MACHINE LEARNING

In Q1 to Q5, only one option is correct, Choose the correct option:

1. In which of the following you can say that the model is overfitting?
A) High R-squared value for train-set and High R-squared value for test-set.
B) Low R-squared value for train-set and High R-squared value for test-set.
C) High R-squared value for train-set and Low R-squared value for test-set.
D) None of the above

2. Which among the following is a disadvantage of decision trees?
A) Decision trees are prone to outliers.
B) Decision trees are highly prone to overfitting.
C) Decision trees are not easy to interpret
D) None of the above.

3. Which of the following is an ensemble technique?
A) SVM
B) Logistic Regression
C) Random Forest
D) Decision tree

4. Suppose you are building a classification model for detection of a fatal disease where detection of the disease is most important. In this case which of the following metrics you would focus on?
A) Accuracy
B) Sensitivity
C) Precision
D) None of the above.

5. The value of AUC (Area under Curve) value for ROC curve of model A is 0.70 and of model B is 0.85. Which of these two models is doing better job in classification?
A) Model A
B) Model B
C) both are performing equal
D) Data Insufficient

In Q6 to Q9, more than one options are correct, Choose all the correct options:

6. Which of the following are the regularization technique in Linear Regression??
A) Ridge
B) R-squared
C) MSE
D) Lasso

7. Which of the following is not an example of boosting technique?
A) Adaboost
B) Decision Tree
C) Random Forest
D) Xgboost.

8. Which of the techniques are used for regularization of Decision Trees?
A) Pruning
B) L2 regularization
C) Restricting the max depth of the tree
D) All of the above

9. Which of the following statements is true regarding the Adaboost technique?
A) We initialize the probabilities of the distribution as 1/n, where n is the number of data-points
B) A tree in the ensemble focuses more on the data points on which the previous tree was not performing well
C) It is example of bagging technique
D) None of the above

Q10 to Q15 are subjective answer type questions, Answer them briefly.

10. Explain how does the adjusted R-squared penalize the presence of unnecessary predictors in the model?

Ans.: The adjusted R-squared is a modified version of R-squared that adjusts for predictors that are not significant in a regression model.

Compared to a model with additional input variables, a lower adjusted R-squared indicates that the additional input variables are not adding value to the model.

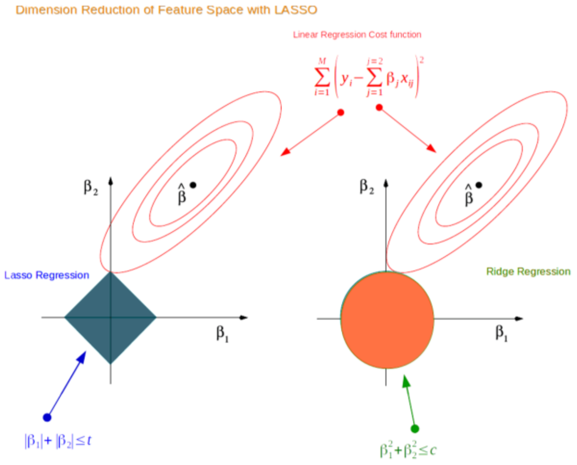
MACHINE LEARNING

Compared to a model with additional input variables, a higher adjusted R-squared indicates that the additional input variables are adding value to the model.

The adjusted R-squared compensates for the addition of variables and only increases if the new predictor enhances the model above what would be obtained by probability. Conversely, it will decrease when a predictor improves the model less than what is predicted by chance.

11. Differentiate between Ridge and Lasso Regression.

Ans.: Ridge and Lasso regression uses two different penalty functions for regularisation. Ridge regression uses L2 on the other hand lasso regression go uses L1 regularisation technique. In ridge regression, the penalty is equal to the sum of the squares of the coefficients and in the Lasso, penalty is considered to be the sum of the absolute values of the coefficients. In lasso regression, it is the shrinkage towards zero using an absolute value (L1 penalty or regularization technique) rather than a sum of squares(L2 penalty or regularization technique).


Dimension Reduction of Feature Space with LASSO
Linear Regression Cost function
$\sum_{i=1}^{M}\left(y_i-\sum_{j=1}^{j=2}\beta_j x_{ij}\right)^2$
β_2
$\hat{\beta}$
β_2
$\hat{\beta}$
Lasso Regression
Ridge Regression
β_1
β_1
$|\beta_1|+|\beta_2|\leq t$
$\beta_1^2+\beta_2^2\leq c$

Since we know that in ridge regression the coefficients can’t be zero. Here, we either consider all the coefficients or none of the coefficients, whereas Lasso regression algorithm technique, performs both parameter shrinkage and feature selection simultaneously and automatically because it nulls out the co-efficient of collinear features. This helps to select the variable(s) out of given n variables while performing lasso regression easier and more accurate.

There is another type of regularization method, which is ElasticNet, this algorithm is a hybrid of lasso and ridge regression both. It is trained using L1 and L2 prior as regularizer. A practical advantage of trading-off between the Lasso and Ridge regression is that it allows Elastic-Net Algorithm to inherit some of Ridge’s stability under rotation.

12. What is VIF? What is the suitable value of a VIF for a feature to be included in a regression modelling?

Ans.: A variance inflation factor(VIF) detects multicollinearity in regression analysis. Multicollinearity is when there’s correlation between predictors (i.e. independent variables) in a model; it’s presence can adversely affect your regression results.

The VIF estimates how much the variance of a regression coefficient is inflated due to multicollinearity in the model.

In VIF method, we pick each feature and regress it against all of the other features. For each regression, the factor is calculated as :

MACHINE LEARNING

$$\mathrm{VIF} = \frac{1}{1 - R_i^2}$$

Where, R-squared is the coefficient of determination in linear regression. Its value lies between 0 and 1.

As we see from the formula, greater the value of R-squared, greater is the VIF. Hence, greater VIF denotes greater correlation. This is in agreement with the fact that a higher R-squared value denotes a stronger collinearity. Generally, a VIF above 5 indicates a high multicollinearity.

Interpreting the Variance Inflation Factor

Variance inflation factors range from 1 upwards. The numerical value for VIF tells you (in decimal form) what percentage the variance (i.e. the standard error squared) is inflated for each coefficient. For example, a VIF of 1.9 tells you that the variance of a particular coefficient is 90% bigger than what you would expect if there was no multicollinearity — if there was no correlation with other predictors.
A **rule of thumb** for interpreting the variance inflation factor:

1 = not correlated.

Between 1 and 5 = moderately correlated.

Greater than 5 = highly correlated.

Exactly how large a VIF has to be before it causes issues is a subject of debate. What is known is that the more your VIF increases, the less reliable your regression results are going to be. In general, a VIF above 10 indicates high correlation and is cause for concern. Some authors suggest a more conservative level of 2.5 or above.

Sometimes a high VIF is no cause for concern at all. For example, you can get a high VIF by including products or powers from other variables in your regression, like x and x2. If you have high VIFs for dummy variables representing nominal variables with three or more categories, those are usually not a problem.

13. Why do we need to scale the data before feeding it to the train the model?

Ans.: Scaling of the data comes under the set of steps of data pre-processing when we are performing machine learning algorithms in the data set.

As we know most of the supervised and unsupervised learning methods make decisions according to the data sets applied to them and often the algorithms calculate the distance between the data points to make better inferences out of the data.

In the machine learning algorithms if the values of the features are closer to each other there are chances for the algorithm to get trained well and faster instead of the data set where the data points or features values have high differences with each other will take more time to understand the data and the accuracy will be lower.

Feature scaling is essential for machine learning algorithms that calculate distances between data. If not scale, the feature with a higher value range starts dominating when calculating distances.

Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step.

To ensure that the gradient descent moves smoothly towards the minima and that the steps for gradient descent are updated at the same rate for all the features, we scale the data before feeding it to the model. Having features on a similar scale will help the gradient descent converge more quickly towards the minima.

14. What are the different metrics which are used to check the goodness of fit in linear regression?

Ans.: There are different metrics that are commonly used for evaluating and reporting the performance of a regression model; they are:

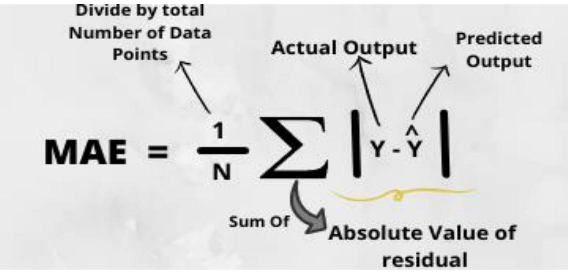
1) Mean Absolute Error (MAE)

MAE is a very simple metric which calculates the absolute difference between actual and predicted values.

To better understand, let's take an example you have input data and output data and use Linear Regression, which draws a best-fit line.

Now you have to find the MAE of your model which is basically a mistake made by the model known as an error. Now find the difference between the actual value and predicted value that is an absolute error but we have to find the mean absolute of the complete dataset.

so, sum all the errors and divide them by a total number of observations And this is MAE. And we aim to get a minimum MAE because this is a loss.


Divide by total Number of Data Points
Actual Output
Predicted Output
MAE = 1/N ∑ | Y - Ŷ |
Sum Of
Absolute Value of residual

Advantages of MAE

1. The MAE you get is in the same unit as the output variable.
2. It is most Robust to outliers.

Disadvantages of MAE

1. The graph of MAE is not differentiable so we have to apply various optimizers like Gradient descent which can be differentiable.

from sklearn.metrics import mean_absolute_error

print("MAE",mean_absolute_error(y_test,y_pred))

Now to overcome the disadvantage of MAE next metric came as MSE.

2) Mean Squared Error(MSE)

MSE is a most used and very simple metric with a little bit of change in mean absolute error. Mean squared error states that finding the squared difference between actual and predicted value.

So, above we are finding the absolute difference and here we are finding the squared difference.

What actually the MSE represents? It represents the squared distance between actual and predicted values. we perform squared to avoid the cancellation of negative terms and it is the benefit of MSE.

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \widehat{y}\right)^2}_{\text{The square of the difference between actual and predicted}}$$

Advantages of MSE

- The graph of MSE is differentiable, so you can easily use it as a loss function.

Disadvantages of MSE

- The value you get after calculating MSE is a squared unit of output. for example, the output variable is in meter(m) then after calculating MSE the output we get is in meter squared.
- If you have outliers in the dataset then it penalizes the outliers most and the calculated MSE is bigger. So, in short, It is not Robust to outliers which were an advantage in MAE.

from sklearn.metrics import mean_squared_error

print("MSE",mean_squared_error(y_test,y_pred))

3) Root Mean Squared Error(RMSE)

As RMSE is clear by the name itself, that it is a simple square root of mean squared error.

$$\mathbf{RMSE} = \sqrt{\mathbf{MSE}}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)}$$

Advantages of RMSE

- The output value you get is in the same unit as the required output variable which makes interpretation of loss easy.

Disadvantages of RMSE

- It is not that robust to outliers as compared to MAE.

for performing RMSE we have to use NumPy square root function over MSE.

print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred)))

Most of the time people use RMSE as an evaluation metric and mostly when you are working with deep learning techniques the most preferred metric is RMSE.

4) Root Mean Squared Log Error(RMSLE)

Taking the log of the RMSE metric slows down the scale of error. The metric is very helpful when you are developing a model without calling the inputs. In that case, the output will vary on a large scale.

To control this situation of RMSE we take the log of calculated RMSE error and resultant we get as RMSLE.

To perform RMSLE we have to use the NumPy log function over RMSE.

print("RMSE",np.log(np.sqrt(mean_squared_error(y_test,y_pred))))

It is a very simple metric that is used by most of the datasets hosted for Machine Learning competitions.

5) R Squared (R2)

R2 score is a metric that tells the performance of your model, not the loss in an absolute sense that how many wells did your model perform.

In contrast, MAE and MSE depend on the context as we have seen whereas the R2 score is independent of context.

So, with help of R squared we have a baseline model to compare a model which none of the other metrics provides. The same we have in classification problems which we call a threshold which is fixed at 0.5. So basically R2 squared calculates how must regression line is better than a mean line.

Hence, R2 squared is also known as Coefficient of Determination or sometimes also known as Goodness of fit.

$$\text{R2 Squared} = 1 - \frac{SSr}{SSm}$$

SSr = Squared sum error of regression line

SSm = Squared sum error of mean line

Now, how will you interpret the R2 score? suppose If the R2 score is zero then the above regression line by mean line is equal means 1 so 1-1 is zero. So, in this case, both lines are overlapping means model performance is worst, It is not capable to take advantage of the output column.

Now the second case is when the R2 score is 1, it means when the division term is zero and it will happen when the regression line does not make any mistake, it is perfect. In the real world, it is not possible.

So we can conclude that as our regression line moves towards perfection, R2 score move towards one. And the model performance improves.

The normal case is when the R2 score is between zero and one like 0.8 which means your model is capable to explain 80 per cent of the variance of data.

from sklearn.metrics import r2_score

r2 = r2_score(y_test,y_pred)

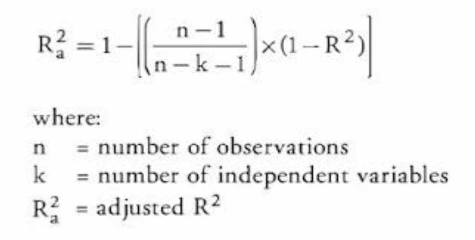
print(r2)

6) Adjusted R Squared

The disadvantage of the R2 score is while adding new features in data the R2 score starts increasing or remains constant but it never decreases because It assumes that while adding more data variance of data increases.

But the problem is when we add an irrelevant feature in the dataset then at that time R2 sometimes starts increasing which is incorrect.

Hence, To control this situation Adjusted R Squared came into existence.



$$R_a^2 = 1 - \left[\left(\frac{n-1}{n-k-1}\right)\times(1-R^2)\right]$$

where:
n = number of observations
k = number of independent variables
R_a^2 = adjusted R^2

Now as K increases by adding some features so the denominator will decrease, n-1 will remain constant. R2 score will remain constant or will increase slightly so the complete answer will increase and when we subtract this from one then the resultant score will decrease. so this is the case when we add an irrelevant feature in the dataset.

And if we add a relevant feature then the R2 score will increase and 1-R2 will decrease heavily and the denominator will also decrease so the complete term decreases, and on subtracting from one the score increases.

n=40

k=2

adj_r2_score = 1 - ((1-r2)(n-1)/(n-k-1))*

print(adj_r2_score)

Hence, this metric becomes one of the most important metrics to use during the evaluation of the model.

15. From the following confusion matrix calculate sensitivity, specificity, precision, recall and accuracy.

Actual/Predicted	True	False
True	1000 **True Positive (TP)**	50 **False Positive (FP)** **Type 1 error**
False	250 **False Negative (FN)** **Type 2 error**	1200 **True Negative (TN)**

Ans.: The total outcome values are: TP = 1000, TN = 1200, FP = 50, FN = 250

1. **Accuracy (all correct / all) = TP + TN / TP + TN + FP + FN**

   = 1000+1200/1000+1200+50+250

   = 0.88 (i.e. 88%)

2. **Misclassification (all incorrect / all) = FP + FN / TP + TN + FP + FN**

   = 50+250 / 1000+1200+50+250

   = 0.12 (i.e. 12%)

3. **Precision (true positives / predicted positives) = TP / TP + FP**

   = 1000 / 1000 + 50 = 0.95

4. **Sensitivity aka Recall (true positives / all actual positives) = TP / TP + FN**

MACHINE LEARNING

= 1000/1000+250

= 0.80

5. **Specificity (true negatives / all actual negatives) =TN / TN + FP**

= 1200/1200+50

= 0.96
