MLDM Group Assignment

Overview and rules

In this assignment, you will solve a machine learning task on real-world data. In detail, the goal is to predict survival on the Titanic; i.e., based on passengers' features, you want to predict whether the person will survive or not. To that end, you should implement a Decision Tree Classifier, and compare achieved results with existing implementations. As always, proper pre-processing, data handling, and evaluation is crucial. The goal of this task is to get confident with the usual machine learning pipeline, and get some deeper knowledge about machine learning algorithms.

The task needs to be solved in **groups of 3-4 students**. The solution is to be written in Python and a single jupyter notebook is to be sent by email to philipp.singer@gesis.org, florian.lemmerich@gesis.org, and vasilev@uni-koblenz.de until **Sunday, 15th of January, 23:59h**. Use [ML-Assignment] as the email subject and do not forget to denote the name of all contributing students in the notebook as well as in the email. The deadline is strict!

After submission, we will conduct **exercise interviews** where each group will be asked about their work in individual slots. Note that each student of a group needs to be familiar about all aspects of the submission. The maximum grade for the submission will be **20% of the course grade** (with the other 80% coming from the course exam)—note that grades might differ for individual students of a group based on performance in the interviews.

Remember that there is no excuse for **plagiarism** and we will consider those submissions as failed. Also, every student of a group is responsible for plagiarism. This does not mean though that you are not allowed to research, get motivated, and influence by other work. However, it is always important to **properly reference and cite work (including code snippets) by others** that you utilize throughout your work, and you need to fully understand all aspects.

Task[1]

The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.
One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class. In this assignment, you should predict which passengers survived the tragedy using a Decision Tree Classifier.

Task 1: Preprocessing and data analysis

As a first step, you should load the provided data. After loading, you should aim at getting a first overview of the data[2]. Study how the data (features) looks like; e.g., by plotting. We then ask you to provide a short summary of your observations, eventually coupled with expectations about the prediction task with respect to which features might contribute. An example of how such an initial analysis could look like can be found on the Kaggle site[3]. You should also think about further data-preprocessing steps that might be necessary, such as handling categorical variables.

Task 2: Prediction

Next, you should aim at predicting survival given the features. To that end, you should use the scikit-learn library and a Decision Tree Classifier[4]. Perform at least the following steps:

- Fit the classifier on training data
- Predict survival on test data
- Evaluate the classifier's performance appropriately (also think about things like cross validation)
- Consider accuracy improvements (e.g., parameter tuning via grid search)
- Compare the performance to an appropriate baseline

[1] The provided task is motivated by the introductory Kaggle challenge presented at `https://www.kaggle.com/c/titanic`. Some text snippets are direct references.The link also provides a multitude of resources to get a better overview of the task at hand.

[2] Description of the fields is provided in `https://www.kaggle.com/c/titanic/data`

[3] `https://www.kaggle.com/omarelgabry/titanic/a-journey-through-titanic/comments`

[4] `http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html`

- Print the classifiers decision rules (or also visualize them) on the complete data.
- Interpret and discuss feature importance

Task 3: Implement your own Decision Tree Classifier

In this task, you should implement your own Decision Tree Classifier for the prediction task at hand.

- Implement the Decision Tree Classifier. You should implement it as a Python class using the default scikit-learn API; i.e., the class should have at least a fit and predict function. For examples refer to the online documentation[5].
- As evaluation criterion you should implement both gini impurity, and information gain (entropy).
- For building the tree, you can use any decision tree algorithm (ID3, C4.5, CART, etc.); an overview and comparison can be found online[6]. Also check a series of Youtube tutorial videos[7] and Google.
- Fit the classifier on training data
- Predict survival on test data
- Evaluate the classifier's performance appropriately
- Compare the results to those obtained by the scikit-learn implementation. Check for discrepancies and consider the parameters of the scikit-learn implementation to rule out certain differences. In case of differences, try to elaborate on them.
- Print the classifiers decision rules (or also visualize them) on the complete data.

Bonus

We might provide bonus points to extraordinary submissions. Examples of bonus work could e.g., include (but is not limited to): improving the classifier performance by utilization of other algorithms and/or feature/parameter tuning, the extension of the decision tree implementation to random forests, other extensions like pruning, termination criteria for tree construction or consideration of further algorithm parameters, or the study of other prediction problems (maybe also considering multivariate outcomes).
We also want to encourage students to submit their solutions to Kaggle and research other challenges that might encourage further studies.

[5] `http://scikit-learn.org/stable/developers/contributing.html#rolling-your-own-estimator`

[6] `http://www.cs.uvm.edu/~icdm/algorithms/10Algorithms-08.pdf`

[7] `https://www.youtube.com/watch?v=eKD5gxPPeY0`

Help

We want to encourage you to actively use the newsgroup in case of any questions. Also, help and respond to questions of other students, and also share resources that you find useful. We might also reward very active students in the newsgroup.
We will also have a dedicated tutorial session after new year for questions. We will also offer an online chat/video call session sometime in the Christmas break for students to ask questions and communicate; details will follow.