Assignment 3

PART A

QA1.What is the difference between SVM with a hard margin and a soft margin?

Vector machines can be used for both classification and regression. A potent machine learning technique is vector machines. Think about a set of data that you wish to categorize first. These facts suggest that they might be differentiable or that the splitting of hyperplanes might not be linear. To guard against incorrect classifications for linear objects, we use SVMs with a big margin. The primary distinction between soft margin and hard margin

Hard margin classification has two key drawbacks. Hard Margins are extremely sensitive to outliers, and Hard Margin Classification only functions if the data can be linearly separated. Hard margins will cause an error-free model to be overfitted. Allowing for errors in the training set can occasionally be advantageous because it may result in a more generalizable model when applied to fresh datasets.

the soft margin SVM uses a similar optimization process. First, we permit misclassifications to occur in this scenario. Therefore, we must reduce the misclassification error, which adds another restriction to our list of issues. Second, we must define a loss function to reduce the error. Hinged loss is a typical loss function used on soft margins.

QA2. What is the role of the cost parameter, C, in SVM (with soft margin) classifiers?

In an SVM you are searching for two things: a hyperplane with the largest minimum margin, and a hyperplane that correctly separates as many instances as possible. The problem is that you will not always be able to get both things.

When C is set to a higher value, the SVM tries to minimize the misclassification rate as much as possible, even if it means creating a more complex decision boundary that may be more prone to overfitting. Conversely, when C is set to a lower value, the SVM allows more misclassifications and tries to find a simpler decision boundary that is more likely to generalize to unseen data.

The choice of the optimal value of C is usually done through hyperparameter tuning, which involves trying out different values of C and evaluating their performance using a validation set or cross-validation.

For each incorrectly classified data point, the C parameter applies a penalty. If c is low, choosing a decision boundary with a high margin comes at the expense of more misclassifications because the penalty for incorrectly classified points is low. The SVM attempts to reduce the number of incorrectly categorized samples when c is big, leading to a decision boundary with a lower margin. The punishment varies depending on the misclassification.

QA3 Will the following perceptron be activated (2.8 is the activation threshold)

This perceptron will not activate as the result is -2.14**. This happens because of 11.1*-0.2 = -2.22, when you add the 0.08 from the resulting math at the top, resulting in -2.14.

QA 4. What is the role of alpha, the learning rate in the delta rule?

The step size, or the amount by which the weights are updated throughout the training process, is controlled by the learning rate, or alpha. To ensure that the network converges to a good solution while avoiding problems like delayed convergence or oscillations, the learning rate is a hyperparameter that needs to be carefully set.

When the delta rule updates the weights of a neuron, it multiplies the input value (or the activation value of the previous layer) by the error term and the learning rate alpha and adds the result to the current weight. In other words, the learning rate determines how much the weight is adjusted in the direction of the gradient of the error function.

A high learning rate causes the weights to change significantly with each iteration, which can speed up convergence but can cause the weights to overshoot the ideal values and miss the error function minimum. While a low learning rate can cause slow convergence, it can also result in a more accurate and stable solution because the weights are only updated a small amount at each iteration.

the choice of learning rate is crucial to the effectiveness of the delta rule, and finding an appropriate value frequently necessitates experimenting. To balance the trade-off between convergence speed and accuracy, it is common practice to utilize a declining learning rate schedule, where the learning rate declines as the training proceeds.

PART B

QB1. Build a linear SVM regression model to predict Sales based on all other attributes ("Price", "Advertising", "Population", "Age", "Income" and "Education"). Hint: use caret train () with The method is set to "svmLinear". What is the R-squared of the model?

In the above model, I added a train control section that adds cross validation to the model.the R square of the model is 0.3702692,The tuning parameter "c" was held constant at a value of 1

QB2 Customize the search grid by checking the model's performance for C parameter of 0.1,.5,1 and 10 using 2 repeats of 5-fold cross-validation.

In this step, we have added a search grid at the desired points. We have also adjusted the cross-validation as instructed. The interesting outcome is that as we increase C, the change decreases. The difference between 1 and 10 is much less than between 0.1 and 0.5. With this, we can see that **0.1** is the best C available.

QB3 Train a neural network model to predict Sales based on all other attributes ("Price", "Advertising", "Population", "Age", "Income" and "Education"). Hint: use caret train() with method set to "nnet". What is the R-square of the model with the best hyper parameters (using default caret search grid) – hint: don't forget to scale the data.

The model selected size = 1 and decay 0.1 as the most optimal model using RMSE. The specific Rsquared for this model is decay 1e-01 the closest Rsquared 0.3402767 for a model with RMSE of 2.319733

QB4 Consider the following input:

- Sales=9
- Price=6.54
- Population=124

- Advertising=0
- Age=76
- Income= 110
- Education=10

What will be the estimated Sales for this record using the above neuralnet model?

Now that we have established the test data that is needed, it is time to predict using our network.

The resulting Pred_sales variable would contain a vector of predicted Sales values, one for each row in the Test data frame, based on the neural network model's estimates of the relationship between the predictor variables and the Sales response variable. The sales predicted value would be "11.46031."