**Computational Intelligence in Business Applications**

**Project 2 Report**

By Shilpa Khichar

Table of Contents

[Problem Description: 3](#_Toc471865482)

[Data Source: 3](#_Toc471865483)

[Technical Specification: 3](#_Toc471865484)

[Models Used: 3](#_Toc471865485)

[Model other then Neral Network: 4](#_Toc471865486)

[For Neural Network we used : 4](#_Toc471865487)

[Parameter Description : 4](#_Toc471865488)

[Neural Network Models for binary classification: 4](#_Toc471865489)

[Results of Neural Network Models for binary classification: 5](#_Toc471865490)

[Neural Network BackPropagation Model for 5 classes: 5](#_Toc471865491)

[Results for Neural Network Model with 5 Classes: 5](#_Toc471865492)

[Results of all Machine Learning Algorithms: 6](#_Toc471865493)

[Neural Network Model Design: 6](#_Toc471865494)

[Conclusion: 7](#_Toc471865495)

[Success And Failures: 8](#_Toc471865496)

[Future Work Suggestion: 8](#_Toc471865497)

# Problem Description:

The goal is to develope a multi class classifier that predicts the labels for the dataset. Observation are Online News Popularity article with 60 features/attributes and the goal is to predict the level of popularity for the article using different Machine Learning Techniques and compare which performs best and why.

The five classes that we have to predict are:

1. Obscure: articles that are shared very few times,
2. Mediocre
3. Popular
4. Super Popular
5. Viral

And we have to measure performance i.e. accuracy of the Models, with which they have predicted classes correctly.

# Data Source:

[**https://inclass.kaggle.com/c/predicting-online-news-popularity**](https://inclass.kaggle.com/c/predicting-online-news-popularity)

# Technical Specification:

* Language : R
* Tool: RStudio
* Library:NeuralNet,Caret

# Models Used:

1. Neural Network
2. Random Forest
3. Desicion tree
4. K-Nearest Neighbours
5. Support Vector Machine
6. Linear Discriminant Analysis
7. GBM :Stochastic Gradient Boosting
8. Multinomial logistic regression

**Model Techniques used:**

### Model other then Neral Network:

* K-Fold ,
* Bootstrap

### For Neural Network we used :

* BackPropagation
* Resilient backpropagation with weight backtracking
* Resilient backpropagation without weight backtracking
* Smallest absolute gradient
* Smallest learningrate

# Parameter Description :

### Neural Network Models for binary classification:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algorithm** | **No. of Hidden Neurons** | **Threshold Value(Stoping Criteria)** | **Learning rate** | **Stepmax**  **(Stoping Criteria)** | **Activation Function** | **Error Function** |
| **BackPropagation** | 3 | 0.01 | 0.01 | 200000 | Logistic | the sum of squared errors(sse) |
| **Resilient backpropagation with weight backtracking** | 3 | 0.01 | 0.01 | 100000 | Logistic | Sse |
| **Resilient backpropagation without weight backtracking** | 3 | 0.01 | 0.01 | 100000 | Logistic | Sse |
| **Smallest absolute gradient** | 3 | 0.01 | 0.01 | 100000 | Logistic | Sse |
| **Smallest learningrate** | 3 | 0.01 | 0.01 | 100000 | Logistic | Sse |

### Results of Neural Network Models for binary classification:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Accuracy** | **Precision** | **Recall** | **F1** | **Kappa** |
| **BackPropagation** | 0.5725094578 | 0.6064257028 | 0.3852040816 | 0.4711388456 | 0.1413938811 |
| **Resilient backpropagation with weight backtracking** | 0.5825977301 | 0.6320346320 | 0.3724489796 | 0.4686998395 | 0.1612219790 |
| **Resilient backpropagation without weight backtracking** | 0.5863808323 | 0.6428571429 | 0.3673469388 | 0.4675324675 | 0.1686568309 |
| **Smallest absolute gradient** | 0.5813366961 | 0.6282051282 | 0.3750000000 | 0.4696485623 | 0.1587604885 |
| **Smallest learningrate** | 0.5775535939 | 0.6540540541 | 0.3086734694 | 0.4194107452 | 0.1499611226 |

### Neural Network BackPropagation Model for 5 classes:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algorithm** | **No. of Hidden Neurons** | **Threshold Value(Stoping Criteria)** | **Learning rate** | **Stepmax**  **(Stoping Criteria)** | **Activation Function** | **Error Function** |
| BackPropagation | 5 | 0.01 | 0.01 | 300000 | Logistic | the sum of squared errors(sse) |
|  | 7 | 0.01 | 0.01 | 300000 | Logistic | Sse |
|  | 10 | 0.01 | 0.01 | 300000 | Logistic | Sse |
|  | 12 | 0.01 | 0.01 | 300000 | Logistic | Sse |
|  | 15 | 0.01 | 0.01 | 300000 | Logistic | Sse |

### Results for Neural Network Model with 5 Classes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **No. of Hidden Layers** | **No. of Hidden Neurons** | **Accuracy** |
| BackPropagation | 1 | 5 | 0.1658291457 |
|  | 1 | 7 | 0.1658291457 |
|  | 1 | 10 | 0.2060301508 |
|  | 1 | 12 | 0.1658291457 |
|  | 1 | 15 | 0.1658291457 |

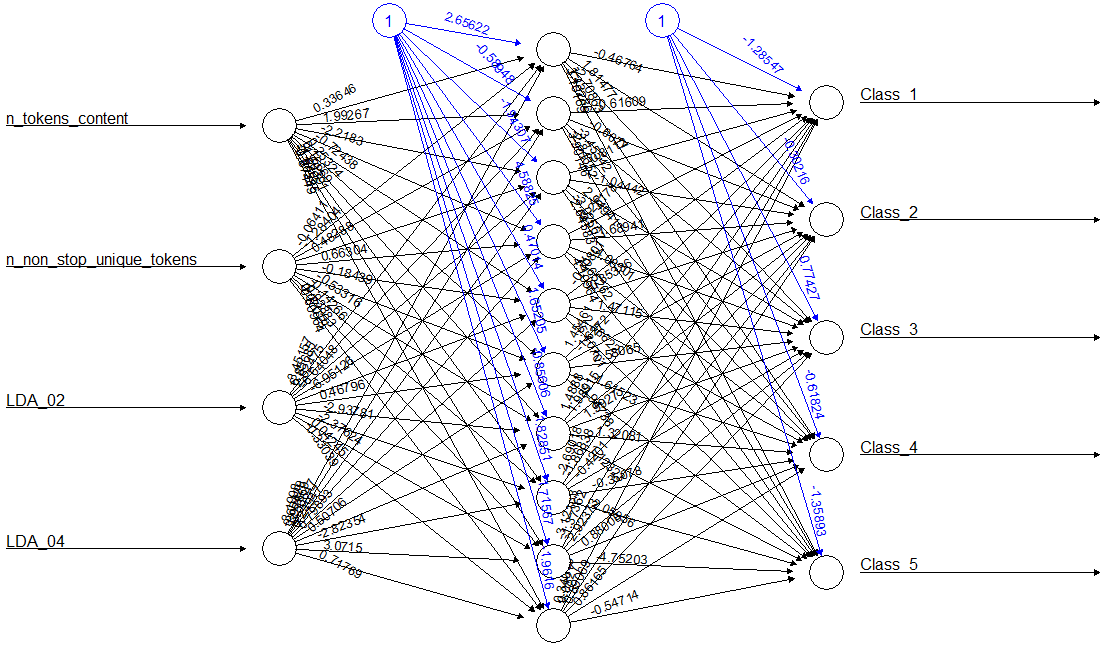
### Results of all Machine Learning Algorithms:

|  |  |  |
| --- | --- | --- |
| **Model** | **Technique** | **Accuracy** |
| **Neural Network** | Backpropagation | 0. 2060301 |
| **Random Forest** | Simple | 0. 4380040 |
|  | K-Fold | 0. 4475806 |
|  | BootStrap | 0. 4501008 |
| **Decision Tree** | Simple | 0. 3996976 |
|  | K-Fold | 0. 4027218 |
|  | BootStrap | 0. 4027218 |
| **Support Vector Machine** | Simple | 0. 4047379 |
|  | K-Fold | 0. 4072581 |
|  | BootStrap | 0. 4072581 |
| **K- Nearest Neighbours** | K-Fold | 0. 3971774 |
|  | BootStrap | 0. 4092742 |
| **Linear Discriminant Analysis** | Simple | 0. 4148185 |
| **Stochastic Gradient Boosting** | Simple | 0.4213710 |
| **Multinomial Logistic Regression** | Simple | 0.4092742 |

# Neural Network Model Design:

The below Neural Network Model Design shows 10 hidden neuron being used for 5 class classification, with their model generated Weights(Black) and Bais(Blue) .And Input Being :

* n\_tokens\_content: Number of words in the content
* n\_non\_stop\_unique\_tokens: Rate of unique non-stop words in the content
* LDA\_02: Closeness to LDA topic 2
* LDA\_04: Closeness to LDA topic 4



# Conclusion:

From the result matrix we can see that rsileint back propagation Neural network model performs best on our dataset. as RPROP is A Fast Adaptive Learning Algorithm.It overcome the inherent disadvantages of the pure gradient-descent technique of the original backpropagation procedure, RPROP performs an adaptation of the weight update-values according to the behaviour of the errorfunction.Hence performs better.

And for the 5 class classification we see that the Random Forest performs best. Question is why Neural Network did not perform as good as Random Forest, because the model fitting process is very different.

Neural network are more flexible and is a supervised learning algorithm,as it can learn to recognize any kind of patterns that are hardly seen by other ML algorithms. Even in case of simple patterns, it is expected that neural networks would adjust its weights in such a way that it would mimic the behaviour of other algorithms which would detect those patterns.This flexibility makes them harder to fit, easier for them to get trapped in local minima, and more likely to overfit to training data (as seen in our case).

# Success And Failures:

For better performance of Neural Network Models :

* Use more number of neurons if more number of classes are to be predicted,
* Normalize your data
* Use more training data then testing data(90:10).

# Future Work Suggestion:

As is seen from the result, no algorithm could reach even upto 50% accuracy .To improve accuracy, there is little room in model selection but much room in feature selection. In the preprocess phase we had 59 features which were extracted from the news articles, from which we extracted 5 features using Feature selection techniques(Random Forest Important attributes and SBF- Selection By Filter technique) and our later work is based on these features.However, the content of news articles hasn’t been fully explored. Some features are related to the content, such as LDA topics, which are convenient to use for learning, but reflect only a small portion of information about the content.

In the future, we could directly treat all the words in an article as additional features, and then apply machine learning algorithms. In this way, what the article really talks about is taken into account, and this approach should improve the accuracy of prediction if combined with our current work..