Data Mining - Assignment 4 Report

# Problem 1:

SVM - polynomial kernel with parameter 2 - average accuracy % = 61.852260198456456 %

SVM - gaussian kernel with parameter 2 - average accuracy % = 66.46453509739067 %

# Problem 2:

Kmeans algorithm is implemented for 3 different k values and executed 10 times for each k value. The average SSE for each k is listed below:

Average SSE for k = 3 is 587.611287710783

Average SSE for k = 5 is 408.9437967905991

Average SSE for k = 7 is 305.0474543165734

These averages mentioned above are not exact and change with each run, but generally they stay around the above values. This is because, kmeans algorithm takes random samples as initial centroids, and so it gets a different result every time.

# Problem 3:

The credit card dataset given for problem 3 has a massive class imbalance. The total number of samples given are 284807. Out of these less than 500 samples have a class ‘1’ i.e., a fraud transaction. Rest all are of class ‘0’ i.e., a valid transaction. So, training a classifier model in the normal may not give good results.

Out of all these samples, only 70% of data is chosen randomly as training data and the rest of 30% data is chosen as test data. While splitting this training and testing data, stratify option is chosen to not lose the ratio of class 0 to class 1 in testing and training data.

After browsing several data mining blogs and research papers on how to address class imbalance, two things have been implemented for problem3. They are:

## Oversampling:

Here, the minority class is sampled, such that we have an equal representation of both the classes. This means duplicating examples from the minority class and adding them to the training dataset. With this technique, it is essential to note that we are artificially reducing the dataset's variance. The resulting model might do poorly upon data shift or data corruption. However, implementing this is easy and will certainly give better results than an unregularized model. Class 1 samples in training data after splitting came to around 344. After oversampling, class 1 samples became 22113, which is around 10% of all training data. This percentage can be varied in the program by changing variable ‘positiveSubSetPercent’.

## Random Forest Classifier:

Random forest model is used instead of any one usual decision tree type model. Random forest model is a bagging-type ensemble (collection) of decision trees that trains several trees in parallel and uses the majority decision of the trees as the final decision of the random forest model.

By performing the above two tasks, the model F score on the test data came out as follows:

F score: 0.8592057761732852