Application of Data Science
Final Report

Introduction The goal of the entire project is to implement text classification on the original cooking dataset and then implementing the same on 4 different datasets. For this a library by the name of "fasttext" has been used which is one of the most efficient ways of learning word embeddings and text classifications. Different parameters such as Learning Rate, Epoch, wordNgrams, Hierarchical Softmax and Multilabel Classification have been used to check the precision values of the model. Github Repository : https://github.com/PriyeshSolanki/ADS_1

1. Replication of Original Work – Bayesian Unsupervised Topic Segmentation

Creating VM – EC2 Instance

•Ubuntu Server 16.04LTS was chosen
•Storage – maximum 30Gb
•Launch instance – connected to ubuntu server

Running code for Bayesian Unsupervised Topic Segmentation

• New directory was created by the name Final using 'mkdir' and then changed to the same directory using 'cd'
• Downloaded the Bayesian Code using 'wget' command in the above created directory
• Uncompressed the above file and then changed into the directory that has been created.
• Then installed java which are a requirement using sudo apt-get update, sudo apt-get default-jre and sudo apt-get default-jdk.
• On the uncompressed file the following commands were executed to change the permission to make the scripts executable:- chmod 700 eval and ./eval config/dp.config.

Text Classification
The main purpose of text classification is to assign documents to one or multiple categories. For this to be done fasttext was installed and build.
• Used wget to extract fasttext from the mentioned url in the documentation.
• Unzipped the file that we got
• Then entered the fastText directory and used make to built it.
• After this we run ./fasttext so as to see different use cases supported by fastText.

Replication of Original Work (On Cooking Dataset)
fasttext was run on the original (Cooking) dataset but there were minor fluctuations in the Precision and Recall Values. Supervised, Test and Predict subcommands have been used which corresponds to learning text classifiers. Following parameters were used to check the precision and recall values: -
• Epoch – the number of times each example is seen. By default, fastText trains the data only 5 times. However, we can increase it upto 50 times to get high accuracy.
• Learning Rate (LR) – used to change (increase/decrease) the learning speed of the

model. By default, LR is 0 which means that there is no change in the model that is it does not learn anything. Range of LR is 0.1 to 1.0.
• Ngrams – To improve the precision of the data, wordNgrams is used. If we use unigram then it will check a single word or if we use bigrams two words will be scanned at a time and so on.
• Hierarchical Softmax – a loss function that approximates the softmax with a much faster computation, it works as a tree
hierarchy in which each word is characterized by leaves or parent nodes.
• Multilabel classification – Used to improve the efficiency. It predicts the probability of the labels for multilabel data.

Cooking Dataset	Original Paper		Replication	
	P@1	R@1	P@1	R@1
Before Pre-processing	0.124	0.0541	0.138	0.0597
After Pre-processing	0.164	0.0717	0.17	0.0737
Epoch 25	0.501	0.218	0.516	0.223
Learning Rate 1.0	0.563	0.245	0.582	0.252
Epoch 25 & Learning Rate 1.0	0.585	0.255	0.586	0.253
Multilabel (At threshold 0.1)	0.591	0.272	0.58	0.348
Multilabel (At threshold 0.5)	0.702	0.2	0.747	0.689

Table 1: Comparing original work and Replication

• wordNgram didn't work on the original 'cooking' dataset and the following error message was received 'std: :bad_alloc'. It is a type of an object thrown as exception by the allocation functions to report failure to allocate storage. So the error is related to memory allocation. The maximum memory allocation that we got while using EC2 instance on AWS was 30Gb, even after this we got the error. So we have used Ubuntu 16.04LTS application on windows where we set the maximum memory allocated to 60Gb so as to run all the parameters successfully.

```
ubuntu@ip-172-31-89-179:~/Dataset/ADS_1/fastText-0.9.1$ ./fasttext supervised -input cookin
g.train -output model_cooking -lr 1.0 -epoch 25 -wordNgrams 2
Read 0M words
Number of words:  8952
Number of labels: 735
terminate called after throwing an instance of 'std::bad_alloc'
  what():  std::bad_alloc
Aborted (core dumped)
```

Figure 1: WordNgram Error

2. Construction of new data We have created four new datasets using stack exchange group of websites. Stack Exchange is a group of question-and-answer websites on topics in diverse subjects, each site covering a specific topic. The four topics selected are: 1. Academia
2. Travel
3. English
4. Gaming

Each of these topics have at least 30,000 questions with multiple labels. All the questions have been extracted with their respective labels using python's beautiful soup library which is used for pulling data out of HTML and XML files and the data is saved in a csv file.

Then again using python, the word 'label' has been added as a prefix to each label that has been extracted and finally the data is converted into a text file (as required for fastText). Example: Each dataset was converted into a pandas data frame, the first column contains questions or text and the second column contains it's labels.

	Questions	Labels
0	Got 0% on a midterm in Math Graduate school	graduate-school,exams
1	When writing a textbook, what percentage of th...	publications,writing,books,publishers
2	I believe my PhD dissertation was unfairly gra...	phd,thesis,germany,all-but-dissertation
3	What universities can I consider for MS in Ele...	graduate-admissions

Figure 2: DataFrame of a Dataset

We need labelled data to train our supervised classifier. Therefore, '__label__' string is added before each label using .replace function. Finally, the data is converted into the format required by fastText (same as original work) as shown below.
Example - __label__graduate-school__label__exams Got 0% on a midterm in Math Graduate school.

```
# Converting to format which can run on fasttext
data['Labels']=['__label__'+s.replace(',',' __label__') for s in data['Labels']]
```

```
# Adding both the columns
data_output = data['Labels']+ ' ' + data['Questions']
```

```
print(data_output,sep=' ', end='', flush=True)
```

```
0        __label__graduate-school __label__exams Got 0%...
1        __label__publications __label__writing __label...
2        __label__phd __label__thesis __label__germany ...
3        __label__graduate-admissions What universities...
4        __label__phd __label__mathematics __label__pos...
```

Figure 3: Code example for fastText

Lastly, the data is divided into training and validation as described in the original paper. This is done using head and tail command. The first 70% data is used for training and the rest 30% is used for validation.

2.1 Academia dataset

Total number of questions: 32500
Total number of labels: 442

```
priyesh_solanki@DESKTOP-302K60E:~/fastText-0.9.1$ head data.txt
__label__engine Do cars which have had recent engine rebuilds tend to be low in or keep their value?
__label__jeep __label__gears Does this gear mesh pattern look good?
__label__honda __label__battery __label__battery-charger Is there a way for the dealership to revive a dying battery?
__label__chevrolet __label__temperature __label__cruze Engine running cold. Warms up at stop lights, drops wile driving
__label__honda __label__brakes __label__civic __label__brake-fluid 2000 Honda Civic ex brake driver brake fluid
__label__honda __label__clutch __label__fit Honda Fit clutch hard on the knees?
__label__starting __label__chevrolet __label__impala 2008 impala won't engage selenoid
__label__brake-rotor __label__brake-calipers What do you do when brand new pads are rubbing on new rotor? It smokes and smells so bad and it seems like it is going to c
atch on fire
__label__engine-theory Helmholtz Resonance
__label__motorcycle __label__starting __label__noise __label__troubleshooting '93 Yamaha SR125 Starting Issues
```

Figure 4: Academia Dataset

2.2 English Dataset

Total number of questions: 32500

Total number of labels: 859

```
priyesh_solanki@DESKTOP-302K60E:~/fastText-0.9.1$ head Dataset_2.txt
__label__pronouns Can 'that' function as the subject in an independent clause?
__label__single-word-requests __label__phrase-requests Need a word for when an event or situation that is very unlikely to happen, happens
__label__single-word-requests Word for the strategy of stating the right answer during a debate
__label__meaning Who has a hurt expression in "He looked at her with a hurt expression"?
__label__quotations single quotes or not [duplicate]
__label__grammar How do we use 'times' that comes before a noun without any preposition?
__label__offensive-language What does it mean to call someone a wipe?
__label__pronunciation __label__archaic __label__scottish-english __label__motto In my pronouns god me defend
__label__single-word-requests __label__phrase-requests Is "making a sound" the most basic verb/verb phrase for the oral like speaking,talking?
__label__single-word-requests __label__nouns __label__academia __label__professions Noun opposite of a theoretician?
```

Figure 5: English Dataset

2.3 Travel Dataset

Total number of questions: 30000

Total number of labels: 1577

```
priyesh_solanki@DESKTOP-302K60E:~/fastText-0.9.1$ head output.txt
__label__visas __label__uk Your application for a visit visa to the United Kingdom has been refused [on hold]
__label__japan Item stuck in Tokyo JR "lost and found" station , not in Japan anymore
__label__visas __label__uk __label__trains __label__public-transport Cheapest way to use Railway Coach and Bus in UK for travel for 2 visitors for a month?
__label__uk __label__visa-refusals __label__standard-visitor-visas __label__spouses Intended stay in UK
__label__public-transport __label__belgium __label__brussels __label__ghent Can I travel from Brussels to Ghent using the JUMP card?
__label__airports __label__airport-security __label__airport-terminals __label__lost-luggage __label__damaged-luggage How can luggage get lost and never recovered if it still has a
 bag tag?
__label__indian-citizens __label__thailand __label__vietnam __label__southeast-asia __label__cambodia Registration process for tourists in South East Asian countries
__label__visas __label__refugees __label__colombia __label__us-permanent-residents __label__stateless-persons USA Permanent Resident through Asylum , visiting Colombia
__label__visas __label__schengen __label__hungary Visa requirements for Hungary for a prospective student pilot
__label__visas __label__uk UK Visa Refusal-Right to appeal-Glaring mistakes by the Visa Officer
```

Figure 6: Travel Dataset

2.4 Gaming Dataset

Total number of questions: 32500

Total number of Labels: 2851

```
priyesh_solanki@DESKTOP-302K60E:~/fastText-0.9.1$ head Gaming2.txt
__label__minecraft-commands __label__minecraft-pocket-edition How to make ranks in MCPE 1.13
__label__nintendo-ds Is possible to check the Health of my Nintendo DS Battery?
__label__heroes-of-the-storm Do blinds affect Kharazim's heal?
__label__oxygen-not-included algae terrarium: automatically empty bottles
__label__warframe Does converting your Kuva Lich count as killing it?
__label__controllers __label__xbox-one Connect multiple wireless Xbox one controllers on Windows 10
__label__williams-pinball-classics Free tables on Williams pinball?
__label__diablo-2 How to close settings in Path of Diablo
__label__minecraft-commands __label__minecraft-bedrock-edition How would I remove an exact amount of named items from a users inventory on MCBE?
__label__minecraft-bedrock-edition In Minecraft Bedrock Edition, can you /summon Arctic foxes?
```

Figure 7: Gaming Dataset

4. RESULTS:

Modelling & Evaluation: -

First, the model with default arguments was implemented on the respective datasets, but the results were pretty low. Then in order to improve the performance of the model over training datasets, we introduced various parameters and tuned the respective values for better performance.

Models along with various parameters are as follows:-
Model 1 – Default
Model 2 - Epoch
Model 3 – Learning Rate (Lr)
Model 4 – Epoch and Learning Rate
Model 5 – Epoch, Learning Rate and wordNgram
Model 6 - Epoch, Learning Rate, wordNgram, Bucket, Dimensions, Hierarchical Softmax
Model 7 - Epoch, Learning Rate, wordNgram, Bucket, Dimensions, One-vs-All

Various parameters were tuned in order to improve the performance of models. After setting the best values for respective parameters, the validation set is used to evaluate how good the model is on new datasets.

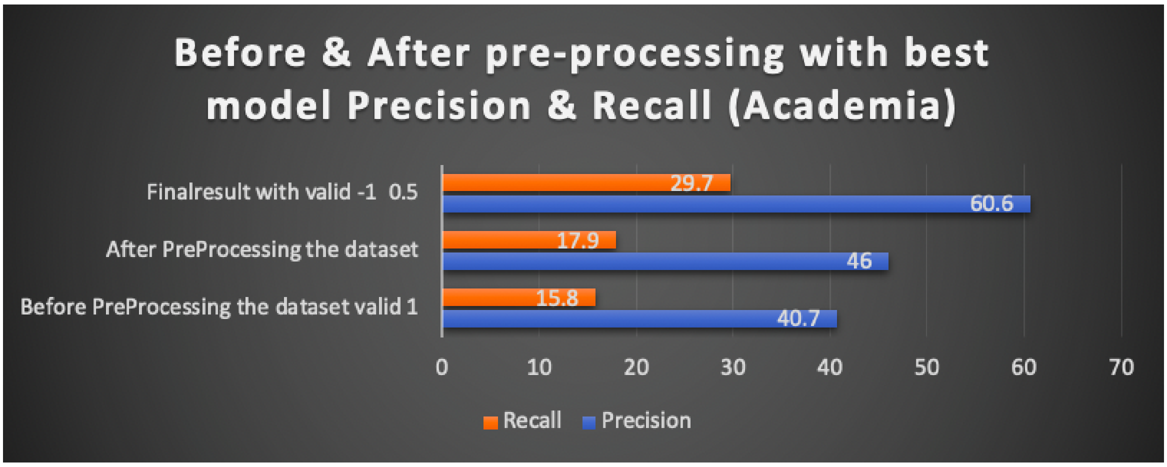
The best values per parameter for respective datasets are tabulated below: (Please refer our github repo https://github.com/PriyeshSolanki/ADS_1 for precision and recall values for every parameter)

Dataset 1 – Academia

Academia Dataset	Before Pre-Processing	After Pre-Processing	Epoch=20	lr=1.0	Lr= 1.0 &Epoch= 20	Loss HS	Loss one v/s all	Test -1 - 0.5
	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	
Precision	40.7	46	60.5	61	57	56.7	59.1	60.6
Recall	15.8	17.9	23.5	24	22	22	22.9	29.7

Table 2: Results of Academia Dataset (in %)

For Academia dataset, the above table contains the values of all the parameters that have been used for fine tuning the Precision and Recall Rate. We can see that for model 1 the precision was at 40.7%. After the pre-processing was done (model 2) there was an approximately 5% increase in the precision value. For model 3 there was a significant increase of 14.5% in the precision values as well as a 5.6% increase in the recall value. Next, model 4 was applied and there was a minor increase of 0.5% for precision. Model 5 decreased the precision and recall rates. Next Loss Hierarchical Softmax was applied(Model 6) and there is a further drop of 0.3% in the precision and recall remained constant. Loss One-vs-All (model 7) increased the precision by 2.4% and recall by 0.9%. Thus, after implementing the threshold Test -1 -0.5 we can say that the overall/final value of Precision Value has increased significantly as compared to the initial i.e. before pre-processing. There is a change of nearly 20% which is good.



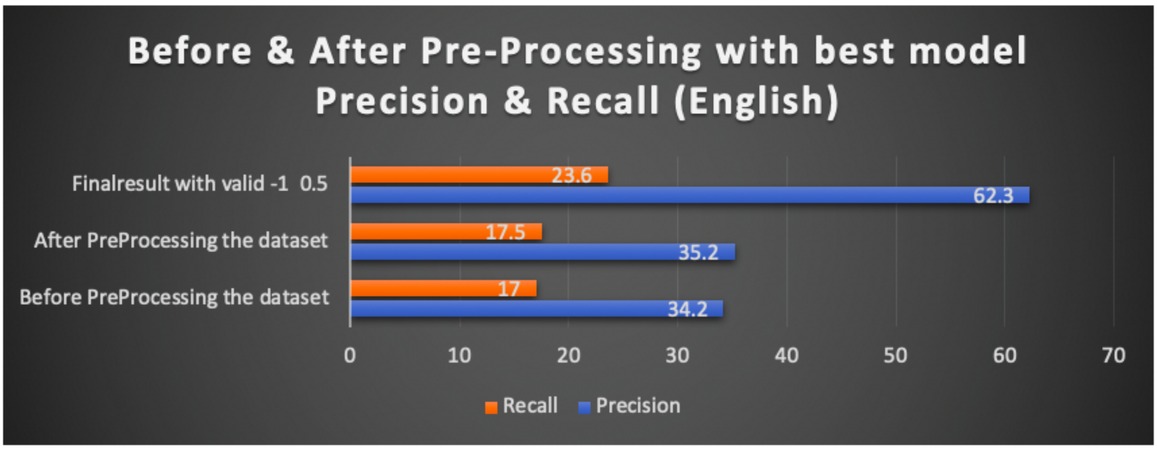
Dataset 2 – English

English Dataset	Before Pre-Processing	After Pre-Processing	Epoch=20	lr=0.7	Lr= 0.7 &Epoch= 20	Loss HS	Loss ova	Test -1 -0.5
	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	
Precision	34.2	35.2	41.6	42	39	39.6	41.1	62.3
Recall	17	17.5	20.7	21	19	19.7	20.4	23.6

Table 3: Results of English Dataset (in %)

The training set is initially fitted by Model 1, and the respective precision and recall rate after running the model over unknown set (Validation set) were insignificant. Model 2 has no parameter defined, rather pre-processing over original dataset and then splitting it into two sets viz., Training and validation set. Model 2 is fitted on training set which then helped to make the precision and recall rate better. Following Model 3 is setting Epoch and tuning until best value is met, evaluating it over validation set which further increase the precision and recall rate (Comparatively better). As described in the above table, each of the remaining models were tuned with best possible parameter's value, and corresponding Precision and recall rates account for making the model better. Finally, the Model with comparatively better precision and recall rates (i.e., Model 7) is used to evaluate on Validation set with setting the threshold value as 0.5, choosing as a trade-off between the best precision and recall value.

Below is the graph which shows the gradual increase in both, precision and recall rate over English dataset. The increase suggests that the model becomes better as we include parameters i.e. best model.

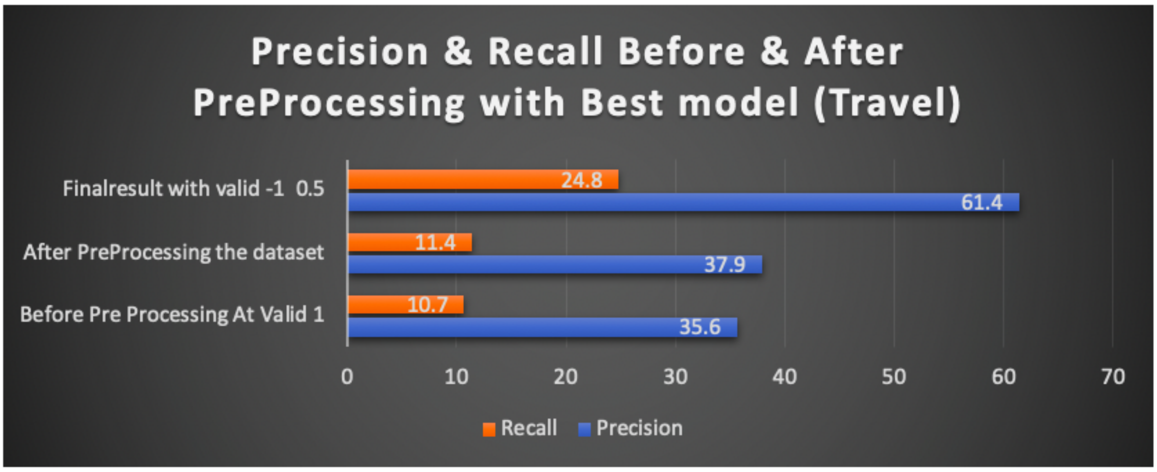


Dataset 3 - Travel

Travel Dataset	Before Pre-Processing	After Pre-Processing	Epoch=50	lr=1.0	Lr= 0.5 &Epoch=25	Loss HS	Loss ova	Test -1 -0.5
	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	
Precision	35.6	37.9	59	59	60	59.8	62.2	61.4
Recall	10.7	11.4	17.8	18	18	18	18.7	24.8

Table 4: Results of Travel Dataset (in %)

Before we start classifying the text, we check Precision for Model 1, it is at 35.6% which is very much insignificant. Model 2 precision is 37.9% bettered after pre-processed. Model 3 has a precision of 59% with epochs and combinations Model has 60% precision. Loss ova and Loss HS has efficient precision and recall values. As described in the below table, each of the remaining models were tuned with best possible parameter's value, and corresponding Precision and recall rates accounted for making the model better. Finally, the Model with better precision 61.4% and recall rate 24.8% (i.e., Model 7) is used to evaluate on Validation set.
Below is the graph which shows the gradual increase in both, precision and recall rate over Travel dataset. This increase suggests that the model becomes better as we include parameters.



Dataset 4 – Gaming

Gaming Dataset	Before Pre-Processing	After Pre-Processing	Epoch=50	lr=1.0	Lr= 1.0 &Epoch=25	Loss HS	Loss ova	Test -1 -0.5
	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	
Precision	20.2	23	38.2	41	41	36.4	37.2	76.8
Recall	15	17	28.3	30	31	27	27.5	26.7

Table 5: Results of Gaming Dataset (in %)

As far as the gaming dataset is concerned, from the above table containing the values of all the parameters that have been used for fine tuning the Precision and Recall Rate we can see from Model 1 the precision was at 20.2%. After the Model 2 was done there was an approximately 3% increase in the precision value. After this Model 3 was applied and there was a significant increase that is of 15.2% in the precision values as well as a 11.3% increase in the recall value. Next Model 4 was checked and there was a minor increase of 2.8% for precision. After Model 5 there was no change in the precision value, it remained fixed at 41% and recall had a minor increase of 1% and reached 31%. Next Model 6 was tried, and we can see there is a drop in the precision and recall value. A drop of nearly 4% in both the values. Even Model 7 almost gave the same value with very minor changes. Thus, after implementing the threshold Test -1 -0.5 we can say that the overall/final value of Precision has increased a lot if compared to the initial i.e. before pre-processing. There is an increase of nearly 56%.