CS 412 Introduction to Machine Learning, Fall 2018
University of Illinois at Chicago

Homework 5: Mini-project

By Mayank K Rastogi

Problem Statement

You are working for a non-profit that is recruiting student volunteers to help with Alzheimer's patients. You have been tasked with predicting how suitable a person is for this task by predicting how empathetic he or she is. Using the Young People Survey dataset (https://www.kaggle.com/miroslavsabo/young-people-survey/), predict a person's "Empathy" as either "very empathetic" (answers 4 and 5) or "not very empathetic" (answers 1, 2, and 3). You can use any of the other attributes in the dataset to make this prediction; however, you should not handpick the predictive features but let an algorithm select them.

Approach

The program uses Python packages such as `scikit-learn`, `pandas`, and `numpy`. Scikit-learn was chosen because it has a good collection of machine learning algorithms that were taught to us in the class, and offers easy APIs for creating machine learning workflows using Pipelines. Using a combination of the above packages, a machine learning workflow was created with the following steps – binarization of dependent variable, splitting into train and test set, missing value treatment, encoding categorical variables, scaling, feature selection, training models with K-fold cross-validation, and finally scoring the best model on the test set.

Data Preprocessing

The supplied dataset contained 1107 samples with 150 features, including the dependent variable (Empathy), out of which 11 were categorical, and most of the remaining features were ordinal – ranging from 1 (lowest) to 5 (highest). Many of these features had missing values, including 5 rows with missing values for the "Empathy" feature.

The rows with missing values for "Empathy" were dropped at the time of loading the dataset since this is our target variable which will be predicted by our model. Additionally, the "Empathy" column was binarized to two classes – 0 and 1 – such that, answers 1, 2, and 3 were mapped to 0, and answers 4 and 5 were mapped to 1. The dataset was then split into training and test sets in 80%-20% ratio. The missing values in the independent variables were imputed with the "mode" of the respective feature since the number of missing values were less and the features were either ordinal or categorical.

The categorical features were transformed using one-hot encoding with the help of pandas' `get_dummies()` function. One-hot encoding was chosen over label encoding, since the categorical variables were not ordinal in nature, e.g. for "Gender", the presence or absence of "male" or "female", using one-hot encoding, makes more sense than saying "male" is superior to "female", if a label encoder was used. The next step was to apply min-max scaling to all the variables to constrain them in the range of 0 to 1 to account for the difference in scales across features, e.g. values of "Music" feature range between 1 to 5, whereas the values for "Weight" range from 41 to 165.

Next, feature selection was done by training a random forest classifier on the train set and then using the feature importance scores to select the most important features. Doing so helps reduce overfitting since duplicate or correlated features get removed by this approach.

Model Selection

Multiple models were trained on the test set using 8 different algorithms – most-frequent (baseline) classifier, KNN, Logistic Regression, Gaussian Naïve Bayes, Perceptron, Decision Tree, Random Forest, and SVM. Parameter tuning was done using GridSearchCV and StratifiedKFold cross-validation technique with K=8 folds, such that each fold sampled about 110 responses. The stratified K-fold technique ensured that the percentage of samples from each class is preserved across folds. Accuracy was chosen as the scoring metric and the cross-validation scores were used to select the best performing model.

Results

With an accuracy of 74.5% on 8-fold cross-validation, **Random Forest** was the best performing model on this dataset, with **SVM** using *RBF kernel*, being the second-best model with an accuracy of 73.5%. Scoring the random forest model on the **test set** gave an accuracy of **68.16%**. In contrast, the most-frequent (baseline) classifier scored an accuracy of 59.7% on the test set.

Find the project source code on ***GitHub****:*
https://github.com/mayankrastogi/empathy-classification

References

- pandas.get_dummies – pandas 0.23.4 documentation
- python - Impute categorical missing values in scikit-learn - Stack Overflow; answer by sveitser
- How to use pd.get_dummies() with the test set – FastML; by Zygmunt Z.
- Documentation scikit-learn: machine learning in Python
- Feature selection – scikit-learn 0.20.1 documentation
- Selecting good features – Part III: random Forests; by Ando Saabas;
- Cross validation – scikit-learn 0.20.1 documentation
- Tuning the hyper-parameters of an estimator – scikit-learn 0.20.1 documentation