Machine learning worksheet 1

Q1 to Q11 have only one correct answer. Choose the correct option to answer your question.

1. Which of the following methods do we use to find the best fit line for data in Linear Regression?
   A) Least Square Error B) Maximum Likelihood
   C) Logarithmic Loss D) Both A and B
   ANS : A) Least Square Error

2. Which of the following statement is true about outliers in linear regression?
   A)Linear regression is sensitive to outliers B) linear regression is not sensitive to outliers
   C) Can't say D) none of these
   ANS: A) Linear regression is sensitive to outliers

3. A line falls from left to right if a slope is_____?
   A)Positive B) Negative
   C) Zero D) Undefined
   ANS: B) Negative

4. Which of the following will have symmetric relation between dependent variable and independent variable?
   A)Regression B) Correlation
   C) Both of them D) None of these
   ANS:A) Regression

5. Which of the following is the reason for over fitting condition?
   A)High bias and high variance B) Low bias and low variance
   C) Low bias and high variance D) none of these
   ANS: C) Low bias and high variance

6. If output involves label then that model is called as:
   A)Descriptive model B) Predictive modal
   C) Reinforcement learning D) All of the above
   ANS: B) Predictive modal

7. Lasso and Ridge regression techniques belong to_____?
   A)Cross validation B) Removing outliers
   C) SMOTE D) Regularization
   ANS: D) Regularization

8. To overcome with imbalance dataset which technique can be used?
   A)Cross validation B) Regularization
   C) Kernel D) SMOTE
   ANS: D) SMOTE

9. The AUC Receiver Operator Characteristic (AUCROC) curve is an evaluation metric for binary classification problems. It uses_____to make graph?
   A)TPR and FPR B) Sensitivity and precision
   C) Sensitivity and Specificity D) Recall and precision

   ANS : C) Sensitivity and Specificity

10. In AUC Receiver Operator Characteristic (AUCROC) curve for the better model area under thecurve should be less.

A) True B) False

Ans : B) False

11. Pick the feature extraction from below:

A. Construction bag of words from a email

B. Apply PCA to project high dimensional data

C. Removing stop words

D. Forward selection

Ans: B) Apply PCA to project high dimensional data

In Q12, more than one options are correct, choose all the correct options:

12. Which of the following is true about Normal Equation used to compute the coefficient of the Linear Regression?

A) We don't have to choose the learning rate.

B) It becomes slow when number of features is very large.

C) We need to iterate.

D) It does not make use of dependent variable.

Ans: A,B,C

In Q13 and Q15 are subjective answer type questions, Answer them briefly.

13. Explain the term regularization?

Regularization

Regularization is one of the most important concepts of machine learning. It is a technique to prevent the model from overfitting by adding extra information to it. Sometimes the machine learning model performs well with the training data but does not perform well with the test data. It means the model is not able to predict the output when deals with unseen data by introducing noise in the output, and hence the model is called overfitted. This problem can be deal with the help of a regularization technique. Simple model will be a very poor generalization of data. At the same time, complex model may not perform well in test data due to over fitting.

We need to choose the right model in between simple and complex model. Regularization helps to choose preferred model complexity, so that model is better at predicting. Regularization is nothing but adding a penalty term to the objective function and control the model complexity using that penalty term. It can be used for many machine learning algorithms.

In the context of machine learning, regularization is the process which regularizes or shrinks the coefficients towards zero. In simple words, regularization discourages learning a more complex or flexible model, to prevent overfitting.

The term 'regularization' refers to a set of techniques that regularizes learning from particular features for traditional algorithms or neurons in the case of neural network algorithms. It normalizes and moderates weights attached to a feature or a neuron so that algorithms do not rely on just a few features or neurons to predict the result. This technique helps to avoid the problem of overfitting.To understand regularization, let's consider a simple case of linear regression. Mathematically, linear regression is stated as below:

y = w0 + w1x1 + w2x2 + ….. + wnxn

where y- value to be predicted.

x1,x2,….xn– features that decides the value of y

Wo – is the bias

w_1, w_2, ….., w_n are the weights attached to x_1, x_2, …., x_n relatively.

To do so, we need to use a loss function and find optimized parameters using gradient descent algorithms and its variants.To know more about building a machine learning application and the process, check out below blog:

How to Develop Machine Learning Applications for Business

The loss function called 'the residual sum of square' is mostly used for linear regression. Here's what it looks like :

$$RSS = \sum_{j=1}^{m} \left(Yi - Wo - \sum_{i=1}^{n} Wi\, Xji \right)^2$$

Next, we will learn bias (or intercept) and weights (also identified as parameters and coefficients) using the optimization algorithm (gradient descent) and data. If your dataset does have noise in it, it will face overfitting problem and learned parameters will not generalize well on unseen data.

Regularization theory

The regularization theory **prescribes that the model with the smallest R(θ) should be chosen**. The first term of the above equation is a measure of the discrepancy between the data and the model's prediction, that is, the empirical risk of the model. The second term, S(z), is a measure of complexity of the model.

Regularization Techniques

There are three main regularization techniques, namely:

- Regression (L2 Norm)
- Lasso (L1 Norm)
- Dropout.

Ridge and Lasso can be used for any algorithms involving weight parameters, including neural nets. Dropout is primarily used in any kind of neural networks e.g. ANN, DNN, CNN or RNN to moderate the learning. Let's take a closer look at each of the techniques.

14.Which particular algorithms are used for regularization?

There are three main algorithms are used regularization namely:

a. Ridge Regression (L2 Norm)

b. Lasso (L1 Norm)

c. Dropout

a.Ridge Regression (L2 Regularization)

- Ridge regression is one of the types of linear regression in which a small amount of bias is introduced so that we can get better long-term predictions.
- Ridge regression is a regularization technique, which is used to reduce the complexity of the model. It is also called as **L2 regularization**.
- The amount of bias added to the model is called **Ridge Regression penalty**. We can calculate it by multiplying with the lambda to the squared weight of each individual feature.

When using this technique, we add the sum of weight's square to a loss function and thus create a new loss function which is denoted thus:

$$\text{Loss} = \sum_{j=1}^{m}\left(Yi - Wo - \sum_{i=1}^{n} Wi\,Xji\right)^2 + \lambda\sum_{i=1}^{n} Wi^2$$

As seen above, the original loss function is modified by adding normalized weights. Here normalized weights are in the form of squares. You may have noticed parameters λ along with normalized weights. λ is the parameter that needs to be tuned using a cross-validation dataset. When you use λ=0, it returns the residual sum of square as loss function which you chose initially. For a very high value of λ, loss will ignore core loss function and minimize weight's square and will end up taking the parameters' value as zero. Now the parameters are learned using a modified loss function. To minimize the above function, parameters need to be as small as possible. Thus, L2 norm prevents weights from rising too high. A general linear or polynomial regression will fail if there is high collinearity between the independent variables, so to solve such problems, Ridge regression can be used.

b. **Lasso Regression (L1 Regularization)**

- Lasso regression is another regularization technique to reduce the complexity of the model. It stands for **Least Absolute and Selection Operator.**
- It is similar to the Ridge Regression except that the penalty term contains only the absolute weights instead of a square of weights.
- Since it takes absolute values, hence, it can shrink the slope to 0, whereas Ridge Regression can only shrink it near to 0.

Also called lasso regression and denoted as below:

$$\text{Loss} = \sum_{j=1}^{m}\left(Yi - Wo - \sum_{i=1}^{n} Wi\,Xji\right)^2 + \lambda\sum_{i=1}^{n} |Wi|$$

This technique is different from ridge regression as it uses absolute weight values for

normalization. λ is again a tuning parameter and behaves in the same as it does when using ridge regression. As loss function only considers absolute weights, optimization algorithms penalize higher weight values. In ridge regression, loss function along with the optimization algorithm brings parameters near to zero but not actually zero, while lasso eliminates less important features and sets respective weight.

Hence, the Lasso regression can help us to reduce the overfitting in the model as well as the feature selection.

c.Dropout

Dropout is a regularization technique used in neural networks. It prevents complex co-adaptations from other neurons. In neural nets, fully connected layers are more prone to overfit on training data. Using dropout, you can drop connections with *1-p* probability for each of the specified layers. Where *p* is called keep probability parameter and which needs to be tuned. With dropout, you are left with a reduced network as dropped out neurons are left out during that training iteration.

Dropout decreases overfitting by avoiding training all the neurons on the complete training data in one go. It also improves training speed and learns more robust internal functions that generalize better on unseen data. However, it is important to note that Dropout takes more epochs to train compared to training without Dropout (If you have 10000 observations in your training data, then using 10000 examples for training is considered as 1 epoch). long with Dropout, neural networks can be regularized also using L1 and L2 norms. Apart from that, if you are working on an image dataset, image argumentation can also be used as a regularization method.

15.Explain the term error present in linear regression equation?

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price,** etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

Error present in linear regression

a. Mean_square error

b. Mean absolute error

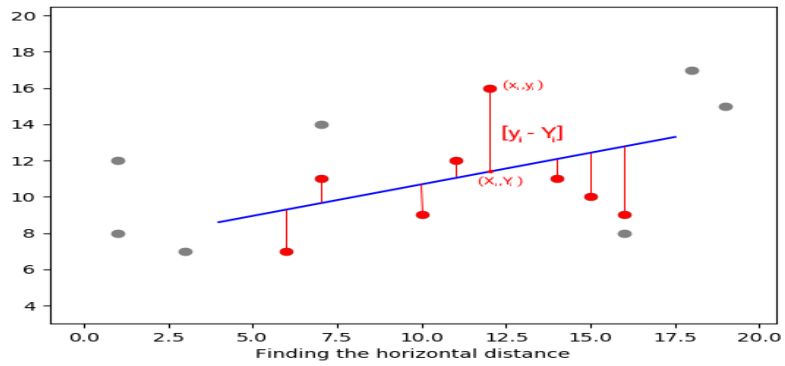
c. Root Mean Squared Error or RMSE

a.Mean-square error

Linear regression most often uses mean-square error (MSE) to calculate the error of the model. MSE is calculated by measuring the distance of the observed y-values from the predicted y-values at each value of x squaring each of these distances calculating the mean of each of the squared distances.Linear regression fits a line to the data by finding the regression coefficient that results in the smallest MSE. It can be written as:

$$\text{MSE}=1\frac{1}{N}\sum_{i=1}^{n}(y_i-a_i x_i+a_o)).^2$$

Where,
N= total number of observation
Yi = Actual value
$(a1x_i+a_0)$= Predicted value



From the figure below let us assume that the blue line is the regression line and then we will try to find the perpendicular distance of the line from that point marked as red present at (x_i, y_i) on the graph.That is, we subtract the height of the line at that point x_i, that is Y_i from the real height of the point y_i and assume the value is e_i

$$e_i = y_i - Y_i$$

Now in proceeding the same way we add all the values of e_i and we get,

$$\sum e_i = \sum(y_i - Y_i)$$

some points could lie on the line, there could be a possibility that there is more number of points below the line which can make the total summation negative but convention decided to let the mse be a positive value for better understanding.

Residuals: The distance between the actual value and predicted values is called residual. If the observed points are far from the regression line, then the residual will be high, and so cost function will high. If the scatter points are close to the regression line, then the residual will be small and hence the cost function.

It is one of the most commonly used metrics, but least useful when a single bad prediction would ruin the entire model's predicting abilities, i.e when the dataset contains a lot of noise. It is most useful when the dataset contains outliers, or unexpected values (too high or too low values).

This can be implemented using **sklearn's mean_squared_error** method:

```
from  sklearn .metrics import mean _squared _error
    actual _values = [3, -0.5, 2, 7]
    predicted _values = [2.5, 0.0, 2, 8]
 mean _squared _error(actual _values, predicted _values)
```

b.Mean absolute error

We know that an error basically is the absolute difference between the actual or true values and the values that are predicted. Absolute difference means that if the result has a negative sign, it is ignored.

Hence, **MAE = True values – Predicted values**

MAE takes the **average** of this error from every sample in a dataset and gives the output.

This can be implemented using **sklearn**'s **mean_ absolute_ error** method:

```
from sklearn.metrics import mean_absolute_error
 # predicting home prices in some area
 predicted_home_prices = mycity_model.predict(X)
mean_absolute_error(y, predicted_home_prices)
```

But this value might not be the relevant aspect that can be considered while dealing with a real-life situation because the data we use to build the model as well as evaluate it is the same, which means the model has no

exposure to real, never-seen-before data. So, it may perform extremely well on seen data but might fail miserably when it encounters real, unseen data.

It is not very sensitive to outliers in comparison to MSE since it doesn't punish huge errors. It is usually used when the performance is measured on continuous variable data. It gives a linear value, which averages the weighted individual differences equally. The lower the value, better is the model's performance.

C .Root Mean Squared Error or RMSE

RMSE is the standard deviation of the errors which occur when a prediction is made on a dataset. This is the same as MSE (Mean Squared Error) but the root of the value is considered while determining the accuracy of the model.

```
from sklearn.metrics import mean_squared_error
from math import sqrt
 actual_values = [3, -0.5, 2, 7]
 predicted_values = [2.5, 0.0, 2, 8]
mean_squared_error(actual_values, predicted_values)
# taking root of mean squared error
 root_mean_squared_error = sqrt(mean_squared_error)
```

In RMSE, the errors are squared before they are averaged. This basically implies that RMSE assigns a higher weight to larger errors. This indicates that RMSE is much more useful when large errors are present and they drastically affect the model's performance. It avoids taking the absolute value of the error and this trait is useful in many mathematical calculations. In this metric also, the lower the value, better is the performance of the model.