) and a density of 0.4678243
##
## most frequent items:
##       1       5       8       2       6 (Other)
##    5763    5746    5656    4283    4065   26859
##
## element (itemset/transaction) length distribution:
## sizes
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   8  41 164 355 552 637 632 576 553 483 415 388 351 248 239 159  74  17
##
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   6.000   8.000   8.889  12.000  18.000
##
## includes extended item information - examples:
##   labels
## 1      1
## 2     10
## 3     11
##
## includes extended transaction information - examples:
##   transactionID
## 1       1000001
## 2       1000002
## 3       1000003
```

```
itemFrequencyPlot(Blackfriday_arules2,topN=18,type = "absolute",col = "#FFDEAD",xlab = "Categories",yla
```




```
rules2 <- apriori(Blackfriday_arules2, parameter = list(supp = 0.7, conf = 0.3, target = "rules"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.3    0.1    1 none FALSE            TRUE       5     0.7      1
##  maxlen target   ext
##      10  rules FALSE
##
```

```
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 4124
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[19 item(s), 5892 transaction(s)] done [0.00s].
## sorting and recoding items ... [7 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [137 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
#inspect(head(sort(rules2, by = "lift")))

rules.tbl2 <- inspect(tail(rules2,20))
```

```
##      lhs          rhs  support   confidence lift     count
## [1]  {14,16,2} => {8}  0.7001018 0.9758694  1.024741 4125
## [2]  {16,2,8}  => {14} 0.7001018 0.9305211  1.041929 4125
## [3]  {14,16,8} => {2}  0.7001018 0.9085903  1.038893 4125
## [4]  {14,2,8}  => {16} 0.7001018 0.9122070  1.048931 4125
## [5]  {14,16,2} => {}   0.7121521 0.9926662  1.005811 4196
## [6]  {,16,2}   => {14} 0.7121521 0.9242291  1.034884 4196
## [7]  {,14,16}  => {2}  0.7121521 0.9052859  1.035114 4196
## [8]  {,14,2}   => {16} 0.7121521 0.9088152  1.045031 4196
## [9]  {16,2,8}  => {}   0.7450781 0.9903000  1.003413 4390
## [10] {,16,2}   => {8}  0.7450781 0.9669604  1.015386 4390
## [11] {,16,8}   => {2}  0.7450781 0.8992216  1.028180 4390
## [12] {,2,8}    => {16} 0.7450781 0.9005128  1.035484 4390
## [13] {14,16,8} => {}   0.7652749 0.9931718  1.006323 4509
## [14] {,14,16}  => {8}  0.7652749 0.9728155  1.021534 4509
## [15] {,16,8}   => {14} 0.7652749 0.9235969  1.034176 4509
## [16] {,14,8}   => {16} 0.7652749 0.8953535  1.029552 4509
## [17] {14,2,8}  => {}   0.7603530 0.9907121  1.003831 4480
## [18] {,14,2}   => {8}  0.7603530 0.9703271  1.018921 4480
## [19] {,2,8}    => {14} 0.7603530 0.9189744  1.029000 4480
## [20] {,14,8}   => {2}  0.7603530 0.8895949  1.017173 4480
```

To run this algorithm,I have used read.transactions which converts the two given columns into transactions.

When we execute the apriori algorithm with support equal to 0.7 and confidence equal to 0.3,we find that for 1st transaction if the user purchases from category 14,16 and 2 then there is a high probability that he will purchase from category 8.The value of confidence is 97.5% which means that 97.5% of the times the users who purchase from categories 14,16 and 2 also purchased from category 8.For transaction 2,if he purchases from categories 16,2 and 8 then there is a high probability that he will purchase from category 14.The confidence value is 93% which shows that 93% of the times the users who purchased from categories 16,2 and 8 also purchased from category 14.Similar explanation can be given for all other transactions

Conclusion

- The maximum purchase was done by the users between age group 26-35
- On an average Males has done more purchase than females.
- City Category B has maximum purchase and males from city category C has maximum purchase
- Product Category 1 and 2 was purchased maximum number of times.

- Maximum people live in a city not more than a year.

Conclusions from the analysis

- Linear Discriminant Analysis. In LDA we find that maximum data is contained in city category B which means that maximum users living in city category B has highest purchase and which also supports our 3rd point in the conclusion above.LD1 contains has 83.71 % of the data.
- Multinomial Logistic Regression. For user 2 the probability that he will purchase from product 2 is 89%.For user 7 the probability that he will purchase from product 1 is 75%.
- Apriori Algorithm In Product category 1, we find that the confidence that products 2,5,8 and 1 are purchased together is higher.In Product Category 2, we find that the confidence that products 14,2 and 8 are purchased together is higher.

Suggestions For the Seller.

- He should sell more products in City category B as there is maximum sales.
- He should keep in stock more items which males purchase.
- The seller should keep the products which are much appreciated and in demand for the people who are either students or are working professionals.
- The seller should sell the products from categories 1,2,5 and 8 together.