CYBR 520 Lab 5: : Dimensionality Reduction through Recursive Feature Elimination - Spambase Dataset (100 points)

Due on Sunday 11/27/2022 @11:59PM

Instructions:

The is a group lab, each team is to submit one submission on eCampus. After the lab is submitted, each group member is to submit a [group member evaluation](https://forms.gle/TvsvercxLsb7sgev9) for each group member (this is worth 15 % of the total grade). We will be using R and Rstudio for this lab. Please read the following the document and provide your answer below each question. Keep the formatting of the document as is.

Dataset: Spambase

R Packages: caret, e1071, randomForest, Hmisc, corrplot, ggcorrplot

Background: The spambase dataset is a classic example dataset for exploring machine learning applications for developing cybersecurity controls (specifically spam filters). The spambase dataset was developed in 1999 by Mark Hopkins, Erik Reeber, George Forman, and Jaap Suermondt of Hewlett-Packard Labs. The spambase dataset is a corpus of spam emails and non-spam (“ham”) emails. 4601 unique instances comprise the dataset. Each instance contains 57 unique attributes which can be used to train machine learning algorithms to automatically classify the data as “ham” or “spam”. It is commonly assumed that this dataset does not contain any error, and all 4601 instances are correctly labeled based upon expert knowledge.

Our objective for this lab will be to try a dimensionality reduction of the spambas dataset and explore its effect on the classification performance. The Recursive Elimination Feature Elimination.

“Recursive Feature Elimination, or RFE Feature Selection, is a feature selection process that reduces a model’s complexity by choosing significant features and removing the weaker ones. The selection process eliminates these less relevant features one by one until it has achieved the optimum number needed to assure peak performance” ~[simplelearn](https://www.simplilearn.com/recursive-feature-elimination-article)

# Get Lab5.r code:

Using GitHub Desktop or PyCharm, pull latest Lab5.r file from the Github repository of the CYBR520 class and the Derby5.csv dataset. The code runs a feature reduction functionality to reduce the number of features used in the DERBY dataset to explore prediction performance (accuracy) enhancement. The idea is to explore whether using a subset of the features (ideally significantly smaller than the original) to obtain a higher accuracy. The code runs for the DERBY dataset to predict whether a file is buggy or not.

# Utilize code for the spambase dataset

Using the provided for Lab5.r on GitHub, modify the code, and utilize it to predict whether a given email is spam or nonspam using the spambase dataset. [15 points]

Run the model and answer the following questions:

1. What does correlationMatrix generate? [5 points)

**As defined by *How to Create a Correlation Matrix in R*. (2022, August 23), “A correlation matrix is a table of correlation coefficients for a set of variables used to determine if a relationship exists between the variables.”**

**The correlation matrix compares each value to every value in the matrix. In this example make is compared to make, make is compared to address, and so on. This relationship is shown with a value of +1 to -1. When make is compared to make it is given as a value of +1, this represents a perfect correlation. The lower the number the less of a relationship between the two values have.**

1. Name three features with the highest positive correlation with the type feature.
   1. How can you tell which features to select? [5 points)

|  |  |
| --- | --- |
| **Feature** | **Value** |
| **type** | **1** |
| **your** | **0.38323382** |
| **num000** | **0.33478704** |
| **remove** | **0.33211742** |

**The highest correlations are self, then “your”, “num000”, and “remove”. The higher the value the stronger the correlation.**

1. How many values are in the correlationMatrix? [5 points)

**There are 3364 values.**

1. How many of those value’s matter? Do we need all of them to? [5 points)

**At face value you can eliminate all correlations of self. This allows you to remove 58 values. Leaving you 3306 values.**

1. What are the highest correlated features? [5 points)

**"num857"**

**"num415"**

**"technology"**

**"labs"**

**"telnet"**

**"num650"**

**"direct"**

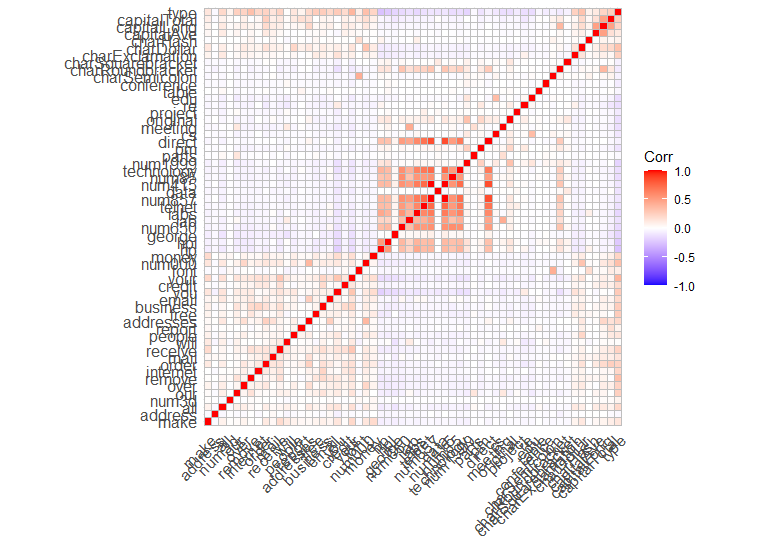
**"hp"**

1. What are the most important features in the dataset? [5 points)

Only 20 most important variables shown (out of 57)

|  |  |
| --- | --- |
| Feature | Overall |
| charExclamation | 0.32453 |
| your | 0.14687 |
| num000 | 0.11208 |
| remove | 0.1103 |
| charDollar | 0.10474 |
| you | 0.07489 |
| free | 0.06928 |
| business | 0.06928 |
| hp | 0.06591 |
| capitalTotal | 0.06208 |
| our | 0.05853 |
| receive | 0.055 |
| hpl | 0.05427 |
| over | 0.0541 |
| order | 0.05362 |
| money | 0.0467 |
| capitalLong | 0.0467 |
| internet | 0.04277 |
| email | 0.0417 |
| all | 0.0388 |

1. Provide the correlation plot, what do you think the dark areas reflect? [5 points)



**The darker areas are the points of higher correlation.**

1. What were the chosen features after running the random forest model? [5 points)

**The top 5 variables (out of 42):**

**capitalLong, report, order, num1999, charHash**

1. Use the code provided in the previous labs and run one classification model (e.g., svm, knn, or decision tree) using the full data, the selected features in step 6, and the selected features in step 8. [15 points)
   1. Report Accuracy and Recall [10 points)
   2. Are there any differences among the three models in terms of accuracy and recall? [10 points]

# Submission:

Submit the source code of your code to your group GitHub Repository, and submit this files along with the answers to eCampus. [10 points]

References

*How to Create a Correlation Matrix in R*. (2022, August 23). Displayr. https://www.displayr.com/how-to-create-a-correlation-matrix-in-r/

*Concept: Correlation Matrix — Dataiku Knowledge Base*. (n.d.). https://knowledge.dataiku.com/latest/courses/statistics/correlation-matrix/correlation-concepts-summary.html