Deep Learning based Email Spam Filter

Introduction :

We will create the email spam filter model using deep learning and evaluate the model with other currently popular machine learning methods like xgboost, random forest, svm etc.
For this sample project, we will use Enron dataset in English. However this approach works well for other languages also which i had empiricially tested in my job.
This approach is combining unsupervised learning with Supervised learning. We will generate the features in unsupervised way using TF-IDF algorithm and then use this to features to train Models on labeled enron data.

Steps :

1. Preprocessing:
Here we will generate a pandas dataframe from the enron dataset . We will tokenize and also do some data analysis
2. Features Generation (Unsupervised Learning)
We will use TF-IDF as features to be used for training the models.
3. Model Training
We will train a 3-layered deep learning model.
We will also train Random forest, SVM and Xgboost for comparison purpose.
We will the same tf-idf features for all the models
4. Result Analysis and iterate to improve the performnce
We will present our results in nice and informative way to provide good comparison information.
Enron data combined with Spam assasin dataset has been obtained from :
https://www.cs.bgu.ac.il/~elhadad/nlp16/spam_classifier.html
5. Prepare training and test data
We will split data into test data and data for model training and validation. We do this step to keep test data out of both tf-idf and classifier models.
We will keep 10000 emails for testing and rest for the model building process.

Build models :

Deep learning model
We will build our 3 layer deep learning model using Keras and tensorflow.
Network
Input -> L1 : (Linear -> Relu) -> L2: (Linear -> Relu)-> (Linear -> Sigmoid)
Layer L1 has 512 neurons with Relu activation
Layer L2 has 256 neurons with Relu activation
Regularization : We use dropout with probability 0.5 for L1, L2 to prevent overfitting
Loss Function : binary cross entropy
Optimizer : We use Adam optimizer for gradient descent estimation (faster optimization)
Data Shuffling : Data shuffling is set to true
Batch Size : 64
Learning Rate = 0.001

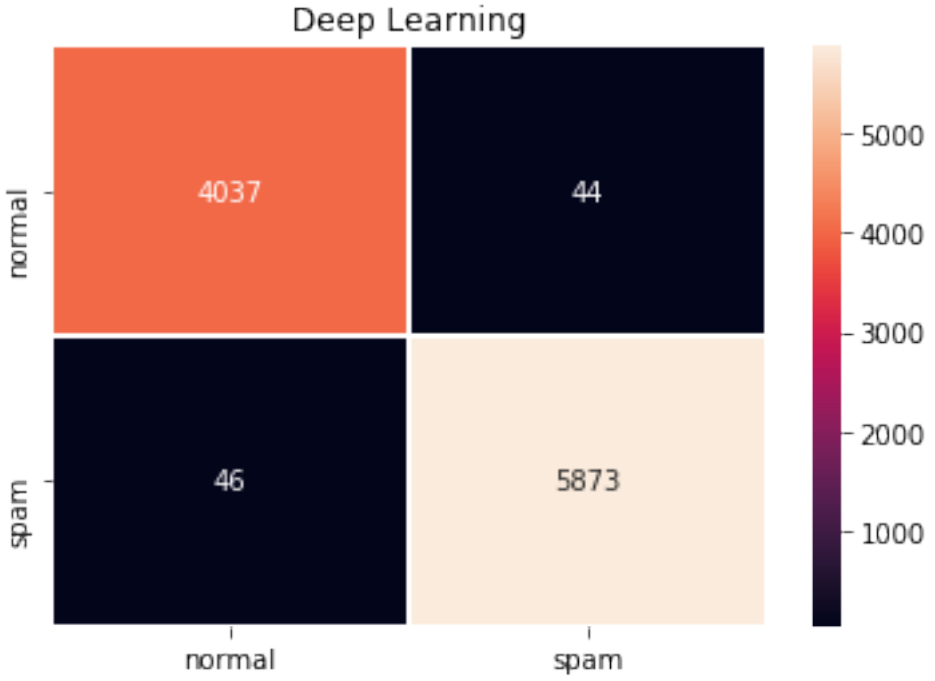
Other Machine Learning Models
We will build 3 more models and compare the performance in the same way. For this purpose we will use the same tf-idf as input feature . We will train following models :

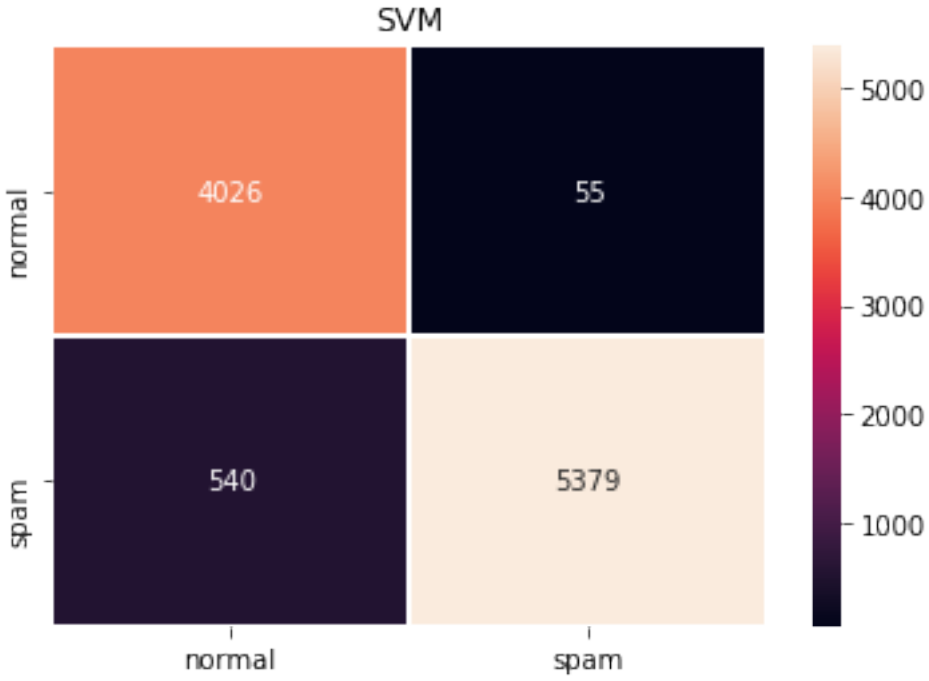
- **SVM**
- **Random Forest**
- **XGboost**

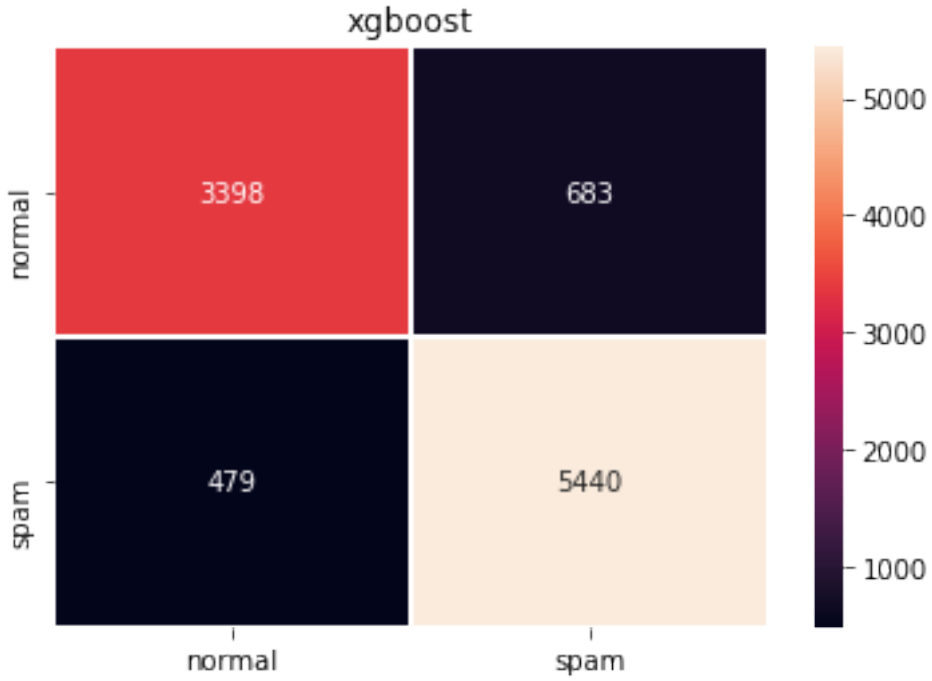
Results :

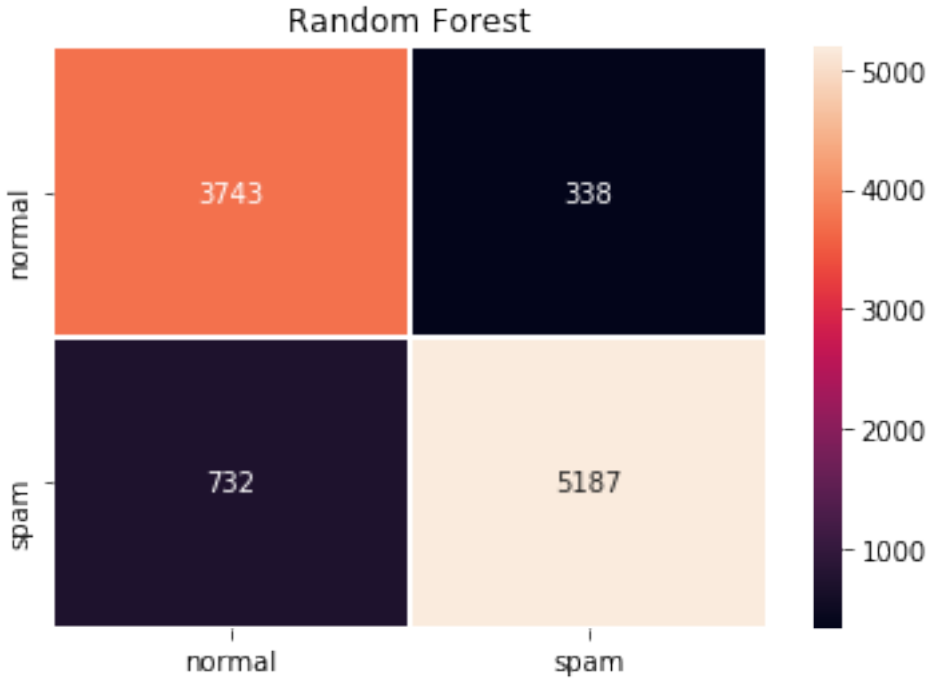
As we see, deep learning model does very well on the test data. The results from other models are close. I have tried this approach over multiple language emails and deep learning model is very consistent with the performance. XGboost also does very well. Please note that i have not optimized random forest and SVM much beyond the defaults. So they may have better performance with tuning.

Confusion matrix :

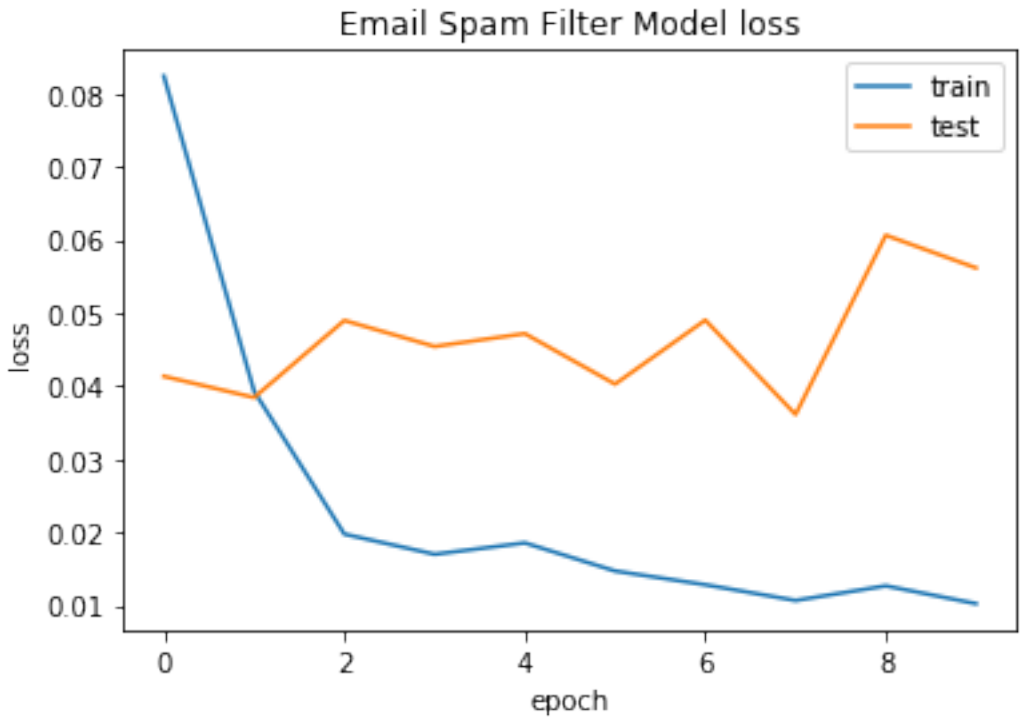





xgboost
3398
683
479
5440
normal
spam
normal
spam
5000
4000
3000
2000
1000


Random Forest
3743
338
732
5187
normal
spam
normal
spam
5000
4000
3000
2000
1000

Error function :



Performance of various models :

	model	precision	recall	f1_score	Total_samples	TP	FP	FN	TN	execution_time
0	random_forest	0.887624	0.896754	0.890721	10000	3743	338	732	5187	0.5935
1	svm	0.935807	0.947646	0.939390	10000	4026	55	540	5379	318.8586
2	deep_learning	0.990649	0.990723	0.990686	10000	4037	44	46	5873	5.4259
3	xgboost	0.882452	0.875857	0.878744	10000	3398	683	479	5440	0.4664