```
## array([0.81666667, 0.83076923])
```

```
f1_score(y_test, clf_log.predict(test_x_vectors), average=None)
```

```
## array([0.68531469, 0.57943925])
```

Testing on a new set

```
test_set = ['very fun', "bad book do not buy", 'pls reply 2 this text with your valid name',
'for your inclusive text credits']
new_test = vectorizer.transform(test_set)

clf_svm.predict(new_test)
```

```
## array(['Non-Spam', 'Non-Spam', 'Spam', 'Spam'], dtype=object)
```

As you can see above that I entered 4 random messages to test the model and the model is predicting the message type correctly. Overall, I am satisfied with the model.

TUNING OUR MODEL (with Grid Search)

```
from sklearn.model_selection import GridSearchCV

parameters = {'kernel': ('linear', 'rbf'), 'C': (1,4,8,16,32)}

svc = svm.SVC()
clf = GridSearchCV(svc, parameters, cv=5)
clf.fit(train_x_vectors, y_train)
```

```
## GridSearchCV(cv=5, estimator=SVC(),
##              param_grid={'C': (1, 4, 8, 16, 32), 'kernel': ('linear', 'rbf')})
```

```
print(clf.score(test_x_vectors, y_test))
```

```
## 0.92
```

CONCLUSION

I built 4 different models (linear SMV, Decision Tree, Naive Bayes, Logistic Regression) for predicting spam vs non-spam messages. Mean accuracy and f1 score for linear SMV is the best out of the 4 models. I also tuned the model with grid search to improve the score further.