NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Tutorial on Weka

http://www.cs.wekato.ac.in/ccd/weka/

Hello and welcome to this tutorial on Weka. Weka is an open source and freely available software package containing a collection of machine learning algorithms. The algorithms present in Weka are all coded in java and they can be used by calling them from your own java pod. However with also provides a graphical user interface from which the algorithms can directly be applied to data sets. For the programming assignments in the introduction to machine learning course, we will mostly be using Weka in its new I phone.

This will allow us to spend more time on understanding how the algorithms which we come across in the lectures actually work and how to use them in analyzing data. You can download different versions of Weka for different operating systems from the website of the University of a Cartoon. This tutorial is mainly aimed at people who have never used by far before we will look at some of the basic features and options provided by the software and also do some linear regression experiments to help you in solving the questions in the third assignment.

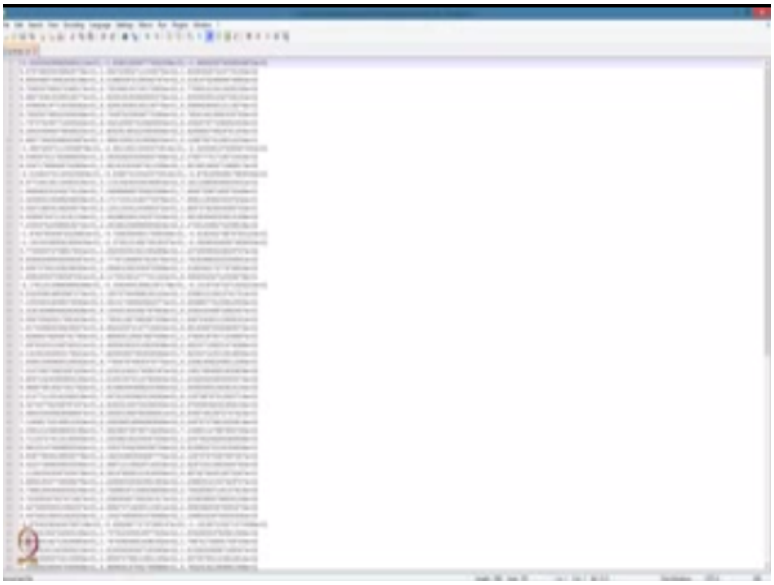
Before we start with Weka let us create a synthetic data set on, which we can then apply linear regression. Since this is just for illustration purposes we will create a simple one dimensional data set. We will create this data head using a few lines of Python code, so here we have important a numpy package. The statement creates the input data which ranges from -25 to 100 and consists of 100 data points. Now we will get the output data which will have a linear relation with the input.

Since we are creating a data set here we know how the input and the output are related. However if this data was given to us then our objective would be to try to run this teach, this relation that is the parameters β not equal to 1 and β 1 equals 3. As we will see when this input that is xy pair problem is provided to the linear regression algorithm. It will be able to learn a perfect model;

this is because there is no noise in our data. So to make things a bit more challenging we will add some noise to the output.

The variable z is essentially the noise character output we will be used noise with parameter 0 and 3. We will now save this data and apply linear regression on it using Weka. We have saved our data in a text file.

(Refer Slide Time: 04:09)

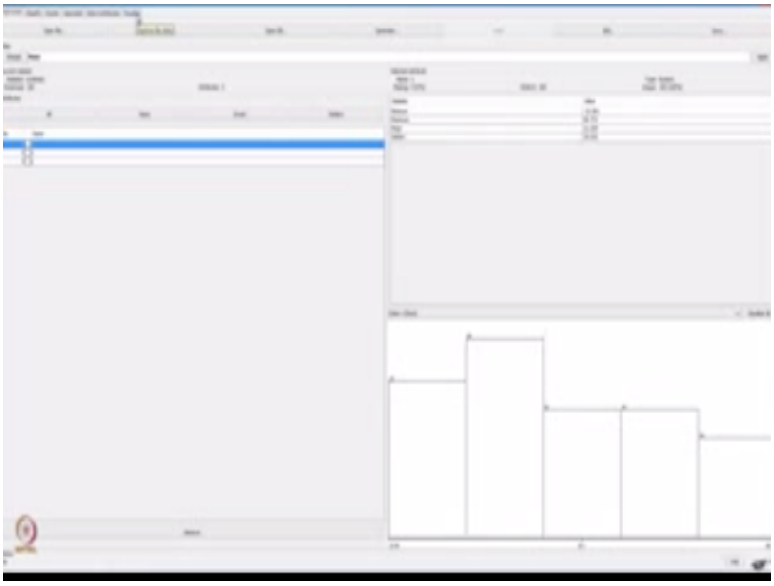


So let us have a look now Weka uses a specific format for its input it this format is known as a RFF and we have to add a few bits of information to what is essentially a CSV file to make it suitable for use in the car. We essentially have to provide three pieces of information this first one just gives a name to the data. Eventually specifying what relation this data is showing, so since we have cooked up this data. We are just given it the name synthetic next we provide the attribute information.

We have used x y and z as the names of the three columns and specified the data type as numeric. There are other data types which will be seen a little later such as nominal and string data type. The final piece that has to be provided is the data which we have already listed and data specify the start of the data. Now we should save this in the ARX a format that is with the extension dot a RF. Hopefully this gives you an idea of how to represent data in the AR FF format suitable for use with Weka. β

Just to recap you have to provide the relation and attribute information and the data is listed row wise that is each row specifies one data point with the values being separated by commas. We will now open this data in Weka and apply linear regression on it.

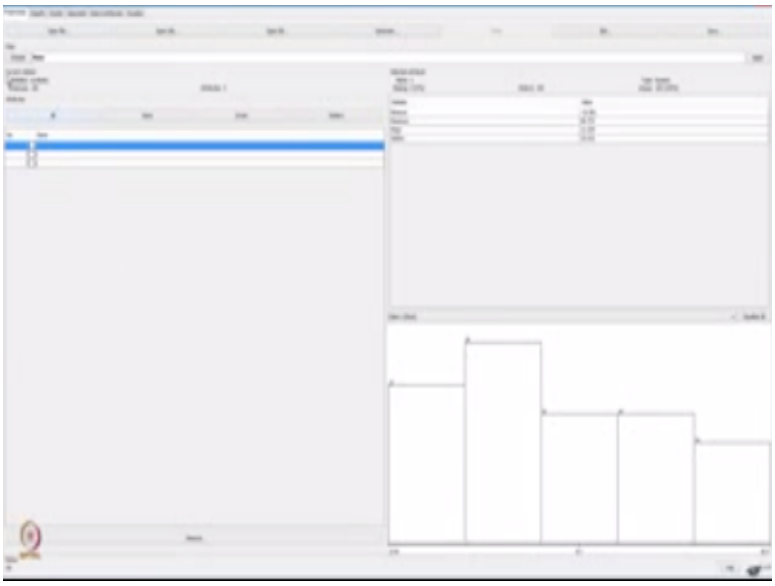
(Refer Slide Time: 06:17)



So this is the opening screen for the Weka application. We will be using explorer right so this the start screen for the explorer application. As you can notice most of the options are grayed out this is because we do not have any data selected yet. So let us do that this is the synthetic data that we had just created, so let us open that right a lot of things to notice here. First of all at the top we have these tabs which allow us to specify different actions. So the first tab is pre process where we can do different pre processing activities such as normalizing the data, filling in missing values, in case the input as missing values and so on.

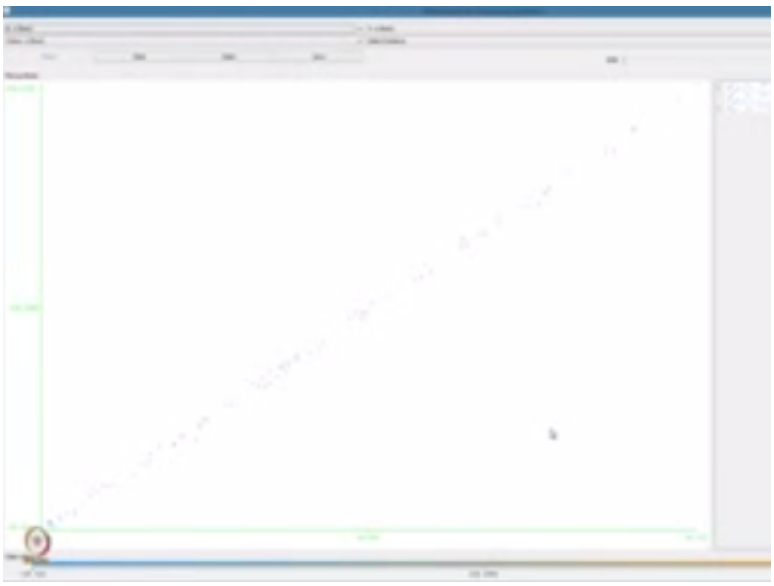
Under the classified tab are listed all the supervised learning algorithms, that is both classification and regression algorithms. Which will be looking at soon clustering algorithm was listed on the cluster tab. Association rule mining algorithms under the associate tab in under the select attributes tab. We can perform attribute selection activities such as is subset selection ECA and soon and finally visualize allows us to visualize the data.

(Refer Slide Time: 07:42)



Let us have a look at that here we have the scatter plots between each pair of attributes in the data; this allows us to visualize the distribution of the data. For example if we look at the scatter plot between y and x, we can observe the perfect linear relation between the two variables. Since that is how we created the data, however if we look at the scatter plot between z and x.

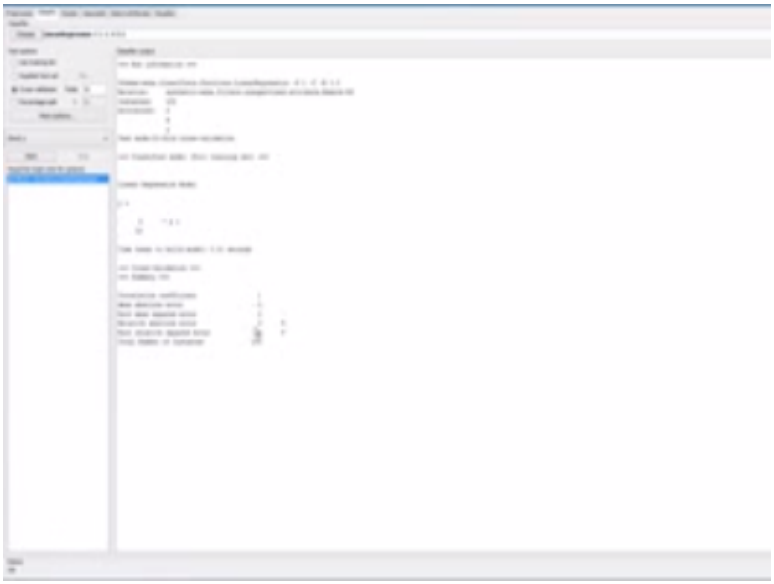
(Refer Slide Time: 08:10)



We can see the effect of adding noise to the output getting back to the three process tab here. We have the relation information the name of the relation is synthetic num there are 100 instances and three attributes. The attribute window gives the list out the attributes. We have three attributes x y and z, on the right-hand side for the selected attribute, we can use some information. So right now the a droid x is selected, so it has 0% missing data there are 100 distinct values with each value being unique. Some basic statistics which has mean maximum mean and standard deviation and at the bottom we have a histogram.

Now let us go ahead and apply linear regression on the data. For our first attempt we will use the first two columns that is x and y and remove the noise character output set right. So first we come to the classified tab.

(Refer Slide Time: 09:17)



Here we have to choose the algorithm, so which we will choose functions and linear regression note that there is a simple linear regression function which is actually suitable for this specific task because we have only one dimensional input but when the dimensionality of the input is more, we need the linear regression function. So let us just use this function these are the default

parameters, for the linear regression function. We can change them by clicking here; the first parameter is the attribute selection method that is the method used to eliminate attributes.

Which not contribute to the learning of the model? Since we would like to handle this ourselves, we will select no attribute selection we will not be using the debug mode the third attribute the third parameter is eliminate collinear attributes which essentially allows Welka to identify and remove attributes, which have a high correlation. We will set this too false for now the final parameter is the value of the regularization parameter note, that we are using Ridge regularization here.

So initially we will take this to zero, that is we will not be using any regularization having set the parameters of the linear regression algorithm. We now look at the different evaluation options, the first option is to use the training data that is we use the training data to build a model and then use the same data to evaluate the model. In case we have a separate data set for testing that is a portion of the data which has not been used in training the model. We can supply that here.

More common you will be using cross-validation, in cross-validation. We will iteratively partition the training data into testing and training slits each iteration, we will train on the training split and evaluate on the testing split the partition will be done in such a way that each data point will appear in the tree, in the testing split at least once. The purpose of doing this is to get a robust estimation of the performance of the model. We will be discussing the concept of cross-validation in more detail in upcoming lectures.

For now we can go ahead and use this evaluation method note that the number of folds simply indicates the number of your iterations and the sea and the sizes of each player here. We have ten folds which mean we will partition, the data and build models ten times with each partition being a 90,10 split between training interesting. Only the percentage slit option allows us to split the training data and keep a portion for testing. Next this drop-down box allows us to select the output attribute.

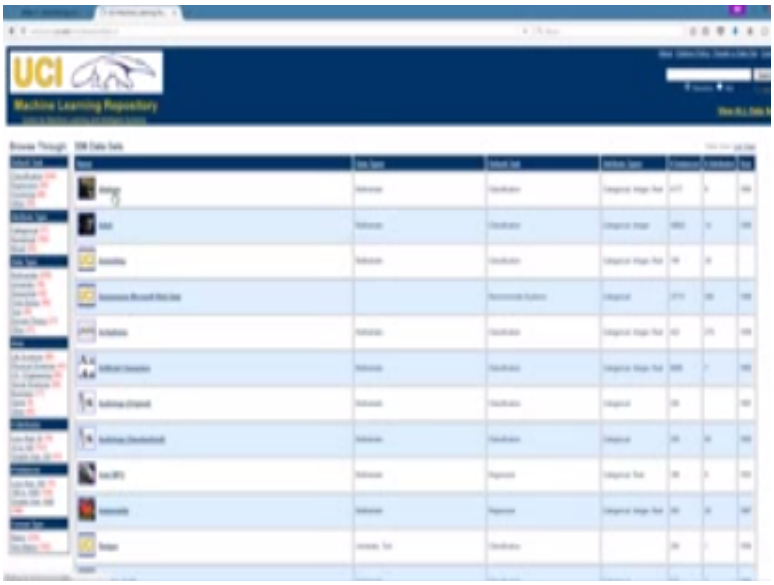
That is the attribute which we are trying to predict with all settings in place we can execute. The main window displays the results of the execution of the algorithm. Going through the output we see the linear regression function used with the following parameters the relation synthetic on which the filter applied is removing of the column 3 there are 100 instances two attributes

namely x and y we use 10-foldcross-validation for evaluation and this is the linear regression model, that we obtained. Note that we are able to recover, the exact parameters values used in constructing the data.

The correlation coefficient is 1the correlation coefficient specifies, the correlation between the predicted output and the actual output here it is perfectly correlated and each error term is 0. Now for more meaningful evaluation we will use the noise corrupted output. So we can recover the column which we deleted by selecting the undo button and now we will remove the y column going back to classify. We will have all the parameters will leave a same and just execute the word.

Note that Welka use the Z attribute for prediction in general Welka will use the last attribute listed for prediction or classification. Here we observe the effect of noise in the output variable. The parameters estimated parameters differ from the actual parameters, due to the noise. We also observe that the noise the error is no longer zero, hopefully you will now be a little comfortable with the vector interface and the steps involved in performing linear regression on the data set up to. Now we have worked with a very simple synthetic data, so next we will apply regression on the more realistic data set.

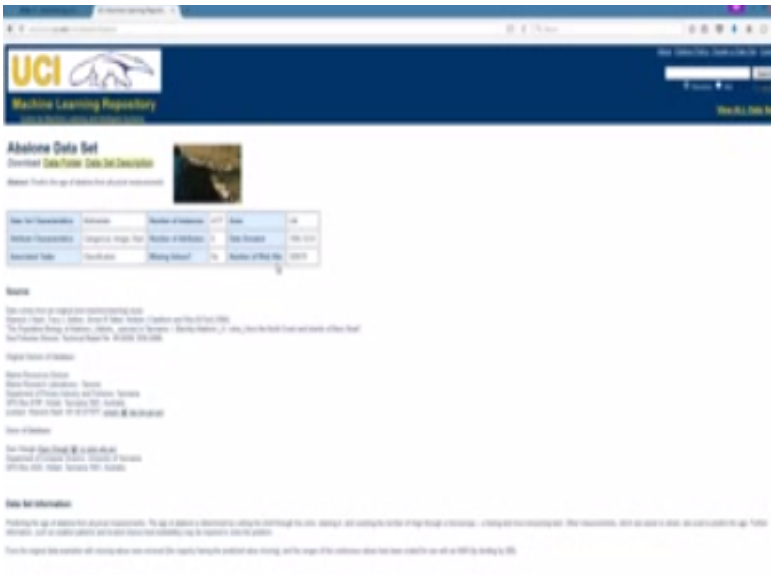
(Refer Slide Time: 15:08)



The UCI machine learning repository is home to a number of real-world data. Here you can see the different data sets characterized by the primary tasks. They support the type of data they

contain the area from which the data has been generated and so on. We will use the abalone data set for our next experiment.

(Refer Slide Time: 15:36)

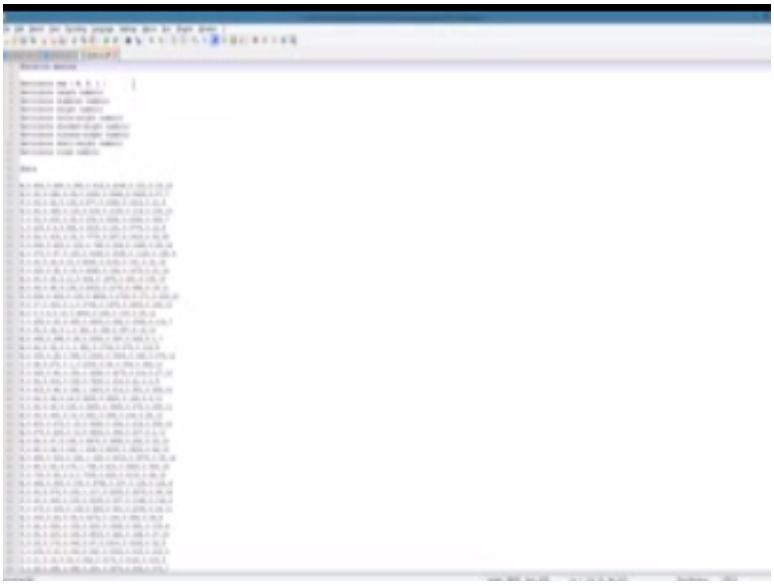


Here you can see the basic information about the data set such as the associated tasks the number of instances a number of attributes whether the dataset contains any missing values and so on. The date description page contains more detailed information.

(Refer Slide Time: 15:56)

Most importantly lists out the attributes and their data types. This information will be needed for creating the AR FF input format.

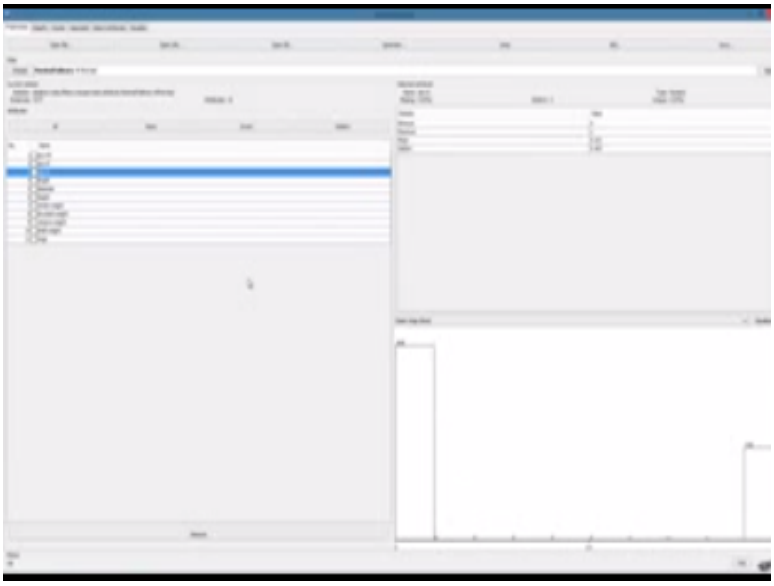
(Refer Slide Time: 16:11)



This is the raw data and here we have added the information necessary for the AR FF representation note, that this data set contains categorical attributes. For such attributes we specify their data type by listing all the possible values they can take. In this case the first attribute can take one of three possible values and fri, note also that the last attribute is of integer type ranging from 1 to 29. Here we have specified it as a numeric attribute but in case, we wanted to consider this as the class level for a classification task.

We would have specified it as a categorical attribute. Let us get back to occur and apply linear regression on this data set here.

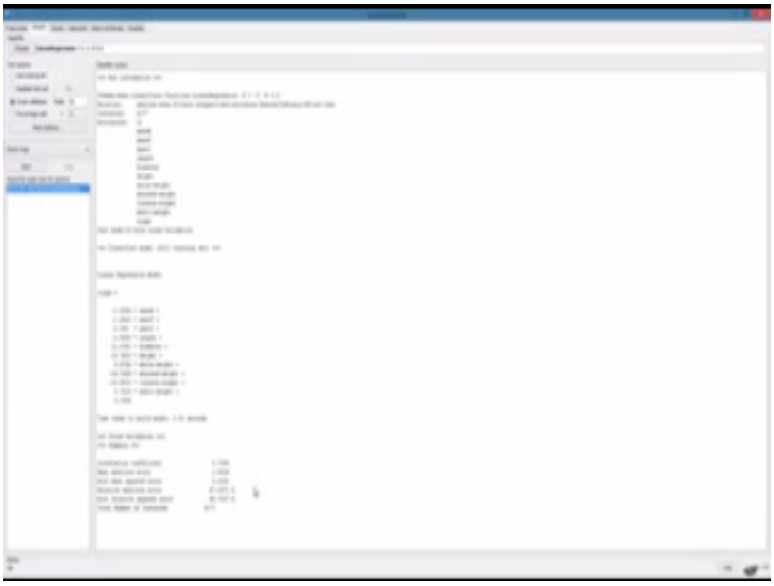
(Refer Slide Time: 17:07)



We see the nine attributes along with the associated information on the right, the first thing to do is to handle the categorical attribute, recall from the lectures we learned about one hot encoding this can be done in the pre pro stab using an appropriate filter. First of all we select the attribute then choose the appropriate filter this filter comes under the unsupervised attribute folder and it is the nominal to binary filter. On applying this filter we observe that from one attribute we have now created three attributes.

One corresponding to each of the possible values that the original attribute would have taken also from the histograms, we can see that each of the attribute is now 01 or that is binary variable, we can now apply linear regression on this data set.

(Refer Slide Time: 18:09)



We will start by using the same parameter values, as used previously we will stick with cross validation for evaluation and try and predict the Rings attribute. Here we observe the results these are the estimated β parameters and below. We see the error measures the question here is can, we improve this result. One idea is to apply regularization, however before applying regularization, we should normalize the data since rigorous rich regression is not invariant, to scale.

This can be done in the preprocessed tab by selecting the normalization filter note that except for the nominal attributes and the output attribute each of the other attributes has been normalized to arrange within 0 & 1 now we can try rigid regression. Let us start with a value of 0.5, however to compare this result we should apply linear regression on the normalized data. So we will run linear regression again with H parameter.

Set to zero here we have the result of running late linear regression with regularization parameter set to zero. On the normalized data looking at the β parameters as well as the error values. We notice some slight changes especially if we look at the β parameters, for some of the attributes we see that in the result for rigid regression the parameter values have shrunk this is understandable, since in which regression we are adding a penalty term ways to are on the magnitude of the parameters.

We can also observe a slight change in there or terms by trying out different values of the regularization parameter, we can attempt to improve on the performance however trying out

parameters manually is not feasible for this, we will use a meta learning algorithm called CV parameter selection essentially CV parameter selection will take as input or learning algorithm. A parameter of that algorithm and a range of values to try out for that parameter, let us specify this, so first we select the algorithm which is linear regression.

We set its parameters notice that the regularization parameter is specified with the letter R, so this will allow us to specify the regularization parameter along with the range. So let us say we want to vary the regulation parameter between 0 and let us say 5 in 50 steps. To stick with the same cross-validation valuation and execute the result of the execution of the β learning algorithm shows the optimal value of the regularization parameter swallows subject to the range constraints provided by us.

The β parameters and the corresponding error measures are shown here comparing, this result with the results of the previous two executions, where the reversion parameter was set to 0 and 0.5. We see that there are small changes but not nothing drastic this seems to suggest that regulation does not seem to have much effect on this model or perhaps. That we have not found the right range of parameters in case of the latter. We can run the β learning algorithm again and specify a larger range.

One very useful technique when searching for the optimal parameters for any learning algorithm is to start with a large range and the large step size, this initial step performs a coarse-grained search over the range of parameter values. Next we perform a fine-grained search in the vicinity of the value which gave the best results in the previous step, in that is we restrict the range but decrease the step size. We can leave it to you to apply these two-stage parameters each on this data set this concludes the tutorial on Welka.

We hope that people encountering Welka for the first time will feel a really comfortable with the basic features of the software in this tutorial we covered most of the concepts which will be required for the first set of programming assignment questions in future assignments we will simply mention the algorithms that need to be used and expect you to apply them on the datasets applied using Welka. We also encourage you to explore the and algorithms provided in the package as and when we cover them and related one in class for this the UCI machine learning repository is a very good source for data sets that can be used for all kinds of learning experiments.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in