INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 6.078
(Volume 6, Issue 4)
Available online at: https://www.ijariit.com

Stacklite – Stack overflow tag prediction

Gada Shrenika
shrenikagada@gmail.com
BVRIT Hyderabad College of Engineering for Women, Hyderabad, Telangana

Ch. Ramya
ramyachowdarychilukuri@gmail.com
BVRIT Hyderabad College of Engineering for Women, Hyderabad, Telangana

A. Rajashekar Reddy
raja.akuthota@gmail.com
BVRIT Hyderabad College of Engineering for Women, Hyderabad, Telangana

ABSTRACT

Nowadays data plays a major role in every aspect of our life. The past data that is available can be used for analysis and to predict the future. For websites that are based on learning, the old data which the users are posting and tagging can be used to analyze and predict what new implementations can be done to increase the user experience. Similarly, Stack Overflow is the largest learning forum that is used by most of the developers to learn and share their programming knowledge. To post a question, users need to enter the tags related to the question manually. Here we are analyzing the past available data to predict the tags automatically based on the question a user enters which increases the enhancement of user experience.

Keywords: *Machine Learning, Multi-class classification, Text Classification, Binary Relevance, Classifier Chains, Label Powerset, K-Nearest Neighbours, Logistic Regression, Tags, TF-IDF vectorizer, Countvectorizer*

1. INTRODUCTION

The world is completely turning into a technological world. It is being transformed into a place where you can predict the future easily based on the past. Let the prediction be related to weather, health, technology, behaviour, etc., but when the term prediction comes into picture then comes the term "DATA". Now-a-days data plays a major role in every aspect of our life.The past data that is available can be used to perform analysis and predict the future. It helps us understand things well, it helps us to group data based on key terms which helps our further analysis easier. Stack Overflow is the largest and widely used online forum for developers to learn and also to share their programming knowledge which was launched on 15th September, 2008. It serves as a platform for users to ask and answer questions, to upvote or downvote questions and answers. Also the user will have privileges to comment and edit posts. The questions on Stack Overflow consist of a wide range of topics and are categorized based on appropriate tags. These tags are entered manually by the user that relates the question they wanted to post. This categorization of data helps users who answer questions posted, to painlessly search for the questions based on the topic they are interested to answer. The information provided to the users must be relevant to their search. With this came the concept of tagging. So whenever a user searches for questions based on a tag, the questions that are displayed must be relevant to their search. But while posting a question the beginners who are new to programming and also the new users of Stack Overflow may not know what to enter as tags. So they may enter tags that are unrelated to the question they post. If such a thing happens, the users search for the questions based on tag, they may get unrelated questions which decreases the user experience. As for any application or forum or platform, the enhancement of user experience is more important.

As in Stack Overflow the tags are manually entered by the user, here we are trying to auto-suggest tags to the users based on the questions they post. This auto-suggestion helps the new users or beginners to the programming users to easily add tags related to their questions. This further helps the other users who answer the questions posted get relevant questions based on their search.

2. ARCHITECTURE AND DATA PRE-PROCESSING

Here, we are using the dataset from kaggle and the dataset consists of all the relevant information like title, body(description), tags, etc., that is needed for the prediction.

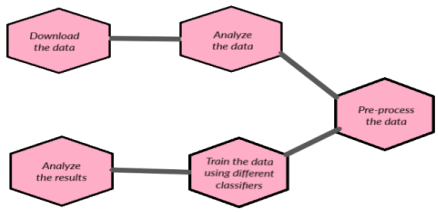

Fig. 1: System Architecture

As the data is vast, we have considered 100000 records for the analysis and there were 7755 unique tags.

```
Number of questions : 100000
Number of unique tags : 7755
```

Fig. 2: Number of unique tags

3	30891
4	24361
2	21430
5	17496
1	5822

Fig. 3: Questions count based on number of tags

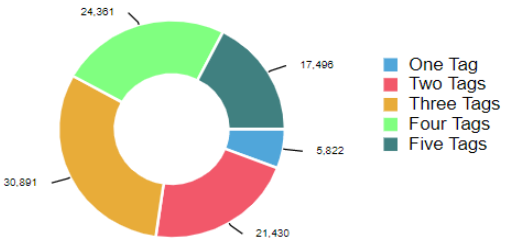

Fig. 4: Visualized view of questions count based on number of tags

Then we have pre-processed the data by taking a new column which is the combination of the title and body of that particular row. On the new column we have performed stemming that removes or adds suffixes or prefixes of all the relevant words like "chocolates", "chocolatey", "choco" are replaced with "chocolate" and then removed stopwords from that column by using the SnowballStemmer method for stemming and Stopwords method to remove stopwords that are available in the nltk library.

3. CLASSIFIERS

Then we have used TF-IDF Vectorizer to find the maximum repeated words by calculating their offset values. Here the TF-IDF value is the product of Term Frequency(TF) value and Inverse Document Frequency(IDF) value. Term frequency is the number of times a word appears in a document divided by the total number of words in the document. Every document has its own term frequency..

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

Inverse document frequency is the log of the number of documents divided by the number of documents that contain the word. Inverse document frequency determines the weight of rare words across all documents in the corpus.

$$idf(w) = log(\frac{N}{df_t})$$

Finally the obtained term frequency value and the inverse document frequency value are multiplied to obtain the tf-idf value. This data is further used for the analyses.

Here, as we can add a minimum one tag and maximum of five tags, the problem here is related to multi-class classification. We have trained the model using problem transformation techniques and adapted algorithms.

3.1 Problem Transformation Techniques

As the name itself says, here we transform our multi-class classification into single- class classification. We have carried out this method in three different ways

3.1.1 Binary Relevance: This is a simple technique which considers every label as a separate single class classification problem. This converts the multi-label classification task into a number of independent learning tasks. The drawback with this is that it does not consider label correlation as it treats every target variable independently.

3.1.2 Classifier Chains: Here it considers the policy of chains. Initially the first classifier is trained based on the input data, then the next classifier is trained based on this input space of the first classifier and it continues.It combines the computational efficiency of binary relevance method. It considers label correlation irrespective of binary relevance.

3.1.3 Label Powerset: Label powerset transforms the problem into a multi-class problem with only one classifier problem based on the unique label combinations available in the data. If the training data is more and there are more number of unique labels in the data, the complexity in the model increases, decreasing the accuracy.

3.2 Adapted Algorithms

Adapted algorithms directly perform the multi-class classification by changing its behaviour at the time of run based on the information available rather than transforming the problem into different subsets. We have implemented it in two different ways.

3.2.1 K-Nearest Neighbours(KNN): KNN is an algorithm used in machine learning for regression and also for classification.It uses the available data and classifies the new data points based on the similarity measures which uses a distance function. A relationship is captured between different samples available. The relation is found using the distance function and we used the Euclidean Distance function to find the relation.

$$d(x, x') = \sqrt{(x_1 - x_1')^2 + \ldots + (x_n - x_n')^2}$$

It mainly performs 2 major steps.

- It runs through the whole dataset computing distance(d) between a point(x) and each training observation.
- It then estimates the conditional probability for each class, that is, the fraction of points in A with that given class label.

$$P(y = j|X = x) = \frac{1}{K} \sum_{i \in A} I(y^{(i)} = j)$$

The cost of calculating the distance between the new point and each existing point is huge which degrades the performance of the algorithm.

3.2.2 Logistic Regression: Logistic regression in its basic form is a statistical model which uses a logistic function to model a binary dependent value. It cannot handle target vectors with more than two classes. Hence multi-class classification using logistic regression can be done using one-vs-rest classification. If we have a classification problem and there are N distinct classes, one-vs-all or one-vs-rest classification is a method which involves training N distinct binary classifiers. Then those N classifiers are collectively used for multi-class classification.

4. RESULT

We trained our model and predicted the tags using all the above mentioned algorithms, i.e., binary relevance, classifier

chains, label powerset, k-nearest neighbours and logistic regression. Out of all the algorithms we used, the results of logistic regression were more accurate compared to the remaining algorithms and then comes the k-nearest neighbours algorithm. The least accuracy was found using binary relevance. We can observe the variations in the accuracy from the below graph.

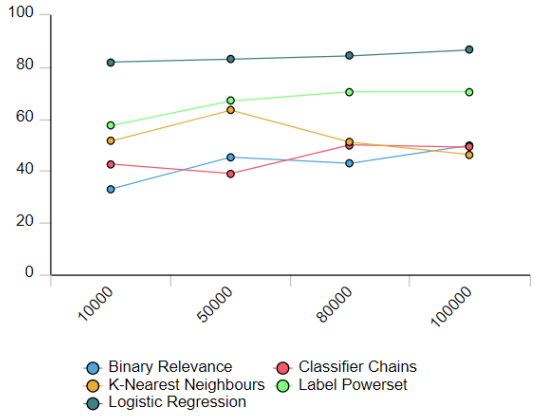

Fig. 5: Comparison of accuracy

5. CONCLUSION

We have implemented different classifiers that will be able to predict the tags based on the question entered by the user. The classification techniques we used are binary relevance, classifier chains, label powerset, k-nearest neighbours and logistic regression. We have performed the analysis by training the model by varying the number of questions for every model. As our future work, we planned to perform our analysis using SVM which uses a pipeline structure of vectorization, transformation and classification, to find whether it gives more accurate results.

6. REFERENCES

[1] Steven Bird, Ewan Klein & Edward Loper. Natural Language Processing.
[2] Sanjay Sood, Sara Owsley, Kristian J Hammond, and Larry Birnbaum. 2007. Tag assist: Automatic tag suggestion for blog posts. In ICWSM.
[3] David G. Kleinbaum, Mitchel Klein (2002). Logistic Regression, A Self-Learning Text, Third Edition.
[4] Clayton Stanley and Michael D Byrne. 2013. Predicting tags for stack-overflow posts. In Proceedings of ICCM 2013
[5] Tania Saini and Sachin Tripati. 2018. Predicting Tags for Stack Overflow Questions Using Different Classifiers. 4th Int'l Conf. on Recent Advances in Information Technology | RAIT-2018
[6] https://www.youtube.com/watch?v=yIYKR4sgzI8
[7] https://scikit-learn.org/stable/modules/sgd.html
[8] https://en.wikipedia.org/wiki/Multi-label_classification

