Tag Suggestion For Stack Overflow

Group Members :
Badadhe Shruti Ekanath
Bhatambre Shruti
Dhulipala Krishna Priya

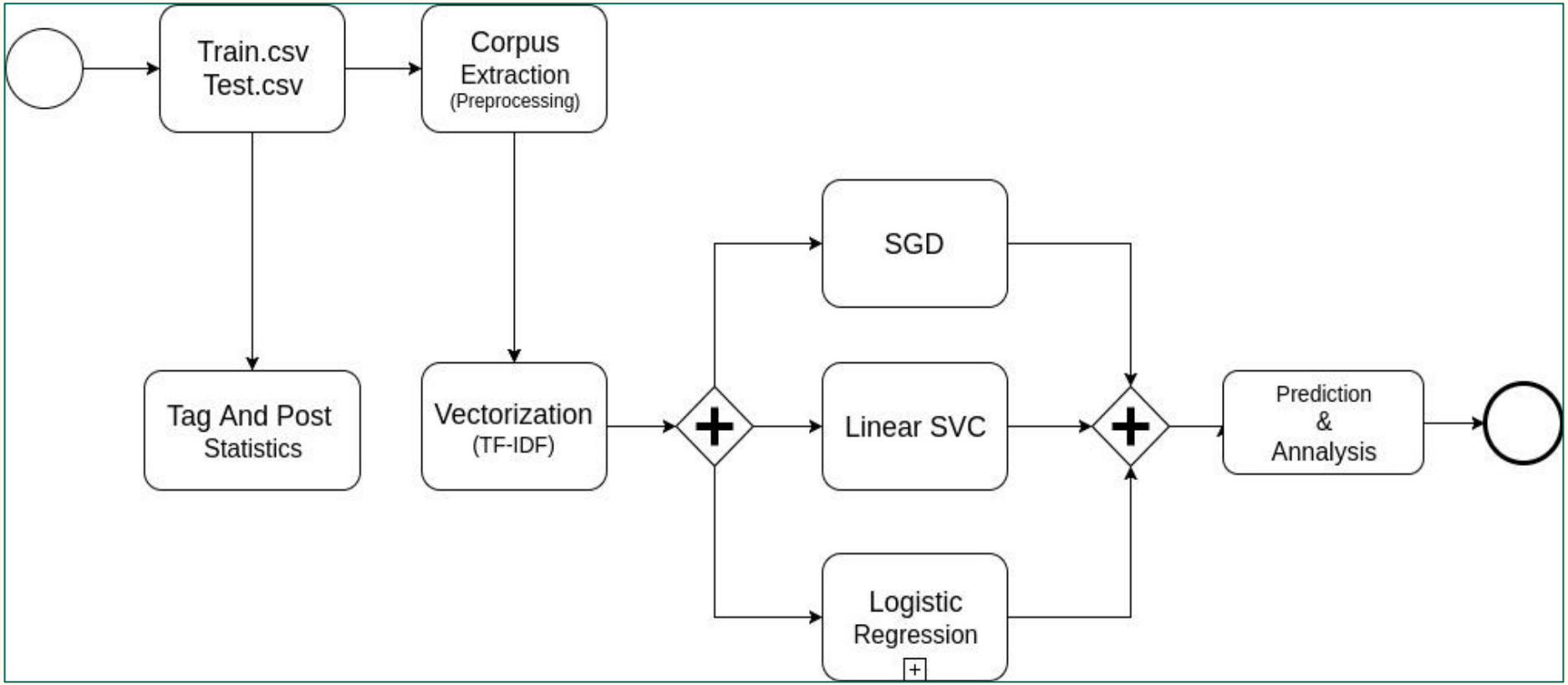
Problem Statement :

- Implementing a tag recommendation system for Question - Answers knowledge system like StackOverflow.
- Since a Question – Answer site may host **millions of questions** with tags and other data, this information can be used as a training and test dataset for approaches that automatically suggest tags for new questions based on the historical similarity of the old Question - Answers.

Introduction :

- Authors provide tags for their content which can be used to find user's goal and interest to provide better experience to user.
- Datasets of website like facebook, twitter, stack overflow studied to relate tags to user interest.
- Stack overflow : website of computer programming related questions and answers.
- Tag : word which describes topic in question, also help in finding related question
- Manual tagging is difficult, so need automated system.

Our Work Flow-chart



Action Taken

Statistics

Processing the data provided on Kaggle

Link :
https://www.kaggle.com/c/facebook-recruiting-iii-keyword-extraction

Dataset :

- Dataset contains a large number of posts and associated tags.
- Each question/post in the training set contains various fields like id, title, body, tags.
- Stack overflow releases its dataset quarterly which can be downloaded and used.

Our Dataset Stats

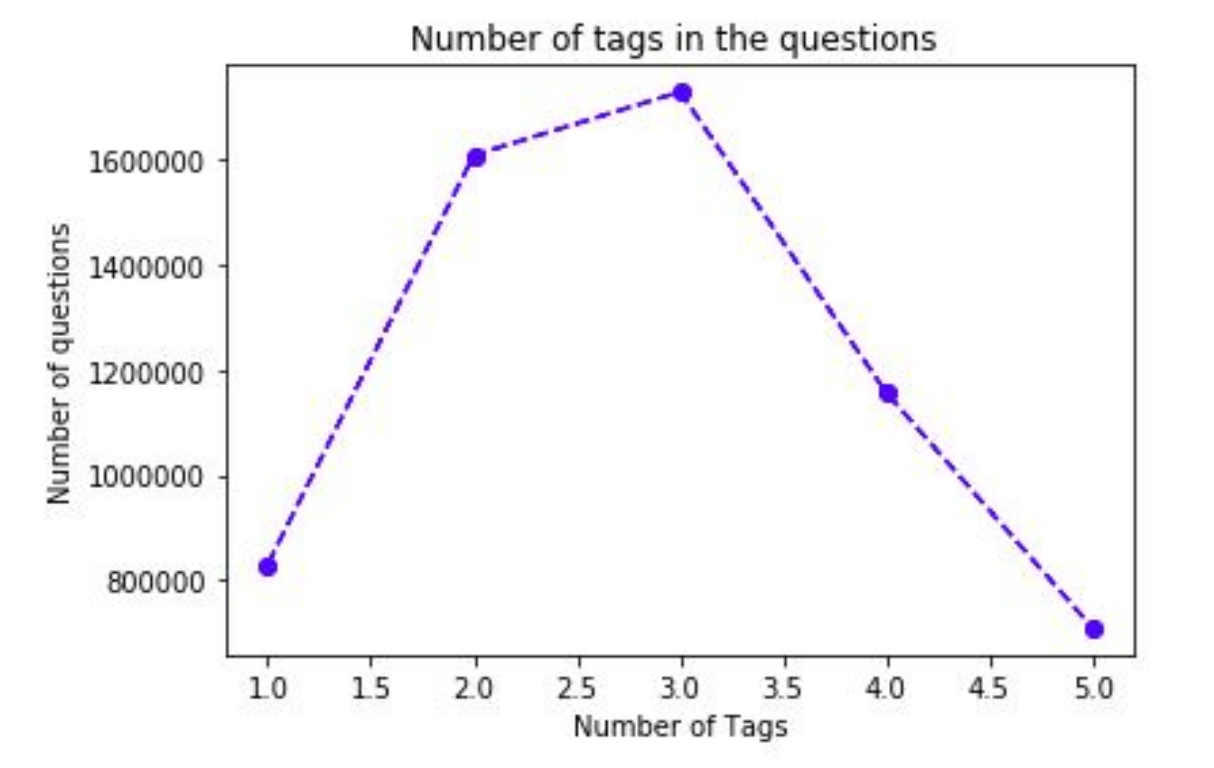
- 6.75 GB in size for Train Data and 2 GB for Test Data
- 6,034,195 questions in Train data & 2 million questions in Test Data
- Also there is no Null value in the tag column.
- Data was dumped into sqlite database.

Tags Stats

```
Number of Unique Tags : 42084
There are total 234 tags which are used more than 10000 times.
19 tags are used more than 100000 times.
Most frequent tag 'c#' : 463526 times.
Maximum number of tags per question: 5
Minimum number of tags per question: 1
Avg. number of tags per question: 2.885222
```

- Unique tags are much smaller compared to total datapoints.
- Majority of the most frequent tags were programming language.

Tag Stats



- Each Question has different number of tags.
- So we did a histogram of for different frequency of tags in each question.

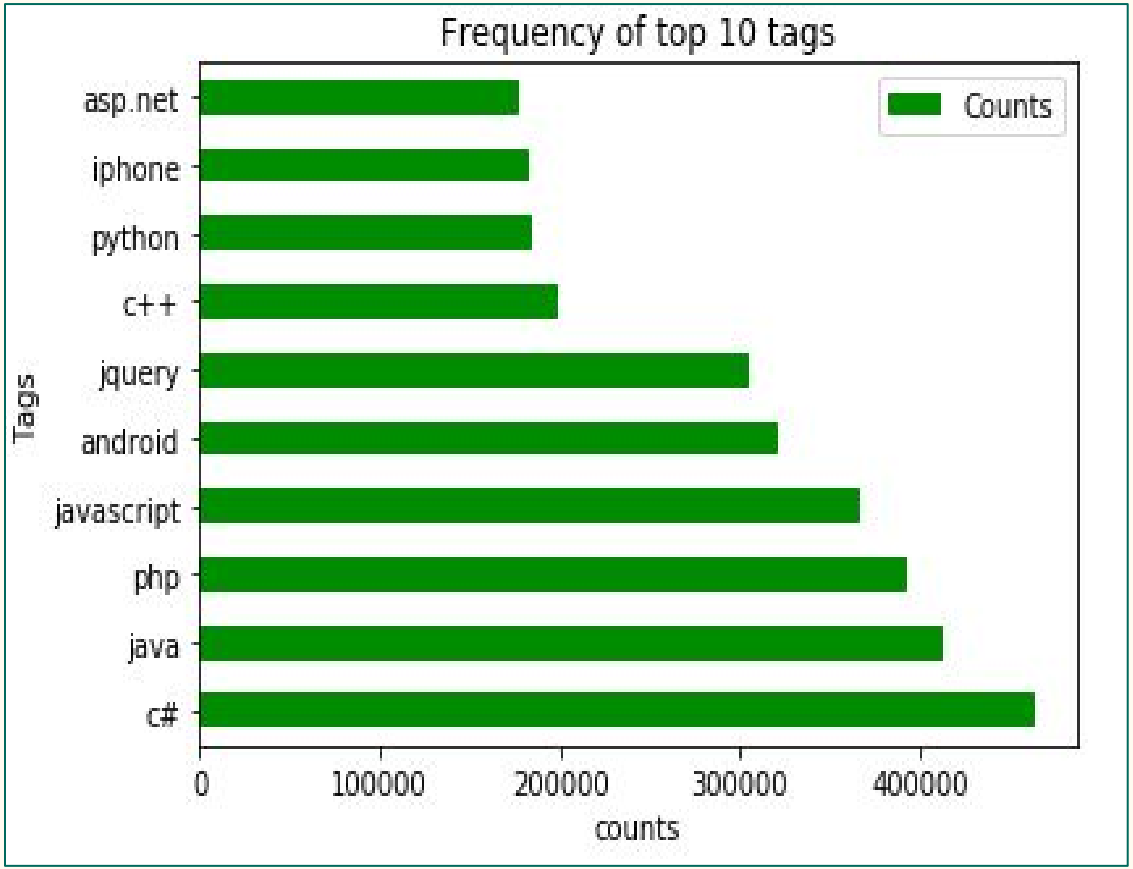
Max # of tags per question: 5

Min # of tags per question: 1

Avg. # of tags per question: 2.885222

- Most of the questions are having 2 or 3 tags

Top 10 Tags



We obtained this data after the following steps:

- Scanned each row
- Accumulated all tag of each post.
- Finally after removing duplicates found the count for each tag
- Found these top 10 tags.

Action Taken

Preprocessing

Pol : *What do u “ read “, my lord ?*

Ham: *Words, Words, Words ….*

Preprocessing of the Posts

- We Took 1 million lines of posts from the total data.
- Removed StopWords, HTML tags, particular punctuations.

```
pre processing Train.csv...
**** Before Removing stopwords ****

<p>I'd like to check if an uploaded file is an image file (e.g png, jpg, jpeg, g
if, bmp) or another file. The problem is that I'm using Uploadify to upload the
files, which changes the mime type and gives a 'text/octal' or something as the
mime type, no matter which file type you upload.</p>

<p>Is there a way to check if the uploaded file is an image apart from checking
the file extension using PHP?</p>

**** After HTML tags & Removing stopwords ****

I'd like check uploaded file image file e g png jpg jpeg gif bmp another file pr
oblem I'm using Uploadify upload files changes mime type gives 'text octal' some
thing mime type matter file type upload way check uploaded file image apart chec
king file extension using PHP


finished
```

Preprocessing of the Posts (cont...)

- Almost many questions consists of text and code .
- Approx. 50% of questions contain <code> & we removed it from body.

Before removing code part

```
<p>You can also describe this data structure by the rules for its growth:</p>

<ol>
<li>Start with node 0 with associated empty set <code>C(0) = {}</code> = <em>cre
ating new object</em></li>
<li>For any node x create a new node y where (x,y) is oriented edge from x to y
and <code>C(y) = C(x) union { (x,y) }</code> = <em>versioning and branching</em>
</li>
<li>For any nodes f and t, create node r, where (f,r) and (t,r) are oriented edg
es and <code>C(r) = C(f) union C(t) union { (f,r), (t,r) }</code> = <em>merging<
/em></li>
</ol>
```

```
**** After removing the code part ****

Define Revision Tracking Graph RTG oriented graph without circles node x set C x
 associated C x contains edges paths node 0 C 0 {} edge set also describe data s
tructure rules growth Start node 0 associated empty set creating new object node
 x create new node x oriented edge x versioning branching nodes f create node r
f r r oriented edges merging Note see mathematical difference creating new versi
```

Action Taken

Feature Extration

Conversion of each question into a feature vector is done using ‘tf-idf vectorizer’

TF-IDF

- **Term frequency** TF(t)

= (Number of times term t appears in a document) / (Total number of terms in the document).

- **Inverse Document Frequency** IDF(t)

= log_e(Total number of documents / Number of documents with term t in it).

Action Taken

Training the data

For training the data, three different approaches have been employed :

- SGD
- Linear SVC
- Logistic Regression

Training the feature Vector

- Following the paper on multi class classifications, we trained the data using three one vs Rest Classifiers. (http://ieeexplore.ieee.org/document/7395121/)
- Then calculated the accuracy, precision, recall and hamming loss for each.
- In the following slides, we show the result for the following question:
- Title: **Can you use sample weights in pystan or pymc3?**
- Tags: python, pymc3, stan, rstan, pystan
- Body: ‘ If my observed dataset has weights (for example tracking multiplicity) is it possible to provide this either to pystan or pymc3, similar to the rstan function signature (http://mc-stan.org/rstanarm/reference/stan_glm.html):

```
stan_glm(formula, family = gaussian(), data, weights, subset, na.action = NULL, offset = NULL, model = TRUE, x = FALSE, y = TRUE, contrasts = NULL, ..., prior = normal(), prior_intercept = normal(), prior_aux = exponential(), prior_PD = FALSE, algorithm = c("sampling", "optimizing", "meanfield", "fullrank"), adapt_delta = NULL, QR = FALSE, sparse = FALSE) ‘
```

SGD Classifier

- Took Hinge loss and log loss functions as possible cases of loss function and various alpha for SGD Classifier Algorithm. Used GridSearch(Scikit) to tune hyperparameters.
- Hinge loss: loss less than zero is taken as zero loss.
- Log Loss: Classifier must assign a probability to each class

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log p_{ij}$$

- Log loss performed better on grid search. alpha = 0.0001
- Predicted Tags: python, stan, pystan, pymc3, rstan, gaussian
- Accuracy over test set: 0.44

Linear SVC

- Used dual SVM with squared hinge loss after grid searching over various parameters with l2 loss.
- Predicted Tags: python, stan, pystan, pymc3, gaussian, matrix
- Accuracy = 0.48

Logistic Regression

- Input values are combined linearly using weights to predict an output value(y).
- y = e^(b0 + b1*x) / (1 + e^(b0 + b1*x))
- Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x).
- Used with L2 regulariser with l1 loss
- Accuracy = 0.52
- Predicted Tags: python, stan, pystan, pymc3, rstan

Results

Classifier	accuracy	F1 score	Precision	Loss
SGD classifier	0.44	0.53	0.72	0.0017
Linear SVC	0.47	0.55	0.76	0.0015
LogisticRegression	0.52	0.65	0.76	0.0014

Other attempts

1. Multilayer Neural Networks(https://goo.gl/r2Hp51)
2. Labeled LDA algorithm and a spreading activation algorithm for finding associated tags followed by a weight tuning algorithm were already implemented online but the accuracy for this method was very close to the accuracy we got.(https://github.com/PoorvaRane/Tag-Prediction)
    a. Accuracy = 0.45
    b. Precision = 0.73
    c. F1 score = 0.67
3. These methods work better for small test and train set than what we considered. But do not scale up as much compared to the one we tried.

Thankyou