Project: Classification Models Analysis

Sowmya Narra (sxn153730)
Tapasya Gutta (txg150730)
Mahesh Paramati (mxp150830)
Manasa Rao Kondrolla (mxk150930)

Aim:

The aim of the project is to generate models on the provided dataset using the various classifications techniques present in the R repository, analyze the technique and results of the models, and present one which is the most accurate.

Data Set Description:

Dataset contains information of customers of an insurance company. The data consists of 86 variables and includes product usage data and socio-demographic data.The data was supplied by the Dutch data mining company Sentient Machine Research and is based on a real-world business problem.

Training Dataset Size: 5000
Test Dataset Size: 4000
Link to Dataset:
https://archive.ics.uci.edu/ml/datasets/Insurance+Company+Benchmark+%28COIL+2000%29

Data Cleaning:

The dataset contains 86 attributes in the training dataset, 85 attributes in the test dataset and one attribute in the test target dataset. No N/A values are found when the summary of all the datasets was performed. Building a model, any model, with 86 attributes will take up too much of processing time, and frankly, no involves no analysis in this case. Hence, we decided to reduce the number of features that are being used to build the classification models. To help us select the required, important attributes, we use the package “Boruta” in the “CRAN” repository. By just mentioning the classification result attribute along with the list of the other attributes in the dataset, and a few other values, it ran 100 iterations building randomForests models to decide which attributes are important. The result of running this method is attributes divided into three levels, “Confirmed”, “Rejected”, and “Tentative”. We ran this on three different machines, and the results were quite similar with a few discrepancies with the attributes labelled tentative. We reject all attributes that are tagged tentative on the three machines, and upon further scrutiny of the remaining attributes, we decided to discard them too. Finally, the number of attributes being considered is 42 plus the output attribute.

Models:

1. **Logistic Regression**
   a. **Description:**

      Logistic Regression is a type of classification model. In classification models, we attempt to predict the outcome of categorical dependent variables, using one or more independent variables. The independent variables can be either categorical or numerical.Logistic regression is based on the logistic function, which always takes values between 0 and 1.

   b. **Approach:**
      i. A general classification logistic regression model is built on the training dataset using glm() method and the results obtained are as shown in Fig 1.

```
Degrees of Freedom: 5821 Total (i.e. Null);
5779 Residual
Null Deviance:      2635.54
Residual Deviance: 2300.723     AIC: 2386.723
```

Fig 1: Summary of General Logistic Regression Model

      ii. This model is used to classify the 4000 entries of the test dataset, and these predictions are matched with the actual classification present in the file 'tictgts2000.txt'. The above procedure is performed to test the accuracy of the model, and it has produced good results, as can be seen in Fig 2.

```
> mean(lr_pred == tictgts2000)
[1] 0.9405
> table(lr_pred, tictgts2000$X1)

lr_pred    0    1
      0 3759  235
      1    3    3
```

Fig 2: Results and Accuracy of Logistic Regression Model

2. **Naive Bayes**
    a. **Description:**

        Naive Bayes classification is a kind of simple probabilistic classification methods based on Bayes' theorem with the assumption of independence between features. Bayes' theorem can be used to make a prediction based on prior knowledge and current evidence. With accumulating evidence, the prediction is changed. In technical terms, the prediction is the posterior probability that investigators are interested. The prior knowledge is termed prior probability that reflects the most probable guess on the outcome without additional evidence.

    b. **Approach:**
        i. The Naive Bayes model is built on the training dataset using the naivebayes() method in the 'e1071' package.
        ii. To test the accuracy of the built model, the model is used to classify the 4000 entries of the test dataset, and these predictions are matched with the actual classification present in the file 'tictgts2000.txt'. The accuracy of Naive Bayes is shown in Fig 3.

```
> table(nb_pred, testTarget$X1)

nb_pred    0    1
      0 3212  154
      1  550   84
> mean(nb_pred==testTarget$X1)
[1] 0.824
```

Fig 3: Results and Accuracy of the Naive Bayes Model

3. **Decision Tree**
    a. **Description:**

        The classification technique is a systematic approach to building classification models from an input data set.Decision Tree Classifier is a simple and widely used classification method. It applies a straightforward idea to solve the classification problem. Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receives an answer, a follow-up question is asked until a conclusion about the class label of the record is reached.

b. Approach:

i. The Decision Tree model is built on the training dataset using the rpart() of the 'rpart' package.

ii. This decision tree model is used to classify the 4000 entries of the test dataset, and these predictions are matched with the actual classification present in the file 'tictgts2000.txt'. The above procedure is performed to test the accuracy of the model, and it has produced good results, as can be seen in Fig 4.

```
> table(tree_pred, testTarget$X1)

tree_pred    0    1
        0 3762  238
        1    0    0
> mean(tree_pred == testTarget$X1)
[1] 0.9405
```

Fig 4: Results and Accuracy of the Decision Tree Model

4. Support Vector Machine

a. Description:

Support Vector Machine (SVM) is primarily a classier method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables.

b. Approach:

i. The Radial SVM model is built on the training dataset using the svm() of the 'e1071' package.

ii. To test the accuracy of the built model, the model is used to classify the 4000 entries of the test dataset, and these predictions are matched with the actual classification present in the file 'tictgts2000.txt'. The predictions are a broad range of values with no positive result. Moreover, the prediction probabilities are pretty less, with none higher than 0.45024. Hence, we changed the threshold value to 0.22 and obtained the following results, in Fig. 5.

```
> table(svm_rad_pred, testTarget$X1)

svm_rad_pred    0    1
           0 3733  229
           1   29    9
> mean(svm_rad_pred==testTarget$X1)
[1] 0.9355
```

Fig 5: The results and accuracy of Radial SVM

iii. The same procedure is used to build a linear SVM model which performed even worse than Radial SVM, and the highest prediction probability is produced 0.01016375. The wide range of the predictions can be seen in Fig 6. After taking the average of the maximum and minimum prediction probability, and using that as the threshold value, the accuracy is still low. The accuracy results can be seen in Fig 7.

```
> table(svm_lin_prob, testTarget$X1)

svm_lin_prob          0 1
  0.0989016042915217 1 0
  0.0989297816371198 1 0
  0.0989375028444591 1 0
  0.0989771509280025 1 0
  0.0989835852778732 1 0
  0.0993010535978364 1 0
  0.0993378315867893 1 0
  0.099355081907769  1 0
  0.0993831561595278 1 0
  0.0993855375208831 1 0
  0.0994040290861257 1 0
  0.0994064574704033 1 0
  0.0994169433922515 1 0
[                                  -- omitted 3987 rows ]
```

Fig 6: The prediction probabilities of Linear SVM model

```
> min(svm_lin_prob)
[1] 0.0989016
> max(svm_lin_prob)
[1] 0.1016372
> svm_lin_pred[svm_lin_prob>0.1] = 1
> svm_lin_pred[svm_lin_prob<=0.1] = 0
> table(svm_lin_pred, testTarget$X1)

svm_lin_pred    0    1
           0 2541  134
           1 1221  104
> mean(svm_lin_pred==testTarget$X1)
[1] 0.66125
```

Fig 7: The results and accuracy of Linear SVM Model

iv. When we tried to utilize the tuning method, tune() in the same package, the processing time took more than an hour, and we had to shut down the R session forcefully.

5. Neural Network

a. Description:

A Neural Network (NN) is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a vast number of highly interconnected processing elements working in unison to solve specific problems. NN is configured for a particular application, such as pattern recognition or data classification, through a learning process.

b. Approach:

i. The Classification Neural Network model is built on the training dataset using the neuralnet() of the 'neuralnet' package.

ii. Two models are generated, one with six hidden nodes and the other with seven hidden nodes.

iii. These neural network models are used to classify the 4000 entries of the test dataset, and these predictions are matched with the actual classification present in the file 'tictgts2000.txt'. The Mean Square Error of the model with six hidden nodes is calculated, and the result is shown in Fig 6.

```
> nn.MSE = sum((tictgts2000$X1 - nn_pred$net.
result)^2)/nrow(tictgts2000)
> nn.MSE
[1] 0.05914765653
```

Fig 8: MSE of Neural Network Model

iv. Scaling of the datasets is not possible since the training data and testing data are from two different tables, which can lead to different scaling properties. Also, since the maximum value of any of the features is never more than 9, and the minimum value is higher than or equal to 0, scaling the data sets seems redundant.

v. An error is being produced when trying to build a neural network with hidden nodes greater than 7.

6. LDA

a. Description:

Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. LDA works when the measurements made of independent variables for each observation are continuous quantities. When dealing with categorical independent variables, the equivalent technique is discriminant correspondence analysis

b. Approach:

i. The model is built using the function ‘lda’ using the package ‘MASS’. The model is trained on the dataset containing 5822 instances. The variable to be predicted is factorized. The labels for the test data containing 4000 instances are predicted.

ii. The predicted labels are compared against the given labels dataset ‘tictgts2000.txt’. For this, a table is constructed using both the variables.

```
> table(try.class,valueLabels$X1)

try.class    0    1
        0 3753  235
        1    9    3
> mean(try.class==valueLabels$X1)
[1] 0.939
```

Fig 9: Accuracy of model built using LDA

iii. Then mean value is calculated and the accuracy obtained on test data for the model built is shown in Fig. 9.

7. Random Forest

a. Description:

A Random Forest consists of a collection or ensemble of simple tree predictors, each capable of producing a response when presented with a set of predictor values. For classification problems, this response takes the form of a class membership, which associates, or classifies, a set of independent predictor values with one of the categories present in the dependent variable.

b. Approach:

i. The model is built using the function ‘randomForest’ using the package ‘randomForest’.

ii. The model is trained on the initial dataset containing 5822 instances, and the labels for the test data containing 4000 instances are predicted.

iii. The predicted labels are stored in ‘sdata’ and are compared to the ‘tictgts2000.txt’. The results obtained by constructing a table and the accuracy of the model built are shown in Fig. 10.

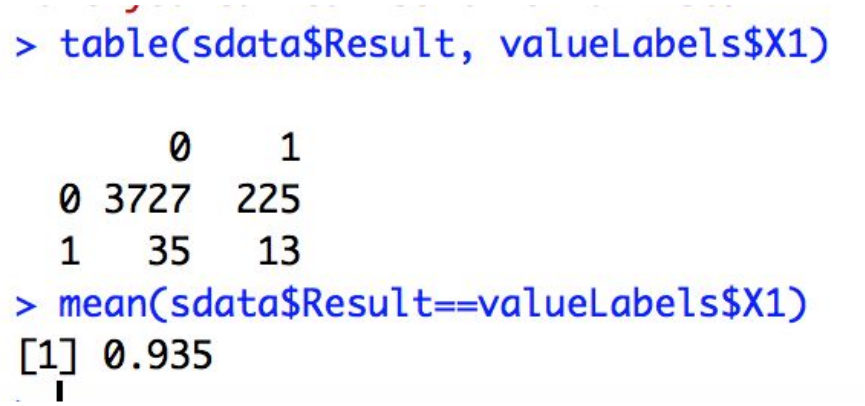

Fig 10: Accuracy of model built using Random Forest

Summary

- Though the number of attributes provided in the dataset is 85, we only considered 42 of them since the rest 43 are not essential for building the models. Of these 43 features, quite a lot of them consist of features which are not classified correctly, i.e., over 5000 entries are classified as the same type.
- We realized that there are a lot of restrictions involved in building a neural network and since it contains hidden layers, the computational time involved is pretty high.
- The accuracy of the models performed is as follows:
    - Logistic Regression: 94.05%
    - Naive Bayes: 82.4%
    - Decision Tree: 94.05%
    - LDA: 93.9%
    - Radial SVM: 93.55%
    - Linear SVM: 66.125%
    - Random Forest: 93.5%

  Looking at these accuracies, we can say that the best model is Logistic Regression and Decision Tree, but all the models except for Linear SVM and Naive Bayes have produced good accuracy, and their accuracy varies mostly in the tenth's region.