

PREDICTION & CLASSIFICATION OF CLOSED QUESTIONS ON STACK OVERFLOW

UNDER THE GUIDANCE OF DR. AJAY KUMAR

PRESENTED BY :

ANSHUL VYAS (2013IPG-024)
KANISHK AGRAWAL (2013IPG-057)
SHREYA SAHU (2013IPG-102)

ABV-INDIAN INSTITUTE OF INFORMATION
TECHNOLOGY & MANAGEMENT

- 1 Contents
- 2 Introduction
- 3 Motivation
- 4 Problem Statement & Description
- 5 Current Fallouts
- 6 Bridging the Gap
- 7 Literature Review
- 8 Solution Approach
- 9 Results
- 10 Results
- 11 Results
- 12 Results
- 13 Results
- 14 Conclusions
- 15 References

INTRODUCTION

- When it comes to knowledge sharing Stack Exchange has emerged out to be most prominent in the past decade which houses more than 160 domains in various fields as science, programming etc.
- It is a community of 4.7 million programmers Stack Overflow (SO) helping each other and share their knowledge to help and guide more than 8.5 millions Stack Exchange data (2016)visitors per day.
- Users of SO can post their query, share screenshots, post solutions, comment and evaluate their overall rating as well.

- There are two broad categories of questions on SO namely:

Open questions are accessible to discussions and may have the further scope of improvement.

Closed questions are monitored by the community and those which don't satisfy their norms are marked as Closed

Questions which don't satisfy their norms are marked as Closed Correa and Sureka (2013). Answering the closed question is not permitted, though editing can be done.

- *What parameters decide whether a question is closed or not?*

Closed questions are judged on following parameters.

- **Off Topic**
- **Not Constructive**
- **Too Localized**
- **Not a Real Question (NRQ)**
- **Duplicate**

MOTIVATION

- **Automatic classification** will enhance quality of questions on the forum. would not only reduce stress on community moderators, also would drive a better user experience.
- **Reduced stress** on community moderators as every minute five questions are posted on SO and they need to be reviewed.
- **Better user experience** by improving the search results.

PROBLEM STATEMENT & DESCRIPTION

- To propose a proper filter to eliminate poor quality questions.
- To implement neural networks & Support Vector Machine for better classification.

CURRENT FALLOUTS

- Textual features were not taken into consideration, or were provided just a minor weightage.
- None of the previous works indicated the importance of neural networks in classifying data of such high volume, noise and irregularity.

BRIDGING THE GAP

- Our work takes into account numerous features extracted from text like codeblock count, punctuation ratios, final thanks etc.
- With our work, we put forward a fast, reliable and more efficient way of approaching this problem.

LITERATURE REVIEW

- We deal with mainly two literatures to predict closed questions on Stack Overflow and their contribution can be summarised as follows:
 - Correa ¹ propose an one against all classification using the original six features to train their classification model.
 - Mukerjee ² propose an improvement in the classification model by taking into account eighteen features

1

2

NOTEWORTHY CONTRIBUTION

- Kaggle dataset ³ helped us in moving in the right direction towards finding the right features.
- The features used in this work are a compilation of all the valid features used in researches before and a few more based on the gaps in the papers.

³Kaggle dataset: 2012, <https://github.com/sasd/kaggle-stackoverflow2012>. [Online; accessed 25-September-2016]

SOLUTION APPROACH

The solution methodology has been divided into following 3 approaches:

- **DATA GATHERING**

- Kaggle dataset

- **DATA PREPROCESSING**

- Feature Extraction
- Normalization

- **TRAINING THE MODELS**

- SVM
- ANN

TRAINING THE MODELS

- **SVM**

This project uses Scikit learn to implement SVM with LibLinear library with LinearSVC model.

This class supports both dense and sparse input and the multiclass support is handled according to a one-vs-the-rest classification scheme.

Table: Customisable parameters of SVM

Parameters	Value
C	1.0
Dual	False
Multiclass	ovr

TRAINING THE MODELS

- **ANN**

This project uses TensorFlow ⁴ library for implementing ANN.

The library is built for implementing ANN and it is more than a black box implementation.

Table: Table depicting the architecture of ANN

Layer	Number of neurons	Activation function
Input	35	-
Hidden	10	Rectified Linear Unit (ReLu)
Output	5	Softmax

⁴<https://www.tensorflow.org/>

Training algorithm implemented is Back Propagation Algorithm ⁵

- **Log loss (or cross-entropy)** The difference between the output value and the desired value is known as error signal.
- **Adaptive Moment Estimation (Adam's) Algorithm** computes adaptive learning rates for each parameter and is a better variant of gradient descent. The learning rate is set to 0.01
- **Regularization** is important to introduce to avoid overfitting of the model during prediction occurs when a model is excessively complex.

Regularization parameter = 0.0001

Regularization function = L2loss

ACCURACY

- **SVM**

The accuracy obtained by SVM is 97.91%

- **ANN**

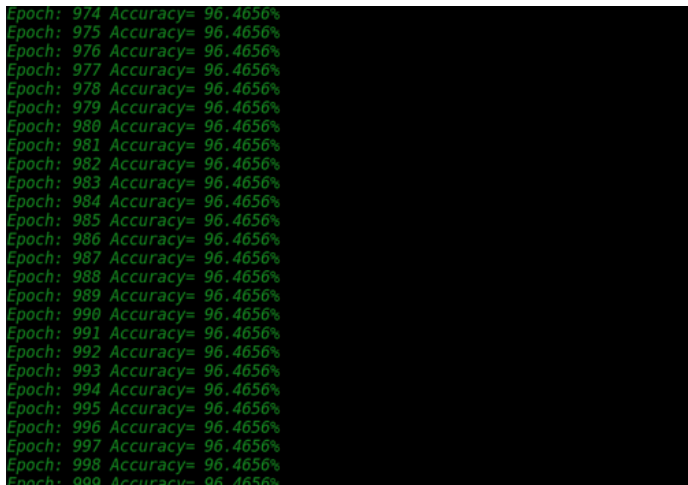
The accuracy obtained by ANN is 96.46%

ACCURACY

```
shreya@shreya-Inspiron-3537: /media/shreya/New Volume1/btp/code/ourCode$ python  
trainSVM.py feature_scaled.csv result.csv  
reading input feature vector...  
read input feature vector...  
converting dict to numpy array..  
starting to train on linear svm....  
linear svm trained...  
predicting output done.. mapping it to values now..  
saving to result file now.. accuracy now..  
0.979109595673
```

Figure: Screenshot for accuracy of SVM

ACCURACY



```
Epoch: 974 Accuracy= 96.4656%  
Epoch: 975 Accuracy= 96.4656%  
Epoch: 976 Accuracy= 96.4656%  
Epoch: 977 Accuracy= 96.4656%  
Epoch: 978 Accuracy= 96.4656%  
Epoch: 979 Accuracy= 96.4656%  
Epoch: 980 Accuracy= 96.4656%  
Epoch: 981 Accuracy= 96.4656%  
Epoch: 982 Accuracy= 96.4656%  
Epoch: 983 Accuracy= 96.4656%  
Epoch: 984 Accuracy= 96.4656%  
Epoch: 985 Accuracy= 96.4656%  
Epoch: 986 Accuracy= 96.4656%  
Epoch: 987 Accuracy= 96.4656%  
Epoch: 988 Accuracy= 96.4656%  
Epoch: 989 Accuracy= 96.4656%  
Epoch: 990 Accuracy= 96.4656%  
Epoch: 991 Accuracy= 96.4656%  
Epoch: 992 Accuracy= 96.4656%  
Epoch: 993 Accuracy= 96.4656%  
Epoch: 994 Accuracy= 96.4656%  
Epoch: 995 Accuracy= 96.4656%  
Epoch: 996 Accuracy= 96.4656%  
Epoch: 997 Accuracy= 96.4656%  
Epoch: 998 Accuracy= 96.4656%  
Epoch: 999 Accuracy= 96.4656%
```

Figure: Screenshot for accuracy of ANN

LABEL PREDICTION: SVM

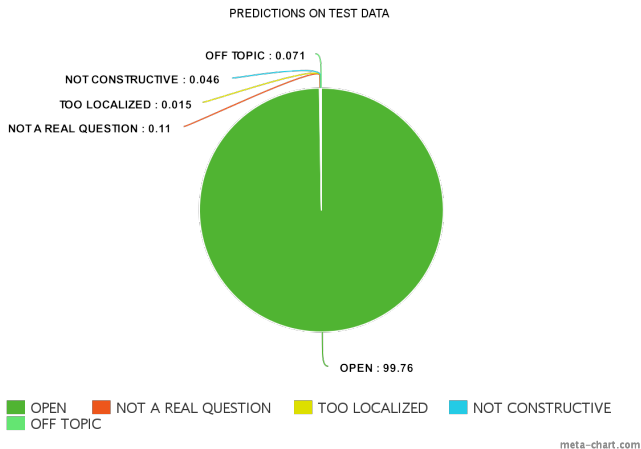


Figure: Prediction using SVM

LABEL PREDICTION: ANN

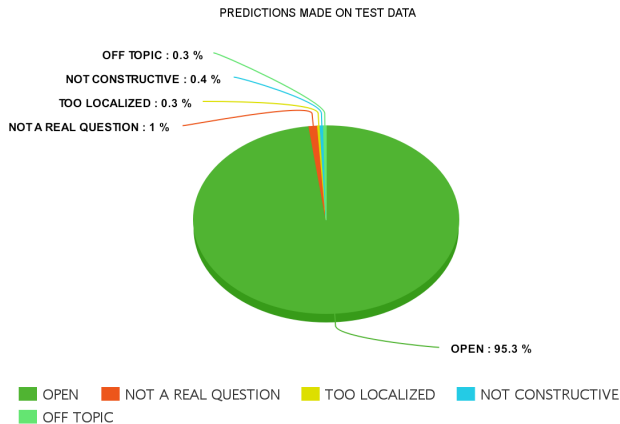






Figure: Prediction using ANN

CONCLUSIONS

- For a large data, ANN is better approach as it predicts the data more accurately.
- Though SVM gives more accuracy, but it overfits towards the skewed data.
- Hence, ANN is a better approach to classify the questions.

REFERENCES

-  Moore, A. W.: 2001, Support vector machines, Tutorial. School of Computer Science of the Carnegie Mellon University. Available at <http://www.cs.cmu.edu/~awm/tutorials>. [Accessed August 16, 2009]
-  Hirose, Y., Yamashita, K. and Hijiya, S.: 1991, Back-propagation algorithm which varies the number of hidden units, Neural Networks 4(1), 6166.
-  Dunne, R. A. and Campbell, N. A.: 1997, On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function, Proc. 8th Aust. Conf. on the Neural Networks, Melbourne, 181, Vol. 185, Citeseer.
-  Kingma, D. and Ba, J.: 2014, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.