Zaid Ur-Rehman 3955500
Zohair Aashiq 3955464

Fourth Assignment

1.

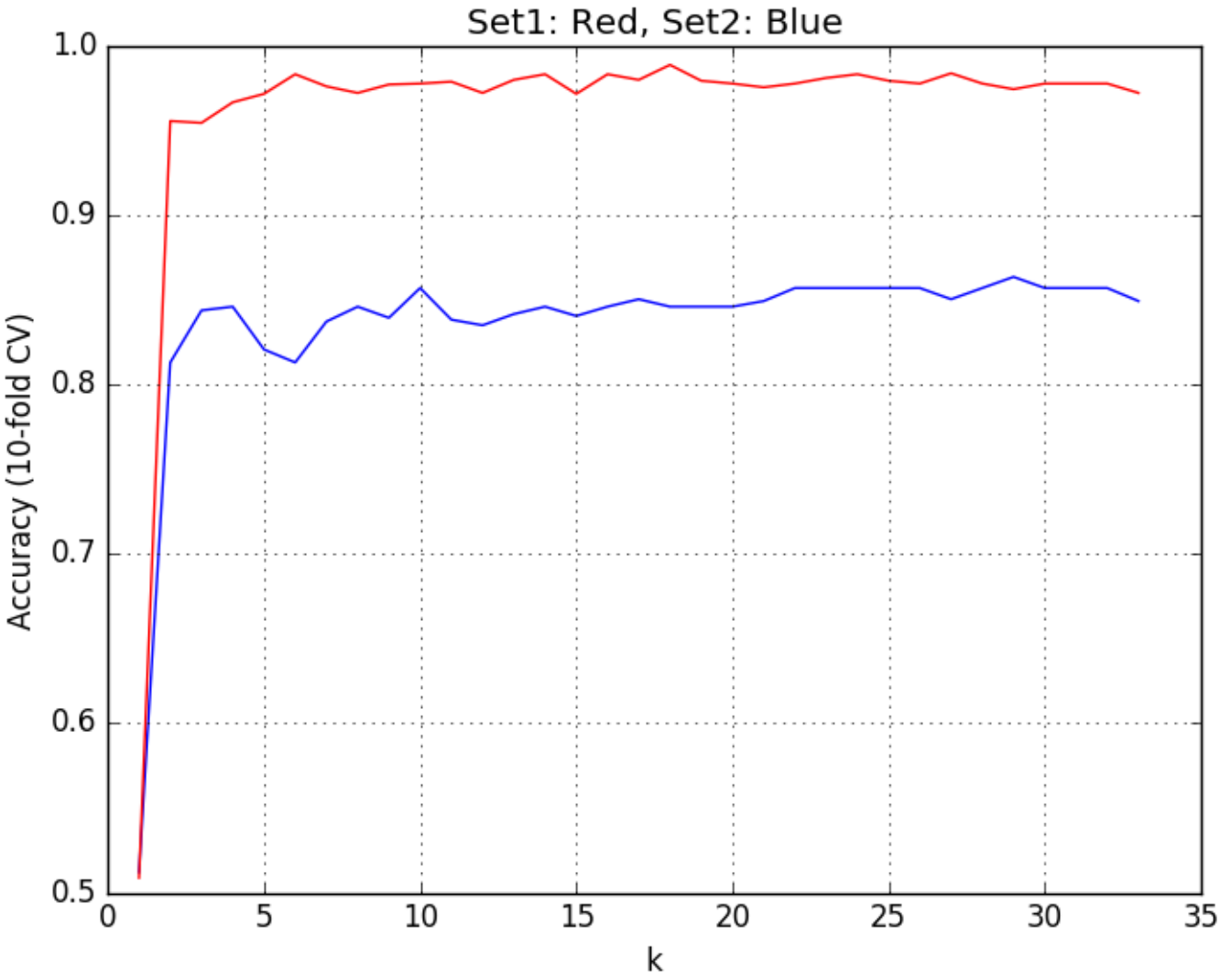
- k = 1
- Preprocessor: MinMaxScaling
- Distance criteria: Euclidean distance
- Cross validation: 10-fold
- Average over 10 iterations

Set#	**Precision Accuracy**
Set 1	0.50659
Set 2	0.51160
Set 3	0.50443
Set 4	0.49870
Set 5	0.49780

2.

- For Set1:
  - k = 2
  - Preprocessor: No feature scaling
  - Distance criteria: Euclidean Distance
  - Precision Accuracy using 10-fold cross validation and averaged over 3 iterations: 0.934065934066
- For Set2:
  - k = 17
  - Preprocessor: Min Max Scaler
  - Distance criteria: Euclidean Distance
  - Precision Accuracy using 10-fold cross validation and averaged over 3 iterations: 0.988950276243
- Plot given below
  - Set1 in Blue
  - Set2 in Red

3. Parameter configuration found by auto-sklearn is as follows.



- Accuracy = 0.520000
- Preprocessor: polynomial (degree = 2, include_bias = 'False', interaction_only = 'True')
- Rescalling: Min/Max
- Classifier: Adaboost (algorithm = 'SAMME', learning_rate = 1.105, max_depth = , n_estimators = 246, max_depth = 2)
- Balancing strategy: Weighting
- One hot encoding: use_minimum_fraction = False

4. Program name: q4.py
    - Preprocessing step: MinMaxScaler
    - Classifier: RandomForestClassifier with n_estimators=100, max_depth = 20, class_weight = {0:1, 1:2}) i.e. 1's have double weight as compared to 0's
    - Pipeline used to implement the above procedure
    - 10-fold Cross validation used to find accuracy_score
5. Feedback:
    - Q1: time ~ 4 hours, improvement: comparing our implemented k-NN with sklearn.neighbours.kNearestNeighbours and reporting the comparison
    - Q2: time ~ 3 hours, Improvement: it should have been mentioned to report average accuracies as the accuracy changes for different iterations with 10-fold CV

- Q3: time ~ 6 hours, Learned: explored auto-sklearn, Improvement: More exercises on auto-sklearn to develop better understanding
- Q4: time ~ 6 hours, improvement: data sets with more samples would have been better, and some suggestion on choice of classifiers / preprocessors and other methods as we usually work with the classifiers we have already studied in ML courses.