Project : Online B2B Retail

– **MD Naseem Ashraf, T12, CS-686 Data Mining, University of San Francisco, Fall 2016**

Aim: Classify products as per their sale priority considering various aggregated attributes.

Classification: 1 (High Priority), 2(Medium Priority) & 3(Low Priority)

Introduction

This project covers building classification models to *classify* the **Sale Priority Class** according to agregated attributes of proudcts sold via B2B site [PUT]. The data preprocessing and actual training and test dataset generation (Steps 1 & 2) is covered in the **Appendix** at the end of this project report.

Step 3. Data Exploration

DataExploration.r does basic statistical exploration of the dataset with attriubtes;

ProductID – Integer, Unique, Generated

Description – Char, Unique

TransactionFreq – Double, Counted for each occurance of each product

TotalQuantity – Integer, Counted for each product

Customers – Integer, Counted for each product

MeanQuantityPerTransaction – Double, Computed for each product

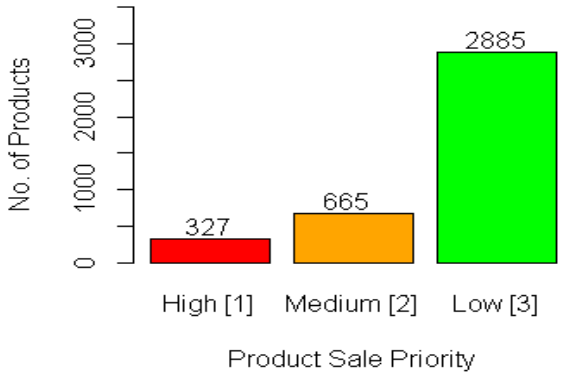
MeanQuantityPerCustomer – Double, Computed for each product

UnitPrice – Double

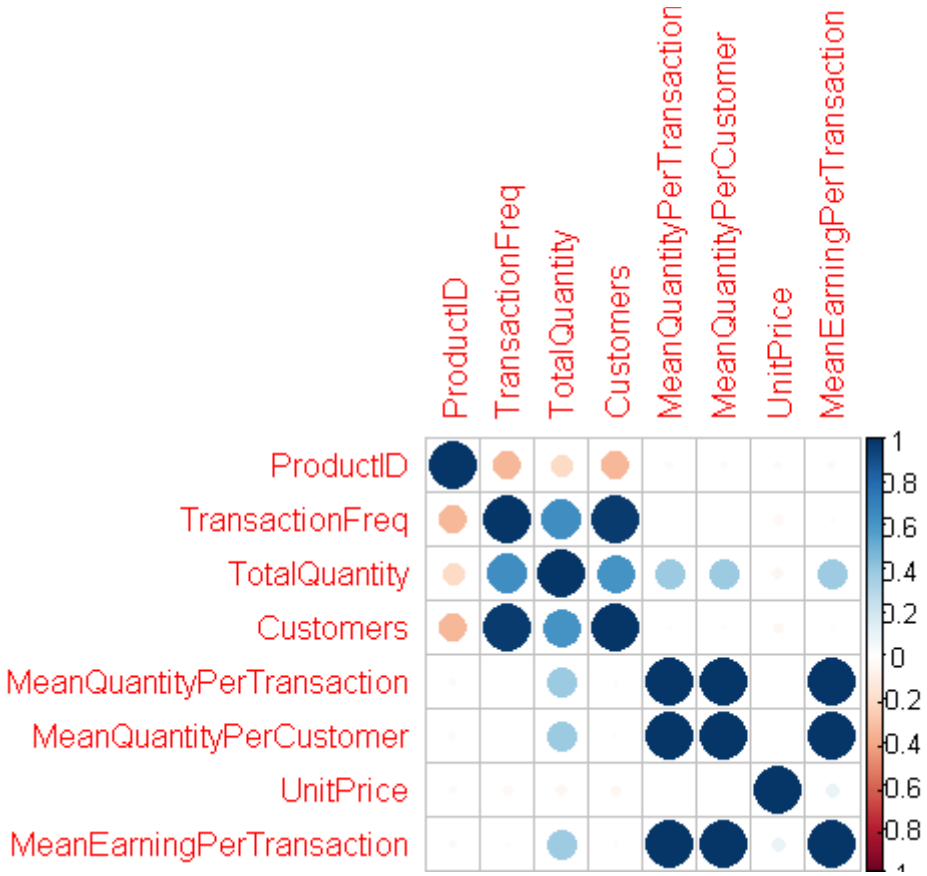
MeanEarningPerTransaction – Double, Computed for each product

SalePriorityClass – Factor, Generated, Levels 3: 1, 2, 3

See below/next page for a break down of classification of the SalePriorityClass of products in dataset.



Further, analytics on the dataset is as follows;



Correlation Table (to two digits after decimal)

1	-0.35	-0.2	-0.35	0.03	0.03	0.02	0.3
-0.35	1	0.65	0.97	-0.01	-0.01	-0.04	-0.01
-0.2	0.65	1	0.62	0.39	0.39	-0.06	0.38
-0.35	0.97	0.62	1	-0.01	-0.01	-0.05	-0.01
0.03	-0.01	0.39	-0.01	1	1	0	0.99
0.03	-0.01	0.39	-0.01	1	1	0	0.99
0.02	-0.04	-0.06	-0.05	0	0	1	0.09
0.03	-0.01	0.38	-0.01	0.99	0.99	0.09	1

Note: Scatter plot added to appendix.

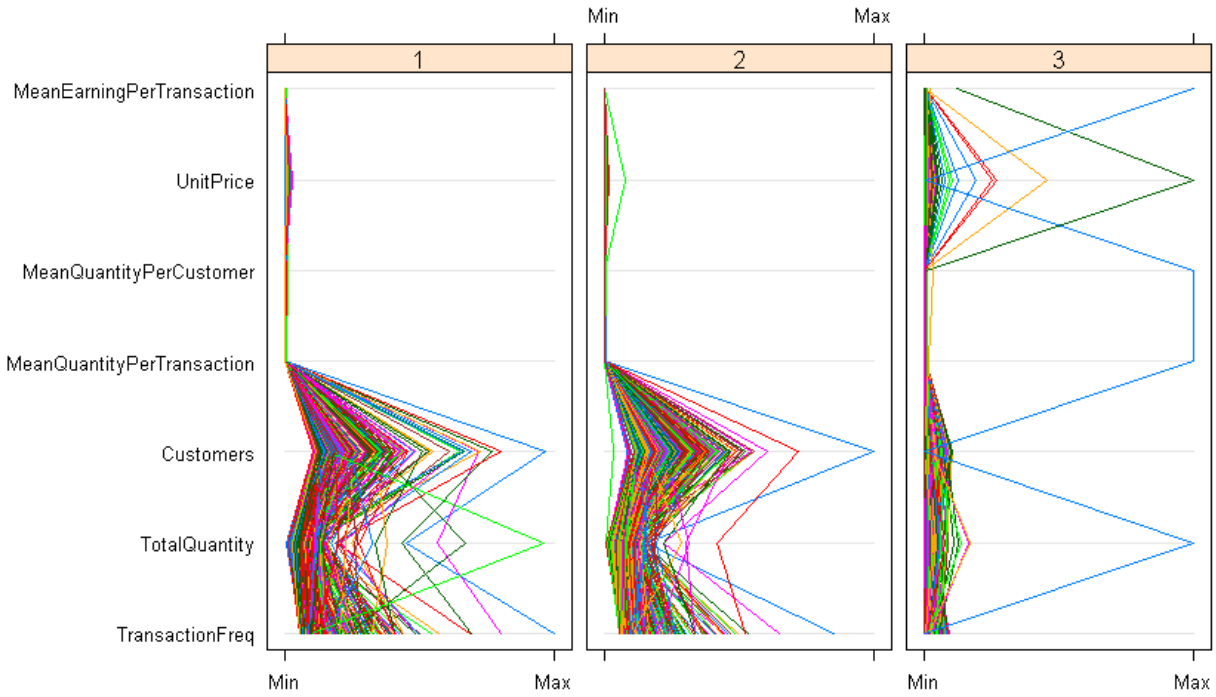
Observation

We can see the high levels of correlation between certain attributes. We can select a few of the attributes as per their corelations and previous knowledge that they are either similar or similarly aggregated form other observations. Like,

MeanQuantityPerTransaction = (TotalQuantity/Total Number of Transactions).

<u>TransactionFreq, TotalQuantity, Customers (Total) and the mean computations.</u>

Class wise Attributes Parallel Plot:



Observation:

We can see that each class has a very small range of Mean computed attributes. They may be the defining contraints to the classification. But, this is not true for some of the strange cases in Class 3 [Low Sale Priority Products].

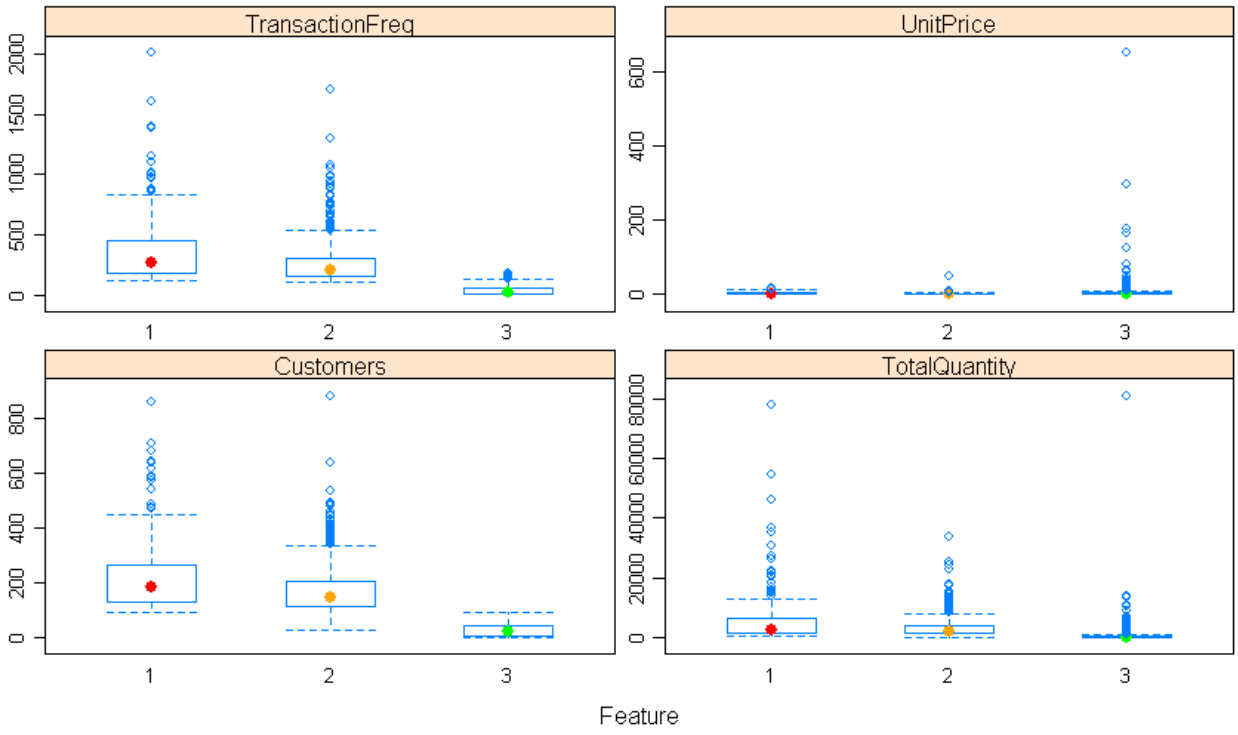
Let's see the light blue line, outlier in Class 3. We can see that the it has a very low number of customers ordering that product, but has a large quantity of order which skews the mean computations and is still correctly classified as a Low priority sale product, despite having large total earning, which shows there is a strong correlation between the Class 3 classification and the low number of Customers (Total). This can also be seen by the wide variance of Customer count for Class 2 [Medium] and Class 1 [High] priority products, which have larger number of customers.

Furthermore, we can see that Class 3 [Low] products have smaller range of TransactionFreq. i.e Very few people are purchasing these products.

In fact on delving in this outlier we see;

```
TransactionFreq = 1
TotalQuantity = 80995
Customers = 1 [Sale of this product is dependent on only one customer and hence
               penelized as Class 3.]
MeanQuantityPerTransaction = 80995
MeanQuantityPerCustomer = 80995
UnitPrice = 2.08
MeanEarningPerTransaction = 168469.6
SalePriorityClass = 3
```

Box and Whisker Plots of highly corellated Attributes:



Observation

Class 1 products have a higher number of transactions over the year, TransactionFreq, followed by Class 2 and Class 3.

UnitPrice is variant most in Class 3, wiz. Class 3 products range from costly products to cheap ones relative to each other.

Class 1 has higher number of total quantities ordered in addition to higher transaction frequency and higher number of customers. The same pattern is seen in Class 2. While Class 3 has the lowest value range of these features.

Other Attribute Information of the Classes;

MeanEarningPerTransaction – [As seen from the dependance on TransactionFreq]

Class 1 [High] = `24.11086, 409.25576`

Class 2 [Medium] = `1.940245, 79.951064 [Note: The overlapping of Class 2 to 1]`

Class 3 [Low] = `0.001, 168469.600 [Note: Maximum Outlier discussed earlier]`

UnitPrice – [Less price variances & difference in Class 1 Products while most in Class 3]

Class 1 [High] = `0.29, 18.00`

Class 2 [Medium] = `0.1, 50.0`

Class 3 [Low] = `0.001 649.500`

MeanQuantityPerCustomer -

Class 1 [High] = 2.754545, 564.608696 [Covers high demand products for most customers]

Class 2 [Medium] = 2.242424, 168.580882

Class 3 [Low] = 1, 80995 [Note: Outlier of one customer ordering 80995 products]

MeanQuantityPerTransaction -

Class 1 [High] = 2.278195, 393.515152 [Relatively higher Quantitie ordered per Transaction]

Class 2 [Medium] = 1.007519, 110.225962

Class 3 [Low] = 1, 80995 [Note: Outlier of one customer ordering 80995 products]

Customers (Total) – [Similar observations as for above features.]

Class 1 [High] = 94, 856

Class 2 [Medium] = 29, 881

Class 3 [Low] = 1, 93

TotalQuantity -

Class 1 [High] = 303, 77916

Class 2 [Medium] = 134, 33670

Class 3 [Low] = 1, 80995

TransactionFreq -

Class 1 [High] = 113, 2015

Class 2 [Medium] = 104, 1714

Class 3 [Low] = 1, 185

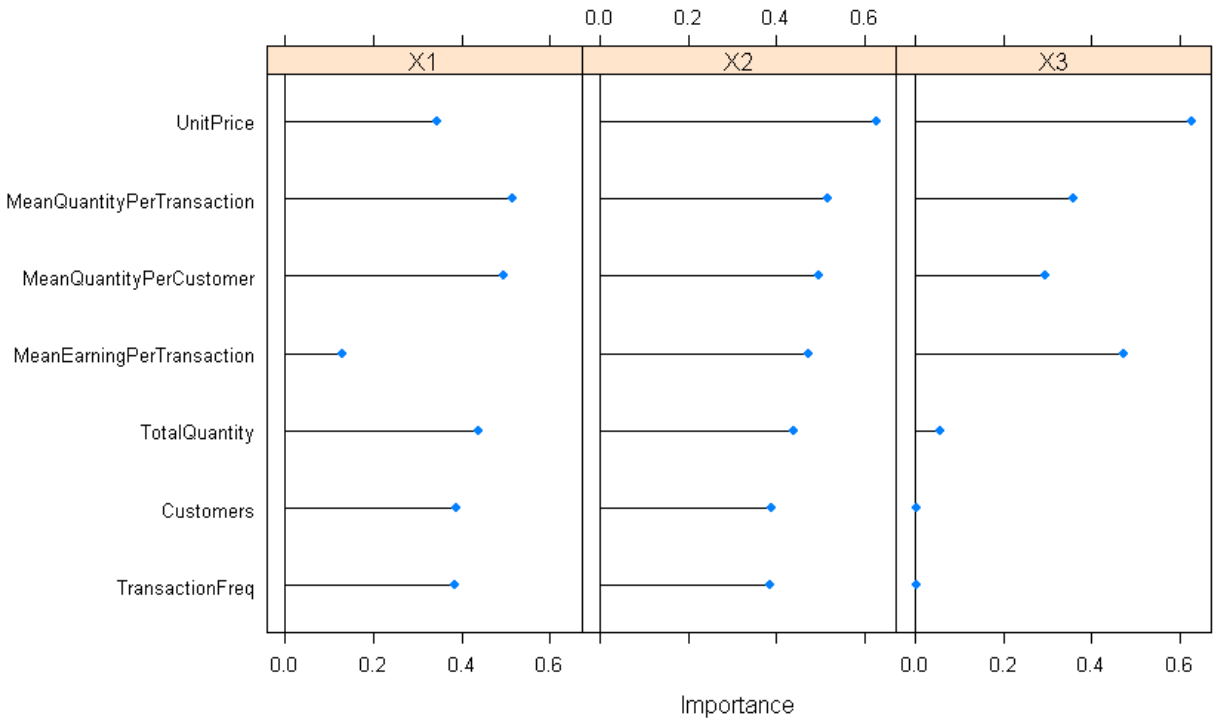
Model Building Hypothesis:

As we can see that Class 3 can be easily singled out with lower values of most features, we see a clear overlapping of ranges between the Class 1 & 2 classification over individual attributes. Perhaps interactions of multiple attributes is invovled in classifying 1 & 2. As also seen from Parallel Plots of the classes and their feature ranges.

We can deduce further about our most relevant and significant features for our model which decides the SalePriorityClass of the produts. I used two feature importance analysis methods to deduce the significant attributes to compose our possible models with.

The first method uses Cross Validation on a classification model taking the entire set of attributes/features of the entire system and finding out which are important and by how much in a relative score. This is done using Cross Validation from library "*mlbench"*.

Feature Importance Plot (Built using CrossValidation):



Note: Refer appendix FeatureImportanceScore in Appendix for numerical score outputs.

The second method of analysis is using a library utilizing Random Forests cross validated for a recursive feature elimination and checking if the model has high prediction accuracy and so on with dropping one feature/attribute form the full system model one by one.

Recursive Feature Elimination (Random Forest method of finding best significance features):

Note: See RFE Plot in Appendix.

```
Recursive feature selection

Outer resampling method: Cross-Validated (10 fold)

Resampling performance over subset size:

 Variables Accuracy  Kappa AccuracySD  KappaSD  Selected
         1   0.6639 0.1662   0.028045 0.065013
         2   0.9969 0.9924   0.003182 0.007879
         3   0.9987 0.9968   0.003288 0.008162
         4   0.9987 0.9968   0.002516 0.006235        *
         5   0.9984 0.9962   0.002501 0.006195
         6   0.9982 0.9956   0.002740 0.006770
         7   0.9979 0.9949   0.003408 0.008442

The top 4 variables (out of 4):
   MeanEarningPerTransaction, Customers, TransactionFreq, TotalQuantity
```

We can see that this method predicts the significane of the aforementioned features which yield the highest statistically significant accuracy. But, we can also see that over all most or all of the features of the model are relevant and give really really high accuracy (approx. ~ 99%).

Histograms of other attributes are irrelevant for this classification data exploration. All DataExploration steps are collected in **DataExploration.r.**

Models in Consideration

Model 1:

SalePriorityClass Relation to (TransactionFreq, TotalQuantity, Customers, MeanQuantityPerTransaction, MeanQuantityPerCustomer, UnitPrice, MeanEarningPerTransaction)

Model 2:

SalePriorityClass Relation to (TransactionFreq, Customers, MeanQuantityPerTransaction, MeanEarningPerTransaction)

Model 3:

SalePriorityClass Relation to (TransactionFreq, Customers, MeanEarningPerTransaction)

This can in itself be one of the predictors for Class 1/2 and Class 3 differentiation. But, not delved into in detail as it's useless for Multiclass classifier.

Note: Most of the models have been run on a test and training dataset sampled randomly into 1:9 ratio.

Methods in Consideration

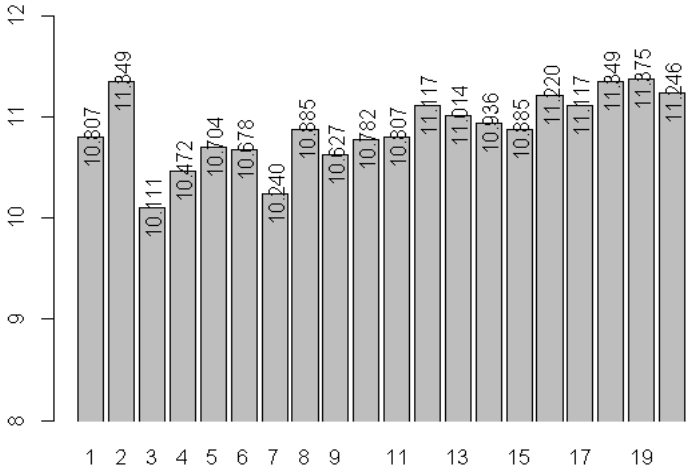
For a multiclass classification model we use supervised learning with the following methods:

1. **KNN** (k-Nearest Neighbors Algorithm) – classifierknn.r
2. **SVM** (Support Vector Machine) – classifiersvm.r
3. **LDA** (Linear Discriminant Analysis) – classifierlda.r

Classifier Design & Performance

KNN -

Cross Validation Model 1



Model 1 Performance (K=3)

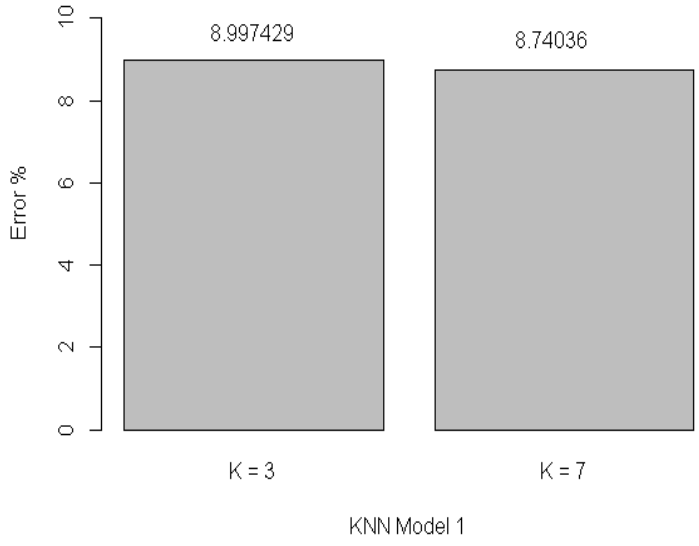
knn.pred	1	2	3
1	13	9	0
2	21	43	3
3	0	2	298

Accuracy = 0.9100257

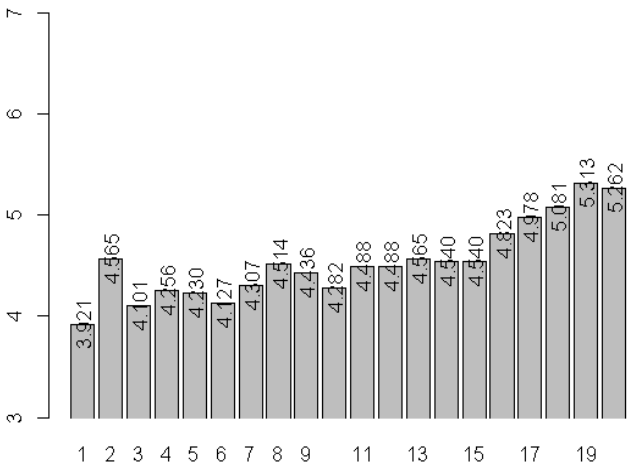
Model 1 Performance (K=7)

knn.pred	1	2	3
1	13	5	0
2	21	48	7
3	0	1	294

Accuracy = 0.9125964



Cross Validation Model 2



Model 2 Performance (K=1)

knn.pred	1	2	3
1	24	6	0
2	10	48	1
3	0	0	300

Accuracy = 0.9562982

Model 2 Performance (K=3)

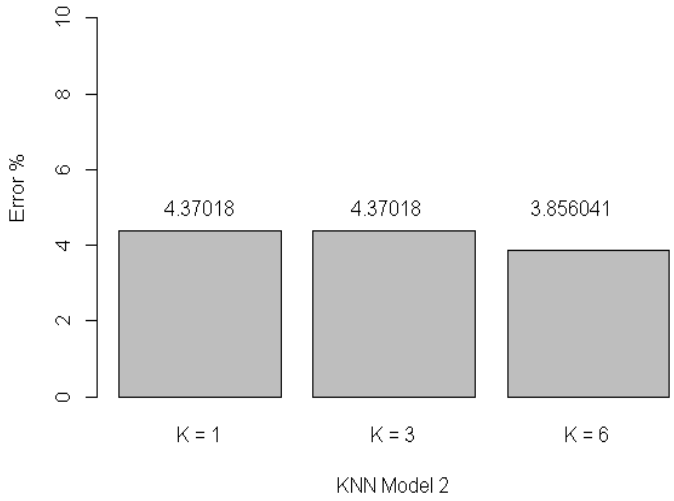
knn.pred	1	2	3
1	22	3	0
2	12	51	2
3	0	0	299

Accuracy = 0.9562982

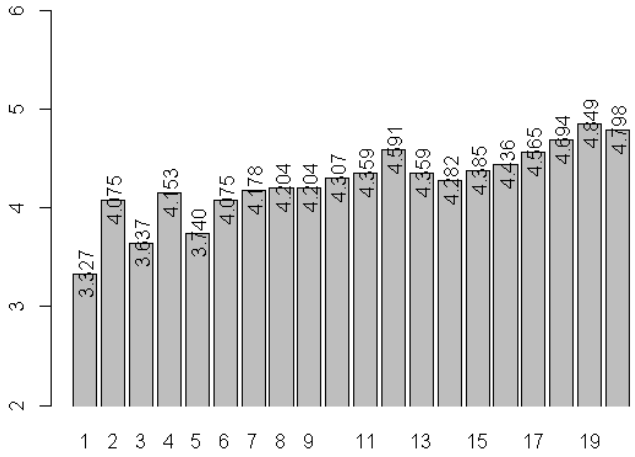
Model 2 Performance (K=6)

knn.pred	1	2	3
1	23	2	0
2	11	51	1
3	0	1	300

Accuracy = 0.9614396



Cross Validation Model 3



Model 3 Performance (K=1)

knn.pred	1	2	3
1	26	5	0
2	8	49	1
3	0	0	300

Accuracy = 0.9640103

Model 3 Performance (K=3)

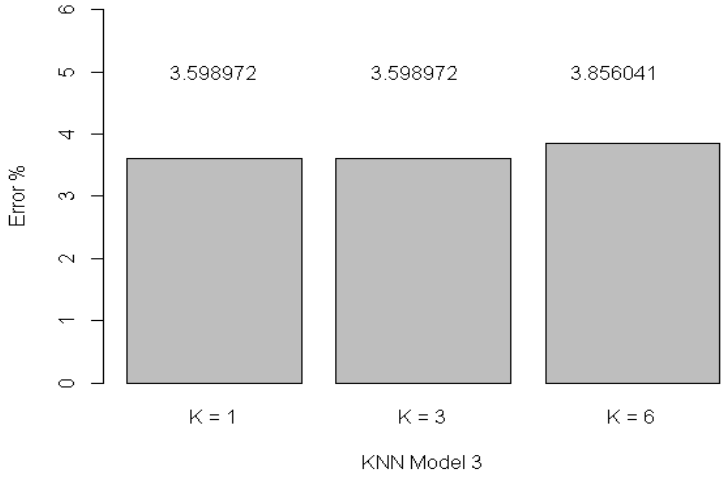
knn.pred	1	2	3
1	24	2	0
2	10	52	2
3	0	0	299

Accuracy = 0.9640103

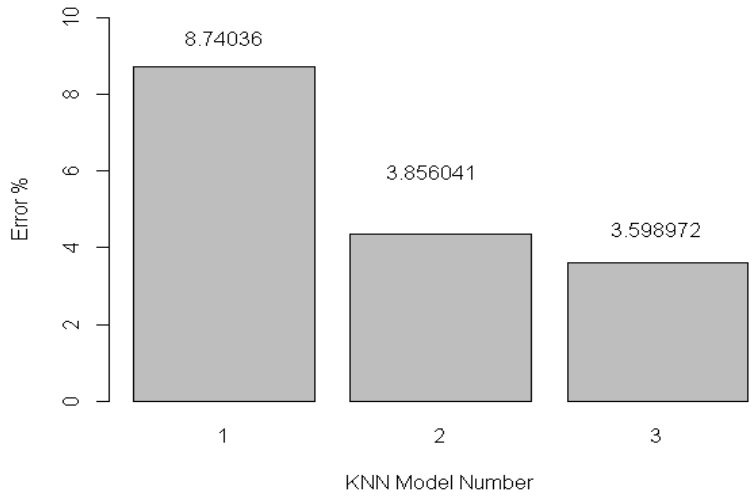
Model 3 Performance (K=5)

knn.pred	1	2	3
1	23	2	0
2	11	52	2
3	0	0	299

Accuracy = 0.9614396



KNN Best of All Models



SVM – [Radial]

Model 1 Performance

pred	1	2	3
1	5	0	1
2	29	52	1
3	0	2	299

Error = 0.0848329

Tuned Model 1 Performance (C=1.1, G=(0.5,1,2))

pred	1	2	3
1	9	1	1
2	25	52	1
3	0	1	299

Error = 0.07455013 (Svc = 659)

Model 2 Performance

pred	1	2	3
1	7	1	0
2	27	52	1
3	0	1	300

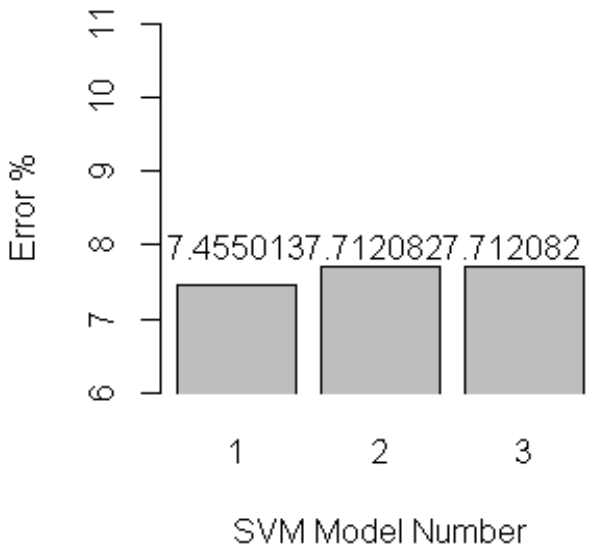
Error = 0.07712082

Tuned Model 2 Performance (C=1.1, G=(0.5,1,2))

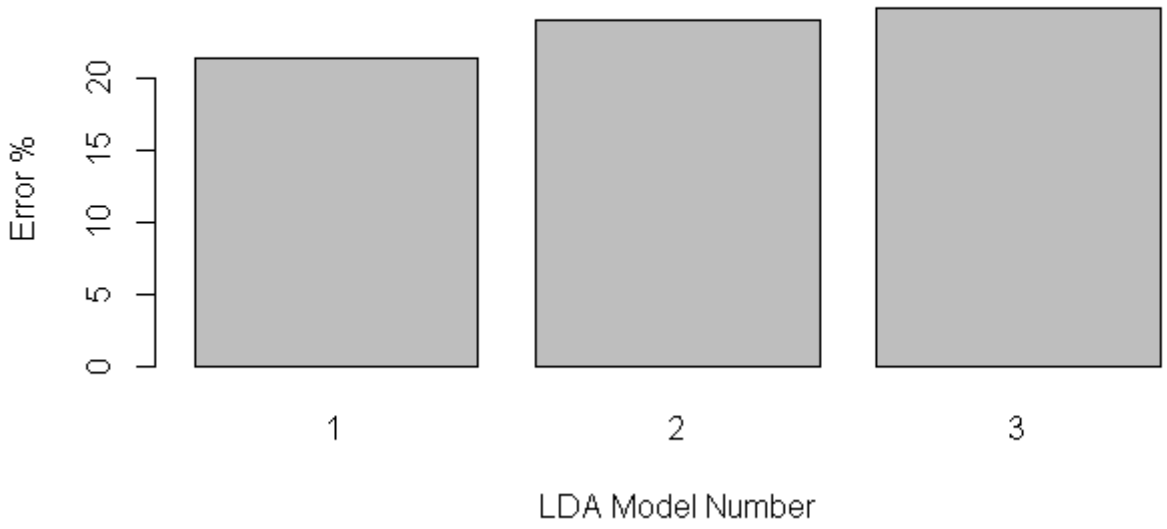
pred	1	2	3
1	6	1	0
2	28	53	1
3	0	0	300

Error = 0.07712082 (Svc = 718 / 724)

SVM Best of All Models



LDA –



LDA Model 1

pred	**1**	**2**	**3**
1	6	5	0
2	8	16	1
3	0	11	70

Accuracy = 0.7863248 ~ 78.63%

LDA Model 2

pred	**1**	**2**	**3**
1	4	4	0
2	9	15	1
3	1	13	70

0.7606838 ~ 76.06%

LDA Model 3

pred	**1**	**2**	**3**
1	5	5	0
2	9	12	0
3	0	15	71

0.7521368 ~ 75.21%

Conclusion

Model 3 of KNN is the best model of the various tried and tested.

APPENDIX

Step 1: Data Preprocessing & Generation

Input dataset (*OnlineRetail.xlsx*) is a set of transactions over a year of a B2B wholesale supplier with 406,828 complete recorded transactions. From which 401,603 are unique transactions. Out of which 128 products were cancelled, lost or damaged product entires disregarded in analysis. After removing all cancelations and erroneous transactions we get 392,731 transactions over the year.

OnlineRetail.xlsx attributes:

InvoiceNo - Alphanumeric

StockCode – Alphanumeric, non-unique

Description – String, non-unique

Quantity – Numeric, Integer

InvoiceDate – Numeric, Decimal

UnitPrice – Numeric, Decimal

CustomerID – Alphanumeric, non-unique

Country – String, non-unique

1. Preprocessing

Preprocessing.r covers entire data aggregation and cleaning from OnlineRetail.xlsx to orderedallproducts.xlsx and finaldataset.xlsx. It also shows subsets the three subsets of high frequency, medium frequency and low frequency transaction products.

Transactions are aggregated per each product, which is assigned a unique key, ProductID. Aggregated data (i.e. New attributes) are;

Orderedallproducts.xlsx attributes:

ProductID – Numeric, Integer, Unique

Description – String, Unique

TransactionFreq – Numeric, Integer (Sum of all occurance of the Product except cancelations). Dataset is sorted in decreasing order of this attribute.

TotalQuantity – Numeric, Integer (Sum of all quantites of the Product ordered over the year).

Customers – Numeric, Integer (Total number of customers who ordered this Product as per CustomerID).

MeanQuantityPerTransaction – Numeric, Decimal

MeanQuantityPerCustomer – Numeric, Decimal

Finaldataset.xlsx attributes:

All attributes of Orderedallproducts.xlsx with additional atrributes from OnlineRetail.xlsx.

UnitPrice

MeanEarningPerTransaction – Numeric, Decimal (Product of Mean of Quantity per Transaction and Unit Price).

2. Data Generation

DataGeneration.r performs the actual filtering of High and medium demand/transaction products chopping of the long tail of low frequency products and classifying dataset to produce our actual dataset to Data Mine on.

DataGeneration.r uses *classifyPrioritySale* function to assign priorities of 1 (High Sale/Profitability), 2(Medium Sale/Profitability) or 3(Low Sale/Profitability), in turn using the statistic of number of customers, mean quantity orderd per transaction and mean eraning per transaction in the given dataset.

This **finaldataset** generated with classification of 1170 Products which have high to medium Transaction Frequency, dropping the rest of the long tail, is saved with **SalePriorityClass** attribute with factors of 1,2 or 3 in **newproductdatasetclassified.xlsx**.

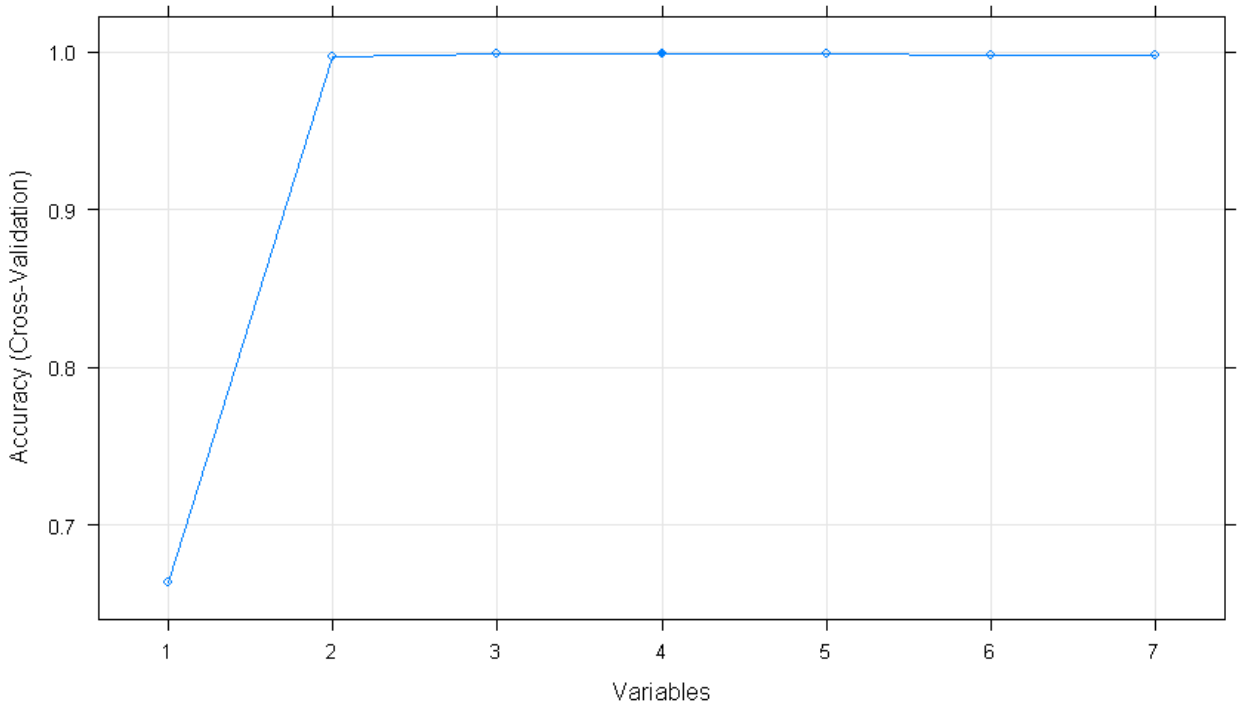
All Data Mining is done on this cleaned & preprocessed dataset **newproductdatasetclassified.xlsx**.

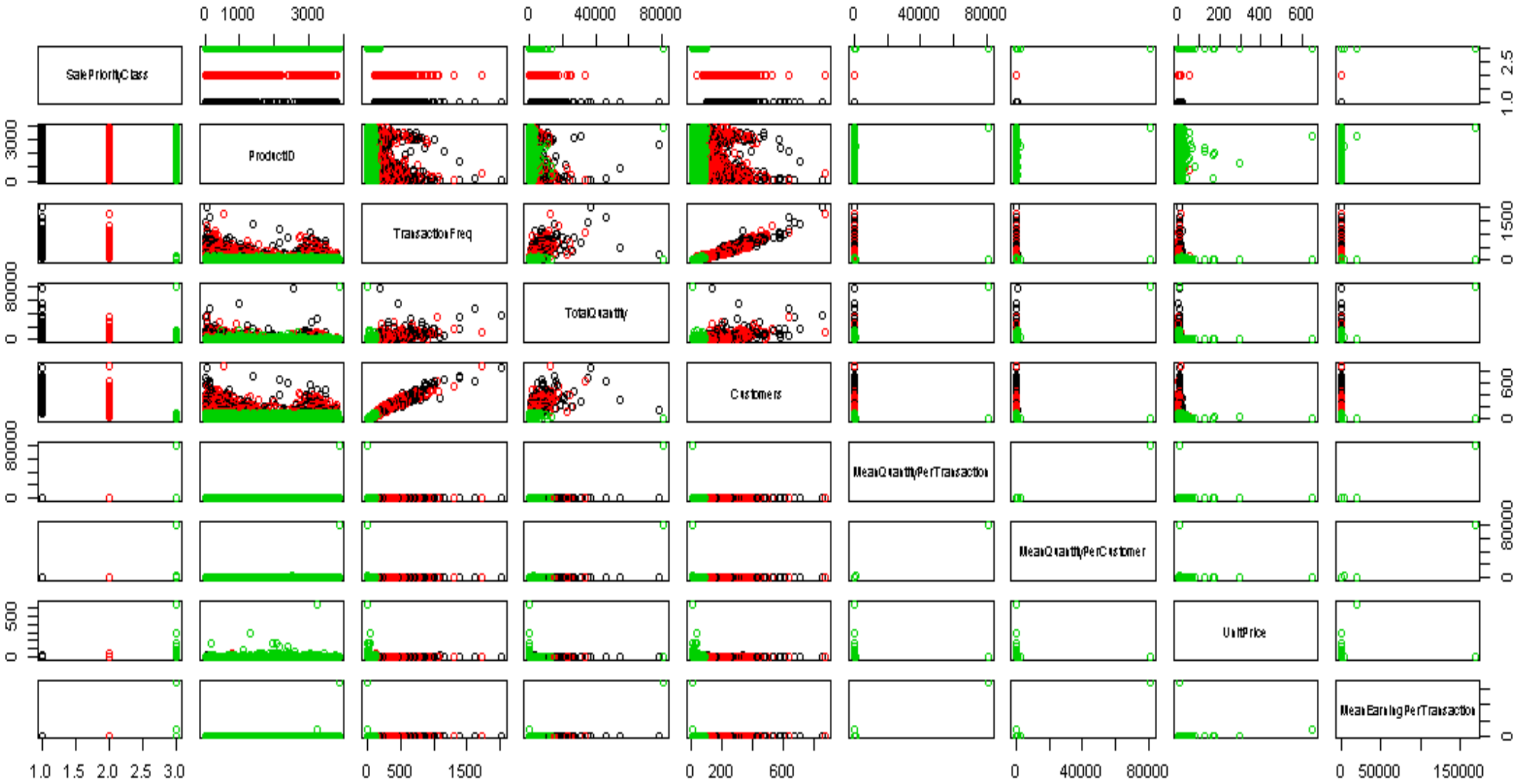
3. Feature Importance Score

```
ROC curve variable importance

  variables are sorted by maximum importance across the classes
                              X1     X2       X3
UnitPrice                   0.3460 0.6279 0.627929
MeanQuantityPerTransaction 0.5164 0.5164 0.358084
MeanQuantityPerCustomer    0.4950 0.4950 0.295861
MeanEarningPerTransaction  0.1298 0.4717 0.471711
TotalQuantity               0.4385 0.4385 0.056460
Customers                   0.3871 0.3871 0.001033
TransactionFreq             0.3834 0.3834 0.002981
```

4. RFE Plot




SalePriorityClass
ProductID
TransactionFreq
TotalQuantity
Customers
MeanQuantityPerTransaction
MeanQuantityPerCustomer
UnitPrice
MeanEarningPerTransaction