Machine Learning from Data – IDC
HW1 – Linear Regression

*** **This assignment can be submitted in pairs.**

In this assignment you will implement a Linear Regression algorithm that uses Gradient Descent. In order to do so you need to first install WEKA:

1. Download WEKA from this link.
2. Run the .exe file that you downloaded and install WEKA.

Prepare your Eclipse project:

1. Create a project in eclipse called HomeWork1.
2. Create a package called HomeWork1.
3. Move the LinearRegression.java and MainHW1.java that you downloaded from the Moodle into this package.
4. Add WEKA to the project:
   a. Right click on the java library that is used and click Build path -> configure build path.
   b. On right hand side click 'add external jar'.
   c. Select the file weka.jar that is in the Weka program directory that was created during the installation.

The 'wind' data consists of 14 features (3 date features & 11 wind speed features) and the target – wind at MAL Station. <u>The goal is to predict the wind speed in MAL Station with the best 3 features, from amongst the 14 available, and using a linear model</u>.
In order to do so you will need to implement the Linear Regression Gradient Descent algorithm.
First, you should run the model with all the features in order to find the best learning rate alpha.
You should also calculate the Training & Test error after learning with all the features.
Second, you will use the alpha that you found in the previous step in order to build Linear Regression models each time with different 3 features. For each combination of 3 features you should calculate the training error. The best 3 features are the ones that will give you the lowest training error. They represent our best choice of 3 out of the 14 features, based on the training data.
For these features, and the associated coefficients, you should calculate the test error as well.
We provide an excel file for the results (you should copy the results to the excel file).

Implement the Linear Regression Gradient Descent algorithm:

1. In LinearRegression.java fill in the code for the methods:
   a. private double[] gradientDescent(Instances trainingData) – where you will implement the gradient descent algorithm learned in class. The method should return the weights (or coefficients) θ of a linear regression predictor.
   b. public double regressionPrediction(Instance instance) – which takes as an input an instance of data and uses the coefficients that were calculated already to return the linear regression prediction on that instance.
   c. public double calculateMSE(Instances data) – which takes as an input the all data instances and returns the average squared error of that predictor on the data. As in class, the average squared error is the total squared error divided by the number of data instances multiplied by 2.
   d. You should think how to edit the buildClassifier method and when to call the findAlpha method.

   Choosing alpha:
   Implement the choosing alpha phase in the findAlpha method.
   Create a for loop with a variable i which goes from -17 up to 0. Set alpha equal to 3^i and run gradient descent with this value for 20,000 iterations. Every 100 iterations calculate the new error and compare it to the previous error (100 iterations ago). If the current error is bigger than the previous error stop the iterations and return the previous error. If you finish the 20,000 iteration the reported error will be the last one. For each alpha you got the best error from your gradient descent algorithm. The alpha which gave you the lowest error, will be the chosen alpha.

   Stopping condition for gradient descent:
   When running gradient descent, after choosing alpha, you should calculate the error every 100 iterations and compare it to the error you calculated 100 iterations ago. If the difference between the errors is very small, say smaller than 0.003, then stop, otherwise continue.

2. Now you can use the methods you wrote in order to predict the wind speed in MAL Station. Included in the folder are training and test data files which include information on many wind measurements – all explained in the text file (open it with your favorite text editor).

3. In your main method:
   a. Load the wind training data using the loadData(String fileName) provided.
   b. Build Linear Regression model with all the attributes and find the best alpha as well as the training and test error.
   c. Iterate over all 3 features combination and train your linear regression predictor by running the public void buildClassifier(Instances data) each time with 3 different features. Use the alpha that you found in the previous step and the stopping condition as above.
   d. After finding the best 3 features calculate the mean squared error on the training & test data, considering all sets of 3 features tested.
   e. Print to the console the following:
      The chosen alpha is: <best alpha>
      Training error with all features is: <Training error with all features >
      Test error with all features is: <Test error with all features >
      List of all combination of 3 features and the training error
      Training error the features <best 3 features >: <Training error>
      Test error the features <best 3 features >: < Test error>
   f. Copy the results to the excel file in the appropriate cells and calculate the Mean, Std & Median of the training error for all the 3 features combinations.

You should hand in a LinearRegression.java, MainHW1.java and hw1results.xls files which the grader will use to test your implementation. All of these files should be placed in a hw_1_##id1##_##id2##.zip folder with the id of both of the members of the team.

*** Submitting in groups on Moodle does not work. Please only submit one zip folder per pair