Data Mining Project-3 (Report)

(Outlier Detection)

Group Members:

Afroz Ahamad Siddiqui	2015A7PS0119H
Ankit Anand	2015A7PS0145H
Govind Bhambhani	2015A7PS0053H

Dataset Used:

For the purpose of the assignment, http://archive.ics.uci.edu/ml/datasets/Statlog+(German) has been used. The description of the dataset and its attributes are as follows:

Data Preprocessing:

Given dataset was already in a clean format, so much preprocessing was not required. For algorithms that need numerical attributes, Strathclyde University produced the file "german.data-numeric". This file has been edited and several indicator variables added to make it suitable for algorithms which cannot cope with categorical variables. Several attributes that are ordered categorical (such as attribute 17) have been coded as integer.

Algorithm Used:

The algorithm used is a Density based outlier detection, with Local Outlier Factor.
Local outliers: Outliers comparing to their local neighborhoods, instead of the global data distribution.

- *k-distance* of an object o, $dist_k(o)$: distance between o and its k-th NN
- *k-distance neighborhood* of o, $N_k(o) = \{o' | o' \text{ in } D, dist(o, o') \leq dist_k(o)\}$
  - $N_k(o)$ could be bigger than k since multiple objects may have identical distance to o

1

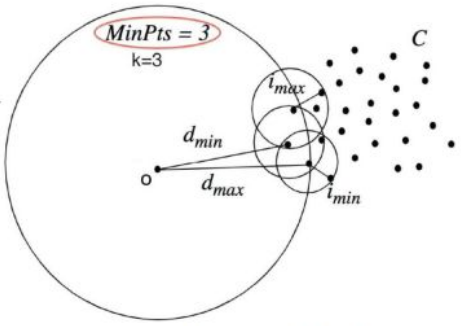
Local Outlier Factor: LOF

- Reachability distance from *o'* to *o*:

$$reachdist_k(o \leftarrow o') = \max\{dist_k(o), dist(o, o')\}$$

  - where k is a user-specified parameter
- Local reachability density of *o*:

$$lrd_k(o) = \frac{\|N_k(o)\|}{\sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)}$$



- LOF (Local outlier factor) of an object o is the average of the ratio of local reachability of *o* and those of *o*'s k-nearest neighbors

$$LOF_k(o) = \frac{\sum_{o' \in N_k(o)} \frac{lrd_k(o')}{lrd_k(o)}}{\|N_k(o)\|} = \sum_{o' \in N_k(o)} lrd_k(o') \cdot \sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)$$

- The lower the local reachability density of o, and the higher the local reachability density of the kNN of o, the higher LOF
- This captures a local outlier whose local density is relatively low comparing to the local densities of its kNN

Results:

```
~/Desktop/assignments/DM/Outlier-Detection  master  python3 detect_outlier.py
Experiment: 0 , k = 50 , num_outliers = 300
Precision 0.592 Recall 0.32 F1-Score 0.4154385964912281

Experiment: 1 , k = 100 , num_outliers = 300
Precision 0.588 Recall 0.31333333333333335 F1-Score 0.4088165680473373

Experiment: 2 , k = 200 , num_outliers = 300
Precision 0.64 Recall 0.4 F1-Score 0.4923076923076923

Experiment: 3 , k = 250 , num_outliers = 300
Precision 0.64 Recall 0.4 F1-Score 0.4923076923076923

Experiment: 4 , k = 300 , num_outliers = 300
Precision 0.64 Recall 0.4 F1-Score 0.4923076923076923

Experiment: 5 , k = 500 , num_outliers = 300
Precision 0.562 Recall 0.27 F1-Score 0.3647596153846154

Experiment: 6 , k = 50 , num_outliers = 150
Precision 0.66 Recall 0.18333333333333332 F1-Score 0.2869565217391304

Experiment: 7 , k = 100 , num_outliers = 150
Precision 0.656 Recall 0.17666666666666667 F1-Score 0.2783666933546837

Experiment: 8 , k = 200 , num_outliers = 150
Precision 0.658 Recall 0.18 F1-Score 0.28267303102625296

Experiment: 9 , k = 250 , num_outliers = 150
Precision 0.658 Recall 0.18 F1-Score 0.28267303102625296

Experiment: 10 , k = 300 , num_outliers = 150
Precision 0.658 Recall 0.18 F1-Score 0.28267303102625296

Experiment: 11 , k = 500 , num_outliers = 150
Precision 0.63 Recall 0.13333333333333333 F1-Score 0.2200873362445415
```

Some of the results are compiled here.

Experiment	K	num_outliers	Precision	Recall	F-value
1	100	300	0.59	0.31	0.41
2	200	300	0.64	0.40	0.49
3	250	300	0.64	0.41	0.50