-----Perceptron and Collaborative Filtering----

Name: Aditya Borde

Net Id: asb140930

Perceptron:

Data Set 1:**Accuracy vs Iterations:**

For η = 0.0008

	No of iterations	Accuracy After Removing Stop Words (%)			Accuracy Before Removing Stop Words (%)		
		SPAM	HAM	Overall	SPAM	HAM	Overall
1	50	31.53%	95.68%	78.24%	9.23%	97.12%	73.22%
2	100	34.61%	95.11%	78.66%	25.38%	91.66%	73.64%
3	200	40.00%	92.24%	78.03%	40.76%	91.37%	77.61%
4	300	67.69%	90.52%	84.30%	44.61%	93.67%	80.33%
5	400	66.15%	91.95%	84.93%	51.53%	89.36%	79.07%
6	500	68.57%	92.39%	87.16%	59.23%	90.20%	84.30%
7	600	70.51%	93.39%	92.16%	62.23%	86.20%	85.30%

Data Set 2:**Accuracy vs Iterations:**

For η= 0.001

	No of iterations	Accuracy After Removing Stop Words (%)			Accuracy Before Removing Stop Words (%)		
		SPAM	HAM	Overall	SPAM	HAM	Overall
1	50	69.53%	80.68%	75.24%	68.23%	77.12%	74.23%
2	100	74.61%	86.11%	78.66%	71.38%	81.66%	79.64%
3	200	78.43%	89.08%	81.68%	72.67%	80.24%	81.64%
4	300	71.92%	90.39%	83.96%	77.64%	82.81%	83.56%
5	400	73.82%	91.53%	85.74%	79.86%	85.99%	83.99%
6	500	75.51%	91.39%	85.86%	79.23%	86.20%	84.30%
7	600	77.18%	88.27%	84.64%	77.18%	91.20%	86.62%

Data Set 3:**Accuracy vs Iterations:**

For η = 0.3

	No of iterations	Accuracy After Removing Stop Words (%)			Accuracy Before Removing Stop Words (%)		
		SPAM	HAM	Overall	SPAM	HAM	Overall
1	100	93.60%	89.47%	92.44%	94.62%	84.86%	91.89%
2	200	93.86%	85.52%	91.52%	94.11%	84.86%	91.52%
3	300	94.62%	86.84%	92.44%	94.37%	82.89%	91.16%
4	400	94.91%	87.51%	92.87%	95.02%	83.08%	92.03%
5	500	93.07%	88.96%	93.28%	94.15%	85.81%	92.56%
6	600	93.68%	88.79%	93.36%	94.78%	87.67%	92.92%
7	50	95.65%	84.21%	92.44%	93.35%	84.21%	90.79%

Naïve Bayes:

Data Set 1:

Accuracy before Removing Stop Words:

1) Accuracy on Spam Mails : 98.46%
2) Accuracy on Ham Mails : 94.83%
3) Overall Accuracy on Test Set : 95.82%

Accuracy after Removing Stop Words:

1) Accuracy on Spam Mails : 98.46%
2) Accuracy on Ham Mails : 93.96%
3) Overall Accuracy on Test Set : 95.18%

Logistic Regression:

Data Set 1:

For η = 0.0008

	No of iterations	Value of λ	Accuracy After Removing Stop Words (%)			Accuracy Before Removing Stop Words (%)		
			SPAM	HAM	Overall	SPAM	HAM	Overall
1	100	0.0003	80.77%	85.63%	84.31%	73.08%	83.33%	80.54%
2	100	0.001	75.38%	89.36%	85.56%	77.69%	84.19%	82.42%
3	200	0.0003	72.58%	90.34%	85.14%	74.33%	87.67%	90.43%
4	200	0.001	78.43%	89.08%	87.68%	78.67%	85.24%	84.64%
5	300	0.0003	76.92%	93.39%	89.91%	77.64%	92.81%	90.56%
6	300	0.001	83.84%	89.08%	87.65%	75.38%	89.65%	85.77%
7	400	0.0003	82.51%	93.39%	90.16%	79.23%	86.20%	84.30%
8	400	0.001	78.92%	91.09%	89.23%	76.15%	84.54%	85.38%
9	500	0.0003	79.12%	89.16%	89.13%	78.15%	87.97%	89.64%
10	1000	0.0003	83.32%	92.81%	91.68%	82.46%	93.97%	92.74%

Conclusions:
Perceptron convergence to the minimum error value is faster than that of logistic regression. So, perceptron gives more accurate result for the lesser number of iterations and using higher learning rate as compared with logistic regression. Naïve Bayes gives more accuracy than the Logistic Regression and Perceptron in both cases considering with or without removing stop words.

Neural networks - WEKA:

Script/Commands to execute:

1) To convert data sets (training/test) into .arff:

java -cp weka.jar weka.core.converters.TextDirectoryLoader -dir<*pathContainingHamSpamDirectory*>**> finalFile.arff**

E.g.: C:\Program Files\Weka-3-6>*java -cp weka.jar weka.core.converters.TextDirectoryLoader -dir enron1_test\enron1\test > enron1_test.arff*

2) To filter the .arff files and making compatible:

java -cp weka.jar weka.filters.unsupervised.attribute.StringToWordVector -b -i<*pathOfTraining arff*> -o <*finalPathOfTrain arff*> -r <*pathOfTest arff*>-s <*finalPathOfTest arff*>

E.g.:C:\Program Files\Weka-3-6>*java -cp weka.jar weka.filters.unsupervised.attribute.StringToWordVector -b -i enron1_train.arff -o set1_enron1_train.arff -r enron1_test.arff -s set1_enron1_test.arff*

1)Data Set 1:

*Observations - on change in *Momentum*.

	No of iterations	Learning Rate	Momentum	Hidden Layers	Accuracy (Correctly classified)%
1	500	0.3	0.2	1	95.8159 %
2	500	0.3	0.2	6,4	94.9791 %
3	500	0.3	0.4	6,4	73.0126 %
4	500	0.3	0.6	6,4	72.8033 %
5	500	0.3	0.8	6,4	72.8033 %
6	1000	0.01	0.04	4,4	93.9331 %
7	1000	0.01	0.1	4,4	93.9331 %
8	1000	0.01	0.2	4,4	93.9331 %
9	1000	0.01	0.4	4,4	93.7238 %
10	1000	0.01	0.7	4,4	93.5146 %
11	1000	0.01	0.9	4,4	94.5607 %
12	1000	0.01	1.0	4,4	72.8033 %

2)**Data Set 2**:

*Observations - on change in *Hidden Layers*.

	No of iterations	Learning Rate	Momentum	Hidden Layers	Accuracy (Correctly classified)%
1	500	0.3	0.2	1	94.0789 %
2	500	0.3	0.2	4,2,4	68.2018 %
3	500	0.3	0.2	2,2,2	67.3246 %
4	500	0.3	0.2	2,2	69.5175 %
5	500	0.3	0.2	2,3,3,2	67.3246 %
6	500	0.3	0.2	8,6,8	94.0789 %
7	500	0.01	0.1	4	93.8596 %
8	1000	0.01	0.1	2,2,2	67.3246 %
9	1000	0.01	0.4	4,2,4	67.3246 %
10	1000	0.01	0.4	4,4	93.8596 %
11	1000	0.01	0.4	6,4	94.5175 %
12	1000	0.01	0.4	6,6	93.4211 %

3)**Data Set 3**:

*Observations - on change in *Learning Rate*.

	No of iterations	Learning Rate	Momentum	Hidden Layers	Accuracy (Correctly classified)%
1	500	0.01	0.5	4,4	96.6851 %
2	500	0.05	0.5	4,4	93.9227 %
3	500	0.08	0.5	4,4	97.4217 %
4	500	0.2	0.5	4,4	72.0074 %
5	500	0.4	0.5	4,4	72.0074 %
6	1000	0.01	0.5	4,4	96.6851 %
7	1000	0.05	0.5	4,4	96.5009 %
8	1000	0.08	0.5	4,4	96.8692 %
9	1000	0.2	0.5	4,4	72.0074 %
10	500	0.4	0.5	4,4	72.0074 %

Conclusions:

1)Majority times, Increase in momentum decreases the accuracy on test set. (Considering other parameter are kept constant). So, lesser momentum with higher number of iterations would probably suitable for the gaining accuracy.

2) Keeping the Learning Rate and Momentum constant,increasing the number of hidden layer will not accomplish to more accurate result. Layers with same number of perceptron at input and output layer will result more accuracy in the results.

3) When Hidden Layers and Momentum are kept constant, at particular value of Learning Rate, it gives the best accuracy than rest all. So, this follows the Water Holding Parabola curve, at certain point of learning rate it provides highest accuracy and increment and decrement in learning rate from that point will reduce the accuracy.

Perceptron Vs Neural Network:

1) Perceptron gives more accuracy than that of Neural Network. Also, Neural Network behaves similar to the single perceptron when the neural network has same number of perceptron on input and output layer keeping the learning rate and momentum less (with some minimum value).

Collaborative Filtering:

1)Mean Absolute Error = (1/n) $\sum_{i=0}^{n} e(i)$

2) Root Mean Squared Error = $\sum_{i=0}^{n} \sqrt{\frac{e(i)^2}{n}}$

Where e(i) = Abs(prediction(i) – actual value(i))

There are two possible scenarios in case of user ***i***not rated movie ***j*** , then value of V***ij*** can be considered to zero or we can calculate using some average rating to the item ***j***. I have implemented considering the first case i.e. value zero when there is no rating provided.

Values Observed From the Test Set:

1) Mean Absolute Error = 0.17432
2) Root Mean Squared Error = 0.19328