1 Text Classification using Naive Bayes

f) 52.34%

Business is hard to classify and has the worst accuracy often classifiet as Sports Sci Tech has a better accuracy but is classified as Sports as well in a lot of the cases.

1151	672	32	45
181	1671	9	39
433	768	541	158
373	798	114	615

g)

1.2 Sentiment Classification of Movie Reviews

We used the k-fold cross-validation approach to evaluate the performance of the naive Bayes classifier in the task of Sentiment Classification of Movie Reviews.

c) The accuracy values obtained for each fold and summary statistics:

	Accuracy	Mean	Standard deviation
1	0.5776	0.5743	0.02897
2	0.573		
3	0.5804		
4	0.5878		
5	0.5524		
6	0.5724		
7	0.5544		
8	0.5722		
9	0.5822		
10	0.591		

The accuracy values obtained for all folders are in range from 0.55 to 0.6 and the mean equals to 0.5743, which is not much higher than ½. Considering that we have 2 classes, we can conclude that the NB classifier implemented is just slightly better than random guessing.

d) The main advantage of using k-fold cross-validation when building predictive models and evaluating model's performance is that it allows to average results over the entire dataset. Intuition is that it is like training the model over all data we have and test on the entire dataset.
Moreover, k-fold cross-validation ensures that the characteristics of the dataset are plain over its different parts and it allows strengthening the model in the context of outliers.

e) Sentiment analysis of texts is more difficult and tricky for computational approaches compared to the topic categorization task. In the topic categorization task the text could be evaluated by presence of topic-related words, whereas sentiment classification task is complicated by semantic phenomena that change the sentiment class. For example, negation (like 'not') switches the meaning to opposite: from positive to negative or vice versa.

2 Instance-based Text Classification

Exercise 2.1: k-Nearest Neighbours Classification

a) In the problems of text classification K-Nearest Neighbours (KNN) is widely used supervised technique. It can be seen as a memory-based approach while instance-based learning is applied. The term K of K-Nearest Neighbours simply means how many nearest neighbours to be considered while classifying. The optimal selection of K is totally data dependent. Generally, a larger K reduces the effects of noise, but it turns the classification boundaries less distinct. The classification follow some necessary steps to have its functionality. Transforming the documents, reducing the dimension and pre-process them to a level so that they can be used for classification task are some of the initial steps. Each document can be represented by a vector of words. A word-by-document matrix can be used for a collection of documents, where each entry represents the occurrence of a word in a document. Weights of words in a document can be measured in several ways. Boolean weighting, term frequency and inverse document frequency can be measured as the weighting factor.

To classify a document, the KNN algorithm ranks the document's neighbours among the training document vectors, and uses the class labels of the k most similar neighbors to predict the class of the new document. The classes of these neighbors are weighted using the similarity of each neighbor to the new document, where similarity can be measured by Euclidean Distance, Hamming Distance, Manhattan Distance or Minkowski Distance.

The classifier has the ability to adapt with the new training data which allows the algorithm to respond quickly to changes in the input during real-time use.In addition, KNN does not explicitly build any model, it simply tags the new data entry based learning from historical data. New data entry would be tagged with majority class in the nearest neighbor. However, KNN has the cons of being too slow with large data and it doesn't work well with imbalanced data.

From exercise 1.1 we can see that Naive Bayes is a probabilistic estimation which is much more faster than the KNN. Naive Bayes assumes that each class is distributed according to a simple distribution and it does its functionality using the approaches like considering prior and likelihood which are completely absent in the process of KNN.

b)

1: Classify (X, Y, x) ; X: Training Data, Y: Class Label of X, x: Unknown Sample
2: **for** i =1 to m **do**
3: ComputeDistance $d(X_i, x)$
4: **end for**
5: Compute set I containing indices for the K smallest distances $d(X_i, x)$
6: **return** majority label for $\{Y_i \; where \; i \in I\}$

Algorithm: K-Nearest Neighbour algorithm

c) KNN computes the distance between the new data point with every training example for labelling the new sample. It evaluates the distance measures to learn the model parameters. The parameter K is chosen accordingly to declare number of nearest neighbours to be considered while labelling classes by majority voting.