**Final Project**

**TAI CS549**

**Project Documentation**

**Topic:** The topic I have chosen as my final project is unsupervised simple competitive learning model. This model was discussed in the class of CS549 as part of the module 8. Unsupervised simple competitive learning is a way of grouping of data based on the dataset given and the weights between the datasets and the output groups. Learning coefficient is also set before starting the training, this learning coefficient will be used to update weights after each data set is trained and processed. The problem I have chosen Is early stage diabetes risk prediction. This topic is important and using unsupervised neural network for each and every data which is of patients can be organized into groups and predictions and proper actions can be taken (precautions) based on the symptoms of patients and proper care can be provided. This project represents one of the way in which Artificial Intelligence is helping to solve issues but not only after the problem is identified but before it is even there by predicting it beforehand.

**Dataset:** The data set used for this project is from UCI machine Learning Repository. The dataset is the real time in person taken from Sylhet Diabetes Hospital in Sylhet, Bangladesh and approved by a doctor. There are in total 17 attributes in this dataset but I have taken out only 14 attributes based on my requirements and they are as follows Polyuria ,Polydipsia ,sudden weight loss , weakness, Polyphagia, Genital thrush, visual blurring, Itching, Irritability, delayed healing , partial paresis , muscle stiness , Alopecia , Obesity . All the attributes are of type Boolean and the data was in Yes or No format for Boolean format. I have changed the values for Yes to 1 and No to 0. The number of datasets provided is 520 but I have removed the dataset which has NO for all the above listed attributes making it to 467. The link for the dataset I have chosen is: <https://archive.ics.uci.edu/ml/machine-learning-databases/00529/diabetes_data_upload.csv> . The dataset was first processed and was kept in json file from which it is copied on to the project file. The dataset is stored in the multidimensional array in java program file.

**Description of the project:** the programs works like this, it has been given with the 467 datasets in the multidimensional array format. It is then provided with initial weights for three groups of output. Which has about 14 values each for the 14 values of each dataset for each of the group. So three groups and 14 values which is we have provided 38 weights. We also have an array called group with size 467 which stores record of the group each dataset belongs to. After each iteration set of iteration this values gets updated based on the weights changed. Next there is an iteration loop which iterates the number of time we want to train the network . lets see what it iterated every single time. There is this nested loop which iterates through the number of dataset provided. For each dataset we calculate the equation 1 . from that the highest value are computed using standard competitive learning rule (with normalization step) and only for that group weights are updated. And then for the next iteration the updated weights are used if needed. The grouping of the dataset is also updated in this phase. After each iteration updated weights are printed out. this process is repeated for all the iteration we have provided and tries to update the weight and eventually update the grouping of the datasets provided. In Implementation of the project I have not used any of the libraries and have used java as the core program. The program was developed using the netbeans IDE and no machine learning libraries were used in development of the program.

**Results:**

The programs gives out the output for the values of equation 1 for each and every dataset provided and then followed by the updated weights and then we print out the grouping which is the group in which each and every dataset belongs to. The entire things listed above is printed for all the number of iterations provided by the developer. At the end of all the iteration we get the groups in which each of the dataset provided.

Example first 5 dataset values for the output can be :

0.528 - 0.433 - 0.876 are the equation 1 values for data set 1

0.424 - 0.272 - 0.495 are the equation 1 values for data set 2

0.723 - 0.466 - 0.69 are the equation 1 values for data set 3

0.492 - 0.436 - 0.377 are the equation 1 values for data set 4

0.934 - 1.002 - 0.997 are the equation 1 values for data set 5

Updated weights can be :

0.006 ; 0.047 ; = 0.117

0.001 ; 0.011 ; = 0.117

0.006 ; 0.001 ; = 0.133

0.151 ; 0.068 ; = 0.095

Grouping can be :

dataset 1 belongs to Group: 3

dataset 2 belongs to Group: 3

dataset 3 belongs to Group: 1

dataset 4 belongs to Group: 1

dataset 5 belongs to Group: 2

**Conclusion:**

Datasets from the real work can be used in this amazing computational models and Can be implemented in the real world entities to be prepared for the upcoming challenges. If the initial proper weights are chosen and applied to the neural network with the right learning coefficient we can get the best output from the system. Grouping the data into three different groups.

**Follow-up:** The programming language used in the project is java and there are no libraries which are used in this project. Coming to the completion of the project, After the final iteration I think the data is properly arranged into the groups but still of the initial weights are more properly chosen and the learning coefficient is chosen properly then the output can still be optimized.