Auto Tagging of StackOverflow Questions

Team Members:

Aditya Pujari (11491374)
Gagan Sai Ram Anvesh Achanta (11447940)
Sai Tarun Gunda (11516657)
Nithish Reddy Boddhi Reddy (11555383)

Goals and Objectives:

Goals:

We want to develop a classification model for categorizing the tags available. Based on the Body & Title of the question we will be building two vectors and combining them. Using the vectors generated to train the classifier model, which will suggest the top 25 Tags which match the Stack Overflow question. We want to use Label Powerset & Classifier Chains and select the model which gives the best accuracy.

Motivation:

In today's Programming world, one of the best ways to discuss programming related errors/issues would be Stack Overflow. If a user posts his question in Stack Overflow, he must manually tag the related topic he is looking for to get the issue resolved/ redirected to the correct community of people who are looking/ already working in that field. For example, if we are looking for some machine learning topic and have a query related to some training of SVM model after posting the question we have to tag it as Machine learning, so that it reaches the appropriate set of people. Instead of tagging Machine learning, if we tag Accounting it would be helpful to resolve the problem.

Significance:

If we can develop a model where it automatically tags the question with its respective task i.e., it automatically tags a machine learning problem to a machine learning tag and accounting questions to an accounting tag and so on. By doing so the questions we will get answers/solutions quickly as it reaches the right set of people.

Objective:

Objectives of our project are:

- Selection of Dataset which has a versatile type of questions and Tags.
- Tags Collection: Collection of most relevant Tags from each area.
- Data preprocessing: Preprocessing the questions body and title like removing StopWords, Translating all the text to lowercase, Stemming and lemmatization.
- Vectorization: Creating the Vectors which will be used to train & test the classification model.
- Training: By using the vectors generated we will be training the model to predict the Top 25 tags which best matches the questions.
- Evaluating the best model based on the test data.
- Deploying the Code as Web applications.

Features:

We will be extracting 3 main features from the questions which gives us the overview of the whole scenarios like What the question is dealing with, What Tags need to be assigned and Title of the question. These are the 3 factors which decide which questions belong to an area.

Related Work:

Knowledge and opinion sharing has been widely associated with a lot of web pages on the internet. People post lots of questions, movie reviews and other information on the internet. However, segregation of this information with appropriate tags has become a complicated task. In manual tagging most of the users might not use relevant tags with which the reach of the information would get affected. In Spite of solving such kinds of problems, many researches were conducted to design a model which could classify content. This multilabel classification is employed in many contexts like telling if a movie review is positive or negative. In the article “Deep dive into multi-label classification..! (With a detailed Case Study)” published on Towards Science by Kartik Nooney, the problem is solved in three steps. Initially, in the data preprocessing stage data cleaning is achieved using NLP techniques such as Stemming, Lemmatization and Removal of HTML tags. In the second stage, the cleaned data is divided into Test and Train datasets. In the final stage, classification is achieved using the Label Powerset Classifier. Nonetheless, in order to get higher accuracy there might be a requirement of further implementation of cleaning techniques and classification models which we’d be using in this project.

Dataset:

The dataset we are using consists of total 6 columns

- OwnerUserId: This column gives the information about the user who has created the Stack overflow question. There are a total 875317 unique rows. This column does not give any information about which tags need to assign So we Can neglect this column while training our model.
- CreationDate: This will give information about the creation date when the Stackoverflow question is created. Even this column does not give us information about which Tags need to be tagged. Total unique Rows.
- ClosedDate: This will give the information when the question is resolved which does not add any value to our analysis.
- Score: This column gives the no of people who have upvoted this question i.e., total no of people who have the related question.
- Title: This column gives information about the topic which we are dealing with. This is one of the important features to be considered. Total we have 1263995 unique columns.
- Body: Body of question that is where we get the total information about the question that needs to be dealt with in the area. So, this would be an important feature to work. Combination of Body and Title gives us more accurate information about the question so that we can assign the related Tags to the question.

Detail design of Features:

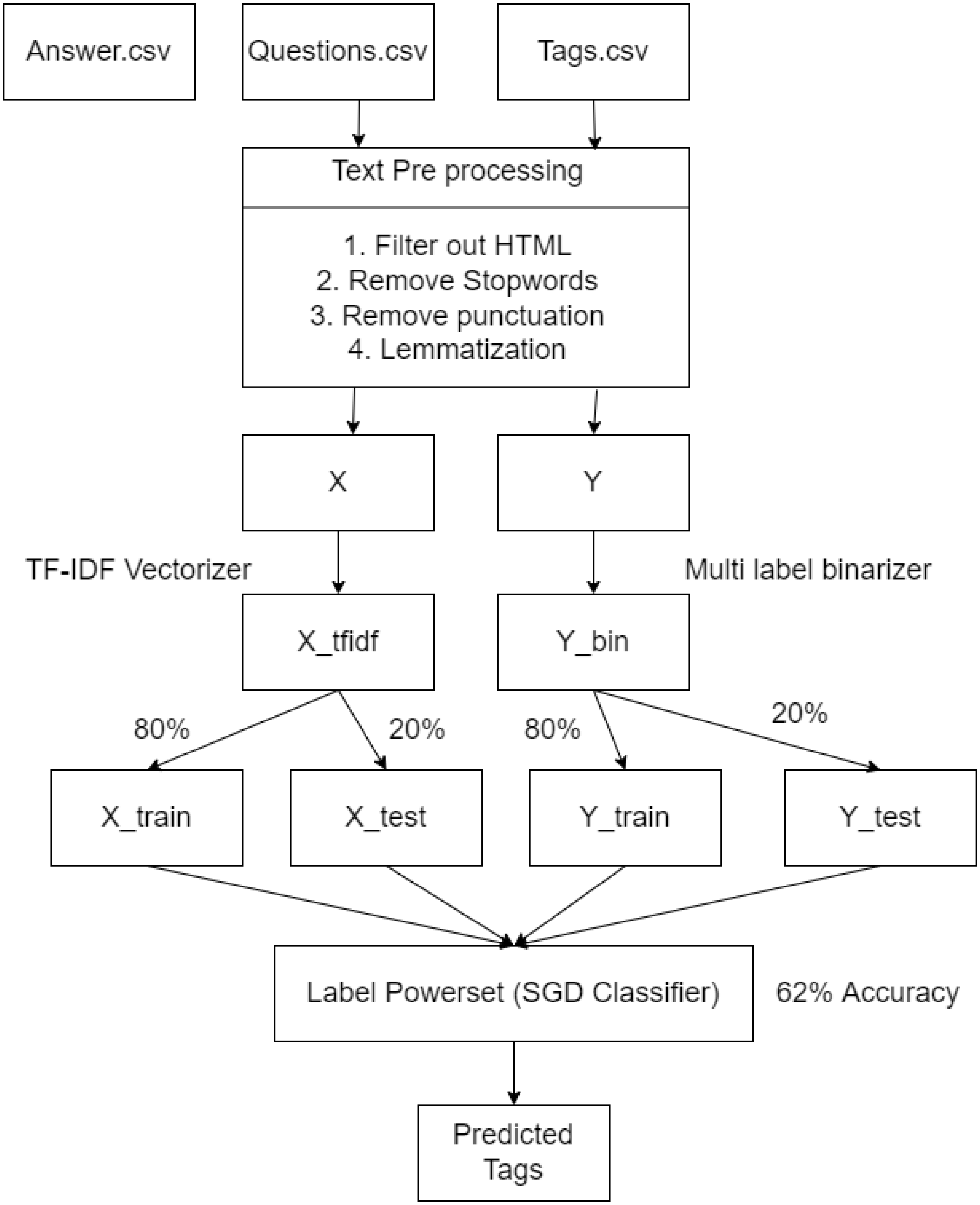
- **Title:** Title of the question contains or gives the basic information of the question which it is dealing with. So, it is necessary to have this information.
- **Body:** Body of the question contains all the information which describes which topics belong to it so that we can assign the relevant tags to it.
- **Tags:** Tags are the main features i.e.,Tags need to be assigned to the questions that we are working on.
- Before fetching the above features, we must preprocess the data like removing the html tags and unnecessary text. In order to do that we are using Beautiful Soup Library

Analysis:

The data set contains three files which are Questions.csv, Answers.csv and Tags.csv. We have over 1264216 questions in the file and the columns in it are Creation Date, Closed Date, Score, Title and Body. In which we would be using Body and Title. These two columns are uncleaned data with HTML tags in them. Also, in all three csv files we have observed that there are no null values in them. In the Tags.csv file there are over 3028 unique tags. Among them top most used tags are Java Script, Java, C#, Php and Android accordingly. For every question in the file the maximum number of tags used is 5. Where 12% of questions have 1 Tag, 26% of questions have 2 tags, 29% of questions have 3 tags, 19% questions have 4 tags and only 12% questions have 5 tags.

Implementation:

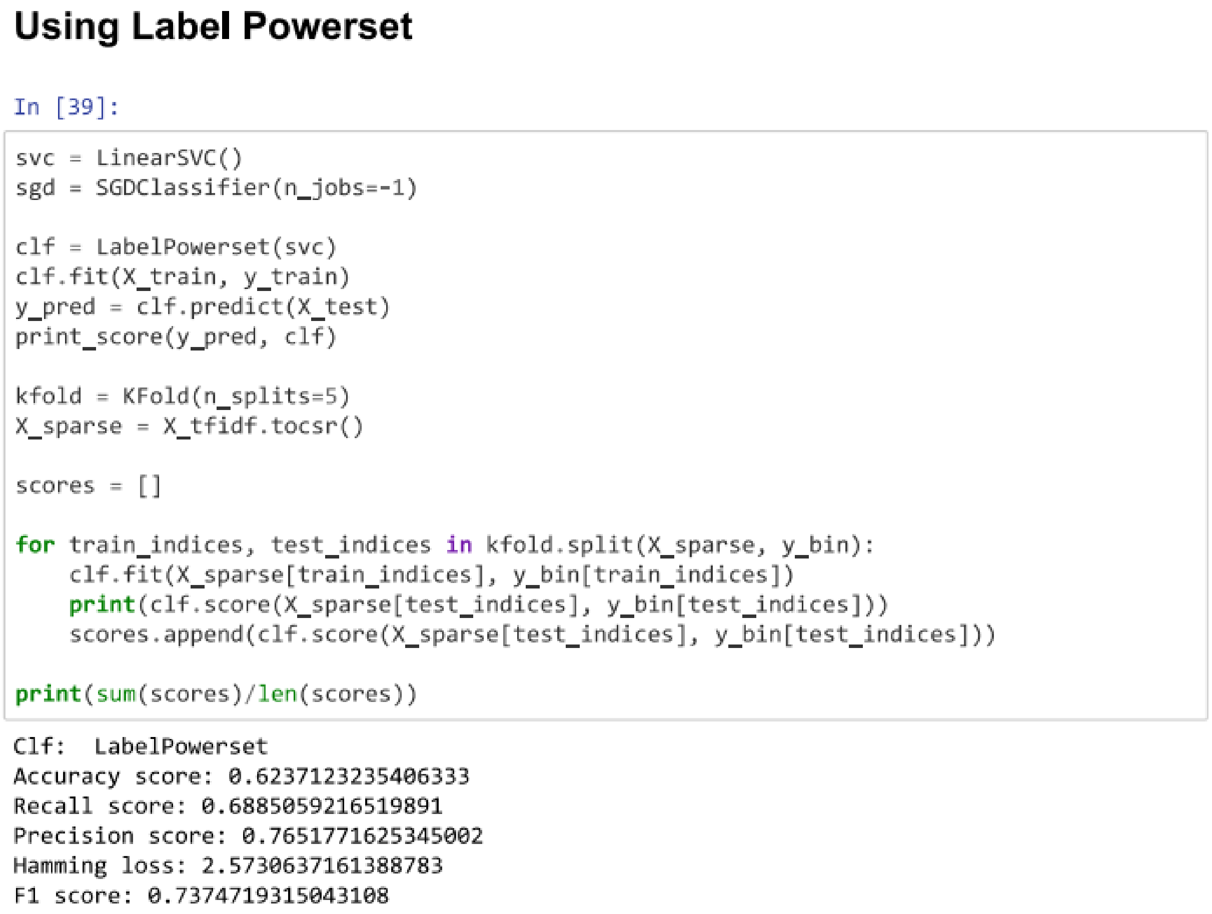
The main objective of this project is to design a classifier which could auto tag the questions in Stackoverflow. We would be using a Kaggle dataset which has a list of Stackoverflow questions and their tags. Initially, the dataset of questions and tags would be extracted into two dataframes one in each. Later, as a part of data preprocessing, the unwanted information in questions is removed using NLP techniques such as Removal of special characters from title and body, Removal of stop words, Removal of HTML tags ,Convert characters to lowercase and Lemmatize the words. In the next step, we would be converting the preprocessed data into vectors using TFIDF Vectorizer. This vectorized data is further divided into test and training data. Later using the Label Powerset classifier we would classify the Stackoverflow questions with tags.



Github Link :
https://github.com/Gagan-Achanta/CSCE_5290_NLP_Project

Preliminary Results:

After the final step of using the Label Powerset classifier we have achieved an accuracy score of 62.3%, Recall Score of 0.68, Precision Score of 0.76, Hamming Loss of 2.57 and finally the F1 Score of 0.73. We have used an SDD classifier with a linear support vector hyperplane to predict the tags of each Stack Overflow question. Also, we have used Tf Idf as a vectorizer to convert word tokens into vectors. For validating the model, we have used K fold cross validation with a number of splits of 5.



Using Label Powerset

In [39]:

```
svc = LinearSVC()
sgd = SGDClassifier(n_jobs=-1)

clf = LabelPowerset(svc)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print_score(y_pred, clf)

kfold = KFold(n_splits=5)
X_sparse = X_tfidf.tocsr()

scores = []

for train_indices, test_indices in kfold.split(X_sparse, y_bin):
    clf.fit(X_sparse[train_indices], y_bin[train_indices])
    print(clf.score(X_sparse[test_indices], y_bin[test_indices]))
    scores.append(clf.score(X_sparse[test_indices], y_bin[test_indices]))

print(sum(scores)/len(scores))
```

```
Clf:  LabelPowerset
Accuracy score: 0.6237123235406333
Recall score: 0.6885059216519891
Precision score: 0.7651771625345002
Hamming loss: 2.5730637161388783
F1 score: 0.7374719315043108
```

Project Management

Implementation status report:

Work completed:

Description: Initially we have collected data from Kaggle in the form of CSV files and have analyzed it. In the next step we performed a few cleaning operations on input and output data(Preprocessing). Later, we trained Label Powerset using the training part of data.

Responsibility (Task, Person):

Aditya Pujari:

- Researched and found an appropriate dataset for Auto Tagging of Stack Overflow Questions from Kaggle.
- Analyzed the data whether it has any missing values before performing preprocessing operations on text data.
- Studied various machine learning algorithms and documented suitable algorithms to be used for this problem statement.

Gagan Sai Ram Anvesh Achanta:

- Analyzed the data and figured out the most frequent tags in the Stack Overflow questions.
- Figured out the most important features in the dataset and removed unnecessary features such as Upwords, I'd in the data.
- Helped in removing HTML tags from features, Title and Body of Stack Overflow questions.

Sai Tarun Gunda:

- Performed most of the cleaning operations on data using NLP techniques such as Removal of Stop words, Removal of special characters in Title, Converting the text into lower case and Lemmatization.
- Split the dataset into Train and Test datasets with 80% data as Training and remaining 20% as Test datasets.
- Converted text information in Train and Test datasets into vectors using TFidf Vectorizer.

Nithish Reddy Boddhi Reddy:

- Designed a classifier using Label Powerset which uses the Support Vector classifier.
- Used K Fold Cross validation for cross validation of the model with a number of splits equal to 5.
- Achieved an accuracy of 62% and F1 score of 0.73 using a Label Powerset. In order to increase the accuracy we would be implementing a new classifier model.

Contributions (members/percentage):

Aditya Pujari	30%
Gagan Sai Ram Anvesh Achanta	25%
Sai Tarun Gunda	22.5%
Nithish Reddy Boddhi Reddy	22.5%

Work to be completed:

Description:

Building new models in order to achieve higher accuracy and F1 score. Most probably we would be using Ensemble models.

Responsibility (Task, Person):

Aditya Pujari:

- Might try to implement additional cleaning methods on dataset such as Spelling Correction.
- Will design a Preprocessing production ready pipeline to use on application end.

Gagan Sai Ram Anvesh Achanta :

- In order to get higher accuracy we might implement Ensemble models on training and test data sets.
- Will build a front end site where users can post a Stack Overflow question, hence tags would get generated automatically.

Sai Tarun Gunda:

- Will try to build a model using binary relevance and Support classifier which is good for multi label classifiers and try to improve the accuracy, precision score and F1 score.
- Try to optimize the hyperparameters of SVM in Label Powerset in order to get higher accuracy.

Nithish Reddy Boddhi Reddy:

- Building a restful API to request Stack Overflow questions and give predicted tags for the question.
- Will create a module for converting vectors to text output as it is useful for getting model vector output to tags.

Issues/Concerns:

- The dataset which we have is insufficient to train and test the model. Hence it is one of the major reasons for getting less accuracy.
- During preprocessing we thought we lost a few important features in the dataset.
- The trained Label powerset has provided an accuracy of 62% which is less in the practical scenario and might predict irrelevant tags to the Stack overflow questions.

Expected Outcome:

The initial classification model is used to determine whether or not there is a ship in the image. And the training classification accuracy should be as close to 98 percent as possible. The pipeline's trained classification model is then connected to an object detection model, which recognizes the ships in the image. We anticipate that the bounding box in which a ship is spotted from a picture will be almost perfect.

References:

- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. CoRR, abs/1301.3781
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. CoRR, abs/1301.3781.
- https://www.kaggle.com/competitions/multilabel-bird-species-classification-nips2013/data
- https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff