Alexandria University
Faculty of Engineering
Computer and Systems Engineering Dept
.

Prof. Dr. Mohamed Abougabal
Prof Dr. Nagwa M. El-Makky
Dr. Marwan Torki
Special Topics in Information Systems

Data Classification

Session # 3 (2018-2019)

Objectives

1. Applying preprocessing techniques learnt before and see their effects on classification accuracy
2. Exploring different classification models and performing tuning of their parameters
3. Exploring different techniques for evaluating classification models
4. Learning how to analyze observed results and explain observations in a detailed report.

Problem Statement

Given the MAGIC gamma telescope dataset that can be obtained using the link below. https://archive.ics.uci.edu/ml/datasets/MAGIC+Gamma+Telescope. This dataset is generated to simulate registration of high energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope using the imaging technique. The dataset consists of two classes; gammas (signal) and hadrons (background). There are 12332 gamma events and 6688 hadron events. You are required to apply preprocessing techniques on this dataset and use the preprocessed dataset to construct different classification models such as **Decision Trees**, **Naïve Bayes Classifier, Random Forests, AdaBoost**, **K-Nearest Neighbor (K-NN)** and **Support Vector Machines (SVM)**. You are also required to tune the parameters of these models, compare the performance of the learned models before and after preprocessing and compare the performance of models with each other.

Lab session

I.Data Balancing and Visualization

Note that the dataset is class-imbalanced. To balance the dataset, randomly put aside the extra readings for the gamma "g" class to make both classes equal in size.

Visualize the dataset using different ways (e.g., histograms, box plots, scatter plots, line plot, correlation matrix, etc.). Use the most discriminative plots and discuss your observations on the data in your report.

II.Data Split

Split your dataset randomly so that the training set would form 70% of the dataset and the testing set would form 30% of it.

III.Preprocessing

Apply any **required** preprocessing techniques (e.g., feature selection/ feature projection, feature normalization, etc.) on both the training and testing sets.

IV. Classification

Choose 5 classifiers from the following models to apply on your dataset, tune parameter(s) (if any), compare performance of each model before and after preprocessing and compare the performance of models with each other

1. *Decision Tree*
   Parameters to be tuned: None
2. *AdaBoost*
   Parameter to be tuned: n_estimators
3. *K-Nearest Neighbor (K-NN)*
   Parameter to be tuned: K
4. *Random Forests*
   Parameter to be tuned: n_estimators
5. *Support Vector Machines (SVM) -linear Kernel*
   Parameter to be tuned: C
6. *Naïve Bayes*
   Parameters to be tuned: None

V. Model Parameter Tuning

As discussed in class, use cross-validation to tune the parameters of classifiers. Test the models trained with best obtained parameter values on the separate testing set.

VI. Report Requirements

For all the requirements mentioned above, you should report the model accuracy, precision, recall and F-measure as well as the resultant confusion matrix using the testing data.

Comment on all visualizations and comparisons made.

Bonus

Students with best 3 F-Score values will get a bonus

VII. Notes

- **This lab session needs more time. So, try to start working on it early.**
- You should work in groups of two. Each student should answer any questions in the lab session.
- You should deliver well documented code as well as a report showing all your work and conclusions.
- Copied assignments will be penalized; so not delivering the assignment would be much better.
- You should write your code in python.

VIII. References

[1] Chapters 8 and 9 of the first reference (J. Han, M. Kamber and J. Pie, "Data Mining: Concepts and Techniques", 3rd Edition, Morgan Kaufmann, 2012).
[2] S. Raschka. Python Machine Learning. Packt Publishing, 2016.
[3] Dua Dheeru and Efi Karra Taniskidou. UCI Machine Learning Repository (2017). http://archive.ics.uci.edu/ml