CSE 4334/5334 – Data Mining

Spring 2015 – Programming Assignment 2

Due: 11:59pm Central Time, Thursday, March 12th, 2015

General Requirements:

Read the following requirements carefully, and make sure you follow every rule. If you fail to meet some requirement, points will be deducted accordingly.

- This assignment must be done individually. You must implement the whole assignment by yourself. Academic dishonesty will have serious consequences.
- You are encouraged to discuss the assignment with other students, but you are not allowed to disclose your solution.
- Your source code must pass compilation. Any non-executable submission is not acceptable and will receive a zero grade.

Problem Scenario:

You will implement a classification algorithm to obtain the required results in this contest on Kaggle: http://www.kaggle.com/c/forest-cover-type-prediction. You are required to build a classifier using the provided train.csv (labels of instances provided) and determine the forest type for all the instances in the provided test.csv (labels of instances not provided). (The terminology of "training" and "test" here is from Kaggle's viewpoint, in that they can use test.csv to evaluate your classification results, since they know the true labels for instances in test.csv. However, in developing and evaluating your classification method, you would need to consider train.csv as given data and partition it into training and test sets, as we explained in lectures.)

Tasks:

1. You need to create an account with Kaggle.com and accept their terms of usage, before you can download their datasets.
2. Download train.csv (http://www.kaggle.com/c/forest-cover-type-prediction/download/train.csv.zip) and test.csv (http://www.kaggle.com/c/forest-cover-type-prediction/download/test.csv.zip).
3. Build a classifier using train.csv (once again, you are supposed to split train.csv into training and test data). The class attribute is "Cover_Type".
4. Predict the class labels of all instances in test.csv. The results should be stored in a file named result.csv. The file should follow the format specified at http://www.kaggle.com/c/forest-cover-type-prediction/details/evaluation.

(See http://www.kaggle.com/c/forest-cover-type-prediction/download/sampleSubmission.csv.zip for a sample file.)

5. Submit your result.csv to Kaggle, by following the instructions at http://www.kaggle.com/c/forest-cover-type-prediction/submit. (To see this page, you need to be logged in.)
6. Kaggle will rank your results and return a rank and a score.

Required Submissions:

You are required to submit a single .zip file into Programming Assignment 2's entry in Blackboard. The .zip file must contain the following files.

1. Source code in a **single** .java or .py file.
   1.1. This assignment must be done using either Java or python. In case of Java, you must use version JDK8; in case of Python, you must use Python 3.4.
   1.2. You can only use standard libraries in Java or Python. You can use Python libraries such as numpy, scipy, sympy for data structures and calculations. But you are not allowed to use classification functions or similarity measures existing in any library. The same rule applies for Java.
   1.3. Your java program must be compiled and executed using the following syntax:

   javac <program>.java

   java <program>.class

   1.4. Your Python program must be executed using the following syntax:
   Python <program>.py
   1.5. Your Java or Python program must read train.csv and test.csv from the same folder as where the above commands are executed and it must write result.csv into the same folder.

2. Documentation in a single .pdf file. The documentation should include the following information:
   2.1. Your Kaggle username;
   2.2. Your Kaggle rank and score for the given task;
   2.3. Screenshot that shows your Kaggle username, rank and score;
   2.4. Design and implementation details of your classification algorithm. This section should have at least 300 words to cover enough details.

Evaluation:

We will execute your program and attempt to replicate the result.csv your submit. We will then submit the result.csv to Kaggle in order to verify your score. You will be evaluated based on your Kaggle score and the design/implementation quality of your program.