UIC CS-412: IML

HW 5: Mini Project by Sankul Rawat [srawat5]

Task: Predict a person's "empathy" on a scale from 1 to 5.

Pre-Processing Steps:

1. Used Pandas for data exploration.
2. Dealt with missing values/NaN in columns such as Height, Weight etc. by replacing them with mean values. Missing values would not have contributed towards learning, so I replaced them with the mean of the values in the column. I could have gone with the mode or median but mean seemed to give the best result.
3. Removed NaN and infinite values in Empathy column (Number of NaN/Infinite values in column = 5). Did so because these examples would not contribute to the learning as their labels are missing and since their number is so less we could remove them without much affecting the size of training sample.
4. Removed categorical features (could have gone with encoding them but chose to remove them as their correlation coefficient was neither negative nor positive i.e. they were neither positive correlated nor negative correlated).
5. Removed Empathy column from the main data and kept it as label column.
6. Scaled the features as there was a huge difference between values of other features and features such as Age, Height, Weight, Siblings.

Experimental Setup:

1. The data was split into Train (80%), Development (Validation) (10%), Test (10%). Went with this scheme of splitting because there are total 1005 examples in the data and removing more than 20% for Development and Test would leave very few examples for learning.
2. Then multiple classifiers were trained over the Train data and their accuracies were compared. There was a fluctuation in the accuracy of most of them as their accuracy is dependent on the seeds of random number generator hence I also used cross-validation (3 fold) to get a stabilized result. Their accuracy was also compared to validation data set. I also compared their F1 scores to make the final decisions.
3. Then the best model (Multi-Layer Perceptron) was picked and its hyperparameters were tuned on validation data set.
4. SelectKBest and GridSearchCV were used for hyperparameter tuning. They were used because they automate the entire process of looping through different parameters and comparing cross-validation accuracies. 10-fold cross-validation was used with GridSearchCV.

Solution:

1. After testing on multiple introductory (Naïve Bayes) and state-of-art (MLP, Ensemble) classifiers it was found that MLP would produce the best result on this data for the said task. Hence after tuning, we got an Accuracy of 52% on Development Data and an accuracy of 40% and an F1 score of 0.34 on Test Data.

Drawbacks & Some examples that got wrong and what to do to correct them:

1. As we can see from the classification report F1 score for the class 1 and 2 is 0. This is so because the data is imbalanced and the number of class 1 and 2 examples are very less. To overcome this we can do resampling (if we want to overcome at data level) or we can use ensemble with bagging and boosting.
2. MLP cannot guarantee to stop at global minima. Due to this the error (say RMS) might end up being high. One resolution is to train it multiple times every time starting from different positions and then pick the one with the best performance.
3. The number of hidden neurons must be set by the user and it is easy to underfit if it is low or to overfit if it is too high. This can be resolved by training it multiple times with a different number of hidden neurons and plotting the performance and identifying the ideal number of neurons for the task.

Repo link: https://github.com/dark-shade/CS-412-IML-HW5-Mini-Project