Mid Project Presentation

Machine Learning

Navya Aggarwal | 2018349
Nitin Gupta | 2018251
Sandeep Kumar Singh | 2018369

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

Mid Project Presentation Template:

Total 10 minutes slot, which includes presentation and QnA session. So ideally presentation should be for around 6-7 minutes. Number of slides for each section can be flexible, but all the sections are mandatory.

1. Motivation (2 slides)

2. Literature review (2 slides) Discuss at least two Research Papers, not blogs or articles.

3. Dataset description (2-3 slides)

    a) Different attributes, some visualization, details regarding the dataset.

    b) Details regarding any kind of preprocessing is required or not.

4. Methodology. (2-3 slides)

5. Results/Analysis/conclusion (2 slides). Based upon work done till now.

6. Timeline, are you able to follow the timeline that you proposed in the proposal, and proposing future timeline. (1 slide)

7. Individual team member's contribution. (1 slide)

Motivation

- StackOverflow is the largest, most trusted online community for developers, students, and other educationists throughout the globe. It is an educational resources available to everybody to clarify doubts, ask questions, answer questions, vote questions, and answers up or down and post questions and answers.
- A question on StackOverflow needs to be assigned tags by users which is used by other people to find solution to their problems. Further, users can subscribe to tags on a certain topic to receive digests of new questions for which they might have an answer. Realising the importance of tags, reveal important consequences and considerations related to tag assignment to questions/ answers.
- The aim for the project is to develop a predictor which can predict the tags based on the content of a question. It can be thought of as a recommendation system for users who pose questions on StackOverflow.

Literature Review

Many articles and similar projects are available on the internet, which gives us an insight into the tag assignment to a question or an answer. Students, educators, and computer science enthusiasts from all across the globe have contributed.

A paper titled ‘Predicting Tags for Stack Overflow Questions Using Different Classifiers’ was presented by Taniya Saini and Sachin Tripathi of ISM Dhanbad. According to the paper, relevant data extraction from the data was a crucial task, and text classification is a multi-class and multi-label classification problem.

There are two ways to implementing the classification.

- ➔ We can decompose the classification problem into k independent binary classification problem or we can train a separate classifier for each of the k possible output labels.
- ➔ The second approach is from adaptive algorithms. Examples of adaptive algorithms include boosting and random forests. Some of the classifiers that can

Literature Review

Another paper titled ‘Predicting Tags for StackOverflow Questions’ by PHD students at Stanford demonstrate an approach which helps in building a predictor. According to the paper, tags for a question can be divided into two parts.

- ➔ Firstly, all questions can be broadly classified using a tag indicating the technology or the programming language they are related to. This type of methodology to classify files according to language is also employed by GitHub.
- ➔ Lastly, the questions can be related to a subtopic or paradigm, and a seperate classifier can be used to predict them.

This dividing approach can be used to predict tags in a more concise manner. Thus, building classifier for seperate types of classification can help predict labels with greater accuracy.

Dataset Description

- The dataset used is available on Kaggle competition on the automatic tagging of StackOverflow posts .
- The original data set consists of 60,34,195 StackOverflow questions. But for the purpose of this project we are using only 20 lakh samples.
- Each datum in the dataset consist of 3 columns specifying Question Title, Body and the related Tags for the question.
- Body contains the extracted HTML text from the web site.

	Id	Title	Body	Tags
0	1	How to check if an uploaded file is an image w...	<p>I'd like to check if an uploaded file is an...	php image-processing file-upload upload mime-t...
1	2	How can I prevent firefox from closing when I ...	<p>In my favorite editor (vim), I regularly us...	firefox
2	3	R Error Invalid type (list) for variable	<p>I am import matlab file and construct a dat...	r matlab machine-learning
3	4	How do I replace special characters in a URL?	<p>This is probably very simple, but I simply ...	c# url encoding
4	5	How to modify whois contact details?	<pre><code>function modify(.....)\n{\n $mco...	php api file-get-contents

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

Pre-Processing

- ➔ Remove HTML tags

  Since the data was directly scraped from the website, it contains HTML tags which are present in every dataset and creating noise and hence needs to be removed.
- ➔ Remove extra whitespaces and other special characters

  There were some users generated extra spaces and special characters which were contributing to noise and we removed them.
- ➔ Lowercase all texts

  To standardise the dataset we converted all the text to lowercase.
- ➔ Convert number words to numeric form

  Since some questions contained numeric data which can’t be processed in text processing hence removed.
- ➔ Remove Stopwords

  For text preprocessing words such as “as”, “is”, “a”, “the” etc creates noise and hence removed.
- ➔ Remove Duplicate entries

  There were duplicate data which might affect our prediction later, needs to be removed.

Pre-Processing

- ➔ Lemmatization

  Removed inflectional endings and stored the base or dictionary form of a word.

- ➔ Tokenization

  Breaks the raw text into words, sentences called tokens. These tokens help in understanding the context and developing the model.

- ➔ Class normalization

  Converting all those words that resembles the same technology for eg converting “ReactJS”, “React-JS”, “React-Js”, “react-JS” etc to “reactjs”.

- ➔ Convert accented characters to ASCII characters

  Convert accented characters like ė, ê, è, ę to its equivalent ASCII “e”.

- ➔ Expand contractions

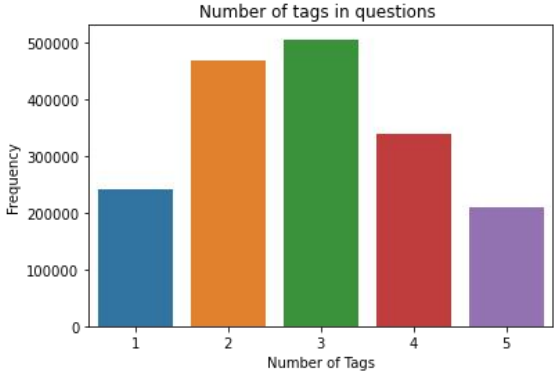
  Expand words like “don’t”, “can’t” to “do not” and “can not” respectively.

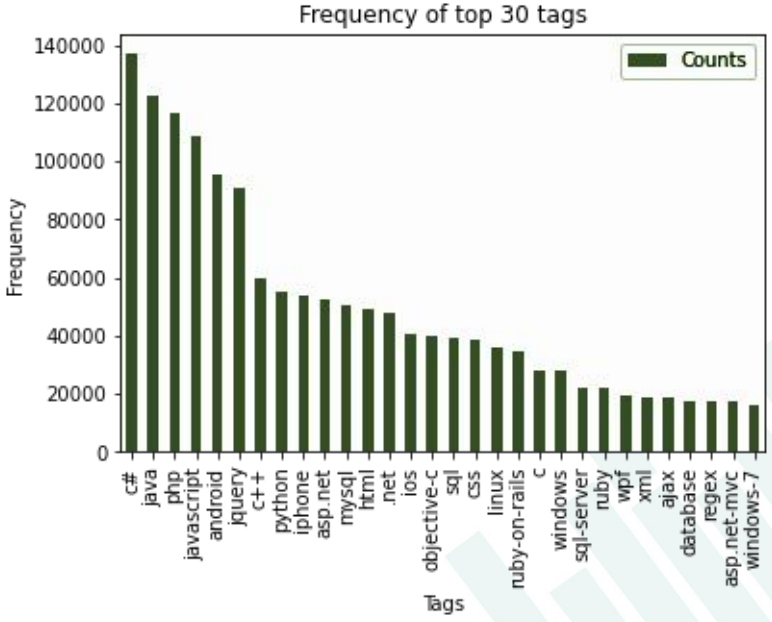
Data Visualization

- ➔ Initial Dataset Size: 20,00,000
- ➔ Dataset after removing Duplicate entries

```
Total number of duplicate questions :  232713
Dataframe shape after duplicate removal :  (1767287, 4)
```

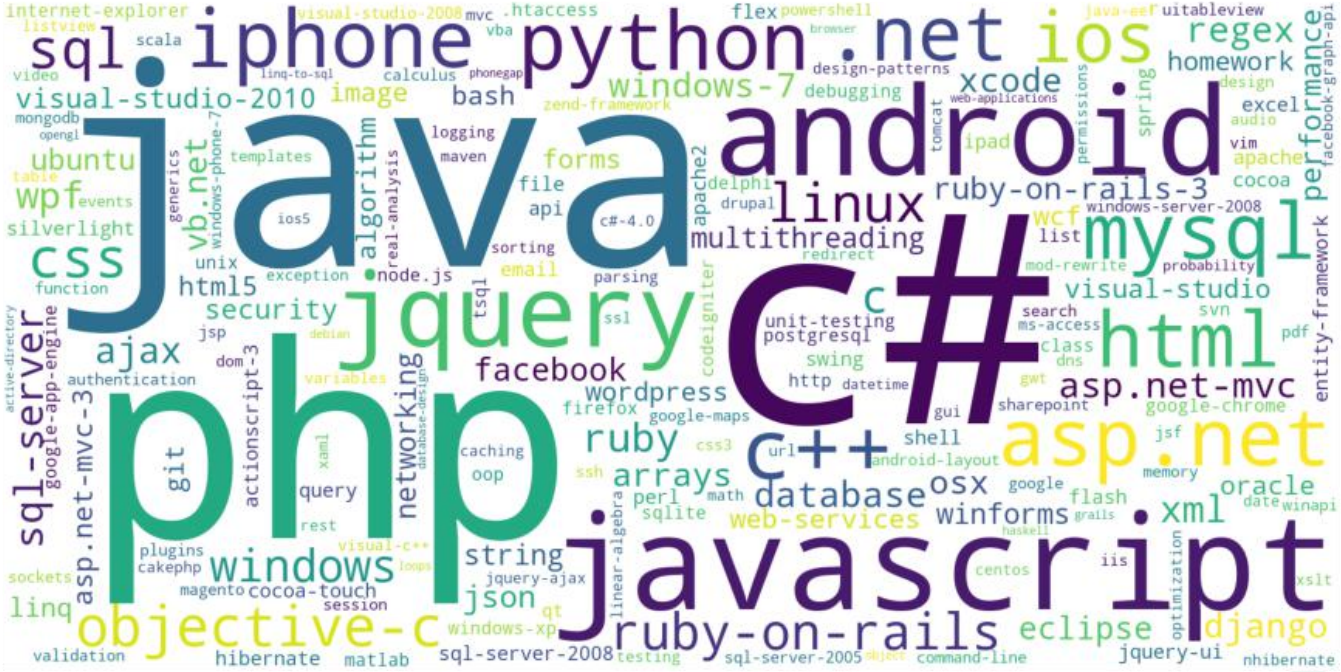
- ➔ Tags:
  - Maximum Tags in a single datum: **5**
  - Minimum Tags in a single datum: **1**
  - Total Number of Unique Tags: **38427**
  - Most occuring Tag: **c#**





Data Visualization

➔ Created this using wordcloud library, for nice depiction of Tag's frequency



Methodology

- **Data Collection**

  Firstly we needed to collect data to start working. We visited different website over the internet to find the data of our need. Finally we were able to find the data on Kaggle.

- **Data Preprocessing**

  We then moved on to processing our data to remove unwanted symbols, spaces , duplicate entries etc. We applied multiple preprocessing technique as mentioned above in the data preprocessing slide.

Methodology

➔ Training different models and analysing each one of them.

Next we will start training our model on the data set. Different classifiers we have planned to use are

- Training using Logistic Regression and analysis
- Training using Random Forest and analysis
- Training using Decision Tree and analysis
- Training using SVM and analysis

All the classifiers were decided after going through the research paper

➔ We will finally compare all the classifier to see which one performing better and then in the end we will mention all our finding in the final report.

Analysis

- So far in the project, our data for training predictors is ready. The dataset required heavy preprocessing.
- Due to this heavy preprocessing need on the dataset, we increased time on preprocessing to remove all kinds of noise and impurities from it.
- Further using this data, we will train text supporting machine learning models like Logistic Regression, Naive Bayes Classifier, Random Forest, and Support Vector Machine (SVM).
- We will use several values of hyperparameters during training, and try to attain model/s with high accuracy and also present an analysis of each model in further reports/evaluations.

Timeline

Major task according to proposed timeline was data collection and preprocessing part. We were successfully able to collect the data by searching different sources and then preprocessed the data using various techniques. Preprocessing took more time than expected. We then finally did some analysis of the features.

Future Timeline

- 13 October - Training using Logistic Regression and analysis
- 22 October - Training using Random Forest and analysis
- 31 October - Training using Decision Tree and analysis
- 9 november - Training using SVM and analysis
- 20 November - Check for Underfitting and overfitting of the models
- 30 November - Report Writing

Individual Member Contribution

Navya	Nitin	Sandeep
➔ Literature Review	➔ Literature Review	➔ Data Preprocessing
➔ Data Preprocessing	➔ Data Visualization	➔ Data Visualization