Natural Language Processing
Spring 2024

Assignment-2: Spam E-mail Filter

Instructions:

1- Students will form teams of **3** students (**Can be from different groups).**

2- Deadline of submission is **- 5 May 2024.**

3- Submission will be on google classroom.

4- No late submission is allowed.

5- No submission through e-mails.

6- File Naming (team ids) □ No grade for wrong ids and missing ids.

7- Cheating cases; you will get a negative grade whether you give the code to someone or take the code from someone/internet.

8- You have to write clean code and follow a good coding style including choosing meaningful variable names, **Also your code must be modular** (All processes are encapsulated into methods & you may use classes If you like) □ **Code Modularity included in the grading criteria.**

9- You have to write a report to summarize & discuss your results □ **Half of The Assignment Grade is on The Report.**

10- The Report must be no more than 2 reasonable pages □ **If it's more than 2 pages it won't be accepted so be concise & into the point don't write irrelevant information.**

Assignment Details:

Write a python notebook to build a spam filtering model using a labeled corpus of spam emails (Spam_Email_Data.csv). The model will classify emails as either spam or non-spam (ham) based on their content. **Submit your code in .ipynb & your Report in .pdf.**

Input:

- Corpus: Labeled corpus of spam emails (Spam_Email_Data.csv)

Output:

- Multiple Different Trained spam filtering models.

Notebook Components:

Data Preprocessing & Features Extraction:

Throughout the course you have been introduced to multiple text processing techniques like text cleansing (removing irrelevant words, symbols, etc.), tokenization (by regex, nltk tokenizer, etc.), Stemming (porter stemmer, snowball stemmer, etc.), lemmatization (wordnet lemmatizer, etc.), part of speech tagging, word-level semantic analysis, text embedding techniques whether neural networks based (word2vec like CBOW, doc2vec, etc.) or not (bag of words, tf-idf, etc.) & more From what you have learned identify what are the text processing techniques that are required for processing the provided data to solve this problem & in **which order.**

Data Splitting:

You will have to split the data into train & test portions, Identify the suitable splitting techniques & portions (50-50, 60-40, 40-60,) & how will you split it(randomly, based on some criteria, ..).

Model Training:

You should choose **at least two** Classifiers (Logistic Regression, Decision Tree, etc.) to train them on the prepared data. If you used text embedding techniques then you should try **at least four techniques** (at least two neural networks based techniques & at least two techniques that aren't based on neural networks) so you would provide in this case **at least 8 models** (4 models for each classification model algorithm)

Model Evaluation:

Choose **at least two** evaluation metrics (Recall, Precision, Accuracy, f1-score, etc.) & evaluate all your models performance based on them. **Summarize all models performances in a single dataframe to compare all of them.**

Report Structure:

Notebook Flow Description:

Short description to the process flow in the notebook for the different components & techniques you utilized in the project (Model Evaluation, Text Embedding, Data Splitting, POS tagging, Data Cleansing, Model Training, Stemming & Lemmatization, Tokenization, etc.) & discuss why you followed this order (e.g. why you applied stemming after tokenization)

Data Preprocessing & Features Extraction:

Briefly discuss why you have chosen each of those data preprocessing & features extraction techniques in your notebook for provided text data current problems.

Data Splitting:

Briefly discuss why you have divided the data into portions of that ratio & sizes & why in this manner.

Model Training:

Briefly discuss why you have chosen those classifiers to train data upon, if utilized text embedding techniques then why those techniques.

Model Evaluation:

Briefly discuss why you have chosen those metrics to evaluate your models.

Dominant Models:

Provide a table summarizing performance metrics of your models & highlight (color their names) the two models with top performance also, briefly discuss why you have chosen those as dominant models.