A Comparative Analysis of Classifiers for Predicting Category of Products in Consumer Packaged Goods Industry

Thasni T
Assistant Professor, Computer Science & Engineering, Presidency University, Bangalore, Karnataka, India

Abstract : Classification is used to find out in which group each data instance is related within a given dataset. It is used for classifying data into different classes according to some constrains. The major kinds of classification algorithms include k-nearest neighbour classifier, SVM, and Random Forest. An organization especially in retail, ecommerce, consumer packaged goods industry have lot of identical product's which are clustered into a higher category. This paper tries to identify which of these algorithms provide a better accuracy on classification of products into higher categories.

***IndexTerms* - Classifiers; decision trees; boosting; random forest; KNN; SVM; Support Vector Machine; Prediction; accuracy;**

I. INTRODUCTION

Classification methods in data mining are able to process a large amount of data. Categorical class labels can be predicted by using this method and it classifies data based on training set and class labels and it can be used for classifying newly available data. The word could cover any context in which some decision or forecast is made on the basis of presently available information. Classification procedure is recognized method for repeatedly making such decisions in new situations. [5].The creation of a classification procedure from a set of data for which the exact classes are known in advance is termed as pattern recognition or supervised learning[6]. Classification task is fundamental in certain contexts like assigning individuals to credit status on the basis of personal and financial information and the diagnosis of a patient's disease in order to select immediate treatment while waiting for perfect test results. The most critical problems arising in science, industry and commerce can be called as classification or decision problems. All groups have some objectives in common. They have all attempted to develop procedures that would be able to handle a wide variety of problems and to be extremely general used in practical settings with proven success.

II. PROBLEM STATEMENT

For any organization, especially in retail and Consumer Packaged Goods industries, that has many thousands of products in their production across various geographies, a strong analysis of their products and classification into its higher category is very important. This helps in understanding the customer behavior across various geographies and helps in forecasting and planning and helps sales teams better regarding their sales targets. But due to diverse and global infrastructure, similar products can get classified as different .Such an inaccurate clustering produces bad results. With better classification the insights generated about the various product ranges becomes usable.

Let's assume there are thousands of products and each product can be described by some attributes. And each product will have different values for such attributes, also known as features. Such an instance is created by this work. We have a number of products described by its attributes and our goal is to classify the different products into its right category by using 3 famous classification algorithms – KNN, SVM and Random Forest and understand which algorithm performs better.

III. ALGORITHMS

3.1. KNN

K Nearest Neighbors is an algorithm which is very simple .It will store all available cases and it will classify new cases by a majority vote of its k neighbors. This algorithm segregates unlabeled data points into well-defined groups. Choosing the number of nearest neighbors i.e. determining the value of k plays a significant role in determining the efficacy of the model[7]. So the selection of k can determine the results of the KNN algorithm by utilizing the data in a good manner. A large k value has benefits which include reducing the variance due to the noisy data; the side effect being developing a bias due to which the learner tends to ignore the smaller patterns which may have useful insights.

Pros:

1. The nature of the algorithm is highly unbiased and there is no prior assumption of the underlying data.
2. KNN algorithm has gained good popularity due to its simplicity and effectiveness.

Cons:

1. Abstraction process is not involved in KNN algorithm.
2. Prediction time is high even though the training time is fast.

3.2. SVM

Support Vector Machine" (SVM) is a well known supervised machine learning algorithm which is simple and can be used for either regression or classification [11]. But, it is commonly used in classification problems. Using this algorithm we are able to plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate[12]. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well.

Pros:

1. It works really well with clear margin of separation
2. It is effective in high dimensional spaces.

Cons:

1. It doesn't perform well, when we have large data set because the required training time is higher
2. It also doesn't perform very well, when the data set has more noise i.e. target classes are overlapping
3. SVM doesn't directly provide probability estimate

3.3. Random Forest

Random Forest is an algorithm which is considered to be a remedy of all data science problems. Random Forest is a flexible machine learning method which is able to perform both classification and regression tasks[18]. It also undertakes dimensional reduction methods, treats missing values, outlier values and other essential steps of data exploration, and does a fairly good job[17]. It is also a type of ensemble learning method in which a group of weak models combine to form a powerful model.

It works in the following manner. We can see that each tree is planted & grown as follows:

1. Here assume number of cases in the training set is N. Then, sample of these N cases is taken at random but with replacement. This sample will be the training set for growing the tree.
2. So if there are X input variables, a number m<X is specified such that at each node, m variables are selected at random out of the X. The best split on these m is used to split the node. The value of m is held constant while we grow the forest.
3. Each tree is grown to the largest extent possible and there is no pruning.
4. Predict new data by aggregating the predictions of the trees (i.e., majority votes for classification, average for regression).

Pros:

1. This algorithm can solve both type of problems i.e. classification and regression and does a decent estimation at both fronts.
2. One of benefits of Random forest which excites me most is, the power of handle large data set with higher dimensionality. Random Forest can handle many of input variables and it will identify most significant variables so it is considered as one of the dimensionality reduction methods. Further, the model outputs Importance of variable, which can be a very handy feature (on some random data set).

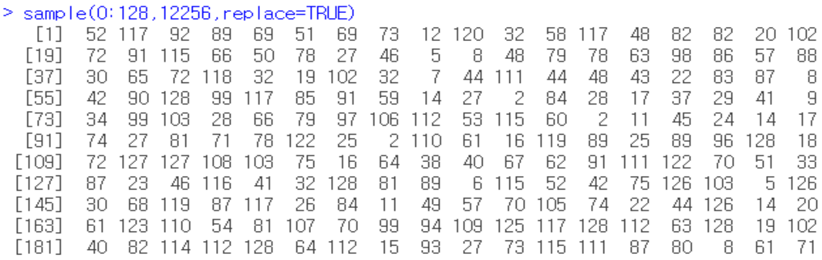
Cons:

1. It surely does a good job at classification but not as good as for regression problem as it does not give precise continuous nature predictions. In case of regression, it doesn't predict beyond the range in the training data, and that they may over-fit data sets that are particularly noisy.

IV. Methodology

Step 1: Data collection

A data set of 12256 observations (records), each record corresponding to a product is used in this work. There are 28 columns/ features in each of the record. These feature are numerical in nature and helps define the properties of the product. Please note that this data set is created solely for the purpose of comparative study and may not represent actual data. The data set has been prepared keeping in mind the results which are generally obtained from organization. All the data manipulations, model training and prediction is done on 3.4.2 version of R Statistical language using R Studio interface v1.0.143. The data set consists of 12256 observations and 28 variables. 27 columns are numerical in nature and one column is categorical .The column "class" is the categorical in nature, It is also the target variable which we want to classify correctly. The columns can represent numerical attributes like Weight, Height, length, width etc. It can be also binary variables like Is_solid, Is_liquid .In real life, there are dozens of important parameters needed to define a product, and it will be beyond the scope of this study to consider and assign all variables. For the purpose of making it simpler, all these features are randomly named and values are also randomly generated using the sample method in R. For the example given below I am creating 12256 values . These values can range from 0 to 128 as mentioned in the below function



```
> sample(0:128,12256,replace=TRUE)
   [1]  52 117  92  89  69  51  69  73  12 120  32  58 117  48  82  82  20 102
  [19]  72  91 115  66  50  78  27  46   5   8  48  79  78  63  98  86  57  88
  [37]  30  65  72 118  32  19 102  32   7  44 111  44  48  43  22  83  87   8
  [55]  42  90 128  99 117  85  91  59  14  27   2  84  28  17  37  29  41   9
  [73]  34  99 103  28  66  79  97 106 112  53 115  60   2  11  45  24  14  17
  [91]  74  27  81  71  78 122  25   2 110  61  16 119  89  25  89  96 128  18
 [109]  72 127 127 108 103  75  16  64  38  40  67  62  91 111 122  70  51  33
 [127]  87  23  46 116  41  32 128  81  89   6 115  52  42  75 126 103   5 126
 [145]  30  68 119  87 117  26  84  11  49  57  70 105  74  22  44 126  14  20
 [163]  61 123 110  54  81 107  70  99  94 109 125 117 128 112  63 128  19 102
 [181]  40  82 114 112 128  64 112  15  93  27  73 115 111  87  80   8  61  71
```

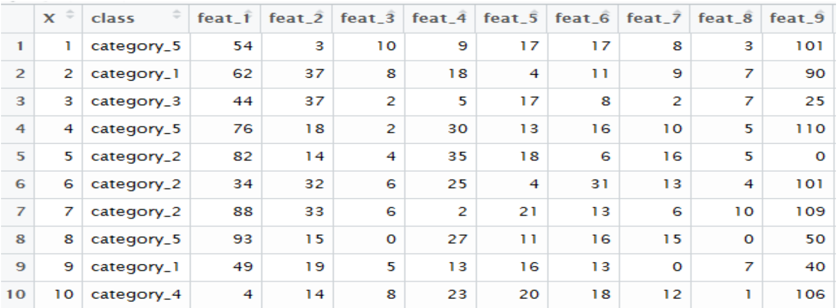
Fig. 1. Sample Data Set

The inner meaning or what a feature represents can be ignored. The first four letters of the "feature" is taken and increased sequentially to arrive at all the column names.

1. ID
2. Class
3. feat_1
4. feat_2
5. feat_3
6. feat_4
7. feat_5
8. feat_6
9. feat_7
10. feat_8
11. feat_9
12. feat_10
13. feat_11
14. feat_12

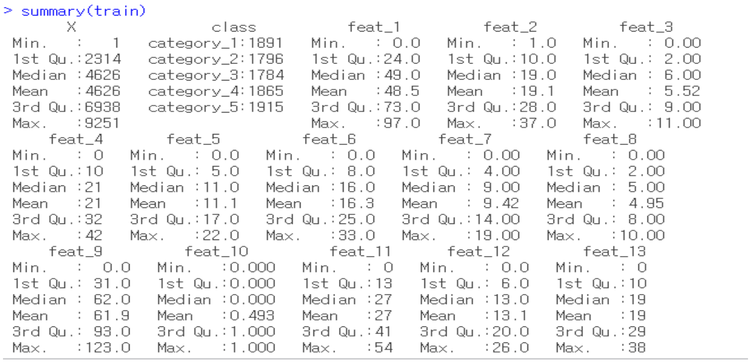
Fig. 2. Column Names

For training the model we divide the data set into 2 – Train data set and test data set. The train data set will have 9251 observations and the test data set will have 3005 observation. The models will be trained on the train data set and final prediction would be done on the test data set. Here's how the train data set looks like:



	X	class	feat_1	feat_2	feat_3	feat_4	feat_5	feat_6	feat_7	feat_8	feat_9
1	1	category_5	54	3	10	9	17	17	8	3	101
2	2	category_1	62	37	8	18	4	11	9	7	90
3	3	category_3	44	37	2	5	17	8	2	7	25
4	4	category_5	76	18	2	30	13	16	10	5	110
5	5	category_2	82	14	4	35	18	6	16	5	0
6	6	category_2	34	32	6	25	4	31	13	4	101
7	7	category_2	88	33	6	2	21	13	6	10	109
8	8	category_5	93	15	0	27	11	16	15	0	50
9	9	category_1	49	19	5	13	16	13	0	7	40
10	10	category_4	4	14	8	23	20	18	12	1	106

Fig. 3. Train Data Set



```
> summary(train)
       X                  class          feat_1          feat_2          feat_3     
 Min.   :   1   category_1:1891   Min.   : 0.0   Min.   : 1.0   Min.   : 0.00  
 1st Qu.:2314   category_2:1796   1st Qu.:24.0   1st Qu.:10.0   1st Qu.: 2.00  
 Median :4626   category_3:1784   Median :49.0   Median :19.0   Median : 6.00  
 Mean   :4626   category_4:1865   Mean   :48.5   Mean   :19.1   Mean   : 5.52  
 3rd Qu.:6938   category_5:1915   3rd Qu.:73.0   3rd Qu.:28.0   3rd Qu.: 9.00  
 Max.   :9251                     Max.   :97.0   Max.   :37.0   Max.   :11.00  
     feat_4        feat_5          feat_6          feat_7           feat_8     
 Min.   : 0   Min.   : 0.0   Min.   : 0.0   Min.   : 0.00   Min.   : 0.00  
 1st Qu.:10   1st Qu.: 5.0   1st Qu.: 8.0   1st Qu.: 4.00   1st Qu.: 2.00  
 Median :21   Median :11.0   Median :16.0   Median : 9.00   Median : 5.00  
 Mean   :21   Mean   :11.1   Mean   :16.3   Mean   : 9.42   Mean   : 4.95  
 3rd Qu.:32   3rd Qu.:17.0   3rd Qu.:25.0   3rd Qu.:14.00   3rd Qu.: 8.00  
 Max.   :42   Max.   :22.0   Max.   :33.0   Max.   :19.00   Max.   :10.00  
     feat_9          feat_10          feat_11        feat_12         feat_13  
 Min.   :  0.0   Min.   :0.000   Min.   : 0   Min.   : 0.0   Min.   : 0  
 1st Qu.: 31.0   1st Qu.:0.000   1st Qu.:13   1st Qu.: 6.0   1st Qu.:10  
 Median : 62.0   Median :0.000   Median :27   Median :13.0   Median :19  
 Mean   : 61.9   Mean   :0.493   Mean   :27   Mean   :13.1   Mean   :19  
 3rd Qu.: 93.0   3rd Qu.:1.000   3rd Qu.:41   3rd Qu.:20.0   3rd Qu.:29  
 Max.   :123.0   Max.   :1.000   Max.   :54   Max.   :26.0   Max.   :38  
```

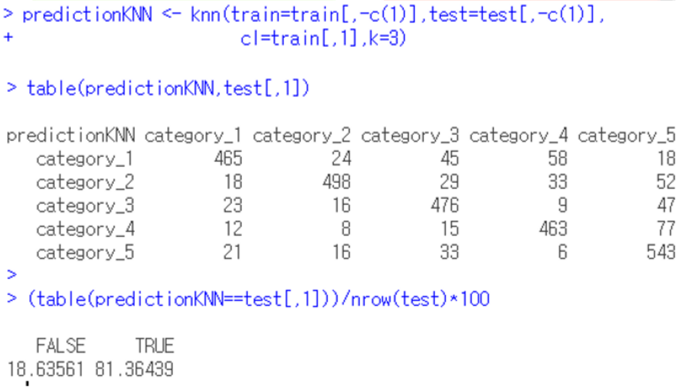
Fig. 4. Summary Train

Step 2: Preparing and exploring the data

We load the train and test data set as follows. The first variable 'id' is unique in nature and can be removed as it does not provide useful information

Step 3 – Training models and predicting the class

a) KNN –The default Euclidean distance measure calculation is used to calculate the distance between various attributes. The parameter K representing the number of neighbors is usually taken as square root of number of parameters So here in our case, it will be sqrt(26)= 5. So we started with K=5. Then we also tested for k=6 and k =3 . The KNN() function from class library was used here. Also ,the output classes are balanced, so a normal accuracy formula is enough to find out accuracy. The highest classification accuracy of 81.36% was achieved with k = 3.

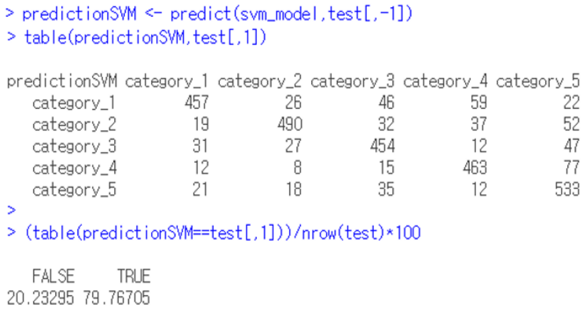


```
> predictionKNN <- knn(train=train[,-c(1)],test=test[,-c(1)],
+                      cl=train[,1],k=3)

> table(predictionKNN,test[,1])

predictionKNN category_1 category_2 category_3 category_4 category_5
   category_1        465         24         45         58         18
   category_2         18        498         29         33         52
   category_3         23         16        476          9         47
   category_4         12          8         15        463         77
   category_5         21         16         33          6        543
>
> (table(predictionKNN==test[,1]))/nrow(test)*100

   FALSE     TRUE 
18.63561 81.36439 
```

Fig. 5. Prediction Accuracy Calculation

b) SVM

The svm() function from e1071 package is used. We set a value of 0.6 for gamma, 0.05 for epsilon, and 0.8 for cost. These parameters helps greatly in training and are explained as below. The gamma is the kernel coefficient for 'rbf', 'poly', and 'sigmoid'. If gamma is 'auto', then 1/n_features will be used instead. C is the cost of constraints violation (default: 1).It is the 'C'-constant of the regularization term in the Lagrange formulation. The prediction accuracy comes out to be 79.77% as shown below.

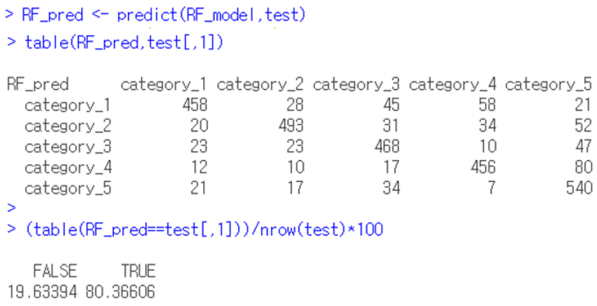


```
> predictionSVM <- predict(svm_model,test[,-1])
> table(predictionSVM,test[,1])

predictionSVM category_1 category_2 category_3 category_4 category_5
   category_1        457         26         46         59         22
   category_2         19        490         32         37         52
   category_3         31         27        454         12         47
   category_4         12          8         15        463         77
   category_5         21         18         35         12        533
>
> (table(predictionSVM==test[,1]))/nrow(test)*100

   FALSE     TRUE
20.23295 79.76705
```

Fig. 6. Prediction Accuracy Calculation

c) Random Forest

We are using the randomForest() function from randomForest() package. We are setting the number of tress to be 500. Ntree represents the number of trees to grow. It should not be set to a small number, to check that every input row gets predicted at least a few times.Training the model as show below. We are getting an accuracy of 80.366%.



```
> RF_pred <- predict(RF_model,test)
> table(RF_pred,test[,1])

RF_pred      category_1 category_2 category_3 category_4 category_5
  category_1        458         28         45         58         21
  category_2         20        493         31         34         52
  category_3         23         23        468         10         47
  category_4         12         10         17        456         80
  category_5         21         17         34          7        540
>
> (table(RF_pred==test[,1]))/nrow(test)*100

   FALSE     TRUE
19.63394 80.36606
```

Fig. 7. Prediction Accuracy Calculation

V. Results

The various prediction accuracy results are put into a tabular format as shown below. The accuracy is low as we had dealt with randomly generated numbers and this could be improved by taking actual working data set from the organizations. As you can see from above table, the KNN algorithm performed better and gave better classification accuracy compared to SVM and Random Forest. In terms of time taken also, the KNN took the least amount of time for This helps in understanding the customer behaviour across various geographies and helps in forecasting and planning and helps sales teams better regarding their sales targets.

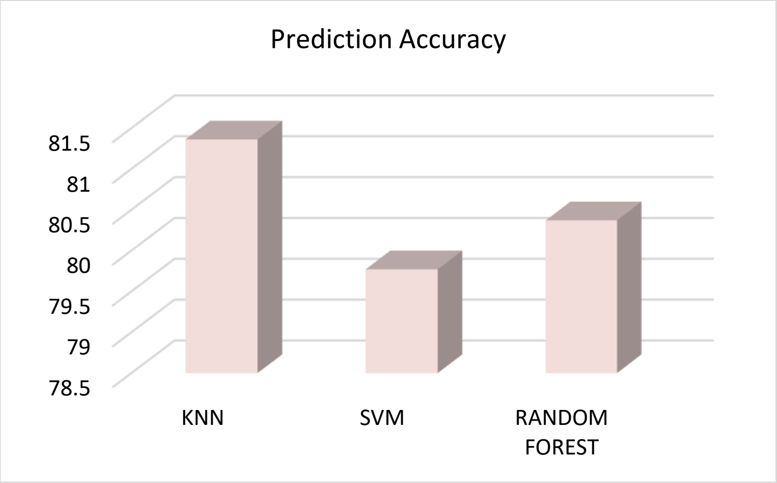


Fig. 8. Maximum accuracy achieved

VI . Conclusion

KNN algorithm performed better and gave better classification accuracy compared to SVM and Random Forest. KNN has taken the least amount of time for training. We have a number of products described by its attributes and so we have achieved our goal to classify the different products into its right category by using 3 famous classification algorithms – KNN, SVM and Random Forest and understood that KNN performed better than SVM and Random Forest.

VII . Acknowledgment

This work was supported by the Department of Computer Science & Engineering,Presidency University. I wish to thank all the faculty members of the department for their suggestions to improve this work.
.

References

[1] Basant Agarwal and Namita Mittal, "Text Classification Using Machine Learning Methods-A Survey," springer, 2014.

[2] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, Kieran Greer, "KNN Model-Based Approach in Classification," springer, 2003.

[3] A. Pradhan, "Support Vector Machine-A Survey," International Journal of Emerging Technology and Advanced Engineering ,vol. 2, no. 8.

[4] Anuradha Patra1, Divakar singh2, "A Survey Report on Text Classification with Different Term Weighing Methods and Comparison between Classification Algorithms," International Journal of Computer Application, vol. 75, no. 7, 2013.

[5] Ramasundram, S.P.Victor, "Text Categorization by Backpropagation Network," International Journal of Computer Applications, vol. 8, no. 6, 2010.

[6] Mahender, Vandana Korde C Namrata, "Text Classification And Classifiers: A Survey," International Journal of Artificial Intelligence & Applications, vol. 3, no. 2, 2012. S.K. Dhurandher, S. Misra, M.S. Obaidat, V. Basal, P. Singh and V. Punia,'An Energy-Efficient On Demand Routing algorithm for Mobile Ad-Hoc Networks', 15 th International conference on Electronics, Circuits and Systems, pp. 958-9618, 2008.

[7] K.T.Khaing and T.T.Naing, "Enhanced Feature Ranking and Selection using Recurisive Featue Elemination and k-Nearest Neighbor Algorithms in SVM for IDS", Internaiton Journal of Network and Mobile Technology(IJNMT), No.1, Vol 1. 2010.

[8] Boser, B. E., I. Guyon, and V. Vapnik (1992). A training algorithm for optimal margin classifiers . In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pages. 144 -152. ACM Press 1992

[9] Chih-Wei Hsu, Chih-Chung Chang, and Chih- Jen Lin. "A Practical Guide to Support Vector Classification" . Deptt of Computer Sci. National Taiwan Uni, Taipei, 106, Taiwan http://www.csie.ntu.edu.tw/~cjlin 2007

[10] C.-W. Hsu and C. J. Lin. A comparison of methods for multi-class support vector machines. IEEE Transactions on Neural Networks, 13(2):415-425, 2002.

[11] . C.J.C. Burges, "Simplified Support Vector Decision Rules," Proc. 13th Int'l Conf. Machine Learning, Morgan Kaufmann, San Francisco, 1996, pp. 71–77.

[12] A. Smola and B. Schölkopf, "From Regularization Operators to Support Vector Kernels," Advances in Neural Information Processing Systems 10, M. Jordan, M. Kearns, and S. Solla, eds., MIT Press, 1998.

[13] F. Girosi, An Equivalence between Sparse Approximation and Support Vector Machines, AI Memo No. 1606, MIT, Cambridge, Mass., 1997.

[14] J. Weston et al., Density Estimation Using Support Vector Machines, Tech. Report CSD-TR-97-23, Royal Holloway, Univ. of S. Berchtold, B. Ertl, D. A. Keim, H.-P. Kriegel, and T. Seidl. Fast nearest neighbour search in high dimensional space. In Proceedings of the Fourteenth International Conference on Data Engineering, ICDE '98, pages 209–218, Washington, DC, USA, 1998. IEEE Computer SocietyLondon, 1997.

[15] Fritz, J. (1975) "Distribution-free exponential error bound for nearest neighbour pattern classification", IEEE Trans. Inform. Theory, 21: 552–557.

[16] Gyorfi, L. (1978) "On the rate of convergence of nearest neighbor rules", IEEE Trans. Inform. Theory, 24: 509–512

[17] Bernard S, Heutte L, Adam S, Towards a Better Understanding of Random Forests Through the Study of Strength and Correlation, ICIC Proceedings of the Intelligent Computing

[18] Kulkarni V Y, Sinha P K, "Random Forest Classifiers: A Survey and Future research Directions", International Journal of Advanced Computing, Vol 36, Issue 1, 1144-1153.