MACHINE LEARNING

ASSIGNMENT – 7

1. Which of the following in sk-learn library is used for hyper parameter tuning?

A**) GridSearchCV ()** B) RandomizedCV ()

C) K-fold Cross Validation D) All of the above

Ans :A **) GridSearchCV ()**

2. In which of the below ensemble techniques trees are trained in parallel?

A**) Random forest** B) Adaboost

C) Gradient Boosting D) All of the above

Ans: A**) Random forest**

3. In machine learning, if in the below line of code:

sklearn.svm.SVC (C=1.0, kernel='rbf', degree=3)

we increasing the C hyper parameter, what will happen?

A**) The regularization will increase** B) The regularization will decrease

C) No effect on regularization D) kernel will be changed to linear

Ans: A **) The regularization will increase**

4. Check the below line of code and answer the following questions:

sklearn.tree.DecisionTreeClassifier(*criterion='gini',splitter='best',max_depth=None,

min_samples_split=2)

Which of the following is true regarding max_depth hyper parameter?

A) It regularizes the decision tree by limiting the maximum depth up to which a tree can be grown.

B) It denotes the number of children a node can have.

C) both A & B

D) None of the above

Ans: **A) It regularizes the decision tree by limiting the maximum depth up to which a tree can be grown.**

5. Which of the following is true regarding Random Forests?

A) It's an ensemble of weak learners.

B) The component trees are trained in series

C) In case of classification problem, the prediction is made by taking mode of the class labels predicted by the component trees.

D)None of the above

Ans: **B) The component trees are trained in series**

6. What can be the disadvantage if the learning rate is very high in gradient descent?

A) Gradient Descent algorithm can diverge from the optimal solution.

B) Gradient Descent algorithm can keep oscillating around the optimal solution and may not settle.

C) Both of them

D) None of them

Ans: **A) Gradient Descent algorithm can diverge from the optimal solution.**

7. As the model complexity increases, what will happen?

A) Bias will increase, Variance decrease B) Bias will decrease, Variance increase

C)both bias and variance increase D) Both bias and variance decrease.

Ans: **A) Bias will increase, Variance decrease**

8. Suppose I have a linear regression model which is performing as follows:

Train accuracy=0.95 and Test accuracy=0.75

Which of the following is true regarding the model?

A) model is underfitting **B) model is overfitting**

C) model is performing good D) None of the above

Ans: **B) model is overfitting**

Q9 to Q15 are subjective answer type questions, Answer them briefly.

9. Suppose we have a dataset which have two classes A and B. The percentage of class A is 40% and percentage of class B is 60%. Calculate the Gini index and entropy of the dataset.

According to Gini (2005), the Gini index can be calculated as the ratio of the area between the perfect equality line and the Lorenz curve (A) divided by the total area under the perfect equality line (A + B)

Answer: The Gini index for the overall examples is 1 − (5/10) 2 -(5/10)2 = 0.5. ... The error rate using attribute X is (60 + 40)/200 = 0.5; the error rate

10. What are the advantages of Random Forests over Decision Tree?

• Does not suffer from overfitting problem.

• Can use for both classification and regression problems.

• Gives highly accurate predictions.

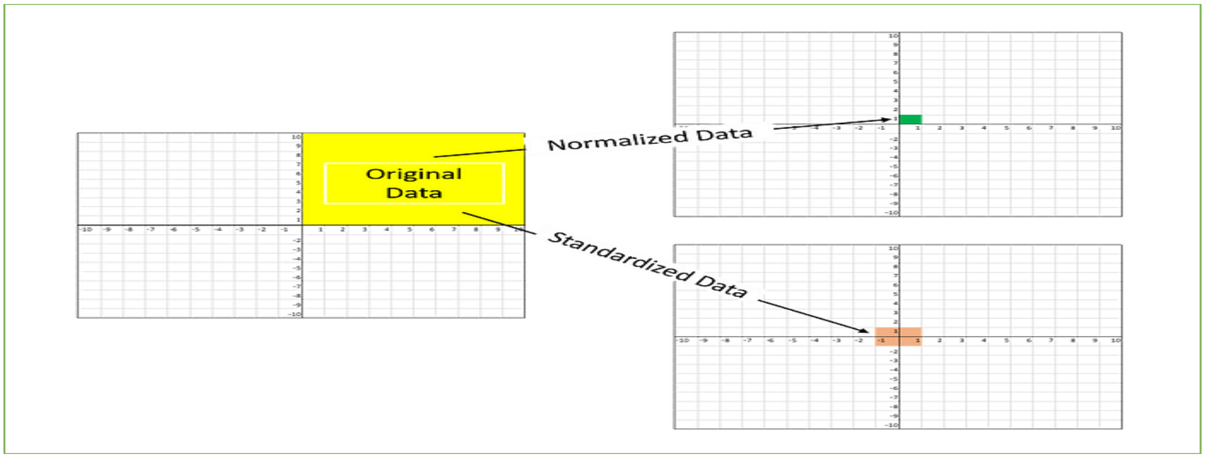
• Powerful than other non-linear models.

11. What is the need of scaling all numerical features in a dataset? Name any two techniques used for scaling.

Feature scaling in machine learning is one of the most critical steps during the pre-processing of data before creating a machine learning model. Scaling can make a difference between a weak machine learning model and a better one.

The most common techniques of feature scaling are Normalization and Standardization.

Normalization is used when we want to bound our values between two numbers, typically, between [0,1] or [-1,1]. While Standardization transforms the data to have zero mean and a variance of 1, they make our data **unitless**. Refer to the below diagram, which shows how data looks after scaling in the X-Y plane.

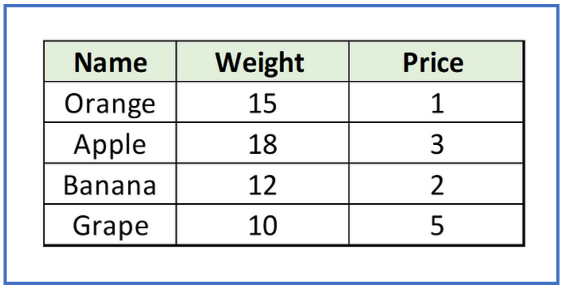


Why the need of scaling because

Machine learning algorithm just sees number — if there is a vast difference in the range say few ranging in thousands and few ranging in the tens, and it makes the underlying assumption that higher ranging numbers have superiority of some sort. So these more significant number starts playing a more decisive role while training the model.

The machine learning algorithm works on numbers and does not know what that number represents. A weight of 10 grams and a price of 10 dollars represents completely two different things — which is a no brainer for humans, but for a model as a feature, it treats both as same.

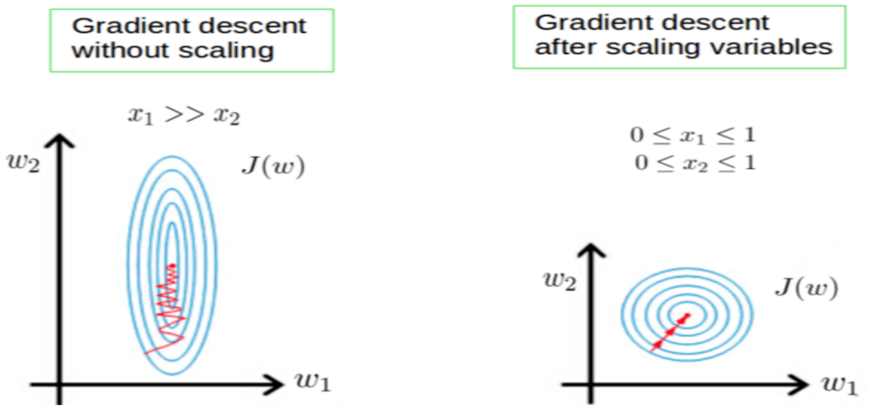
Suppose we have two features of weight and price, as in the below table. The “Weight” cannot have a meaningful comparison with the “Price.” So the assumption algorithm makes that since “Weight” > “Price,” thus “Weight,” is more important than “Price.”



Name	Weight	Price
Orange	15	1
Apple	18	3
Banana	12	2
Grape	10	5

So these more significant number starts playing a more decisive role while training the model. Thus feature scaling is needed to bring every feature in the same footing without any upfront importance. Interestingly, if we convert the weight to “Kg,” then “Price” becomes dominant.

Another reason why feature scaling is applied is that few algorithms like Neural network gradient descent **converge much faster** with feature scaling than without it.



One more reason is **saturation**, like in the case of sigmoid activation in Neural Network, scaling would help not to saturate too fast.

Standardization and Normalization are the 2 techniques we can use for scaling.

12. Write down some advantages which scaling provides in optimization using gradient descent

algorithm.

Optimization Gradient descent is a simple optimization procedure that you can use with many machine learning algorithms.

The main advantages:

- We can use fixed learning rate during training without worrying about learning rate decay.
- It has straight trajectory towards the minimum and it is guaranteed to converge in theory to the global minimum if the loss function is convex and to a local minimum if the loss function is not convex.
- Gradient descent is an optimization algorithm which is commonly-used to train machine learning models and neural networks. Training data helps these models learn over time, and the cost function within gradient descent specifically acts as a barometer, gauging its accuracy with each iteration of parameter updates.
- More stable convergence and error gradient than Stochastic Gradient descent
- Embraces the benefits of vectorization
- A more direct path is taken towards the minimum
- Computationally efficient since updates are required after the run of an epoch

13. In case of a highly imbalanced dataset for a classification problem, is accuracy a good metric to measure the performance of the model. If not, why?

Accuracy is not a good metric for imbalanced datasets.

This model would receive a very good accuracy score as it predicted correctly for the majority of observations, but this hides the true performance of the model which is objectively not good as it only predicts for one class. the most common metrics to use for imbalanced datasets are:

Marco F1 score

AUC score (AUC ROC)

Average precision score (AP)

G-Mean

The common factor for all of these metrics is that they take into account the model performance for each class, instead of looking at the summarised performance.

Accuracy is not a good metric for imbalanced datasets.

14. What is “f-score" metric? Write its mathematical formula.

The F-score, also called the F1-score, is a measure of a model's accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into 'positive' or 'negative'.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

An F-score is the harmonic mean of a system's precision and recall values. It can be calculated by the following formula: 2 x [(Precision x Recall) / (Precision + Recall)].

A good F1 score means that you have low false positives and low false negatives, so you're correctly identifying real threats and you are not disturbed by false alarms.

An F1 score is considered perfect when it's 1 , while the model is a total failure when it's 0 .

Applications of F-score

Classification tasks. The F-score is often used to evaluate the performance of a classifier, particularly when the goal is to balance precision and recall.

Information retrieval tasks. The F-score can be used to evaluate the performance of a search engine or other information retrieval system. Precision quantifies how well-retrieved documents match the query, whereas recall evaluates how many appropriate results were found.

Hyperparameter optimization. The F-score can be used as a performance metric when optimizing the hyperparameters of a Machine Learning model. This can be useful in finding the best set of hyperparameters for a given task.

Model comparison. The F-score can be used to compare the performance of different Machine Learning models on the same task. This can be useful when choosing the best model for a particular application.

It is worth noting that the F-score is just one metric that can be used to evaluate the performance of a Machine Learning model. Other common metrics include accuracy, Area Under the Curve (AUC), and log loss. The appropriate metric will depend on the specifics of the task and the goals of the model.

15. What is the difference between fit(), transform() and fit_transform()?

Estimators:- fit(): –

It is used for calculating the initial parameters on the training data and later saves them as internal objects state.

– This method calculates the parameters μ(mean) and σ(standard deviation) and saves them as internal objects.

– A black box which only does the computation and gives nothing.

Transformers:- transform()

– Use the initial above calculated values and return modified training data as output.

– Using these same parameters, using this method we can transform a particular dataset.

– Used for pre-processing before modeling.

fit_transform()

– It is a conglomerate above two steps. Internally, it first calls fit() and then transform() on the same data.

– It joins the fit() and transform() method for the transformation of the dataset.

– It is used on the training data so that we can scale the training data and also learn the scaling parameters. Here, the model built will learn the mean and variance of the features of the training set. These learned parameters are then further used to scale our test data.

Predictors: fit(): – It calculates the parameters or weights on the training data (e.g. parameters returned by coef() in case of Linear Regression) and saves them as an internal object state.

predict(): – Use the above-calculated weights on the test data to make the predictions.

Difference between fit(), transform(), and fit_transform() methods in scikit-learn

Let's try to understand the difference with a given example:

Suppose you have an array arr = [1,2,3,4,5] and you have a sklearn class FillMyArray that filled your array.

When you declare an instance of your class:

my_filler = FillMyArray()

We have the in hand methods fit(), transform() and fit_transform().

fit(): my_filler.fit(arr) will compute the value to assign to x to fill out the array and store it in our instance my_filler.

transform(): After the value is computed and stored during the previous .fit() stage we can call my_filler.transform(arr) which will return the filled array [1,2,3,4,5].

fit_transform(): If we perform my_filler.fit_transform(arr) we can skip one line of code and have the value calculated along with assigned to the filled array that is directly returned in only one stage.

However, for the testing set, Machine learning applies predictions based on the learning during the training set, due to which it doesn't need to perform calculations and perform just the transformation.

If we perform the fit() method even on test data, we will compute a new mean and variance that will be a Naive scale for each feature and will allow the model to learn on the test data too. However, we will no longer be able to keep it as a surprise to our model and it wouldn't be able to give us a good estimate on model performance on the unseen data, which is certainly our ultimate aim.

It is the general procedure to scale the data when building a machine learning model. So that the model is not biased to a specific feature and prevents our model to learn the trends of our test data at the same time.