Lab Course Machine Learning Kalaiselvan Panneerselvam

Exercise 9 278920

Exercise 1: A spam filter using SVM

Part A: Build a spam filter using a pre-processed dataset

Loading Spambase Dataset

```
spam_data = pd.read_csv("spambase.data", header = None)
spam_data.head()
```

	0	1	2	3	4	5	6	7	8	9	...	48	49	50	51	52	53	54	55	56	57
0	0.00	0.64	0.64	0.0	0.32	0.00	0.00	0.00	0.00	0.00	...	0.00	0.000	0.0	0.778	0.000	0.000	3.756	61	278	1
1	0.21	0.28	0.50	0.0	0.14	0.28	0.21	0.07	0.00	0.94	...	0.00	0.132	0.0	0.372	0.180	0.048	5.114	101	1028	1
2	0.06	0.00	0.71	0.0	1.23	0.19	0.19	0.12	0.64	0.25	...	0.01	0.143	0.0	0.276	0.184	0.010	9.821	485	2259	1
3	0.00	0.00	0.00	0.0	0.63	0.00	0.31	0.63	0.31	0.63	...	0.00	0.137	0.0	0.137	0.000	0.000	3.537	40	191	1
4	0.00	0.00	0.00	0.0	0.63	0.00	0.31	0.63	0.31	0.63	...	0.00	0.135	0.0	0.135	0.000	0.000	3.537	40	191	1

5 rows × 58 columns

Converting to LibSVM Format

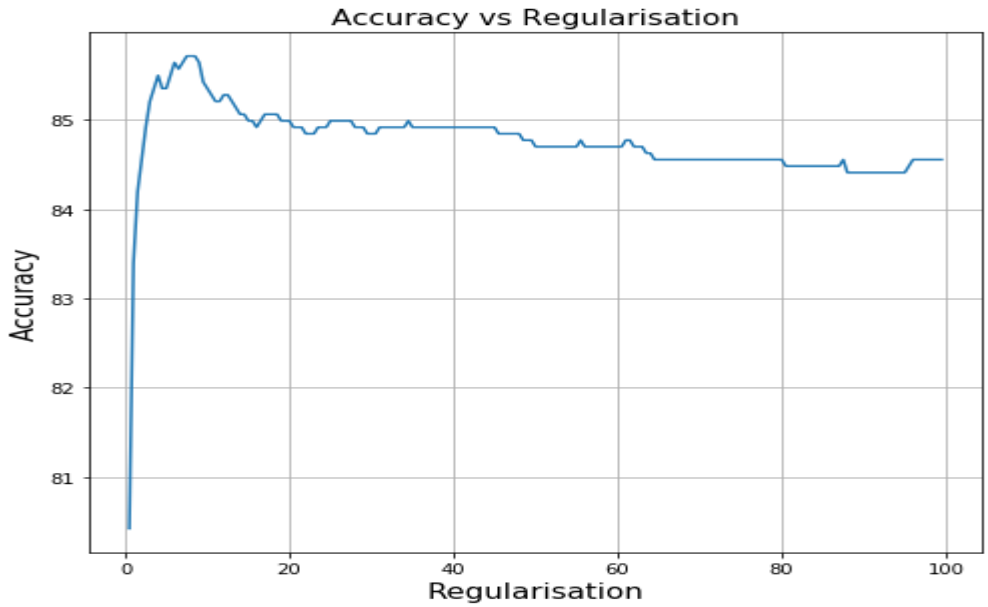
< label > < index1 >:< value1 > < index2 >:< value2 >. . .

```
convertTolibSVM(trainSet, "libSVMProcessed/trainSet")
convertTolibSVM(testSet, "libSVMProcessed/testSet")
```

```
def convertTolibSVM(data, path):
    n = data.columns.size
    for i in range(n-1):
        data[i] = str(i+1)+':'+data[i].astype('str')
    col = data.columns.tolist()
    col = col[-1:] + col[:-1]
    new = data[col]
    new.to_csv(path+'_formated', sep=' ', header=None, index=None)
```

Plots

Cost vs Accuracy



Cost vs MSE

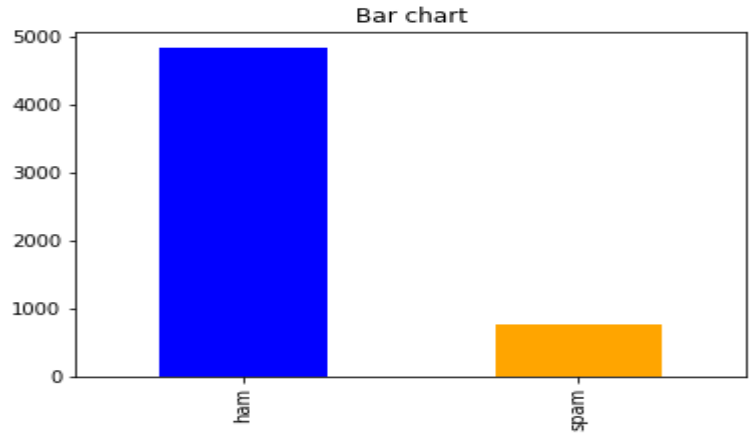
Part B: Pre-processed a dataset and learn SVM

Loadins SMSSpamCollection Dataset

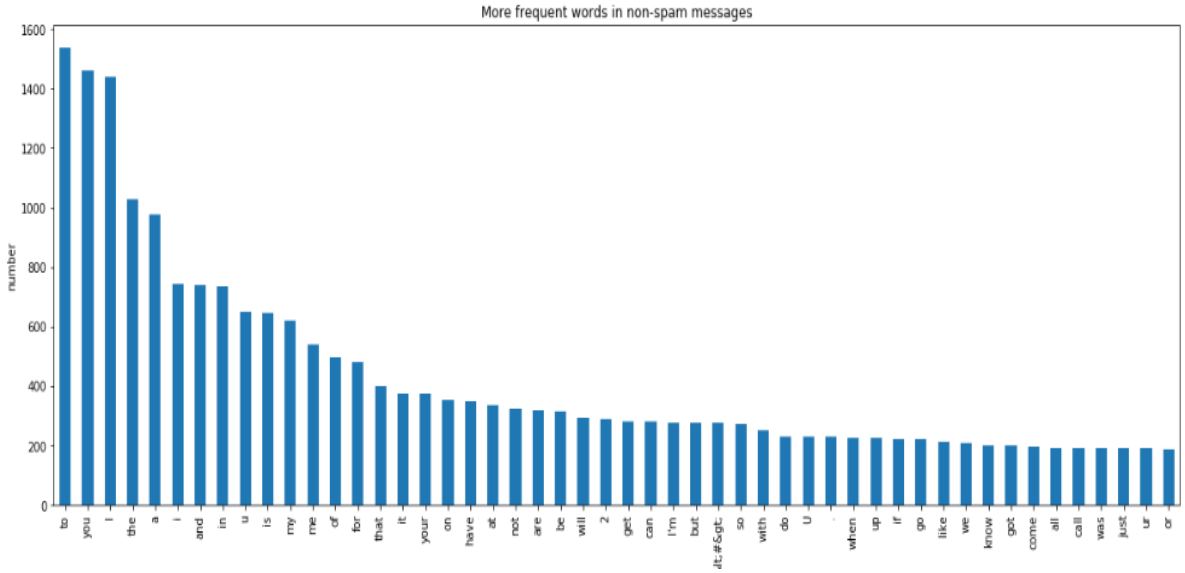
```
sms_data = pd.read_csv('SMSSpamCollection', sep="\t", header=None, encoding='latin-1', names = ['Y', 'X'])
sms_data.head(5)
```

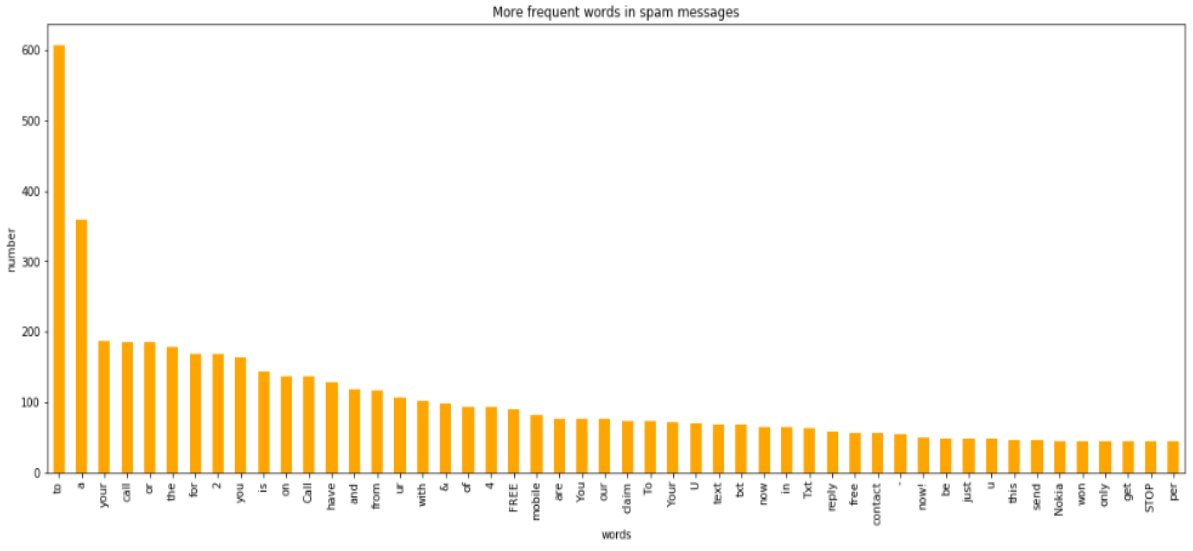
	Y	X
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

Distribution spam/non-spam plots



Frequent words in spam/non-spam messages

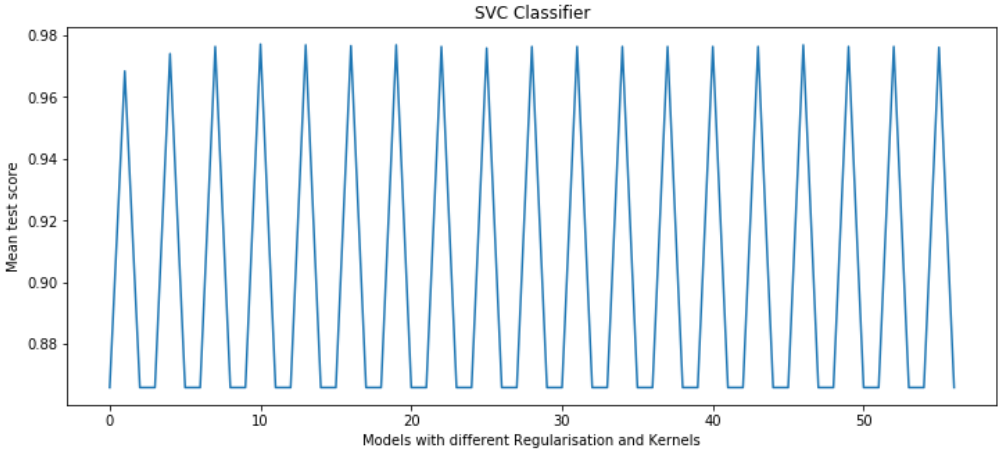




TF-IDF Vectorizer and Stopword Removal

```
#Tf-idf and Stopword Removal
from sklearn.feature_extraction.text import TfidfVectorizer
f = TfidfVectorizer(stop_words = 'english')
X = f.fit_transform(sms_data["X"])
```

Plots of SVC Classifier



Classification Report for SVC Classifier

	precision	recall	f1-score	support
ham	0.99	1.00	0.99	1448
spam	0.99	0.92	0.95	224
avg / total	0.99	0.99	0.99	1672

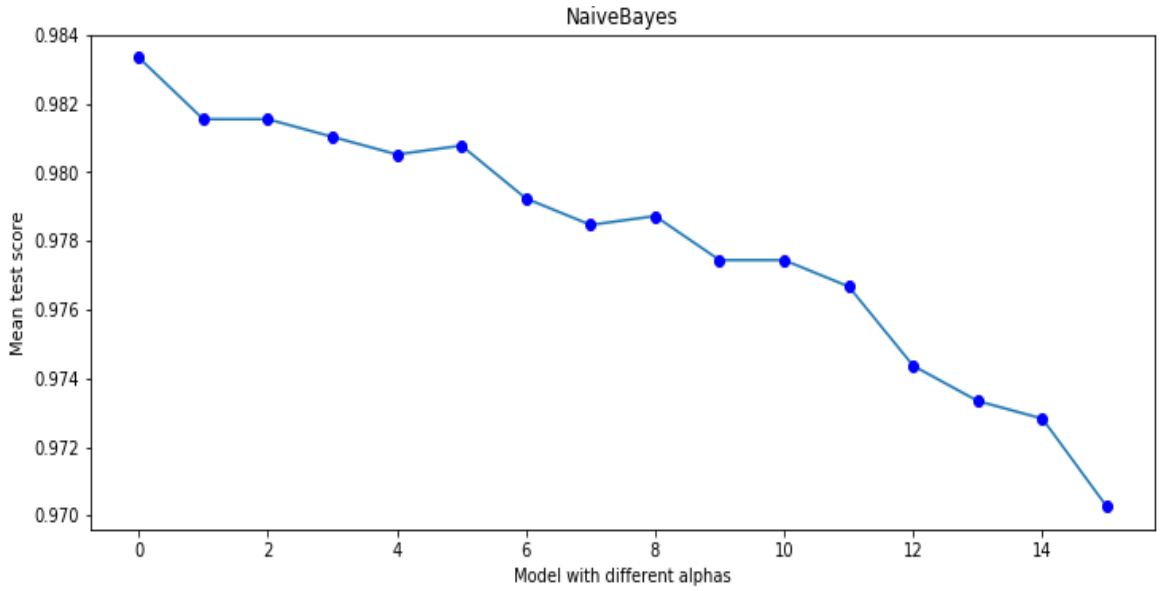
Best Score for SVC Classifier

```
print("Best score " + str(model_SVC.best_params_))
```

```
Best score {'C': 2.0, 'kernel': 'linear'}
```

Exercise 2: Compare SVM based spam filter with another model

Plots of Naïve Bayes Classifier



Classification Report for Naïve Bayes Classifier

	precision	recall	f1-score	support
ham	0.99	1.00	0.99	1448
spam	0.97	0.96	0.96	224
avg / total	0.99	0.99	0.99	1672

Best Score for Naïve Bayes Classifier

```
print("Best score " + str(model_NB.best_params_))
```

```
Best score {'alpha': 0.1}
```

The both SVM and Naïve Bayes classifier seems to be performing good. The F1 score and Recall is high for Spam than ham in SVM whereas there is better precision for NB for spam class as compared to SVM classifier.