**Assignment:  
Compare the performances of machine learning-based email spam detectors.**  
Task Details:  
1. See the reference [1], it uses the support vector machine (SVM) algorithm to identify  
spam emails.  
2. Understand the concepts therein and the workflow.  
3. Practice the code and run on your own machines/Kaggle.  
4. Use two other machine learning algorithms (any two) to detect spam emails.  
5. Measure the accuracy (classification error rate) for each algorithm.

1. **A Review on Machine Learning Algorithms used**
   1. Multinominal Naive Bayes (MNB) (4) is a relatively simple probabilistic algorithm based on the Bayes theorem. It naively assumes that the features are independent. It can be trained rather quickly using supervised learning.
   2. Extreme Gradient Boosting (1) is a decision-tree-based ensemble Machine Learning algorithm that uses a [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting) framework.
2. **Dataset Description**
   1. Dataset used for training contains 5575 samples of which 87% classified as ham,

Train test split is done 75%, 25% respectively.

Where each new set contains 87% of Ham, same spam ham ratio as original(main) data set.

Chart, bar chart

Description automatically generated

* 1. To simulate real life situation both models are tested on **unseen data**.
  2. Data Cleaning: no data cleaning is performed. Vectorization is performed using Scikit- Scikit-learn’s model CountVectorizer (3)

1. **Hyper-Parameters** 
   1. In order to improve performance of the models, the most appropriate hyperparameters are found using the Scikit-learn’s GridSearchCV Cross Validation method (2). Recall that MNB does not have hyper-parameters to tune.
   2. Table 1. Optimal hyper-parameter values Extreme Gradient Boosting Model:

|  |  |
| --- | --- |
| Gamma | 1.5 |
| Learning rate | 0.5 |
| Max\_depth | 4 |
| Reg\_lambda | 1 |
| Scale\_pos\_weight | 0.1 |

1. Performance Observation:

|  |  |  |
| --- | --- | --- |
|  | Main dataset train / test | Unseen data |
| Multinominal Naïve Bayes | 0.98 | 0.72 |
| Extreme Gradient Boosting | 0.98 | 0.75 |

**XGBoost Unseen data**:

Chart

Description automatically generated

Naive Bayes Unseen data:

Chart

Description automatically generated

1. Conclusion:

Both models performed with high accuracy on (train / test split) main dataset and 98 % of emails were classified correctly. Due to high imbalance of Ham-Spam (87% - 13%) emails in dataset and limitations of our approach, we count only repetitions of words in emails and make prediction base on that. Accuracy went down once exposed to unseen data.

On unseen data, False Positive category, which is recognised as important, in email classification, harmless email that ended in spam folder XGBoost performed better with False Positive Rate of 0.10 compering to MNB FPR = 0.14, False Negative Category, spam emails that end up in inbox folder, is where MNB is showing better performance with False Negative Rate of 0.72 compering to XGBoost FNR = 0.76

References:

1 XGBost documentation. Available online: https://xgboost.readthedocs.io/en/stable/

2 GridSearchCV in Scikit Learn. Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html>

3 CountVectorizer <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html>

4 <https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html>