Comparative study of Supervised learning classifiers

Umesh Jadhav
OMSCS
Georgia Institute of Technology
Georgia, USA
ujadhav6@gatech.edu

***Abstract*—Supervised machine learning is a Machine learning technique which uses datasets of instances with pre-labelled outcomes for learning the relationship between the independent features and target feature to predict the outcome of the unseen samples. Classification and Regression are the two domains under Supervised machine learning where the former deals with binary or multi-label outcomes whereas regression method is used to determine the outcome of continuous nature. In this study, three classification algorithms specifically, K Nearest Neighbors (KNN), Support Vector Machine (SVM), and Neural Network are put under test for two different problems (datasets) while first analysing their performance over various sizes of training data and finally studying the influence of hyper-parameters tuning over various ranges respectively. The study begins by exploring the datasets with an initial hypothesis of which algorithm would perform the best; preceded by a brief discussion of experimental methodologies. Finally, the results obtained are analysed to evaluate our initial hypothesis.**

I. Introduction

Machine learning is specialized section of Artificial Intelligence (AI) in which models are developed by training them over one or more datasets to identify and learn the underlying data information and predict outcome of similar but unseen data, distinguished further into two branches namely, Supervised learning and Unsupervised learning. In Supervised learning, the model is trained over data with pre-labelled outcomes, learning the relationship between the independent features and the target variable. Supervised learning is favourable in tasks such as classification where the outcome labels are of discrete nature, and regression in which the outcome is continuous numbers. Some of the prominently used supervised learning algorithms are Decision tree, Naive Bayes, K nearest neighbours (KNN), and Support Vector machine (SVM) among others. On the other hand, Unsupervised learning do not have labelled outcomes and hence, the model itself identifies and establishes the relationship between the independent features with the outcome. K means clustering, Hierarchical clustering, Principle Component Analysis, and Singular Value Decomposition are few of the algorithms under the umbrella of Unsupervised learning. Another section of Machine learning defines a concept of Neural network which is a machine learning technique inspired by a human brain. As the name suggest, the core of this algorithm involves set of neuron units interlinked with each other in the form of layers. Neural network can be applied over wider ranges of problems including text resolution, image resolutions and more complex datasets of higher dimension. In this report, K nearest neighbors, Support Vector Machine, and Neural network algorithms are applied over two different binary classification datasets representing mutually exclusive domains and studied comparatively for their performance. The study explores both the datasets, their insights initially while building the algorithmic models and tuning them based upon their optimal hyper-parameters respectively. Finally, the study presents the results obtained and discusses them in details.

II. Dataset

A. Exploration

For this assignment, Diabetes Classification dataset[1] which identifies whether the person is diagnosed with diabetes depending upon certain physical characteristics, whereas Company Bankruptcy[2] classifies whether the company could face bankruptcy based upon various economic parameters. Both the datasets belong to mutually exclusive domains of work.

1) Dataset A: Diabetes classification dataset has 16 features with 1,00,000 observations. The independent features in this dataset are combination of categorical, ordinal and discrete numbers. The greater accuracy of all the three algorithms makes the dataset interesting to study and analyse them, influenced by the structure of the dataset following linear pattern.

2) Dataset B: On the other hand, Company Bankruptcy dataset contains 96 features including the target variable "Bankrupcty?", while having 6820 observations. Although the dataset falls under the category of classification problem, the independent features are continuous in nature. The structure of this dataset makes it interesting to observe and understand how the three classification algorithms behave on data with overlapping construct making it bit difficult for these algorithms to process. Also, there exists disproportion of the outcome labels of higher degree.

B. Pre-processing

As discussed in data exploration section, Diabetes classification datasets contains combination of categorical, ordinal, discrete and continuous independent features. For smooth functioning, the categorical values, and ordinal values are transformed into discrete numerical atomic values in place by identifying the number of unique values (n) from each of the features and replacing them by a single value in the range (0, n-1).

In case of Company Bankruptcy dataset, none of the independent features contain any nominal or ordinal data, hence data is not encoded. Also, the values here represent the economical rates, standardisation of data is not required as they follow the same fundamental unit.

C. *Hypothesis*

To better understand the datasets and build a hypothesis, pair plot is used which enables to explore the variables. Exploring the data with the help of pair plot distinguished over the binary label of the target variable, overlapping of certain features could be seen in both the datasets.

Beginning with Diabetes classification dataset, as seen in figure 1 majority of the independent variables follow linear pattern with an exception for ”age” and ”bmi”. Such kind of data is highly favourable for Support Vector Machine (SVM) algorithm and Neural Network; while K Nearest Neighbors (KNN) algorithm might under-perform due to its overlapping of data points. Similarly, Neural network works upon one record at a time, updating its weight based upon the learning rate defined by comparing the predicted outcome with the actual outcome iteratively. Since, the dataset in this case is assumed to be linearly separable with less complexity, SVM performance might outperform the other two algorithm in comparison.

Following the data structure of Company Bankruptcy dataset studied in previous section indicate that SVM with rbf kernel will outperform the other two classifiers because of its ability to split the outcome labels into a higher dimensional hyperplane. However, the performance can be debatable with neural network classifier due to its back–propogation strategy.

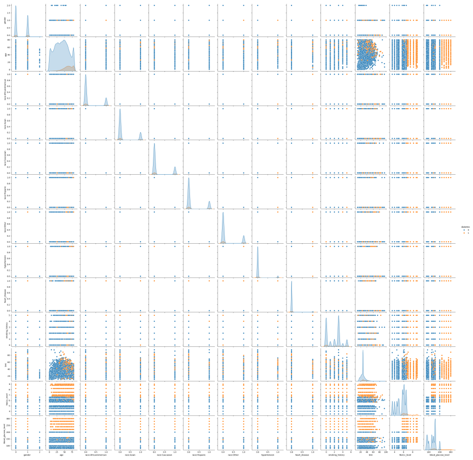

Fig. 1. Pair plot of Diabetes Classification dataset

III. Methodology

Experimental methodology is illustrated in the form of flowchart in figure 2.

A. *Import Dataset*

The experimental methodology follows Object oriented programming in python programming language. Hence, for each of the datasets, two isolated class files are created. First, the datasets are undergoes pre-processing as discussed in Pre-processing section.

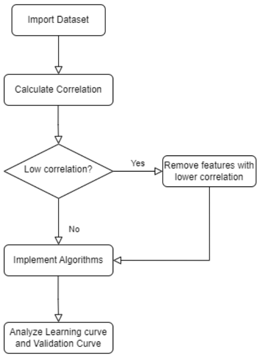
Import Dataset
Calculate Correlation
Low correlation?
Yes
Remove features with lower correlation
No
Implement Algorithms
Analyze Learning curve and Validation Curve

Fig. 2. Experimental Methodology

B. *Calculate Correlation*

Correlation quantifies the strength of influence of independent variables over the outcome variable. Bi-variate analysis between each of the independent feature and target variable is performed to find the best set of highly correlated features. This helps to exclude features that do not influence the outcome and might simply act as noise to the data.

For Diabetes Classification dataset, the features ”year” and ”location” are excluded from further computation due to lowest correlation. After removing these features, the dataset is passed as the input for training the three machine learning models as stated. In case of Company Bankruptcy dataset, multiple independent features are excluded due to their low correlation with the outcome variable.

C. *Implement Algorithms*

The implementation of KNN, SVM and Neural Network is entirely done using Python programming language scikit learn module (sklearn). Here, a common sampling technique is used for cross validation of the three algorithms to evaluate their performance known as Stratified sampling technique while splitting the entire dataset into training set and testing set. Stratified sampling ensures all the split get proportional outcome labels data points. K folds indicate the number of times the entire dataset is split. At each point in the iteration, one of the data split is used as the validation set while the reamining are combined to form the training set.

Diabetes Classification dataset

1) K Nearest Neighbors – As witnessed in pair plot from figure 1, the data points are linearly separable. Therefore, the weights are uniformly assigned instead of based upon distance metrics so that the algorithm identifies the data points in close proximity belonging to same class label. 2 Fold Stratified cross validation is chosen meaning 50 percent of the dataset is used as training set and 50 percent is used as validation set for each iteration. Secondly, Euclidean distance is used (p = 2)

for calculating the distance between the data samples which determines the shortest distance between them, assigning the class label to the data point under test same as that of the closest proximity data point. Manhattan distance (p=1) determines the distance metric based upon the absolute difference between the data points, favourable for regression tasks. n_jobs indicate the number of parallel running jobs for determining the neighbors space set to 10. Finally, the number of neighbors for learning curve is initially set to 100. In case of validation curve, the model is screened upon various values of K (n_neighbors), keeping the values of other hyper-parameters intact. The core equation of the KNN model initialization for plotting learning curve is specified in equation 1.

$$KNeighborsClassifier(n\_neighbors = 100, \\ weights = "uniform", p = 2, n\_jobs = 10) \quad (1)$$

2) Support Vector Machine – For analysing the performance of SVM, it is implemented with Gaussian (rbf), and Sigmoid kernels for learning curve as well as the validation curve. In case of learning curve, the hyper-parameters C value is set to 20 which is the cost metric imposed when the model misclassifies the outcome label of the data point. The initial value of gamma is set to scale for both the kernels. Finally, for evaluating the classifier, it is tested for various values of C instead of gamma as seen in the validation curve of figure.

$$SVC(C = 20, gamma = "scale", \\ kernel = "rbf") \quad (2)$$

$$SVC(C = 20, gamma = "scale", \\ kernel = "sigmoid") \quad (3)$$

3) Neural Network – For building a neural network classifier, tensorflow's keras module is used. A two layered neural network is developed with 100 neurons while the input_dim parameter set to the number of 15. The activation function for the first layer is set to "relu" as it converges the negative values to zero and positive to the exact same value. This provides an initial limitation for the range followed by the predicted output. For the second layer, the output shape from the first layer acts as the input shape generating a single output as the outcome variable is of binary nature. The activation function for the second layer is set to sigmoid as it calculates the probability of the final value being one of the outcome label interpreted as values below 0.5 to be 0 and greater than 0.5 to 1. The model is compiled with the loss metric set to "binary_crossentropy" and optimizer initialised to "adam" while "accuracy" being the optimising metric. The model is fit with 100 epochs with batch sie of 10% data points.

Finally, for learning curve, the model is evaluated over various training sizes where "accuracy" represents the train accuracy and "val_accuracy" representing the validation accuracy extracted from the model's history. In this case, the final value of "accuracy" and "val_accuracy" is calculated by taking the mean of these values over the model's history for that particular training size. For validation curve, the total number of epochs is set to 100 while 80 percent of the dataset contributing towards training set and 20 percent for the validation set. All the values for other hyper-parameters in validation curve remains the same as that of learning curve.

Company Bankruptcy dataset

1) K Nearest Neighbors
First, the weight is set to "uniform" so that the algorithm itself determines the majority of the class labels to be the outcome label for the data point under test. For stratified sampling, 5 Fold cross validation and p = 2 is defined indicating Euclidean distance. These setting of hyper-parameters are common for both learning curve as well as validation curve for comparative analysis. The number of neighbors for learning curve is set to 10. For validation curve, the model is screened upon various values of number of nearest neighbors(n_neighbors) between the range of 1 and 100, keeping the values of other hyper-parameters intact. The final equation of the KNN model initialization for plotting learning curve is specified in equation 4.

$$KNeighborsClassifier(n\_neighbors = 10, \\ weights = "uniform", p = 2, n\_jobs = 10) \quad (4)$$

2) Support Vector Machine The SVM classifiers are implemented with Gaussian (rbf), and Sigmoid kernels with C (regularization) set to 400 for learning curve as well as the validation curve. For learning curve, gamma is set to "auto" for both the kernels. The main equation for learning curve estimator with kernel rbf and sigmoid is defined in equation 5 and equation 6 respectively. Finally, for evaluating the classifier in form of validation curve, it is tested for various values of gamma between the range of 0.1 to 0.2.

$$SVC(C = 400, gamma = "auto", \\ kernel = "rbf") \quad (5)$$

$$SVC(C = 400, gamma = "auto", \\ kernel = "sigmoid") \quad (6)$$

3) Neural Network For building a neural network classifier, two layered architecture with 100 neurons and input_dim set to the number of independent features of the dataset. The activation function for the first layer is set to "relu" while the second layer to sigmoid as its ability to

transform the probabilities to discrete values of 0 and 1. The model is compiled with the loss metric set to ”binary_crossentropy” and optimizer initialised to ”adam” while ”accuracy” being the optimising metric. To fit the model over the dataset, 100 epochs are followed with the batch size of 10 percent of the dataset per epoch. For evaluating the model using learning curve and validation curve, training split is 80 percent of the entire dataset while validation set being the remaining 20 percent. The number of epochs and batch sizes are also set to the exact same values of 100 and 10 percent of the total observations respectively.

D. Learning Curve and Validation Curve

Learning curves and validation curves for KNN and SVM for both the datasets are implemented using learning_curve() and validation_curve() methods from sklearn preprocessing. Both the curves follow similar parameters setting with the difference only in the estimator, which is seen in equations from previous section. These curves are generated using Python's matplotlib library.

1) Learning Curve
   Learning curve helps to understand how the models' performance varies over various training sizes. To plot learning curve for both the datasets, the learning_curve() method splits the dataset into training sizes within the range of 10% to 90% with a step size of 10%. The training space is defined using numpy's arange() function. The independent features of the dataset are denoted by X while the dependent (outcome) feature is indicated by Y, passed as the input to the learning_curve() function. The models are evaluated over ”accuracy” metrics and random_state is set to 29. Finally, the data points from X and Y are shuffled at each iteration. These hyper-paramaters settings are common for both the datasets. However, Stratified K fold cross validation is used, indicated by cv, set to 2 for Diabetes Classification dataset and 5 for Company Bankruptcy dataset. At the end, the models indicated by the equations from previous section, are passed as estimators to the learning_curve(). These setting are same for both KNN, and SVM algorithm and for both the dataset.
   For plotting the learning curve of Neural Network, manual method is used where the dataset is splitted iteratively over the train sizes between the range of 0.1 (10% of entire dataset) to 0.9 (90% of the entire dataset), using sklearn's train_test_split() function. At each iteration the model's history is explored to get the average training accuracy and validation accuracy. Finally, these average accuracies are plotted to form the learning curve.
2) Validation Curve
   On the other hand, validation curve illustrates the models' performance over various ranges of one of their hyper-parameters. For implementing validation curve, sklearn's validation_curve() function is leveraged. Here, the parameter under test is specified using param_name attribute for the algorithm and the range is specified by param_range attribute. Stratified k fold validation is implemented here while accuracy specified as the evaluation metric. Finally, the model is instantiated through estimator parameter with user defined values for the remaining hyper-parameters.
   In case of Diabetes classification dataset, cv is set to 2 indicating 2 fold stratified cross validation for KNN as well as SVM classifier. For KNN classifier, the evaluation is based upon number of neighbors between the range of 1 to 100 with a step size of 10. Whereas, in case of SVM classifier, the influence of C (regularization) parameter between the range of 1 to 10 with a step size of 1 studied for ”rbf” and ”sigmoid” kernels.
   For Company Bankruptcy dataset, 5 fold stratified cross validation is favoured for KNN and SVM classifier. KNN classifier is evaluated over number of neighbors in the range of 1 to 100, with weight set to ”uniform”. The SVM classifier is analysed for the gamma hyper-parameter in the range between 0.1 to 0.2 with step size of 0.01 while C (regularization) set to 400.
   Finally, the validation curve of Neural Network follows a similar architecture for both the datasets. The classifier is evaluated for its accuracy over the number of epochs. The models' history is explored to analyse its performance for training set as well as validation set. The other hyper-parameters are set to the exact same values as discussed in Algorithms section.

Results

The graphs for learning curve for all the three algorithms have training sizes on x-axis while validation curve has the hyper-parameter range shown on x-axis. The y-axis for both the curves indicates accuracy.

1) K Nearest Neighbor (KNN)
   The results for K Nearest Neighbors is shown graphically in the form of Learning curve and Validation curve. Both the curves are discussed in detail sequentially for both the datasets.
   Beginning with Diabetes Classification Dataset, the learning curve shown in figure 3 illustrates a positive trend as the training sizes increases. Although the training set and validation set move away with a minor degree from each other during the training size of 30 percent (indicated by 0.3), the classifier learns quickly in the next iteration making training set and validation set to follow similar path within the close proximity to each other, taking the average accuracy to 94.7 percent.
   Figure 4 illustrates the validation curve for Diabetes classification dataset, indicating the KNN classifier tends to learn better as the number of neighbors for classification increases. Showcasing an inverse movement with respect to each other until 10 nearest neighbors, the training curve and validation curve follows a similar path once the number of neighbors reach to 30. This indicate

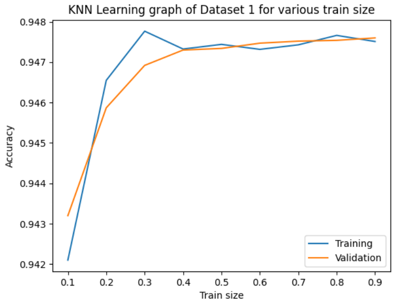

Fig. 3. Learning Curve of KNN classifier for Diabetes Classification dataset

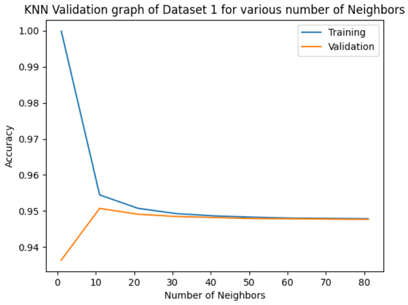

Fig. 4. Validation Curve of KNN classifier for Diabetes Classification dataset

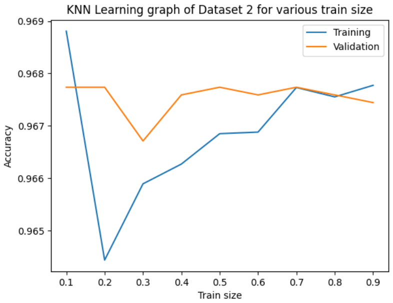

Fig. 5. Learning Curve of KNN classifier for Company Bankruptcy dataset

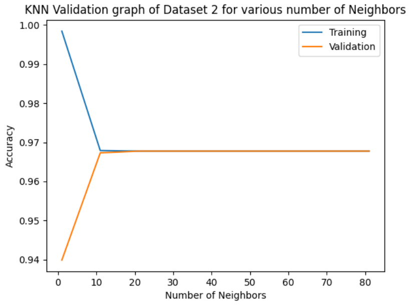

Fig. 6. Validation Curve of KNN classifier for Company Bankruptcy dataset

that the classifier is gradually learning and is able to predict the unseen samples with an average accuracy of 95 percent.

In case of Company Bankruptcy dataset, figure 5 showcase the learning curve of KNN classifier where a mixed performance where no mapping between the training set and validation set can be seen till training size of 30 percent is reached; after which, the learning experience appears. The two curves converge during 70 percent training size before following opposite path at the end.

However, the validation curve of Company Bankruptcy dataset as illustrated in figure 6 indicate that the classifier's accuracy converges towards an accuracy of 96.5 percent following an opposite movement in the direction. After the increase of number of neighbors from 10 percent, the classifier stabilises, showing no influence till the end.

2) Support Vector Machine (SVM)

In figure 7, the learning curve of SVM with rbf kernel for Diabetes Classification dataset is shown, where a positive learning rate in terms of accuracy with an increase in the training sizes is seen. The training curve and validation curve renounce from each other between the training size of 20% and 60%. However, during these iteration the trend maintains its gradual incline finally converging at 70% of training size. The learning curve shows the classifier's average accuracy of 96.2%.

In figure 8, the learning curve of SVM for sigmoid curve illustrates inverse relationship between the learning rate and training size. After a steep decline from training size of 10% to 20%, the momentum of the trend decreases with a gradual decrease in the accuracy. However, the training set and validation set move in the close proximity showing average accuracy of 83.42%.

The validation curve of SVM for rbf and sigmoid kernel for Diabetes Classification dataset is illustrated in figure 11, and figure 12 respectively. First, the rbf kernel shows a positive growth for training, and validation set overlapping over each other. However, the influence of learning rate due to training set is evident from the validation set which follows similar trajectory. This indicate that the increasing the value of Regularization parameter enables the classifier to provide better accuracy. While the rbf kernel performs well, the sigmoid curve shows an absolute negative validation curve. The curve follows a smooth trajectory indicating the negative influence of Regularization parameter (C) as its value increases.

For Company Bankruptcy dataset, the learning curve for rbf kernal and sigmoid kernel is shown in figure 9 and figure 10 respectively. In case of rbf kernel, the classifier shows a huge gap indicating an over-fit model. However, in case of sigmoid kernel, the classifier is fitted better though, the accuracy rate declines significantly over two iterations before increasing gradually thereafter. The average accuracy of the model is approximately 93.5 percent.

While analysing the Company Bankruptcy dataset validation curve of SVM for rbf and sigmoid kernel from

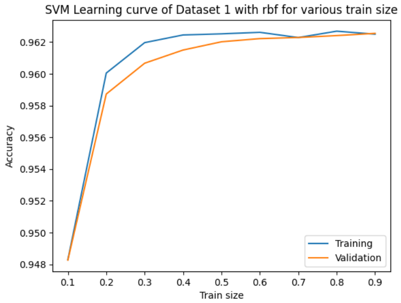

Fig. 7. Learning Curve of SVM classifier with Gaussian (rbf) kernel for Diabetes Classification dataset

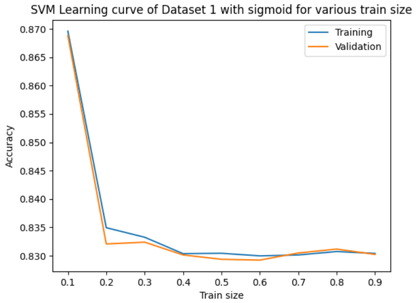

Fig. 8. Learning Curve of SVM classifier with Sigmoid kernel for Diabetes Classification dataset

figure 13 and figure 14, the classifier shows negative learning rate for various gamma values. The over-fitted model is evident by the gap between the training set and the validation set. However, in case of sigmoid kernel, the classifier shows an excellent learning rate as the value of gamma increases. A significant uptrend can be seen as the gamma value reaches to 0.5 from 0.1, maintaining a stable accuracy at 96.75 percent once the gamma value moves from 0.5 to 1.0.

3) Neural Network

Figure 15 showcase the learning curve of Neural network classifier for Diabetes Classification dataset indicating a mixed performance over various training

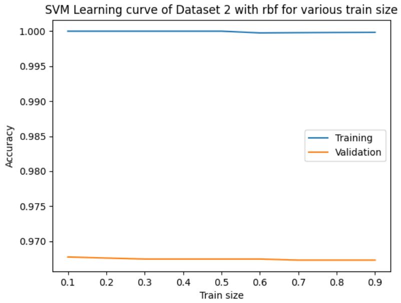

Fig. 9. Learning Curve of SVM classifier with Gaussian (rbf) kernel for Company Bankruptcy dataset

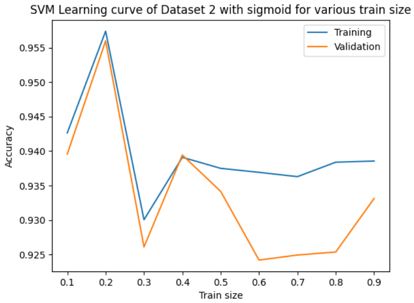

Fig. 10. Learning Curve of SVM classifier with Sigmoid kernel for Company Bankruptcy dataset

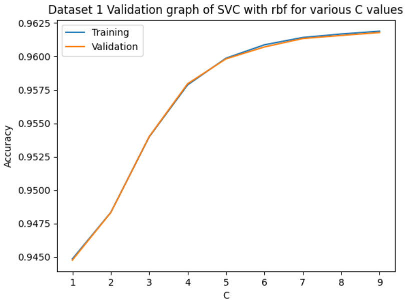

Fig. 11. Validation Curve of SVM classifier with Gaussian (rbf) kernel for Diabetes Classification dataset

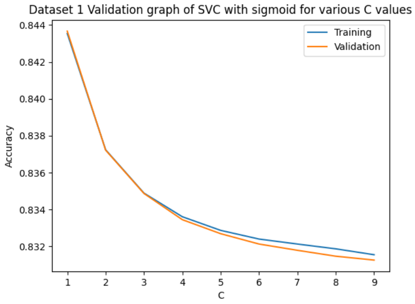

Fig. 12. Validation Curve of SVM classifier with Sigmoid kernel for Diabetes classification dataset

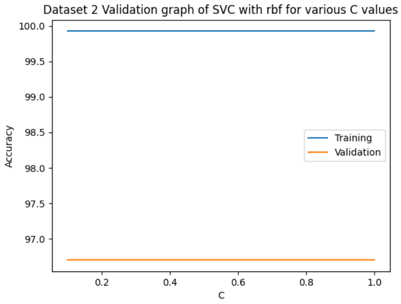

Fig. 13. Validation Curve of SVM classifier with Gaussian (rbf) kernel for Company Bankruptcy dataset

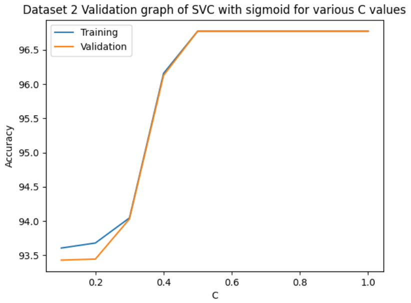

Fig. 14. Validation Curve of SVM classifier with Sigmoid kernel for Company Bankruptcy dataset

sizes. Initially, the classifier performs well showing an uptrend in accuracy up to the training size of 40% with a steep decrease in accuracy in the next iteration. Following trend shows spike at 60% before declining for one iteration while increasing significantly in the next; ending with an inverse movement. The average accuracy is approximated to 92%.

The validation curve of Neural network classifier for Diabetes classification dataset can be seen in figure 16. An exponential learning rate is evident from beginning to 20 epochs followed by a lower momentum while maintaining the uptrend gradually. Both the training set and validation set follow similar trajectory, indicating the excellent learning rate with an average accuracy of 92 percent.

The learning curve of neural network for Company Bankruptcy dataset illustrated in figure 17 shows similar trajectories for training set and the validation set, where the latter shows higher sensitivity to the training size. Both the sets in the first half of the training sizes illustrates a downward trend while spiking towards higher accuracy once training size of 60% is reached, making the average accuracy to 91.5%.

The validation curve of neural network for Company Bankruptcy dataset is seen in figure 18 where the training set and the validation set showcase a steep spike till tenth epoch. Once the classifier reaches twentieth epoch, multiple spikes and trough can be seen until the end while maintaining same momentum by both the sets. This makes the average accuracy of the classifier to 95 percent.

IV. Discussion

As per the results obtained from learning curve and validation curve of both the datasets, the three algorithms are compared over their accuracy.

Diabetes Classification dataset

First, as seen in Data exploration section, Diabetes classification dataset have majority linearly separable data points which influenced the performance of the three algorithms. KNN classifier showcased a rising accuracy as the training size

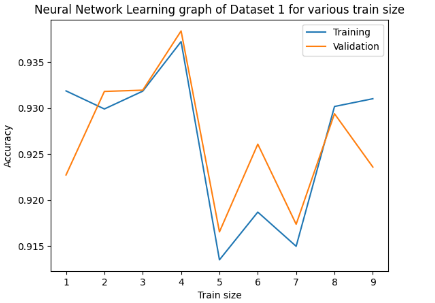

Fig. 15. Learning Curve of Neural Network classifier for Diabetes Classification dataset

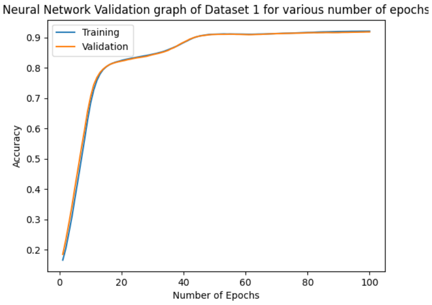

Fig. 16. Validation Curve of Neural Network classifier for Diabetes Classification dataset

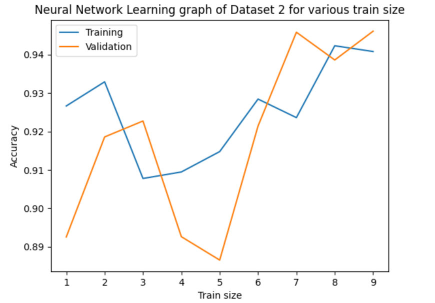

Fig. 17. Learning Curve of Neural Network classifier for Company Bankruptcy dataset

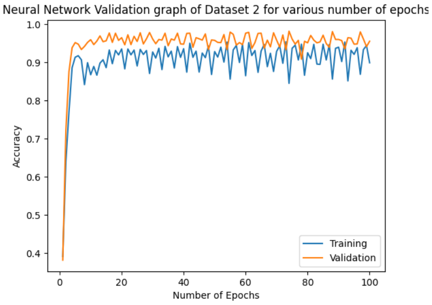

Fig. 18. Validation Curve of Neural Network classifier for Company Bankruptcy dataset

increases due to uniform weights. However, the accuracy in case of validation curve outperforms the accuracy in learning curve because the number of neighbors is not optimal, leading to consideration of data points that acts a noise, giving improper classification. The effect of number of neighbors is studied from validation curve, where the graph follows a slight declining trajectory as the number of neighbors increases.

The learning curve of SVM classifier for rbf kernel performs better than the sigmoid kernel since it is easier for the rbf kernel to classify the data points accurately given the linearly separable structure of the dataset. The sigmoid curve however, tries to plot the hyperplane with sigmoid boundary around the data points and hence as the data size increases, its accuracy decreases. This behavious is validated via validation curve wherein, the increase in the regularization parameter C benefits the accuracy of the rbf kernel while, the accuracy declines in the case of Sigmoid kernel showing no influence of the C. The performance might be debatable if linear kernel is used instead of sigmoid kernel and contrasted with the accuracy of rbf kernel.

On the other hand, the learning curve of neural network classifier shows abrupt spikes and troughs throughout the graph due to large learning rate. This could be balanced if the optimizer is tweaked for its hyper-parameter settings or optimizer is changed. Whereas, the validation curve shows a spontaneous learning experience over the training set and validation set since both tends to overlap each other while gaining positive trajectory indicating the high accuracy rate of the classifier.

Finally, from the results of all the three algorithms, the initial hypothesis of SVM classifier outperforming the KNN classifier and Neural Network holds true based upon their accuracies, standing at 96.2% for SVM, 95% for KNN, and 90% for Neural network. This is due to the fact that the linearly separable data points enabled SVM to classify better.

Company Bankruptcy dataset

The Company Bankruptcy dataset possess all the independent features of continuous nature overlapping over each other. However, out of the 6820 data samples, 6600 belong to class 0 while 220 belongs to class 1, creating a ratio of 1 : 330. Although KNN classifier, SVM classifier, and Neural network classifier show greater accuracies evident from validation curve, the fact that unequal distribution of the outcome lables enable these classifiers to have accuracies of 97%, 96.8%, and 96% respectively. Hence, the existence of continuous indepedent features did not have any influence over these supervised learning classifiers as the proportion of outcome labels has a tremendous difference leading to the pseudo accuracies. To overcome this problem, the number of stratified fold cross validation should be increased that ensures equal proportion of outcome labels in every split of the cross validation sets.

However, the learning curve for all the three classifiers shows a zigzag pattern indicating the shift in accuracies over the training sizes. The sigmoid kernel of SVM showcase better learning rate than that of the rbf kernel which is overfitted, given the disproportionate outcome labels. This problem can be overcomed by extended search of best set of hyper-parameter using modules such as GridSearchCV.

Finally, the hypothesis that Neural network classifier will outperform KNN classifier and SVM classifier based upon the structure of the dataset does not hold true as it is evdident from their accuracies.

V. Conclusion

The three supervised learning algorithms, K Nearest Neighbors, Support Vector Machine (SVM), and Neural network are implemented and compared over Diabetes classification dataset and Company Bankruptcy dataset. Both the datasets are mutually exclusive to each other, while showcasing a different structural patterns. For Diabetes classification dataset, its linear structure is highly favoured by SVM for rbf kernel as it draws a higher dimensional hyperplane distinguishing the data points. While Neural network classifier showcased a spontaneous learning experience, the accuracy is outperformed due to the fact that hyper-parameters such as learning rate and optimizer, are not explored and tuned. On the other hand, the KNN classifier operates in single dimensional space and since the data points are uniformly weighted, the outliers aggravate the performance.

The performance of KNN classifier for Company Bankruptcy dataset outperforms both SVM classifier and Neural network classifier as the effect of outliers does not hamper the accuracy of the classifier. Also, the rbf kernel of SVM is over-fitted while sigmoid curve also suffers from the gamma value tuning leading to the reducing the accuracy. Neural network in this case follows similar architecture as that from Diabetes Classification dataset. Hence, the problem of learning rate tuning is evident as the smoothness in absent from the curve.

Finally,it is understood that SVM leverages its kernel functions have an advantage due to its ability to draw a higher dimensional hyperplane to classify the data points. KNN classifier also works tremendously well for classification problems where the data points are less scattered over the space. Additionaly, an extended search for best number of neighbors elevate can its performance. Also, Neural network has tons of potential of being the best performing algorithm if the optimizer is fine tuned. The importance of learning curve for evaluating the model's performance over various training sizes for determine the fit of the model, and finding an optimal hyper-parameter over a given sample range for its influence on the classifiers' performance metrics is also studied.

References

[1] Priyam Chowksi, Comprehensive Diabetes Clinical Dataset(100k rows), https://www.kaggle.com/datasets/priyamchoksi/100000-diabetes-clinical-dataset, July 2024.

[2] Fedesoriano, Company Bankruptcy Prediction, https://www.kaggle.com/datasets/fedesoriano/company-bankruptcy-prediction.