**Assignment One**

**Part A**

1. Regularization is a technique that is utilized to establish a penalty on the complexity of a training model. The main objective of regularization when training predictive models is to prevent overfitting. Overfitting happens when the training model grows too complex and begins to fit the noise in the data instead of the underlying patterns. The consequence of overfitting leads to a model that performs well in the training data but does not generalize optimally to the unseen data. The penalty regularization imposes is usually added to the loss function in which the model is trying to optimize during the training. There are multiple regularization techniques which include Dropout and early stopping.

Dropout happens when some of neurons in the network are randomly dropped. This assists the model from relying too much on a specific input which can lead to overfitting. Early stopping is a technique which monitors the model’s performance and stops the training when the performance begins to degrade.

1. The role of a loss function in a predictive model is to quantify the difference between the predicted output of the model and the actual output. It measures how well the model is performing and provides a way to optimize the model during training.

In a predictive model the role of a loss function to quantify the difference the actual and predicted output of the model. The loss function measures how efficient the model performs and provides a way to optimize the model during training.

For classification two common loss functions are:

Binary Cross-Entropy: This function measures the difference between the predicted and actual probabilities of the positive class.

Categorical Cross-Entropy: This loss function is typically used for multi-class classification problems and measures the difference between the predicted and actual probabilities for each class and averages them across all classes.

For regression models, two common loss functions are:

Mean Squared Error also known as MSE: This loss function measures the average squared difference between the predicted and actual values.

Mean Absolute Error also known as MAE: This loss function measures the average absolute difference between the predicted and actual values. And it also less sensitive to outliers than the MSE function.

1. No, you cannot fully trust the model because there is a risk of overfitting. Due to the model having a small training error it indicates that the model may have memorized the data instead of continuing to learn underlying patters and is at a high risk of not generalizing well to new data which it has not seen before.

That’s why it’s crucial to evaluate the model’s performance on a separate validation set, which can assist in determining whether the model is overfitting to the data, or it is generalizing well to unseen data.

1. In Lasso regression models the lambda parameter controls the strength of the L1 regularization.

In Ridge regression models the lambda parameter controls the strength of the L2 regularization. The lambda parameter is a hyperparameter which controls the amount of regularization is applied to the model. It continuously needs to be tuned during the training. A small lambda value leads to less regularization and a more complex model, and a large lambda value results in more regularization and a simpler model.

**PART B**

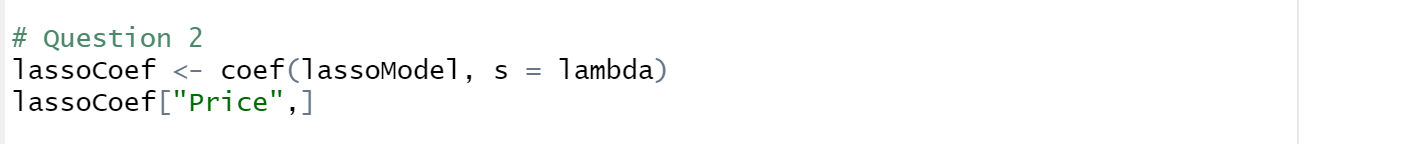
QB1. Build a Lasso regression model to predict Sales based on all other attributes ("Price", "Advertising", "Population", "Age", "Income" and "Education"). What is the best value of lambda for such a lasso model?

**Graphical user interface, text

Description automatically generated**

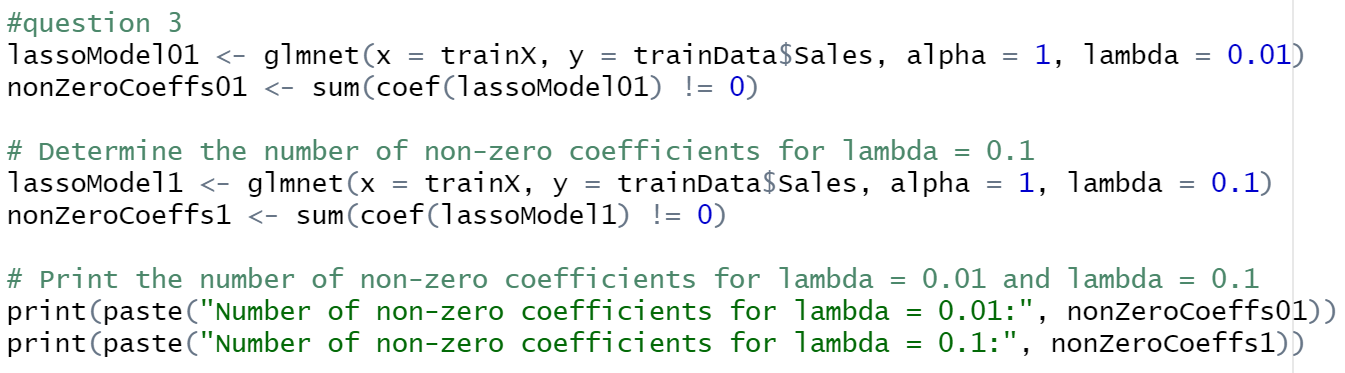
The optimal value of lambda is the value that minimizes the cross-validation error which is 0.00451294034403994

QB2. What is the coefficient for the price (normalized) attribute in the best model (i.e. model with the optimal lambda)? (15% of total points)



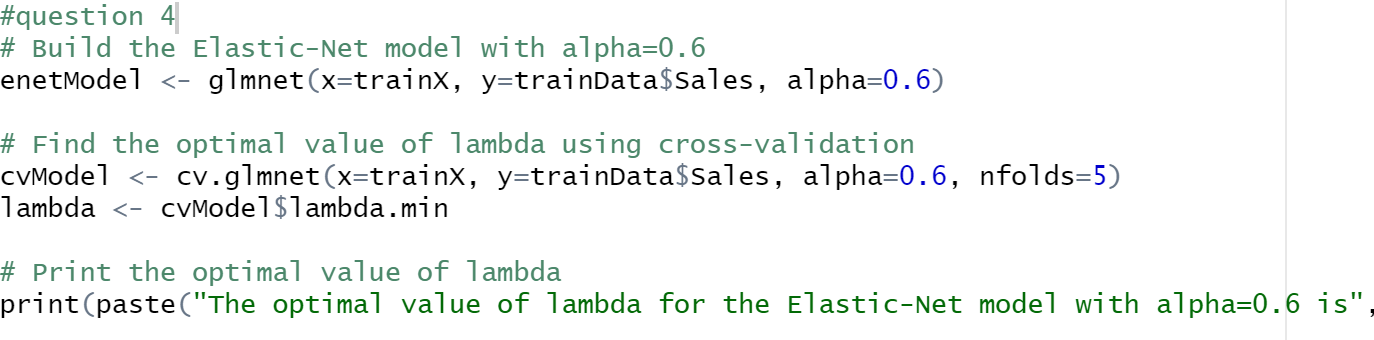
The coefficient for the price attribute in the best model is -1.299902

QB3. How many attributes remain in the model if lambda is set to 0.01? How that number changes if lambda is increased to 0.1? Do you expect more variables to stay in the model (i.e., to have non-zero coefficients) as we increase lambda? (15% of total points)



When the lambda is set to 0.01 there are seven attributes left in the model. When the model is set to 0.1 there are six non-zero coefficients’ attributes left. Based on the model, I expect as the lambda increases, more variables will have zero coefficients and thus be removed from the model.

QB4. Build an elastic-net model with alpha set to 0.6. What is the best value of lambda for such a model? (10% of total points)



The optimal value of lambda for the Elastic-Net model with alpha=0.6 is 0.00624453726258919"