Comparison of MLP and SVM to Predict Length of Stay at Hospitals

Student Name: Seren Turan

Student ID: 200032726

Contents

Abstract
In this research, our objective is to explore how accurately it is possible to predict LoS of patients on case basis. Towards this aim, we introduce two models which are Feedforward Multilayer Perceptron (MLP) and a Support Vector Machine (SVM). We investigate the performance of each model given different hyperparameters and compare them. Finally, we critically evaluate the models and discuss the possible implementations that can be introduced by future work conducted in predicting LoS at hospitals.

1.Introduction:
Length of Stay (LoS) is defined as the patient hospitalization duration, and it is an indication of inpatient **hospitalization costs and resource utilization**. As proven with the ongoing COVID-19 pandemic, hospitals have extremely limited bed capacity and most of them are facing significant financial pressure. Due to the growing number of hospitalized patients, it is of significant importance to predict the average LoS for resource planning, admission scheduling, patience safety, good quality of care and reduced healthcare costs. It would also an important factor to help health professionals know about the expected length of stay of their patients in planning the most appropriate target interventions. Being able to predict LoS is not only important for the healthcare system but also for the patients as studies found that reduced LoS is related with lower death rates, higher responsiveness, better quality of care and meeting the patients 'needs more. However, predicting LoS is very challenging as LoS varies depending on cases with different conditions and complications. Husted et al. [1] found that the among hip and knee replacement cases, various factors like the age, sex, marital status, co-morbidities, time between surgery and mobilization, and others, influence a series of outcomes, including the LOS. Whereas another study found that paediatric liver transplant recipients [2], LoS was highly related with factors such as infants less than one year of age, fulminant liver failure, government insurance, and transplant era. In another example, congestive heart failure patients´ LoS [2], was related to patient demographics, severity of illness, management modalities, response to treatment, and administrative data. Despite being challenging, prediction of LoS is a very attractive task in the healthcare industry. For this reason, machine learning and neural computing techniques are used often to offer a first step and helping hand in extracting useful information from patients' data.

1.1 Multilayer Perceptron (MLP):
MLPs are feed forward neural networks consisting of at least three layers of nodes: an input layer, a hidden layer and an output layers as many as the number of classes that is desired to be predicted. The learning process of MLP is based on N-dimensional input vector x and the M-dimensional output vector d, composed of the desired classes that will be obtained. MLP processes the input vector and produces the output signal vector $y(x, w)$ where w is the vector of adapted weights. Each node in MLP apart from the initial uses a nonlinear activation function. MLPs are generally trained by backpropagation for training, which is a a supervised learning technique to adjust the weights given to different nodes and optimise the model. The learning algorithm of MLP is based on the minimization of the error function using the Euclidean norm which leads to the optimal values of weights. The most effective methods of minimization are the gradient algorithms [1].

1.2 Support Vector Machine (SVM):
Support Vector Machines (SVMs) are supervised binary classifiers, that serve to classify observations belonging to different classes by use of a functional margin. They define the optimal hyperplane that maximises the distance between the classes and the boundaries, which can then be used to determine the most probable label for unseen data [5][6].

In case of nonlinear classification problems, SVMs employ use of kernel method to transform the support vectors to a higher-dimensional input space which serves to convert nonlinearly separable set of features to a set of linearly separable ones [5]. Kernel methods are also often used as a form of dimensionality reduction for linear SVM models. Because of their relative simplicity and flexibility for addressing a range of classification problems, SVMs distinctively afford balanced predictive performance, even in studies where sample sizes may be limited [5].

1.3 Cons and Pros of SVM and MLP:

In MLPs classifiers, the tested data sets need more hidden units, and the complexity is controlled by keeping the number of these units small, whereas the SVMs complexity does not depend on the dimension of the data sets. SVMs based on the minimization of the structural risk, whereas MLP classifiers implement empirical risk minimization. So, SVMs are efficient and generate near the best classification as they obtain the optimum separating surface which has good performance on previously unseen data points. Main difference between MLP and SVM models are their complexities. MLPs employ global approximation strategy and usually introduces very small number of hidden neurons. Whereas SVMs are based on the local approximation strategy and they use large number of hidden units, and they use quadratic function to solve non-linear problems. It greatly reduces the number of operations in the learning mode but takes longer to run [1].

2. Data Description:

The dataset used in this analysis is sourced from Kaggle [4] and it is based on length of stay of patients given different information regarding the patients on case basis, and the hospitals' condition. It contains 318,438 rows and 18 columns making up a total number of 5,731,884 data points. Some of the attributes found in the dataset are age of the patient, severity of the patient´s illness, type of admission, hospital code, city of the hospital, department of the hospital, ward type, etc… The data contains information based on individual cases, and their stay of length are given as a range such as 0-10 days, 20-30 days etc… As they are given as categorical variables, we have a multiclassification problem rather than regression. Among 11 classes of length of stay, the dataset is imbalanced as class 2, 3 and 4 form the 33.9%, 42.4% and 25.4% of the dataset, respectively. The remaining 8 classes make up the remaining portion of the dataset which is quite small compared to the percentage of the top 3 majority classes.

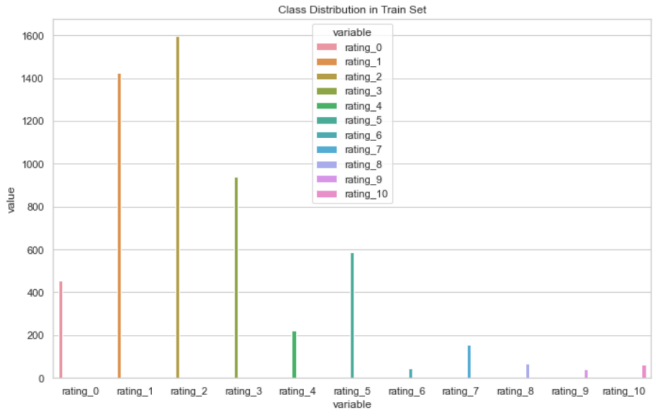

Figure 1: Distribution of Different Classes to be Predicted and the Imbalance in the Dataset

2.1 Initial Data Analysis:

Firstly, the most important features are plotted by using bar and pie chart and histograms to see how they vary for given attributes. Secondly, a heatmap is plotted to see the correlation between the numerical columns. Next, we dropped the missing values as it was only 1.4% of

the data. Then, we created new columns by splitting the Age column into two and obtained lower bound age and lower bound age column to be able to use them as numeric values in our analysis. Next, we checked the distributions and the normality of the numerical features. For avoiding inaccurate results caused by non-normality and skewness, we took the following steps to transform our data: we detected and removed the outliers from the numerical features of the dataset by using IQR method. Secondly, we performed log transformation to make data conform to normality. Please note that these transformations were not performed on the upper and lower age columns as this would cause the original ranges provided in the dataset to vary. We used label encoding and manual encoding for the categorical columns. For the ones that had more than 10 classes, we used one hot encoding. After performing encoding, we obtained 130 columns in total. As the final step, we performed standard scaling to standardize the data values in a standardized format and dropped the unnecessary features such as patient ID, hospital ID, etc…

3. Methodology:

As for the methodology, we created our MLP and SVM models on two different datasets: the original and the SMOTE applied rebalanced dataset. We applied the SMOTE method only on the training and validation set. The reason why we chose SMOTE over undersampling is because if we removed data from the majority class until it is equal with the minority class, the dataset would overly shrink. On the other hand, oversampling would not help removing the imbalance in the dataset but just adding more data points into our dataset which would still not address our problem and would make our analysis even more challenging as we have a dataset of 5 million points already. After that, for both datasets the following was applied: we split the 20% of the dataset for testing and the remaining 80% was used for training and validation of the model. The training and validating process was performed in 3-fold stratified cross-validation batches on 7000 randomly picked samples of the data. This is because the dataset has more than 5 million data points and even 10000 samples took considerably long to train and run. After training the models, we used grid search method to select the best hyperparameters for both algorithms being MLP and SVM. As a result, the train and test accuracies and losses were compared for all the hyperparameters and the best performing models were selected for both methods. For the algorithm comparison and to decide whether SVM or MLP performs better on our dataset, the models were retrained on the best hyperparameters using the training and validation data and evaluated the performance on the test set by looking at the loss, accuracy, and the confusion matrix. We used R^2 to evaluate the loss of the SVM model and cross-entropy function to evaluate the loss of the MLP model as we used softmax activation function in the output layer.

3.1 Hypothesis statement:

A high accuracy is not expected as a result of this analysis as we have a severely imbalanced and high dimensional, large dataset with 11 different classes. Therefore, a high accuracy will not be the ultimate aim of this analysis. We expect the SVM model to take longer time to train and run as we don´t have a linear problem and SVM uses quadratic service to classify non-linear datasets, which would eventually take longer to train due to complexity. Since MLP can introduce multiple layers to analyse data from different perspectives, we expect it to yield better accuracy, precision and recall, however we also know that SVMs are quite powerful classifiers when they reach an optimal separating surface.

3.2 Architecture and Parameters Used for the MLP:

In designing our model, due to the high number of classes to be predicted and the size of the data, we used single hidden layer. We acknowledge that multi-class classification in high dimensional setting can yield better results with higher number of hidden layers, however we have experienced very long training and run times with models more than 1 hidden layer for this

specific dataset, therefore we prefer the run time to be shorter over accuracy and limit the number of hidden layers in this analysis. We used the scorch library when training our model as such, when training the model, backpropagation is implemented via the .fit method. Additionally, we used drop-out of 0.5 in our network to drop the data or noise to improve processing and overfitting. We also implemented the sklearn GridSearchCV function to implement grid search and cross validation. We used stratified k-fold cross validation with 3 folds to assess the results and the accuracy of the model. We decided to do the grid search on the following hyperparameters to select the best performing model: learning rate, maximum epochs, momentum, and weight decay.

3.3 Architecture and Parameters Used for the SVM:

SVM implements decision boundary to solve classification tasks. Trying to correctly classify all the points might end up the decision boundary to be highly sensitive to noise and overfit the model, whereas on the other hand placing the boundary as far as possible to each class might cause misclassification. To handle this tradeoff, we will introduce regularization parameter c as our hyperparameter. Additionally, SVM uses kernel trick to handle data that are not linearly separable, so we will adjust the kernel parameter. Additionally, we will control the groups to be formed together by the gamma. Finally, we will use degree parameter where applicable in case of polynomial kernel function) hyperparameters. We use GridSearchCV function to implement cross validation and grid search on the mentioned hyperparameters. We use 3-fold cross validation to assess the results and the accuracy of the different SVM models.

4. Results, Findings & Evaluation:

For evaluating the models, train, test and validation accuracies are compared, losses were evaluated. Confusion matrix, precision, recall and f1-score are evaluated for all classes. We have also evaluated the models based on how long time it took to train and run them.

4.1 Model Selection for MLP:

Best Performing MLP Model Hyperparameters on the Original Dataset			
Learning Rate	0.1	0.2	0.3
Maximum number of epochs	10	20	30
Momentum	0.10	0.15	0.20
Weight Decay	0.01	0.02	0.03
Best Performing SVM Model Hyperparameters on the SMOTE Rebalanced Dataset			
Learning Rate	0.1	0.2	0.3
Maximum number of epochs	15	20	25
Momentum	0.10	0.15	0.20
Weight Decay	0.01	0.02	0.03

Table 1: Different Hyperparameters Chosen for the MLP Model on the Original and Rebalanced Dataset

We performed grid search over the following hyperparameters values and their combination, fitting 81 different combinations in total for MLP:
Green and yellow colored values seen in Table 1 above are the hyperparameters that gave the best performing MLP model for the original and rebalanced dataset respectively. For the original dataset, the best score for training and test data set was recorded as 39.2% for both sets. Whereas for the rebalanced dataset, it was recorded as 32.8% and 15.0% for the training and test sets respectively. It is observed that the predictive power of the MLP model on the rebalanced dataset is quite poor and there is a big gap between it´s training and testing

accuracy, showing that the model after SMOTE is not as generalized and it has specialized to the structure in the training dataset. Therefore, it is overfitting and SMOTE might not be the best choice to rebalance our dataset. This might be because we have a severely imbalanced dataset with a high number of different classes and rebalancing the dataset might be creating a training set that is not representative of the real data. Another reason might be because the variations within the minority classes in our dataset is very high and similarities and patterns between the classes is very high too. Besides the accuracy, the overall precision and recall of the original and rebalanced MLP models were compared. The MLP with original dataset gave an overall precision, recall and F1-score of 36%, 39% and 39% respectively. Whereas with SMOTE, the precision, recall and F1-score decreased to 27%, 15% and 15%. For this reason, we select the original dataset and green highlighted hyperparameters in Table 1 to be the best performing MLP model.

4.2 Model Selection for SVM

Best Performing SVM Model Hyperparameters on the Original Dataset							
Kernel	***Gamma***		***C***				***Degree***
Rbf	10^{-3}	10^{-4}	1	10	100	1000	-
Linear			1	10	100	1000	-
Polynomial	10^{-3}	10^{-4}	1	10	100	1000	1, 2, 3, 4
Sigmoid	10^{-3}	10^{-4}	1	10	100	1000	1, 2, 3, 4
Best Performing SVM Model Hyperparameters on the Rebalanced Dataset							
Kernel	***Gamma***		***C***				***Degree***
Rbf	10^{-3}	10^{-4}	1	10	100	1000	-
Linear		1	10	100	1000	-	
Polynomial	10^{-3}	10^{-4}	1	10	100	1000	1, 2, 3, 4
Sigmoid	10^{-3}	10^{-4}	1	10	100	1000	1, 2, 3, 4

Table 2: Different Hyperparameters Chosen for the SVM Model on the Original and Rebalanced Dataset

We performed grid search over the above values listed in Table 2 above for both the original and rebalanced dataset. Green and yellow colored values seen in Table 2 above are the hyperparameters that gave the best performing SVM model on the original and rebalanced dataset respectively. For the original dataset, the best score for training and test data set was recorded as 39.88% and 39.86% respectively. Whereas for the rebalanced dataset, it was recorded as 37.60% and 17.7% for the training and test sets respectively. It is observed that a similar decrease in the performance and overfitting issue on the rebalanced dataset that we experienced for the MLP model occurred for the SVM model as well and the same justifications explain the decrease in the performance for the SVM model too. Looking at the precision, recall and F-1 scores of the SVM models, the original dataset performed better with a 32%, 40% and 40% respectively. Whereas the SVM model on the rebalanced dataset had a precision, recall and F1-score of 32%, 18% and 18%. Due to higher overall accuracy, precision and recall, the SVM model performed on the original dataset with the hyperparameters that are highlighted in green in Table 2 above are selected as the best performing SVM model.

4.3 Algorithm Comparison

To decide the best algorithm in classifying length of stay of patients for this dataset, we evaluate the training and run time, the overall accuracy, precision and recall of the best performing MLP and SVM models we selected above.

Referring to results discussed above, we see that the overall accuracy of the models are almost the same and as discussed in the hypothesis statement the accuracy of all the models generated are poor. MLP has a slightly higher precision in predicting different classes, whereas the precision are the same for the both models. Neither of the classifiers were successful in classifying the minority classes due to the heavy imbalance towards 4 majority classes among 11 different classes. The time it took us to train and run the SVM model was 85 seconds whereas it took 13 seconds in total to train and run the MLP model. So for a large data set, it is seen that MLP was much quicker. Due to the significant difference between run times and higher precision, it is found that for this specific dataset, using MLP is preferrable.

5. Conclusion:

As discussed in the hypothesis statement, the accuracy of the models and the ability of predicting the minority classes were quite poor as expected due to the severe imbalance and high dimensional multi class nature of the dataset. It was also seen that rebalancing the dataset with SMOTE might not always yield better results depending on the type of the dataset. As we had a high variation within the minority classes and a very high similarity between different classes, synthesizing fake data points for the minority classes created a misrepresentation of the original dataset and resulted in overfitting the model. In addressing this challenge, it would be suggested to increase the level of complexity of the MLP model by introducing more hidden layers but longer training time and computational limitations should not be overlooked. Another approach might be collecting more real-life data and training the models on a higher number of data points. More importantly, it would be suggested to take a different approach when collecting the data. There might be need for using different variables rather than generic variables like age, sex, hospital type, bed type etc. For different departments and specialization of medicine areas, it would be advised to collect more specific variables that is known to have a greater effect on the length of stay at hospitals, which might require a solid statistical research before data collection. This is because when the variables and the dataset are not very relevant, it is hard to reach a meaningful conclusion and high accuracy models.

References:

[1]: Zanaty, E.A. (2012). Support Vector Machines (SVMs) versus Multilayer Perception (MLP) in data classification. *Egyptian Informatics Journal*, 13(3), pp.177–183.

[2]: Bucuvalas, J.C., Zeng, L. and Anand, R. (2004). Predictors of length of stay for pediatric liver transplant recipients. *Liver Transplantation*, 10(8), pp.1011–1017.

[3]: Garcez, A., 2021. MLP, Neural Computing Lecture.

[4]: kaggle.com. (n.d.). *AV : Healthcare Analytics II*. [online] Available at: https://www.kaggle.com/nehaprabhavalkar/av-healthcare-analytics-ii [Accessed 11 Apr. 2021].

[5]: White, B.W. and Rosenblatt, F. (1963). Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. *The American Journal of Psychology*, 76(4), p.705.

[6]: Leudar, I. (1989). James L. McClelland, David Rumelhart and the PDP Research Group, Parallel distributed processing: explorations in the microstructure of cognition. Vol. 1. Foundations. Vol. 2. Psychological and biological models. Cambridge MA: M.I.T. Press, 1987. *Journal of Child Language*, 16(2), pp.467–470.

[7]: Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, [online] 2(4), pp.303–314. Available at: https://link.springer.com/article/10.1007%2FBF02551274 [Accessed 7 Aug. 2019].

[8]: Yıldırım, S. (2020). *Hyperparameter Tuning for Support Vector Machines — C and Gamma Parameters*. [online] Medium. Available at: https://towardsdatascience.com/hyperparameter-tuning-for-support-vector-machines-c-and-gamma-parameters-6a5097416167.