Machine Learning

Assignment 9.1

Submitted By: Ranji Raj

December 28, 2020

a) Misleading majority class label-Target imbalance problem

- A situation where we are trying to detect two types of signals (normal (class 0) or anomaly (class 1)) in a intrusion detection system and if the proportion of class 0 is higher than that of class 1.
- In such a scenario the model gets more exposed to learn from majority since it dominates in comparison to minority class.

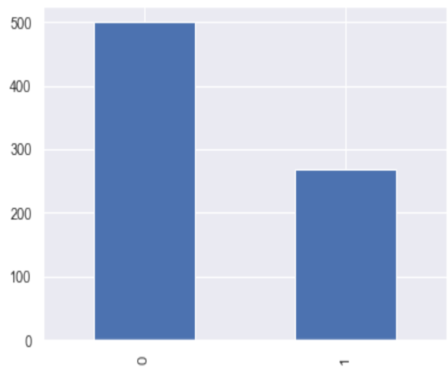

Figure 1: Target imbalance problem

b) TWO Weighting schemes

Distance Weighting

Weights the contribution of k neighbours according to their distance to the query point x_q, giving greater weight to closest neighbours.

$$f(\hat{x_q}) \leftarrow \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{k} w_i \delta(v, f(x_i))$$

where,

$$w_i = \frac{1}{d(x_q, x_i)^2}$$

Classification : To the function with the maximum value is assigned.

Attribute Weighting

To each attribute a weight is assigned, e.g.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} w_i (x_i - y_i)^2}$$

Simple approach to optimize weights for a given classification problem. Classification by adapting weights:

- *Wrong classification:* Increase all weights of attributes with large distance, decrease all weights of attributes with small distance.
- *Correct classification:* Increase all weights of attributes with small distance, decrease all weights of attributes with large distance. Learning with a small gradient in the weight update rule as:

$$w_i := w_i - \Delta \qquad w_i := w_i + \Delta$$

c) kNN as regressor-Locally Weighted Regression

- It uses **distance-weighted** training examples to form local approximation to $f(x)$.
- Given a new query instance x_q, the general approach is locally weighted regression is to construct an approximation to $\hat{f}$ that fits the training examples in the neighborhood surrounding x_q.
- This approximation is then used to calculate the value $\hat{f}(x_q)$, which is the output as the estimated target value for the query instance.
- The description of $\hat{f}$ may be deleted, because a different local approximation will be calculated for each distinct query instance.

d) Importance of normalization in kNN

- For classification algorithms like kNN we measure the distances between pairs of samples and are influenced by the measurement units also.

- For example: Let's say, we are applying kNN on a dataset having 3 features. First feature ranging from 1-10, second from 1-20 and the last one ranging from 1-1000. In this case, most of the clusters will be generated based on the last feature as the difference between 1 to 10 and 1-20 are smaller as compared to 1-1000. To avoid this miss classification, we should normalize the feature variables.

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i} \qquad \text{(Shepard's method)}$$

e) Imputing missing values by kNN-KNNImputer

- At first by using the euclidean distance metric the nearest neighbours are found.
- Each missing feature is imputed using values from nearest neighbors that have a value for the feature.
- The feature of the neighbors are averaged uniformly or weighted by distance to each neighbor.
- If a sample has more than one feature missing, then the neighbors for that sample can be different depending on the particular feature being imputed.
- When the number of available neighbors is less and there are no defined distances to the training set, the training set average for that feature is used during imputation.
- If there is at least one neighbor with a defined distance, the weighted or unweighted average of the remaining neighbors will be used during imputation.
- If a feature is always missing in training, it is removed.