Classifying Poker hands using Weka

Poker is a family of card games. Each player participating in a Poker game is dealt five cards which is also called a *hand*. Though Poker is a complex game, it can be said that the success or failure of a player depends largely on the quality of their hand. Each hand can be classified as one of nine well defined classes. The aim of this project is to develop good quality classifiers for classifying Poker hands. It is of course a trivial task to write a program that takes a Poker hand as an input and outputs its class. But our aim is to see whether classifiers can discover patterns and learn to classify Poker hands. We will use a dataset from the Machine Learning Repository of University of California at Irvine. I have kept a local copy of this dataset. The dataset has two parts, a training set and a testing set. Even though Weka performs a 10-fold cross validation while training, a large training set may cause overfitting, i.e., the classifier will work well for the training set, but may not work well for a separate test set. Here are the local copies of the training and test sets:

Training data
Testing data
Explanation of the attributes

Project tasks:

The purpose of the project is to design classifiers for the Poker dataset. Similar to many data analysis tasks, it is impossible to say which classifier will be appropriate for this dataset. Moreover, it is impossible to say whether the training data set will cause overfitting.

- You should try to explore at least two, preferably three classifiers for this project. We have covered mainly four classifiers in the lectures, *decision tree*, *naive Bayesian*, *artificial neural networks* and *support vector machine*. You can use these or other classifiers available in Weka for the project. However, you have to write a brief explanation of classifiers that you use (so that I can get a feel how well you have understood the classifier). The explanations could be intuitive and one paragraph each will suffice. You should use the training dataset for training the classifiers.
- Next, you should take a small part of the test dataset for testing the classifiers. I would suggest taking 10 sets of 5000 records chosen from the test set. You have to of course remove the class labels from the test sets and eventually compare the class labels that the classifier(s) are giving with the actual class labels and report the accuracy of the classifiers.
- You should check whether there is any overfitting due to the large training set. This will require you to create smaller training sets. However, this is a bit tricky as you should maintain a correct or appropriate proportion of the different hands in each training set. This will require you to write a program in your favourite language. You should perform an analysis with at least three different traing sets for each classifier and report the results.
- The project can be done in groups of at most two students.
- Marking of the project will be comparative, i.e., your project will be judged against the best submitted project(s). The tasks I have mentioned are just the minimum. You will get a higher mark if your analysis is thorough. The marking of this project will be harder compared to the first project.