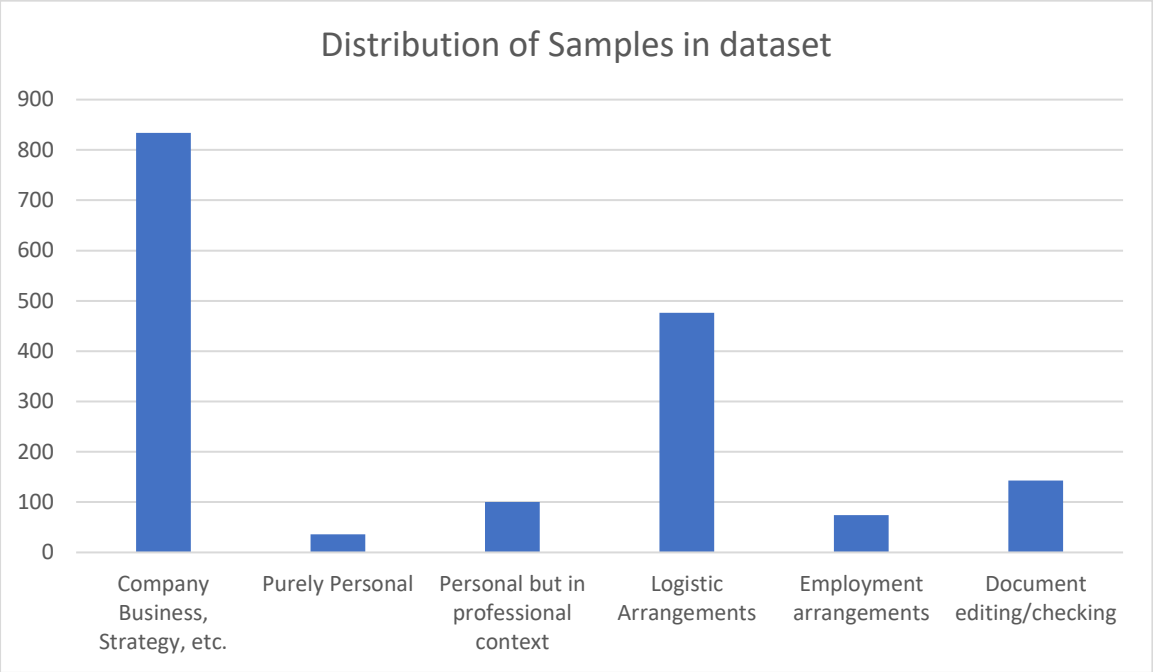
NLP Coding Assignment: - by Arnab Paul Choudhury (arnabpc.oshin94@gmail.com)

Given,

Total Sample Size: 1700 emails

Total number of labels/classes: 6

Distribution of samples in each class:



Suggested ratio for test-validation: 8:2

Observations:

- **Dataset is imbalanced:** As can be seen "Company business, Strategy, etc" has a huge number of samples as compared to other classes
- **Samples are full of junk values:** Upon closer inspection of the data, the samples had multiple junk data as a part of the email which contained metadata about the email, which has little information about the classification classes

Pre-Processing:

The following pre-processing steps were performed in the given order:

- All lines which included the following keywords were dropped:
    - junks = ("Message-ID", "Date", "From", "To", "Cc", "Mime-Version", "Content-Type", "Bcc", "X-From", "X-To", "X-bcc", "X-Folder", "X-Origin", "X-FileName", "X-cc", "Content-Transfer-Encoding", "Email")
- cleantext library of python was used to perform the following transformations
    - fix_unicode=True, # fix various unicode errors

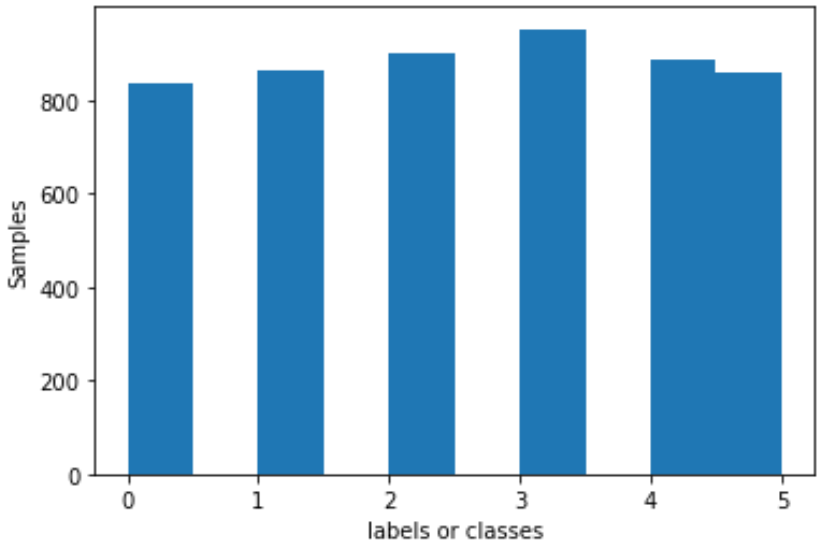
- to_ascii=True, # transliterate to closest ASCII representation
- lower=True, # lowercase text
- no_urls=True, # replace all URLs with a blank
- no_emails=True, # replace all email addresses with blank

- Each sample was transformed to a single sentence containing all relevant information.
- The cleaned data was then loaded into python using pandas library where the class labels were changed into numerals as given below
  - {0: "Company Business/Strategy", 1: "Purely Personal",2: "Personal but in professional context", 3:"Logistic Arrangements",4: "Employment arrangements", 5: "Document editing/checking"}
- The pandas Dataframe has 2 columns, namely ["labels", text] which contains the compiled cleaned data with the labels in numeric form. Given below is a view of the compiled data,

	label	text
0	3	subject: fwd: successful purchase of nber pape...
1	3	subject: re: ene officer elections no. we hav...
2	2	subject: re: i've joined charles river associa...
3	3	patrick.conner@enron.com, blundst@ei.enron.com...
4	2	subject: help!!! linda - i haven't heard of t...
...	...	...
5291	4	subject: fw: interview candidate shirley, p...
5292	2	subject: re: confidential - do not distribute ...
5293	4	subject: re: will hixon - ut candidate i don'...
5294	5	subject: draft letter to paul joskow for ken l...
5295	3	subject: ees staff meeting, in 746 ralph ree...

5296 rows × 2 columns

- The unbalanced classes were balanced using oversampling till each class had similar sample sizes. This increased the total sample size to 5296. Given below is the histogram of the sample size in each class after oversampling,



Modelling:

- Metric used for check model performance: Accuracy
    - Accuracy was chosen as a metric as it gives the overall performance of model on all classes. Since, no additional information was provided regarding the problem statement, this was the best metric to be used.
- Total number of Experiments conducted: 3

Experiment 1:

Using Naïve Bayes classification method, with imbalanced data
Results: Maximum accuracy = approx. 0.3

Experiment 2:

Using Decision Tree classifier, with imbalanced data
Result: Maximum accuracy = approx. 0.5

Experiment 3:

Fine-Tuning pre-trained BERT model, with imbalanced data
Results: Accuracy = 0.75

Experiment 4:

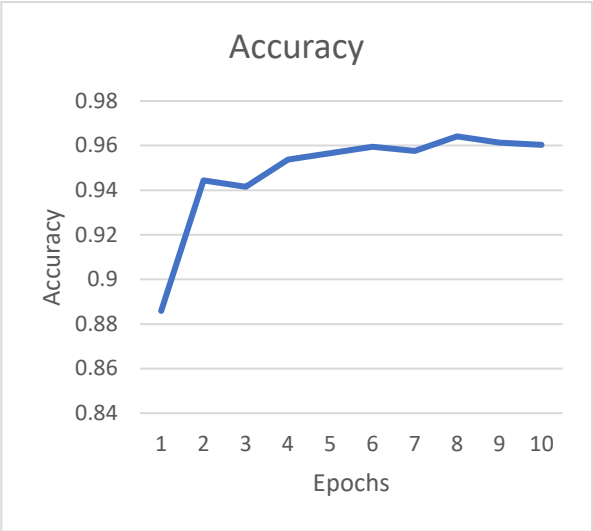
Fine-tuning pre-trained BERT model, with balanced data
Results: Accuracy = **0.96**

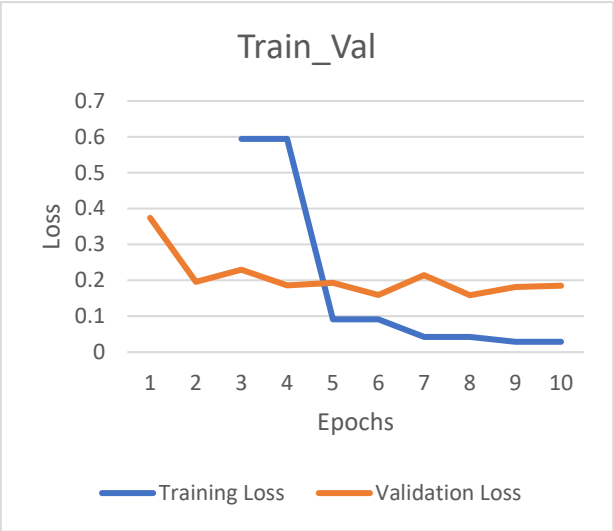
Details on Experiment 4:

- Pre-trained model used: bert-base-uncased
- Hyper-parameters:
    - Epoch: 10
    - Train Batch size: 18
    - Validation batch size: 12
    - Compute metric: Accuracy
- The model weights and biases are saved after every 500 iterations
- Time taken to train: 1 hour 10 minutes 48 seconds

Given below is the Train-validation and accuracy curve as well as the table with numeric values,

Epoch	Training Loss	Validation Loss	Accuracy
1		0.37432	0.885849
2		0.195753	0.94434
3	0.5944	0.229507	0.941509
4	0.5944	0.186385	0.953774
5	0.0912	0.193273	0.956604
6	0.0912	0.159387	0.959434
7	0.0421	0.214659	0.957547
8	0.0421	0.158619	0.964151
9	0.0289	0.181598	0.961321
10	0.0289	0.185037	0.960377





Analysis

As can be seen from the accuracy and Train validation curve, the accuracy keeps increasing with the number of epochs. Additionally, both the train and validation loss follow the expected J curve with validation loss and accuracy starting to plateau around epoch 7-8. This shows that there was no overfitting or underfitting and the model did understand something meaningful. An accuracy of 0.96 can be considered good if the use case is not too critical, however, it is suggested that a value above 0.99 is good to ensure very little wrong predictions. To improve upon the accuracy, the following steps can be taken

Scope for further analysis and experimentation to improve accuracy,

- Check accuracy for each class to check if any particular class performs poorly.
- Use better pre-processing methods
- Use a different pre-trained Large Language Model
- Hyper-parameter optimization
- More time for training
- Using Different method for balancing data