Stack overflow question tag prediction

Team :

Mudra Sahu 193050007
Muttineni Navya 193050017
Ankita Singh 19305R002
Tejal Topre 19319R002

Objective

- Stack-Overflow Tag Prediction using text in the title and description.
- Important in business- correctly send questions to experts based on tags
- Data in Excel format- contains 6034195 rows (6.75 GB)
- Each row can have multiple tags
- Dataset contains following 4 columns:

<ID> < Title>   <Body> < Tags>

ANALYSIS /RELATED LITERATURE

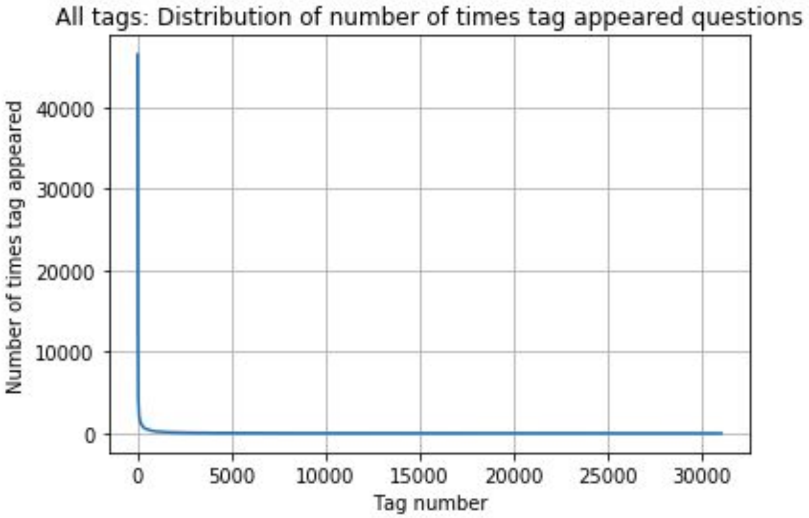
- Assign multiple tags to one question = multi-label classification problem
- Performance measure - ‘Accuracy’ not enough
- Need high precision and recall for predicted tags - F1 score

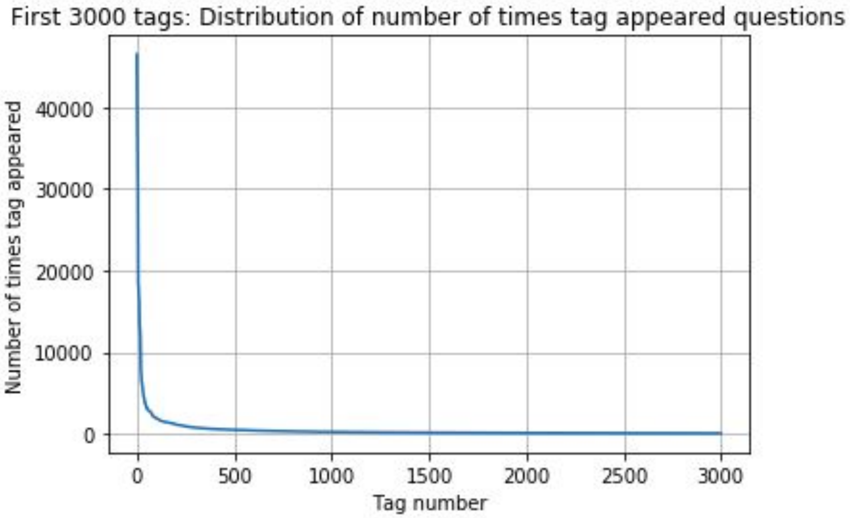
$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Pre-processing - ANalysis of tags

- Total 31k Tags for entire dataset
- Distribution of tags -highly skewed
- Around 4000 tags enough to cover 98% questions
- Better for multilabel classifier to consider fewer tags-faster
- Removed infrequently occurred tags from the data

Pre-processing - ANalysis of tags

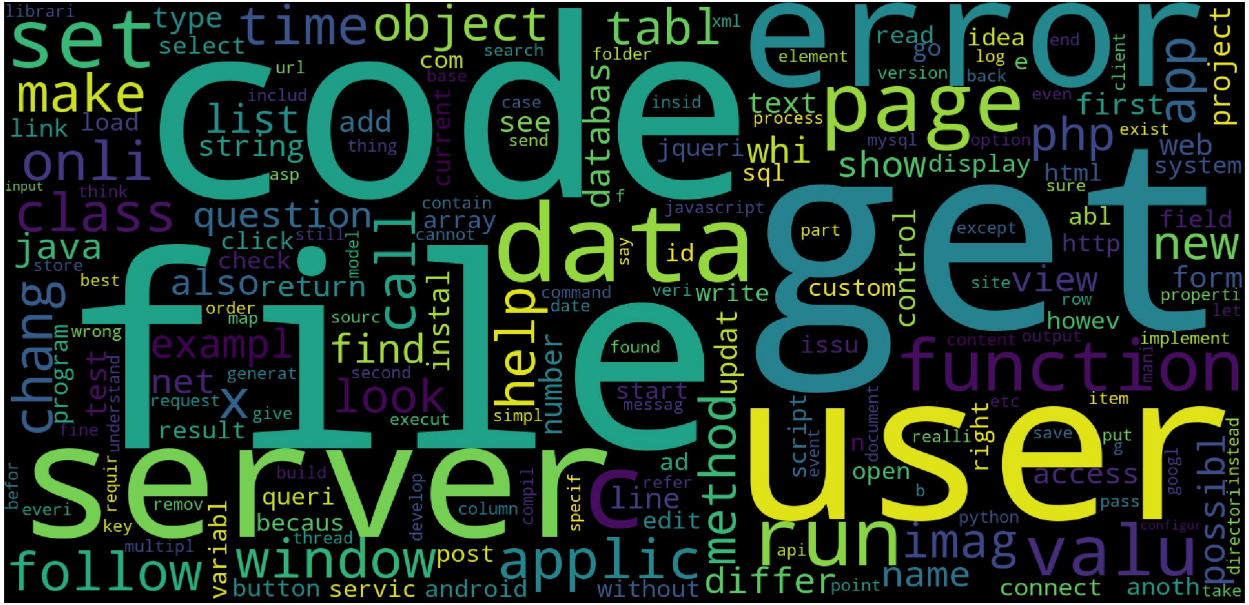




Pre-processing - Analysis of questions

- Stopwords like “the”,”is” etc occurs in high frequency- Removed using nltk stopwords library
- Question title- high info content : given 2x weightage
- Stemmed english words
- Seperated code snippets and Html tags using regex
- Converted all characters to small letters
- Removed extra useless words after analysis via word cloud

Pre-processing - Analysis of questions



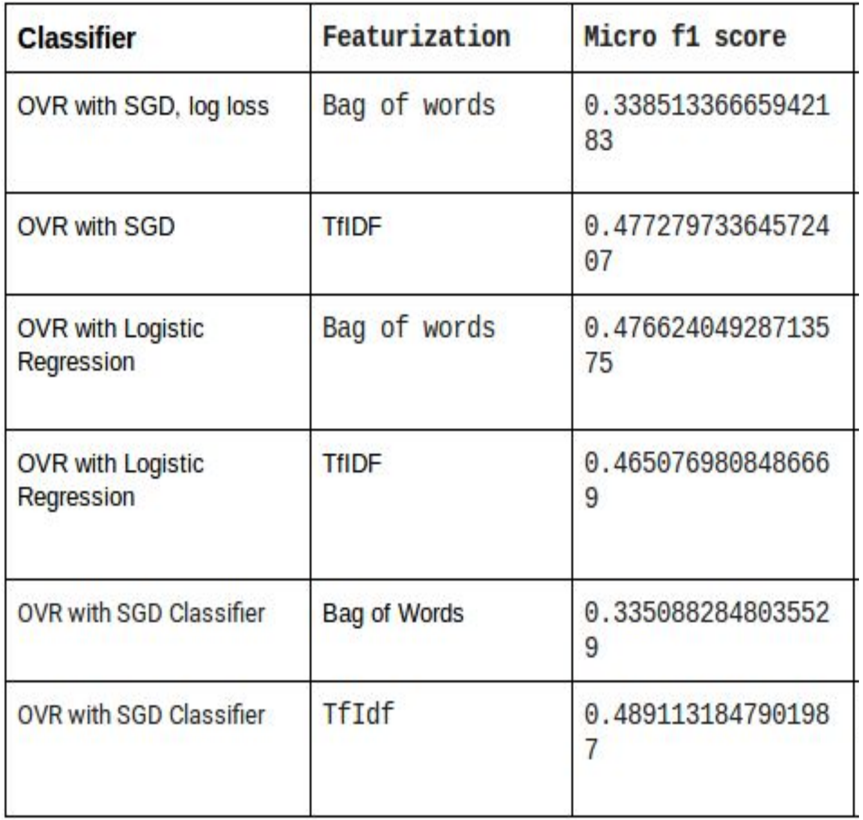
Word Cloud for title+body after pre-processing

Experimental approach for modelling

- Considered two featurizations : Bag of words and TFIDF
- Considered bigram features for both
- Classification : OneVsRest Model used (from Sklearn) for multiclass classification
- Underlying classifcation models tried : SGD with log loss ,SGD with hinge loss, logistic regression and linear SVM

Experimental approach for modelling

- Also performed hyperparameter tuning on all the models considered using gridsearchcv sklearn
- We observed best F1 scores for logistic regression with SGD (for both featurisations)



Classifier	Featurization	Micro f1 score
OVR with SGD, log loss	Bag of words	0.33851336665942183
OVR with SGD	TfIDF	0.47727973364572407
OVR with Logistic Regression	Bag of words	0.4766240492871357 5
OVR with Logistic Regression	TfIDF	0.4650769808486669
OVR with SGD Classifier	Bag of Words	0.3350882848035529
OVR with SGD Classifier	TfIdf	0.4891131847901987

Language & environment

- Python 3 , Jupyter Notebook ,Google Collab
- Tested using various graphical plots
- Initially considered 10k rows and 100 tags
- Total run on 30% of total data (20 lakh rows) due to GPU limitations - 4000 tags threshold

EFFORT

- Most challenging : working on huge datasets , training time and parameter tuning
- Time distribution :

  25% understanding and visualising problem statement

  30% preprocessing (tried GuessLang and some less used libraries for better information retrieval)

  40% Modelling and hyper-parameter tuning

Thank you!