Machine Learning Assignment 3

Manas Gaur
Wright State University

November 11, 2017

1 Introduction

Neural Networks are a class of supervised learning algorithms popular for pattern recognition by modeling the latent features in their hidden layers. Neural Networks (a.k.a Multilayer perceptron) have random weighted connections between the input and hidden layer, and hidden layer and output layer. These weights can be initialized randomly or uniformly (normal distribution). These complex network-based supervised classification algorithms have been developed upon the formulation of some simple supervised classifiers. Logistic Regression and Perceptron are the principle models over which the Neural Networks are developed with an extension of hidden layer and hidden neurons. These extensional features of Neural Network are a part their hyper-parameters and varying their values, influences the power of the Neural Networks.
In this assignment, we will test the power of Neural Networks by varying their hyper-parameters, varying the dataset values (pre-processing) and finally assessing these changes in terms of accuracy (hit rate), confusion matrix and ROC (Receiver Operating Characteristic) Curves. The flow of the assignment is structured as follows; First, we will provide an overview of the two datasets that will be used in the subsequent experiments. Second, we will see the performance of the Neural Networks on these two datasets, keeping these datasets un-normalized. Third, we will perform the normalization over the dataset and see its impact in terms of classification accuracy and confusion matrix. Finally, we will perform regularization of the stochastic gradient functions in Neural Network and repeat the experiment described in the third section.

2 About the Wine Dataset

In this section, we will be providing some descriptive statistics about the datasets that will be used in the subsequent experiments. We will be using two datasets in this assignment, viz. Wine dataset (UCI Respository) and WPBC flu dataset.

2.1 Wine Dataset

This dataset is three class dataset class 1, class 2, and class 3 having 178 samples and 14 attributes. Out of these 14 attributes, we have 13 independent features and 1 target feature, called *Class*. The descriptive statistics of this dataset has been described in table 1. The features which are made bold-faced in the table 1, are the candidates for normalization. This is because their statistical values are relatively outlier with respect to other features. The samples in the dataset are distributed among three classes as, Class 1: 59 samples, Class 2: 71 samples and Class 3: 48 samples.

2.1.1 Box plot Analysis of Wine Dataset

Before we develop a learning model for the wine dataset, it is an encouraging method of performing a box plot analysis of individual features of the data. This plotting strategy will provide a visualization of presence or absence of outliers in each feature. Outliers are some unwanted observations in the dataset which are recorded by the instruments (or manually) and their presence mislead the classifier. For instance, in figure 1, Malic Acid, Ash, Alkalinity of Ash, Magnesium and Color Intensity are some of the features exhibiting the presence of outliers. Whereas, Alcohol, Total Phenols, Flavonoids, Non-flavanoids, diluted wines and Proline does not show the presence of outliers. In the subsequent sections, we will convincingly state that identification these features with the outliers and their removal, really affects the specificity and sensitivity of the classifier. Moreover, it also (not drastically) increases the hit rate of the classifier. In the attached the projects folder, we have provided small (4 line) implementation of the box plot for the wine dataset.

2.1.2 Histogram plot for the Wine Dataset

This is another univariate plotting strategy for understanding the distribution of the individual feature values. In comparison with Box plot, this plot is more frequency based distribution, whereas providing information about the quartiles and medians which support outlier identification. Observing the figure 2, we observed right-skewed distributions of Malic Acid, Magnesium, Color Intensity, and Proline. Some features show edge peak normal distribution (uniform); such as Alkalinity of Ash and Ash. A visualization of this distribution assists in identifying the normalization procedure for the dataset. Moreover, this distributional analysis provides a picture of which feature to keep and which to remove. For instance, a feature following a plateau type distribution in the histogram (e.g. diluted wines, total phenols) are more likely feature contributing to the learning of the classifier as oppose to features have skewed distribution.

After both boxplot and histogram analysis, we are able to say that there are 6 features in the Wine dataset, that will be contributing to the classification of variable *Class*.

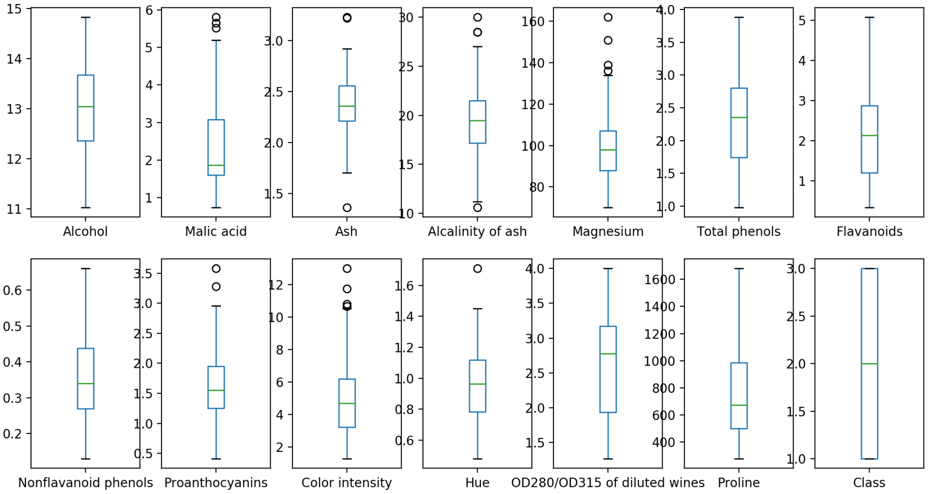


Figure 1: Descriptive Box plot Analysis of the Wine Dataset

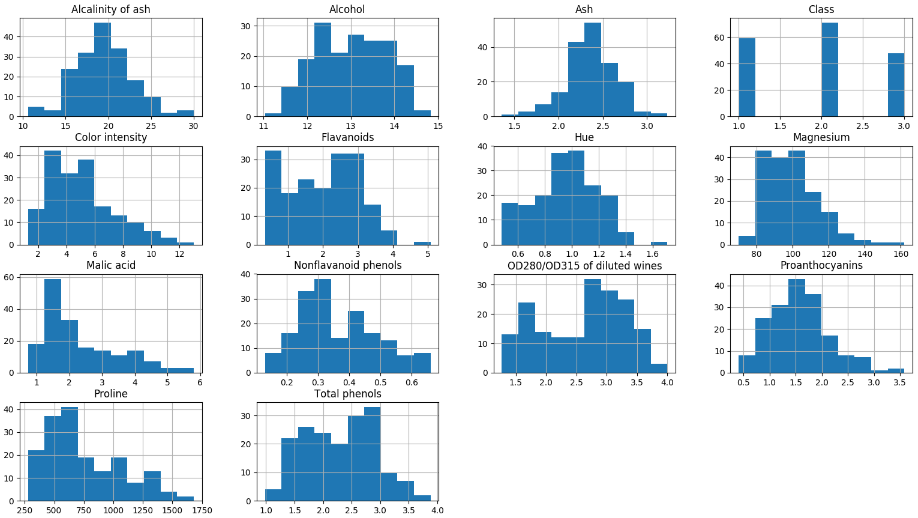


Figure 2: Individual feature histogram plot of the Wine Dataset

	Alcohol	Malic_Acid	Ash	A.Ash	Mag	T.Phenols	Flav.	Non-Flav.	ProA	Col.Int.	Hue	dil.Wines	Proline
mean	13.0	2.4	2.4	19.5	99.7	2.3	2.03	0.4	1.6	5.1	1.0	2.6	746.9
std	0.8	1.1	0.3	3.4	14.3	0.7	1.0	0.12	0.6	2.3	0.23	0.7	314.9
min	11.03	0.7	1.4	10.6	70.0	1.0	0.3	0.13	0.4	1.3	0.5	1.3	278.0
25%	12.4	1.6	2.2	17.2	88.0	1.7	1.2	0.3	1.3	3.2	0.8	1.9	500.5
50%	13.1	1.9	2.4	19.5	98.0	2.4	2.1	0.3	1.6	4.7	1.0	2.8	673.5
75%	13.7	3.1	2.6	21.5	107.0	2.8	2.9	0.4	2.0	6.2	1.1	3.2	985.0
max	14.8	5.8	3.2	30.0	162.0	3.9	5.1	0.7	3.6	13	1.7	4.0	1680.0

Table 1: Basic Statistics of the Wine dataset.Features:{ A.Ash: Alcalinity of Ash, Mag: Magnesium, T.Phenols : Total phenols, Flav: Flavanoids, Non-Flav: Nonflavanoids phenols, ProA: Proanthocyanins, Col.Int.:Color Intensity, dil.Wines: OD280/OD315 of diluted wines }

2.2 WPBC Flu Dataset

The flu dataset has been created from a survey conducted across 21 schools in the year of 2011. The survey covered 417 students and generated a dataset of 18 features. Of these 18 features, 17 independent features include Vaccination, Hand Wash Quality, Hand Wash Frequency, Non-Touch Face, Social Distance, Respiratory Etiquette, Personal Distance, Complication, Barriers, Inefficacy, Knowledge Transfer, Knowledge Management, Sick and Gender. Binary feature Flu is taken as a target variable for the classification. Analyzing the distribution of the samples, we observe that there are 283 students (samples) with non-flu (label 0) and 68 students (samples) with flu (label 1). Though it is good news that majority of children does not have Flu but for the classifier, it is a class-imbalance problem. Moreover, the dataset is not normalized as 8 features follow Likert scale, 7 features have real number values and 2 features are binary (see table 2, for descriptive statistics of Flu dataset).

2.3 Boxplot Analysis of WPBC-Flu Dataset

Aligning with our thoughts on outlier detection before modeling, we performed box plot analysis of Flu dataset in figure 3. We observed that of 17 features in the dataset, SocialDist, Risk, Complic, Barriers, and KnowlMgmt showed the presence of outliers in their individual feature values. Out of these 8 features, *Barriers* showed the presence of large number of outliers. Since, *Sick*, *Gender*, *NotTchFace*, *PrsnlDist*, *RespEtiq*, and *Flu* are discrete integral (likert) type features, their box plot contributes nil towards outlier detection. So, after the boxplot analysis, we have following features for our model: *Risk*, *Complic*, *Inefficacy*, *KnowlTrans*, *KnowlMgmt*, *Sick*, and *Gender*.

2.3.1 Histogram based analysis of WPBC-Flu Dataset

Similar to figure 2, we perform univariate histogram analysis of Flu dataset. One observation that is very transparent is that very few features have a normal distribution as compare to Wine dataset. What I observe is dog food distribution in *KnowlMgmt*, *RespEtiq*, *PrsnlDist*, *KnowlTrans* which indicates some missing values (empty or NaNs). This requires handling of missing data in this dataset. Moreover, features like Barriers, Complic exhibits right-skewed distribution. This type of distribution provides an idea of the presence of outliers in these features. Features having outliers are visualized using the box plot in figure 3. Some simple but strong statistical observations from the histogram plots are enumerated as follows:

1. The percentage of female students in the survey is more than male students.

2. There is an imbalance in the target variable: Flu. The percentage of students having flu is very less as compared not having flu.

3. There is three class of *Sick*, which is mapped to Flu variable. It is evident from both the feature's distribution that student having a cold-like feeling are categorized as not

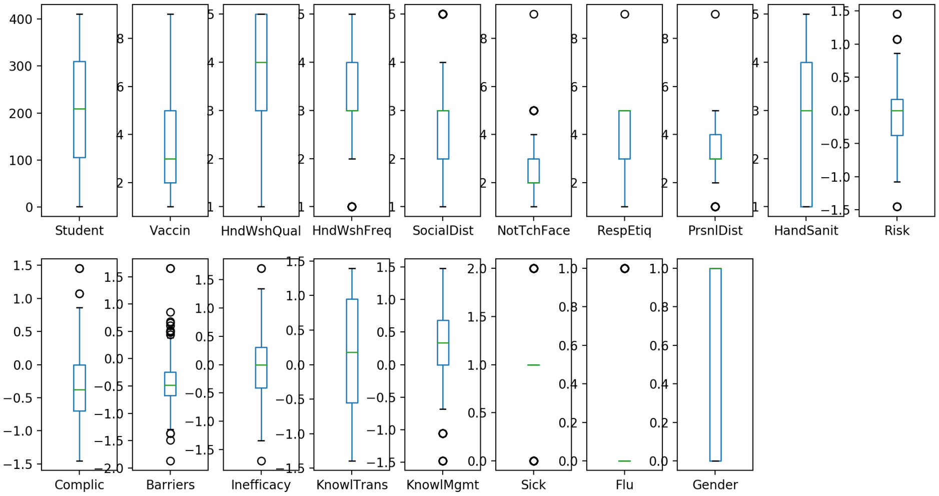


Figure 3: Box plot of Flu Dataset features

having flu.

4. There is a high percentage of students who are neutral on their thoughts regarding keeping a distance from ill students.

5. Students involved in the survey have descent respiratory etiquette scores.

Note: Ignore, Student feature histogram as it is just an anonymized id for the names.

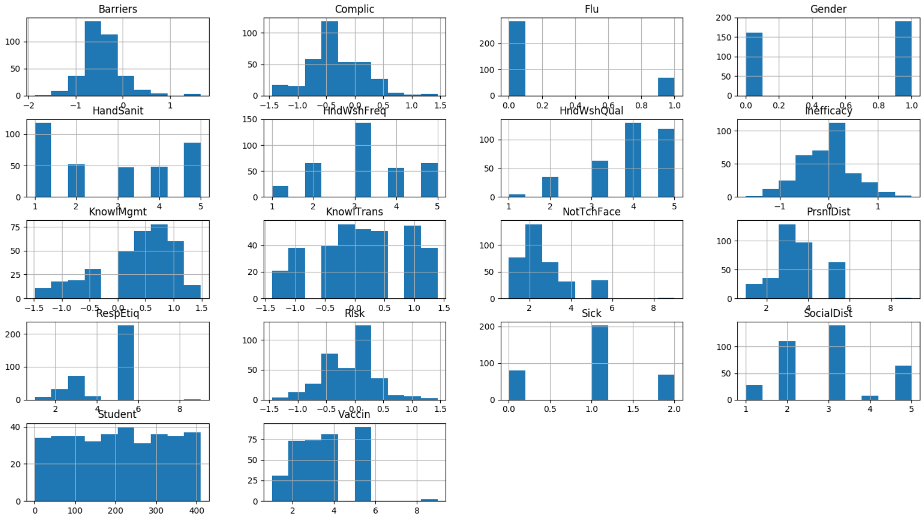


Figure 4: Individual feature histogram plot of the Flu Dataset

	Vacc.	HWQ	HWF	SD	NTF	RE	PD	HS	Risk	Comp.	Barr.	Ineff	KT	KM	Sick	Flu	G
mean	3.4	4.0	3.2	2.9	2.5	4.2	3.4	2.8	-0.12	-0.4	-0.4	-0.1	0.1	0.3	1.0	0.2	0.5
std	1.4	1.0	1.1	1.2	1.3	1.2	1.2	1.6	0.5	0.5	0.4	0.5	0.8	0.7	0.7	0.4	0.5
min	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	-1.5	-1.5	-1.9	-1.7	-1.4	-1.5	0.0	0.0
25%	2.0	3.0	3.0	2.0	2.0	3.0	3.0	1.0	-0.4	-0.7	-0.7	-0.4	-0.6	0.0	1.0	0.0	0.0
50%	3.0	3.0	3.0	3.0	2.0	5.0	3.0	3.0	0.0	-0.4	-0.5	0.0	0.2	0.3	1.0	0.0	1.0
75%	5.0	4.0	4.0	3.0	3.0	5.0	4.0	4.0	0.2	0.0	-0.3	0.3	1.0	0.7	1.0	0.0	1.0
max	9.0	5.0	5.0	5.0	9.0	9.0	9.0	5.0	1.5	1.5	1.7	1.7	1.4	1.5	2.0	1.0	1.0

Table 2: Basic Statistics of the WPBC Flu dataset.Features:{ Vacc:Vaccine, HWQ:HndWshQual, HWF:HndWshFreq, SD:SocialDist, NTF:NotTchFace, RE:RespEtiq, PD:PrsnlDist, HS:HandSanit, Comp:Complic, Bar:Barriers, Ineff:Inefficacy, KT:KnowlTrans, KM:KnowlMgmt, G:Gender}

3 Environment of Neural Network

This section will provide the necessary background to understand the experimental explanations under subsequent sections.

3.1 Programming Environment and Libraries

We created the neural network classifier using some simple libraries in python. The entire programming was conducted using Jupyter Notebook and Pycharm IDE. The libraries used in this experiment are Math, Pandas, Numpy, Matplotlib (for plotting), and Seaborn.

3.2 Parameter Tuning

The neural network is a very complex and powerful non-linear classifier. There are various numbers of hyper-parameters in the neural network which, if properly tuned can drastically increase the performance of the classifier. We have carried out the experiment keep the learning rate to 0.01, the number of epochs was kept varying between 500 and 1000, the number of hidden layers was kept varying between 1 and 3, and the regularization parameter λ was given a value from this set of values - {0.1, 0.01, 0.001, 0.0001}. For generating the Receiver Operating Characteristics (ROC) Curve, we ran for the model for 10 iterations with 10 fold cross-validation and 600 epochs. Our selection of parameter is not a rule, rather, varying this parameter of the neural network can improve the accuracy and precision of the classifier.

3.3 Evaluation Metrics

In this assignment, we have used a couple of metrics for analyzing the performance of our classifier and the validate that its consistency across the two datasets. The metric used in our experiments are; Precision, Recall (a.k.a True Positive Rate), False Positive Rate, Averaged Accuracy (in our context, we used Hits). We used the values of these metrics to generate Confusion Matrix and ROC Curve. Now, we will briefly state the meaning of these metrics.

Precision is defined as the ratio of a number of samples actually predicted positive over the total number of samples predicted positive. The term "positive" is bit ambiguous. It is not always the case that "1" is taken as positive and "0" as negative. The precision of the classifier is calculated individually for both and them averaged to achieve the precision of the classifier. In our assignment, we calculated precision as :

$$Precision = \frac{TP_0 + TP_1}{TP_0 + TP_1 + FP_0 + FP_1}; TP : TruePositive, FP : FalsePositive. \tag{1}$$

{0,1} in the subscript of the terms in 1 are labels in the dataset.

Recall is defined as the ratio of number of samples actually predicted positive over the total number of samples actually positive in the testing set. Recall value is also derived with the same intuition as Precision. The formulation for the recall is :

$$Recall = \frac{TP_0 + TP_1}{TP_0 + TP_1 + FN_0 + FN_1}; FN : False Negative \tag{2}$$

False Positive Rate is defined the specificity of the classifier. False positive is defined as the case when the predicted label and actual label do not match with the predicted label being positive. False positive rate is defined as the ratio of a number of the sample predicted positive and are not actually positive over the number of samples that are actually negative. It is formulated as :

$$False Positive Rate(FPR) = \frac{FP_0 + FP_1}{FP_0 + FP_1 + TN_0 + TN_1}; TN : True Negative \tag{3}$$

Note: True Positive (TP) is defined as the case when the predicted label matches the actual label and both are positive. True Negative (TN) is defined as the case when the predicted label matches the actual label and both are negative. The word "negative" is also ambiguous in machine learning context. If we consider "1" as positive then "0" is negative and if we consider "0" as negative then "1" is positive.

Average Accuracy(AA) is defined as the correct hits made by the classifier. It is calculated the ratio of the count of the number of times when predicted label matches actual label over the total number of samples predicted by the classifier. In our experiment, we calculated AA because we took the average of the accuracy of the classifier over multiple layers.

Confusion Matrix is defined as the contingency table created out of true positive, true negative, false positive, and false negative values. In actual scenario, these values are generated from the confusion matrix. An example confusion matrix:

		Prediction outcome		
		p	**n**	**total**
actual value	**p′**	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	total	P	N	

Reg. Parameter	Average Accuracy
$\lambda = 0.1$	71.43%
$\lambda = 0.01$	73.72%
$\lambda = 0.001$	73.2%
$\lambda = 0.0001$	79.43%

Table 3: Change in the average accuracy values with change in regularization parameter values. This experiment was done on Neural Network trained and tested on Wine Dataset.

Receiver Operating Characteristic(ROC) Curve is defined a curve showing the trend in true positive rate with respect to false positive rate. It is a diagnostic test, to check that the classifier minimize false positive and maximize the true positive. Left most corner of the plot is the ideal point for the classifier when the FPR is 0 and TPR is 1.
Note: We have used macro f-score measure for evaluating the performance of the classifier.

3.4 Regularization and Normalization

Before, detailing more about the regularization and $\lambda' s$, we define regularization.

Regularization is a technique used to prevent the model from over-fitting by adding a high weighted term to the optimization function. There are various ways of formulating this terms, such as , L1 ($||\theta||$), L2 ($||\theta||_2^2$). We have used Frobenius norm [] to formulate our optimization function with regularization.

$$J(\theta) = \sum_{i=1}^{N_{tr}} (h(\theta)_i - Y_i)^2 + \frac{\lambda}{2}||\theta||_2^2 \tag{4}$$

where $J(\theta)$ is the cost function, θ are the weights,
N_{tr} : Number of the training samples. This value changes with the cross folds ratios.
λ: this is the regularization parameter.
$h(\theta)_i$: this is the hypothesis function.
$h(\theta)_i = \frac{1}{1+e^{\theta^T x_i}}$; x_i is i^{th} observation in the dataset. Y_i : is the i^{th} label under *Flu* or *Wine* datasets. Regularization and feature scaling are the techniques used to make model reach its goal of convergence and better prediction without over-fitting and preventing the ill-posed problem. In this experiment we varied values of λ between 0.1 and 0.0001. This variation affected the accuracy rate of the model.

Normalization : Normalization is also called as Feature Scaling. We applied Min-Max normalization for our experiment. There are various ways for performing normalization such as Z-transformation, Scaling to unit length, Re-scaling, and Standardization.

$$scaled - dataset = \frac{feature^i_j - \mu_{feature^i}}{max(feature^i) - min(feature^i)} \tag{5}$$

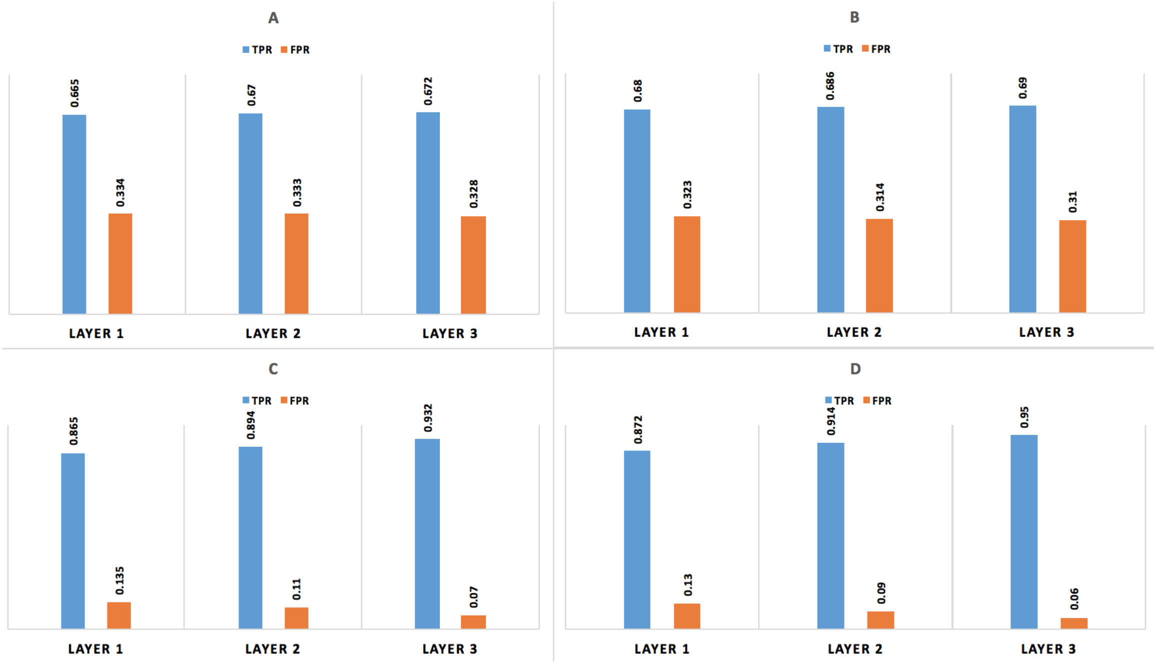


Figure 5: True Positive Rate (TPR), False Positive Rate (FPR) score for Wine dataset in 4 scenarios : A: Un-Normalized Dataset, B: Removed 7 features from the Dataset, C: Normalized Dataset and D: Regularized NN over Normalized Dataset

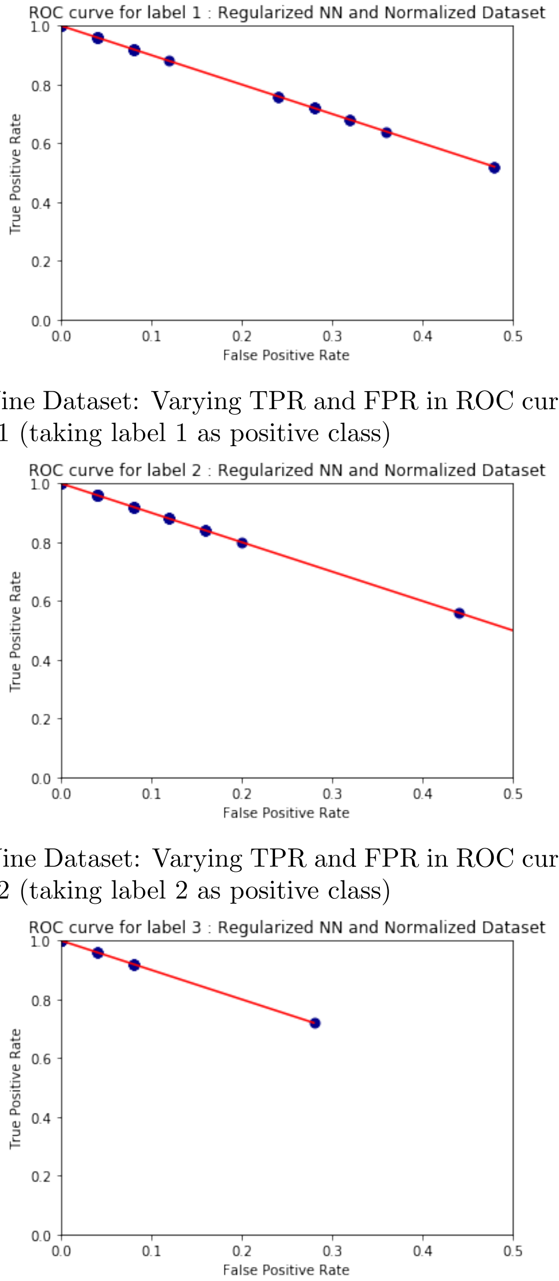
i ϵ { selected features from wine and flu dataset} and j ϵ [0, length($feature^i$)-1]

4 Results and Analysis

In this section, we will describe in details, the performance of the modeled and trained neural network classifier over two different datasets: Wine and Flu.

4.1 TPR and FPR based analysis of Wine Dataset

We created 4 scenarios for illustrating the efficacy of the trained neural network on the Wine dataset. Firstly, we trained and tested the neural network over complete dataset in its vanilla stage. Secondly, we carefully removed some set of features (see section 2). Thirdly, we normalized the dataset and took complete set of features. Lastly, we regularized the neural network, normalized the dataset and assess the performance of the classifier. Based on the analysis shown in the figure 5, we observed that regularized neural network over normalized dataset performed well, achieving a FPR of 6% and TPR of 95%. Also, the figure 5, showed improvement in the results of the classifier as move from scenario A to scenario D. Regularization as stated in previous section, is used to prevent model from learning noise. Value of the regularization parameter should be adjusted appropriately so



(a) Wine Dataset: Varying TPR and FPR in ROC curve for label 1 (taking label 1 as positive class)

(b) Wine Dataset: Varying TPR and FPR in ROC curve for label 2 (taking label 2 as positive class)

(c) Wine Dataset: Varying TPR and FPR in ROC curve for label 3 (taking label 3 as positive class)

Figure 6: ROC curves for the wine dataset.

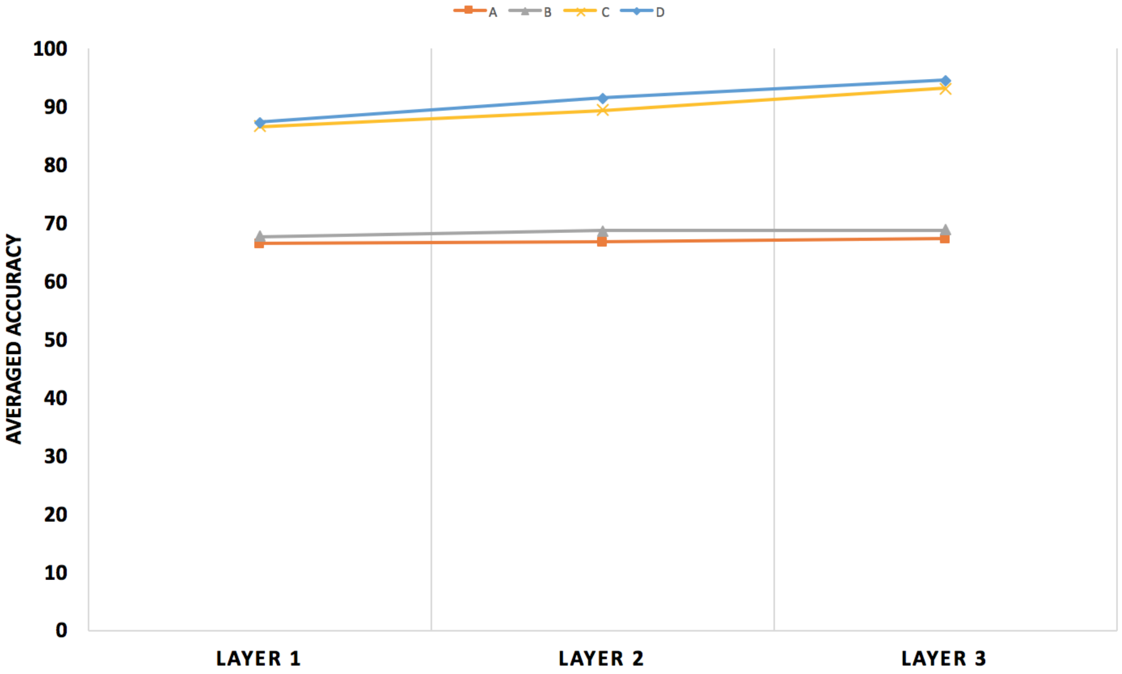


Figure 7: Wine Dataset: Trend in Accuracy of the NN when we add layers to the network under 4 scenarios; A: Un-Normalized Dataset, B: Removed 7 features from the Dataset, C: Normalized Dataset and D: Regularized NN over Normalized Dataset

that model does not over fit (high variance) or under fit while learning patterns in the dataset. Improvement in the performance of the classifier when λ is set 0.0001. It is also called as a supplement bias to the classifier so that model can be generalized. In the figure 5, the plot C,shows the TPR and FPR rate over a normalized dataset. As seen in figure 1 and 2, normalization is required in the wine dataset because some features like Proline and Magnesium, lie way out of the range as compared to other. Hence, we need to normalize so that all the features are in the same range [0,1]. We also observed that normalization over the entire feature set performs better than removing those features which have outliers or have values way bigger than the other relative features. Reason for improvement is two folds : Firstly, performing normalization (using mean) is very sensitive to outlier and there is high probability that values which are in outliers are scaled to [0,1], reducing their impact on the classification process. Secondly, the size of the dataset is small resulting in normalization performing better than feature reduction. Moreover, feature reduction is very hard step, manual based feature reduction will not be very effective. A recommended principal component analysis (PCA) before NN should be compared with the process of normalization before NN.

We also evaluated the efficacy of the classifier using ROC curves. These diagnostic curves illustrates the paired variation of TPR and FPR over an iterative run of the classifier. Since, the wine dataset have three class labels {1,2,3}, we created three ROC curves for three classification strategy : {label 1: 100, label 2: 010, label 3: 001}. From plots in the figure 6a, 6b and 6c, we observed consistency in the performance of the classifier. That means, our classifier achieves a TPR of 1.0 and FPR of 0.0 over the wine dataset. Our ROC curve plot is not logarithmic in shape, this is because, we didn't use any threshold to calculate the TPR and FPR. Rather, we used max function to calculate the predicted label. It can be formulated as :

$$\max_{0<=score<=1}[score_{label1}, score_{label2}, score_{label3}] \tag{6}$$

"score" is the real number value generated from the sigmoid activation function in the Neural Network. Along with this documentation, we have generated a result log for the wine dataset containing the confusion matrix and accuracy values generated from various experiment [see text file: *Result_log_Wine.txt*]. Apart from the ROC curve, we also generated an accuracy plot (see figure 7). As stated before, in this experiment, we consider accuracy as the count of hits made by the classifer and also an approaximate performance measure. We observe that by increasing the number of layer in the neural network, the performance of classifier is improved. It reached above 90% on the wine dataset by increasing the number of layers from 1 to 3. Morover, we also show the improvement in TPR of the classifier with the **increase in the number of layers** (see figure 5) The intuition behind adding more layer and getting improvement is the addition of non-linearity in the model. Non-linearity in the model cannot be achieved with single layer, as output of a single layer is a linear combination of the inputs. Hence, multiple layers in the NN, gains the capacity to understanding some latent non-linear relationships between the feature values in the dataset.

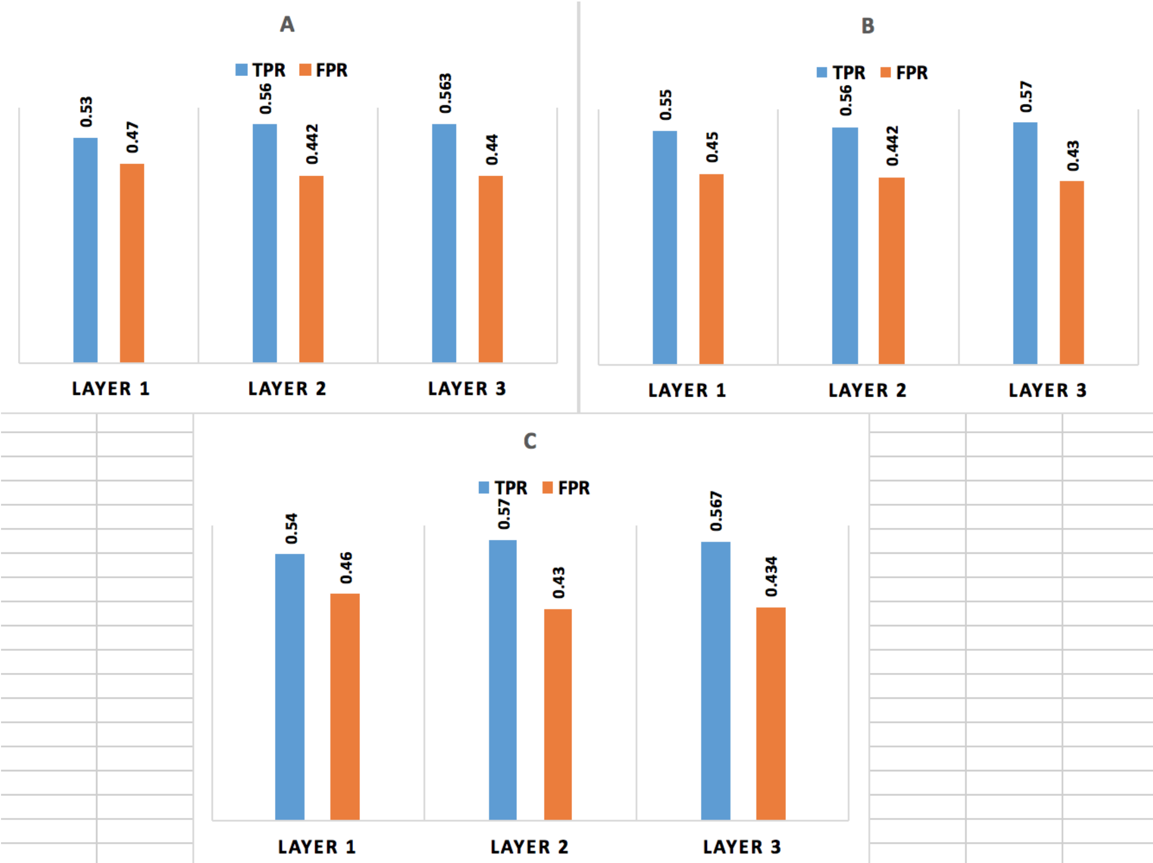


Figure 8: True Positive Rate (TPR), False Positive Rate (FPR) score for Flu dataset in 3 scenarios : A: Un-Normalized Dataset, B: Normalized Dataset, C: Regularized NN over Normalized Dataset

4.2 TPR and FPR based analysis of Wine Dataset

We replicate our model to perform over a more crude and survey-based dataset. The flu dataset is not cleaned and curated dataset as the Wine dataset. This is the biggest reason for biggest difference in the TPR,FPR, and Accuracy score of the NN over the two datasets. Our detailed result log has been stored in a text file: *Result_log_Flu.txt*. The file contains TPR, FPR , Average Accuracy and Confusion matrix generated by the model over multiple layers. One differentiable feature between the two dataset is that Wine dataset is a ternary labeled dataset whereas Flu dataset is binary labeled. We assessed the performance of the classifier over 3 scenario; Firstly, we trained and tested the model over the un-normalized dataset. Secondly, we experimented the same over a normalized dataset and finally, we regularized the model, normalized the dataset and repeated the experiment. Moreover, on this dataset, we did not experimented with manual feature reduction. This is because, we took 8 features from the dataset. These features are: Risk, Complic, Barriers, Inefficacy,

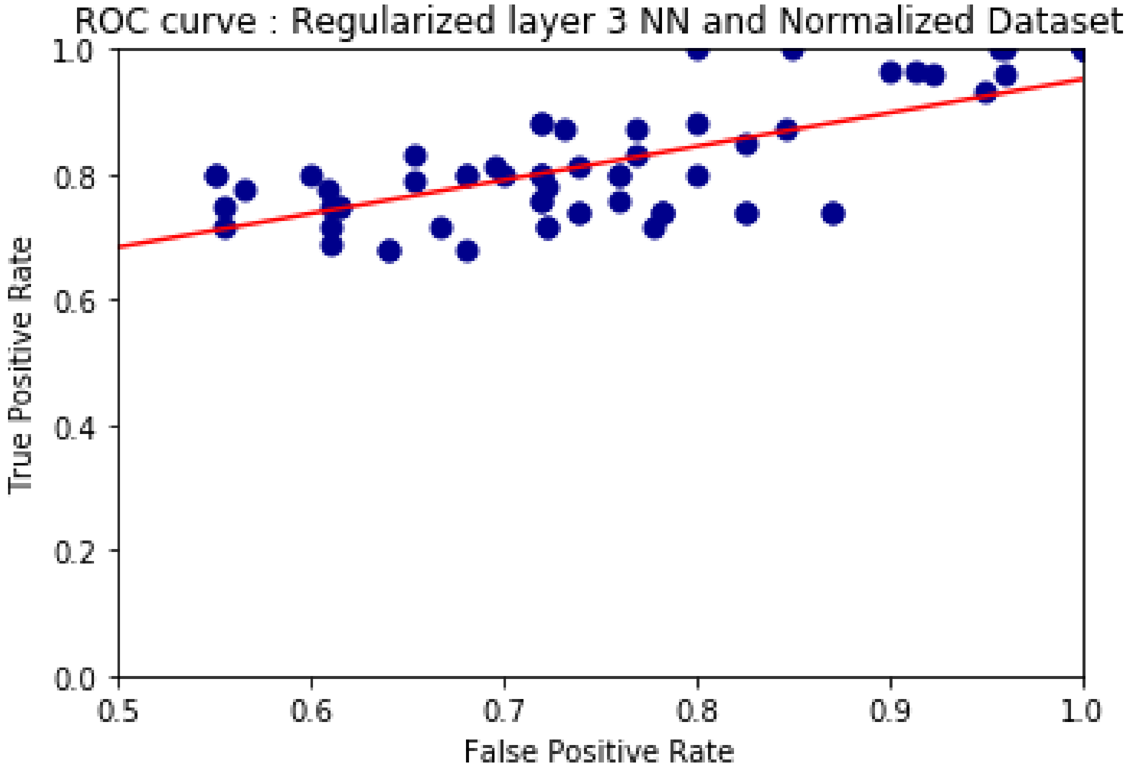


Figure 9: Flu Dataset: Varying TPR and FPR in ROC curve under the influence of Regularized NN and Normalized dataset

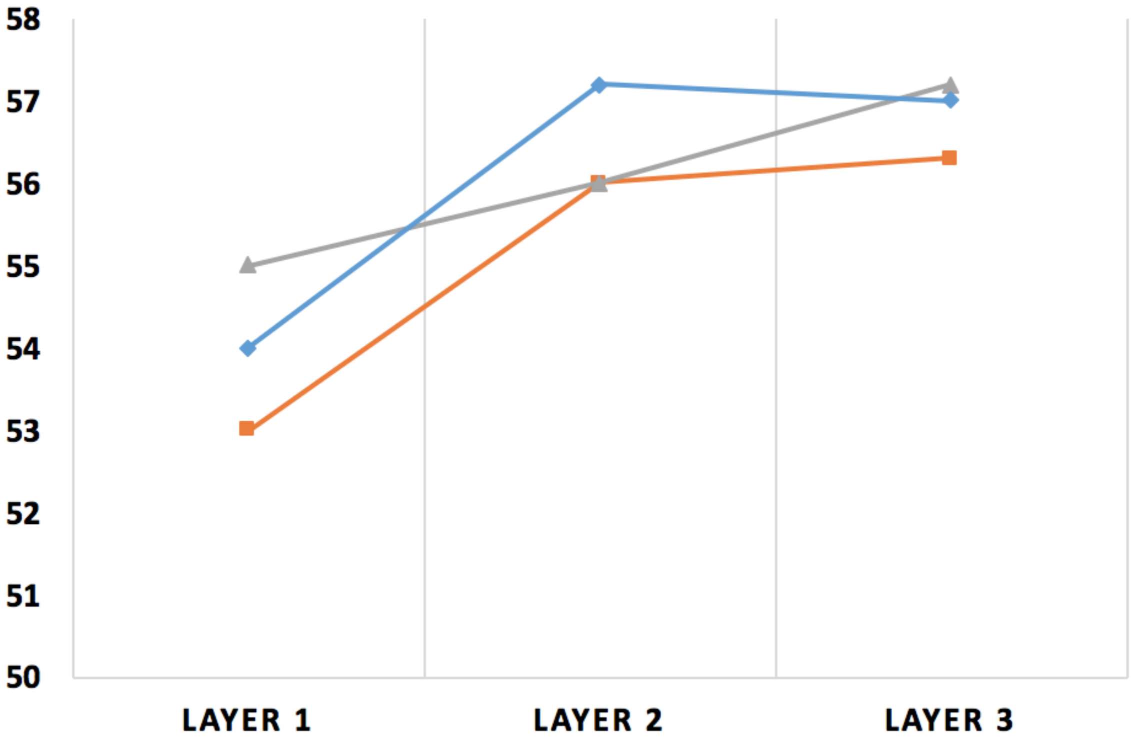


Figure 10: Flu Dataset : Trend in Accuracy of the NN when we add layers to the network under 3 scenarios; A: Un-Normalized Dataset, B: Normalized Dataset and C: Regularized NN over Normalized Dataset

KnowlTrans, KnowlMgmt, Sick and Gender. From figure 8, we observed that there is not major difference between the FPR and TPR which can be seen in figure 5, but with increase in the number of layers in the network, the performance of the classifier is improved (decrease in FPR). Moreover, we see significant trend in the accuracy values of the NN over the changing the number of layers of the network. In figure 10, we observed that on normalized dataset and regularized NN, the performance of classifier is above 57%. One significant development observable from the figure 10 is that regularized NN was consistent over the layers whereas un-regularized NN showed a decline in the performance when the number of layers changed from layer 2 to 3 (blue line in the plot). This shows the requirement of regularization as it prevents the model from high bias or high variance situation.

We analyzed the specificity (FPR) and sensitivity (TPR) of the classifier using the ROC curve. In figure 9, we observe non-linear (logarithmic) trend in the TPR and FPR values when the regularized 3-layer NN was tested over flu dataset. We observed that when the FPR was 0, TPR was 70% which is a considerably significant performance of NN. Moreover, for a FPR of 80%, TPR is 100%. The area under this scatter distribution of points is the measure of accuracy of the classifier (AUC: 0.57).The red trend line over the scatter plot, shows the behavior of FPR which is linearly increasing (not a good idea though, but it is largely governed by the quality of data).

Note: The neural network developed and tested on wine and flu dataset, performed extremely well on Wine dataset (AUC $\approx$ 0.99) as compared to Flu dataset (AUC $\approx$ 0.57). Reason for such behavior is following :

1. The level of Noise in the flu dataset is high as compared to Wine Dataset.

2. Flu dataset have two type of datatype : Likert and Real number. Wine dataset features have mutliple datatype : Real Number, Binary, Ternary and Integer.

3. Presence of missing values in Flu dataset which is absent in Wine dataset.

4. Class imbalance in Flu dataset. There is no class imbalance in Wine dataset. Though proportion of samples for Class 3 is small as compared to Class 1 and Class 2 but is not that low that we call that dataset as class imbalanced dataset.

4.3 Weight Initialization Strategy: A Comparison

In a NN, the weight of the linkages from the input data to hidden layer needed to be defined prior training and testing the model. There are various ways of initialization such as : Random Number Generation between 0 and 1 (strategy 1), Uniform Distribution based Random Number Generation (strategy 2), All the weights initialized to 1 (strategy 3) and Some predefined values generated from rules, priority consideration etc.. In this assignment, we compare the performance of the classifier on Wine dataset under three strategy for weight initialization: strategy 1, strategy 2 and strategy 3. The reason for following strategy 1 and strategy 2 is to break the symmetry problem during weight initialization. If all the weights are initialized with 1 or 0, it seriously affects the performance of the classifier (see table 4).

Weight Init. Strategy	Accuracy
strategy 1	87.4%
strategy 2	88.5%
strategy 3	66.3%

Table 4: Impact on the accuracy of the classifier by changing the weights initialization strategies

This is because, in the forward propagation learning of the network, each hidden unit in the hidden layer will get same amount of signal. Hence, all the hidden units in any layer will be same. By initialization the weights randomly, we tend to get multiple solutions to same problem and performance of the classifier can be assessed.