Ans-1) DataSet Description:

1. The problem in the dataset is the Multi-class classification. We have to predict the folder to which one particular file belongs
2. Y is evenly distributed, hence no imbalance of classes in the dataset
3. No Categorical data found with low cardinality.
4. Identified the following attributes can be used as individual features.

* Nntp-posting-host
* From
* Subject
* Organization
* Lines
* Reply-to
* Distribution

Ans-2) Observations from data:

|  |  |  |
| --- | --- | --- |
| Attributes | Unique items | Missing values |
| Nntp-posting-host | 704 | 1637 |
| From | 2004 |  |
| Subject | 2076 |  |
| Organization | 1308 | 103 |
| Lines | 200 | 117 |
| Reply-to | 304 | 2337 |
| Distribution | 34 | 2171 |

1. Because of the high cardinality in the attributes, we cannot use them as categorical features.

Ans-3) Reasons for selecting the below three models:

**Random Forest Classifier:** Random Forest is easy to implement and in the ideation phase I want my model to run fast and fail.

**XgBoost Classifier:** I have used Xgboost because:

* fast
* Accurate
* To improve model build by Random forest classifier

**Deep Learning Framework**

I have build two neural networks:

1. With a bag of word vectorizer
2. With pre-trained word vectors-used to learn the complex pattern

The first model is performing better than the second because the latter one is not able to converge as we do not have sufficient data points in the dataset(fewer training examples).

Ans-4) Performance of Models:

|  |  |
| --- | --- |
| **Model** | **F1\_score** |
| Random Forest | 48.46% -For validation set |
|  | 38.72% -For the test set |
| xgboost | 63.93% -For validation set |
|  | 59.74% -For the test set |
| Neural Network | 61.01% -For validation set |
|  | 57.38% -For the test set |

Ans-5) I will choose the simplistic neural network above all the models because it is performing best than all the models.

Improvement: If more data can be provided we can build a more complex model using approaches like a transformer, bert, pre-trained word vectors, etc