CS771A: Machine Learning Techniques
Assignment 1
Report

Deepak Kumar
(12228)

QUESTION 1: Naive Bayes Algorithm

(Using CountVectorizer, GaussianNB, MultinomialNB from scikit-learn library)

A: Direct on mail subject + body

	Gaussian	Multinomial
Part1	92.39 %	98.62 %
Part2	93.77 %	98.62 %
Part3	94.46 %	99.31 %
Part4	95.16 %	99.31 %
Part5	95.17 %	100.00 %
Part6	95.50 %	100.00 %
Part7	95.16 %	99.31 %
Part8	94.12 %	98.27 %
Part9	94.12 %	99.31 %
Part10	94.50 %	98.97 %
Average	**94.43 %**	**99.17 %**

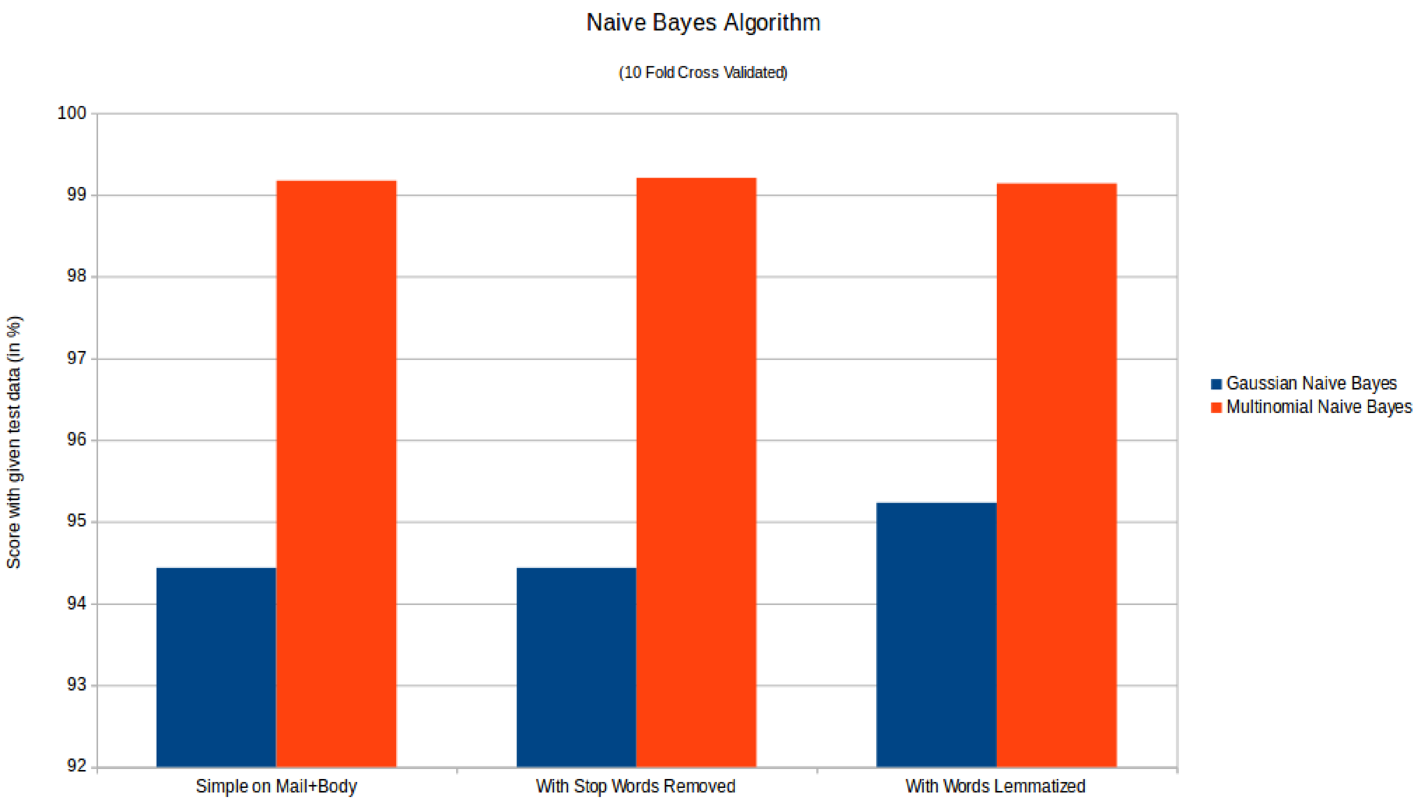
B: With Stop Words Removed (Set stop_words='english')

	Gaussian	Multinomial
Part1	92.39 %	99.31 %
Part2	93.77 %	99.31 %
Part3	94.46 %	98.96 %
Part4	95.16 %	99.65 %
Part5	95.17 %	100.00 %
Part6	95.50 %	100.00 %
Part7	95.16 %	99.65 %
Part8	94.12 %	98.62 %
Part9	94.12 %	97.58 %
Part10	94.50 %	98.97 %
Average	**94.43 %**	**99.20 %**

C: With Words Lemmatized (lemmatize using word_tokenize from nltk library)

	Gaussian	**Multinomial**
Part1	92.73 %	99.31 %
Part2	95.50 %	98.96 %
Part3	96.54 %	99.31 %
Part4	94.81 %	98.96 %
Part5	95.86 %	99.66 %
Part6	95.85 %	100.00 %
Part7	95.50 %	99.65 %
Part8	94.81 %	98.62 %
Part9	96.89 %	97.92 %
Part10	93.81 %	98.97 %
Average	**95.23 %**	**99.14 %**

Code for lemmatizing used from:
http://stackoverflow.com/questions/26126442/combining-text-stemming-and-removal-of-punctuation-in-nltk-and-scikit-learn



QUESTION 2: Bag Of Words Representation

(Using TfidfVectorizer, LinearDiscriminantAnalysis, Perceptron from skikit-learn library, tokenizing after lemmatizing)

A: Binary Bag of Words (Using Countvectorizer with binary=True)

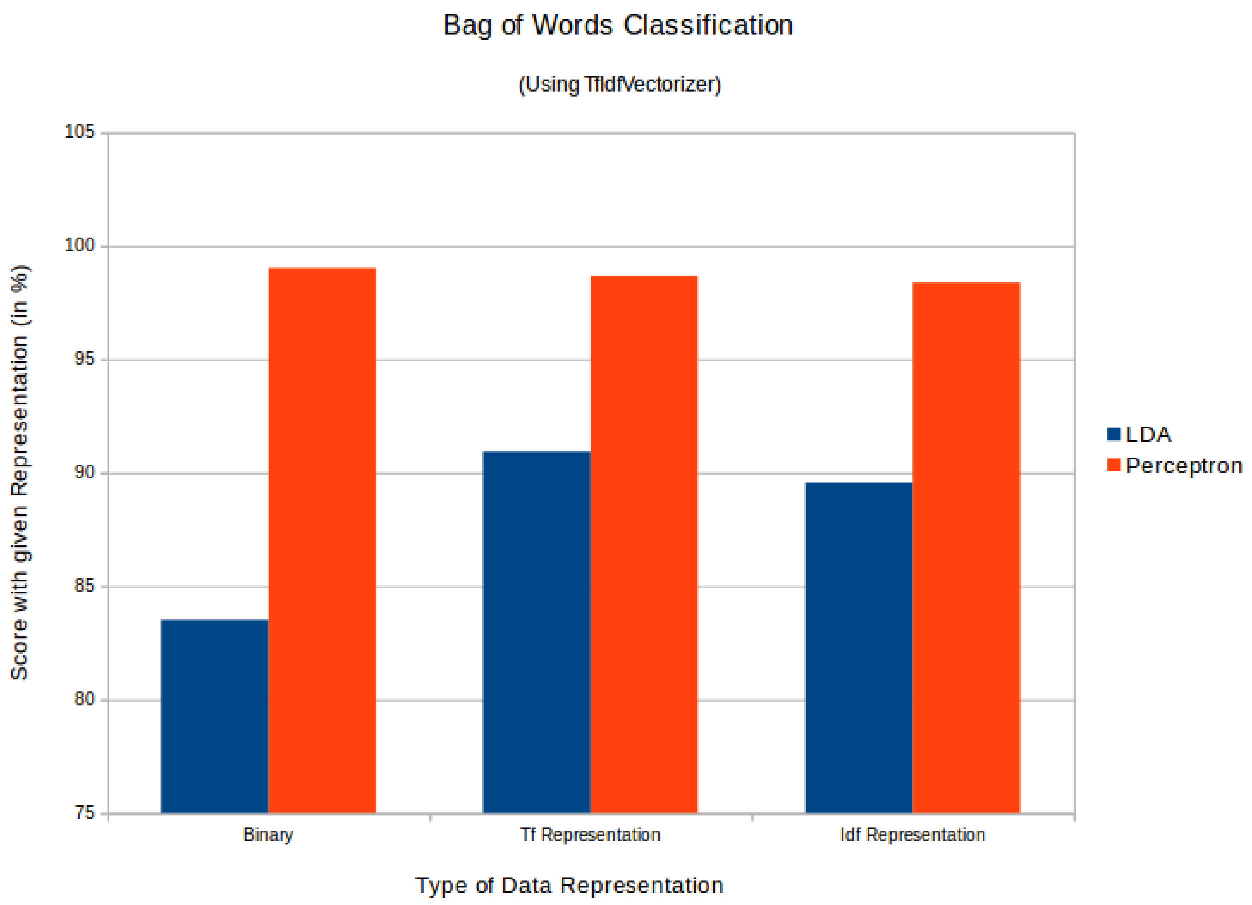
	Linear Discriminant Analysis	**Perceptron**
Part1	83.04 %	97.92 %
Part2	76.47 %	98.62 %
Part3	76.47 %	98.62 %
Part4	84.78 %	99.31 %
Part5	76.90 %	100 %
Part6	93.43 %	99.65 %
Part7	84.43 %	100 %
Part8	82.35 %	98.96 %
Part9	91.00 %	99.31 %
Part10	86.25 %	97.94 %
Average	**83.51 %**	**99.03 %**

B: Term Frequency based Bag of Words (Using TfidfVectorizer with use_idf=False)

	Linear Discriminant Analysis	**Perceptron**
Part1	97.23 %	97.92 %
Part2	94.12 %	97.23 %
Part3	93.43 %	98.62 %
Part4	91.70 %	99.31 %
Part5	96.21 %	99.31 %
Part6	71.28 %	98.27 %
Part7	83.04 %	99.65 %
Part8	94.81 %	98.27 %
Part9	96.19 %	99.65 %
Part10	91.41 %	98.63 %
Average	**90.94 %**	**98.69 %**

C: Inverse Document Frequency (with smooth_idf=True, use_idf=True)

	Linear Discriminant Analysis	**Perceptron**
Part1	94.81 %	99.31 %
Part2	92.73 %	97.23 %
Part3	92.73 %	97.92 %
Part4	91.00 %	97.92 %
Part5	95.17 %	99.31 %
Part6	66.78 %	97.92 %
Part7	82.70 %	98.62 %
Part8	94.12 %	97.92 %
Part9	93.77 %	98.62 %
Part10	91.75 %	98.97 %
Average	**89.56 %**	**98.37 %**

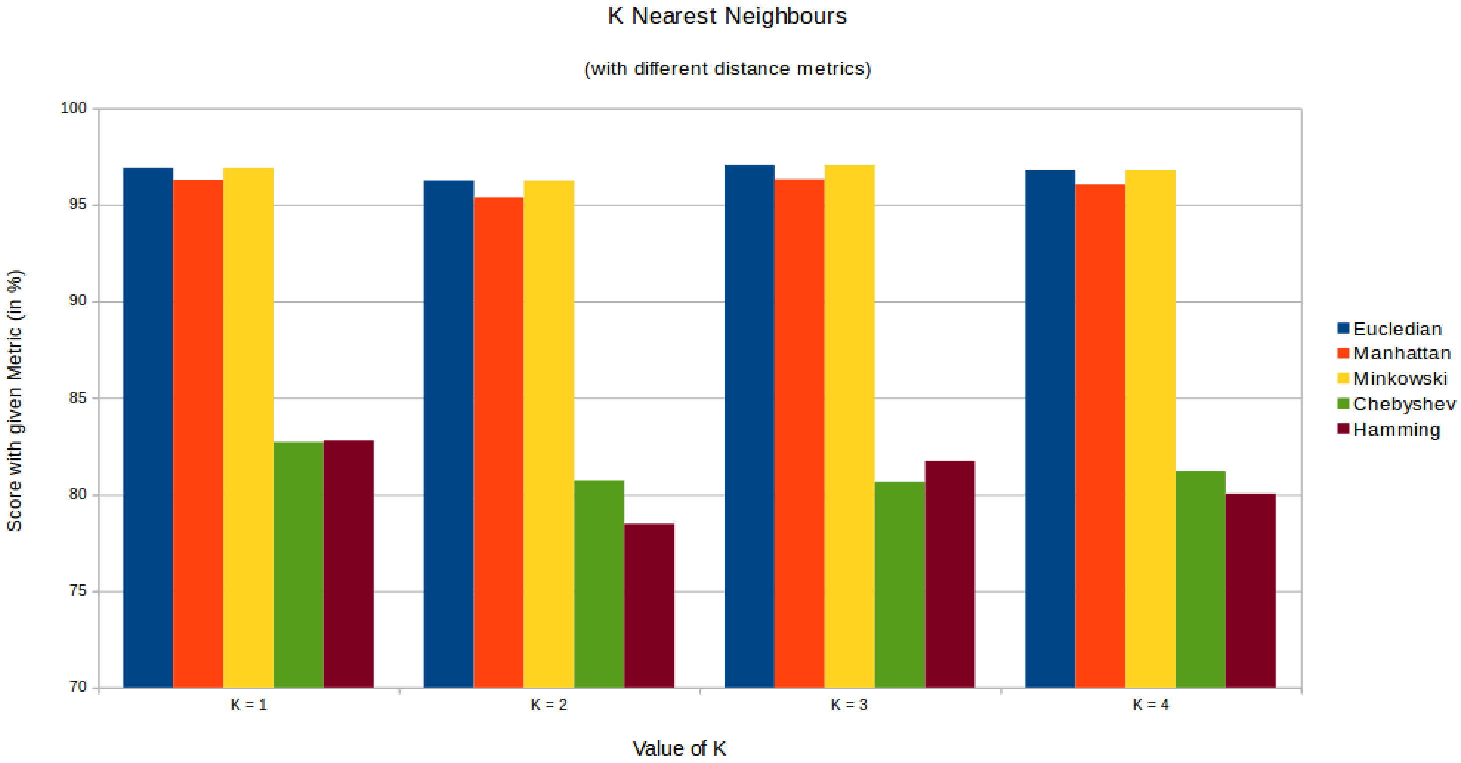


QUESTION 3: K-Nearest Neighbours

(Using KneighborsClassifier from scikit-learn library)

	Distance Metric				
	Eucledian	**Manhattan**	**Minkowski**	**Chebyshev**	**Hamming**
K = 1	96.91 %	96.31 %	96.91 %	82.71 %	82.8 %
K = 2	96.27 %	95.4 %	96.27 %	80.73 %	78.47 %
K = 3	97.05 %	96.33 %	97.05 %	80.64 %	81.73 %
K = 4	96.82 %	96.07 %	96.82 %	81.2 %	80.04 %

Code snippet to read MNIST data taken from: http://g.sweyla.com/blog/2012/mnist-numpy/


K Nearest Neighbours
(with different distance metrics)
Score with given Metric (in %)
K = 1
K = 2
K = 3
K = 4
Value of K
Eucledian
Manhattan
Minkowski
Chebyshev
Hamming