# MACHINE LEARNING – WORKSHEET 4

## In Q1 to Q8, only one option is correct, Choose the correct option:

1. Which of the following in sklearn library is used for hyper parameter tuning?
   1. GridSearchCV() B) RandomizedCV()

C) K-fold Cross Validation D) None of the above

1. In which of the below ensemble techniques trees are trained in parallel?
   1. Random forest B) Adaboost

C) Gradient Boosting D) All of the above

1. 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?

* 1. The regularization will increase B) The regularization will decrease

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

1. 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?

* 1. It regularizes the decision tree by limiting the maximum depth up to which a tree can be grown.
  2. It denotes the number of children a node can have.
  3. both A & B
  4. None of the above

1. Which of the following is true regarding Random Forests?
   1. It's an ensemble of weak learners.
   2. The component trees are trained in series
   3. In case of classification problem, the prediction is made by taking mode of the class labels predicted by the component trees.
   4. None of the above
2. What can be the disadvantage if the learning rate is very high in gradient descent?
   1. Gradient Descent algorithm can diverge from the optimal solution.
   2. Gradient Descent algorithm can keep oscillating around the optimal solution and may not settle.
   3. Both of them D)None of them.
3. As the model complexity increases, what will happen?
   1. Bias will increase, Variance decrease B) Bias will decrease, Variance increase C)both bias and variance increase D) Both bias and variance decrease.
4. Suppose I have a linear regression model which is performing as follows: Train accuracy=0.95

Test accuracy=0.75

Which of the following is true regarding the model?

* 1. model is underfitting B) model is overfitting

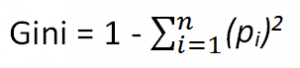
C) model is performing good D) None of the above

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

1. 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.

Ans: Gini index or Gini impurity measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen. But what is actually meant by ‘impurity’? If all the elements belong to a single class, then it can be called pure. The degree of Gini index varies between 0 and 1, where 0 denotes that all elements belong to a certain class or if there exists only one class, and 1 denotes that the elements are randomly distributed across various classes. A Gini Index of 0.5 denotes equally distributed elements into some classes.

**Formula for Gini Index**



In our dataset we have one class 60% and second 40% so P(One class): 6/10 ,P(Second class): 4/10 ,

**So Gini Index= 1 - ((6/10)^2 + (4/10)^2) = 1-0.52= 0.48**

We can think about the entropy of a dataset in terms of the probability distribution of observations in the dataset belonging to one class or another, e.g. two classes in the case of a binary classification dataset.For example, in a binary classification problem (two classes), we can calculate the entropy of the data sample as follows:

Entropy = -(p(0) \* log(P(0)) + p(1) \* log(P(1)))

**So entropy is -(p(6/10) \* log(P(6/10)) + p(4/10) \* log(P(4/10))) = 0.97**

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

Ans: Random forests are a strong modeling technique and much more robust than a single decision tree. They aggregate many decision trees to limit overfitting as well as error due to bias and therefore yield useful results.

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

Ans: 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. Since the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without normalization. For example, many classifiers calculate the distance between two points by the Euclidean distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance.

Two most used scaling techniques are StandardScaler & MinMaxScaler:

StandardScaler follows Standard Normal Distribution (SND). Therefore, it makes mean = 0 and scales the data to unit variance.

MinMaxScaler scales all the data features in the range [0, 1] or else in the range [-1, 1] if there are negative values in the dataset. This scaling compresses all the inliers in the narrow range [0, 0.005].

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

Real-world data can come up in different orders of magnitude. For example, your age ranges from 0 to 100 years, while your yearly income from €10,000 to €10,000,000 (and more). Using such unprocessed data as input features for a linear regression system might slow down the gradient descent algorithm to a crawl.

It happens because — as we will see shortly — such not normalized data warps the cost function the gradient descent has to process, making the minimum point really difficult to reach.

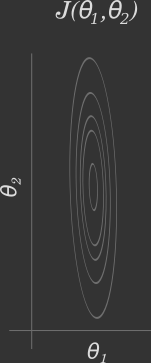
Because of that, an important trick in machine learning and in linear regression is to make sure that all the input features are on a similar scale. This is a preparatory step you do in order to optimize the input data, known as **feature scaling**.

## Feature scaling

In feature scaling you basically normalize your input values. For example, say you have two features:

* x1x1 as the yearly income (10,000-10,000,000);
* x2x2 as the age (0-100).

Below you will find a contour plot for the cost function J(θ1,θ2)J(θ1,θ2) as if we were using the raw, unprocessed values. As you may see the result is a very thin and stretched version of it. The gradient descent algorithm would oscillate a lot back and forth, taking a long time before finding its way to the minimum point.



*1. A stretched contour plot, due to missing input feature scaling.*

With feature scaling we will bring back the original bowl-shaped figure in order to let the gradient descent algorithm do its job efficiently. You have to options here: **min-max scaling** or **standardization**.

### Min-max scaling

The idea is to get every input feature into approximately a [−1,1][−1,1] range. The name comes from the use of minmin and maxmax functions, namely the smallest and greatest values in your dataset. It requires dividing the input values by the range (i.e. the maximum value minus the minimum value) of the input variable:

x′i=xi−min(xi)max(xi)−min(xi)xi′=xi−min(xi)max(xi)−min(xi)

where xixi is the original ii-th input value, x′ixi′ is the normalized version.

For example, say I'm dealing with the yearly income x1x1 and in particular I want to normalize the value of $30,000:

x′1=30,000−10,00010,000,000−10,000≈0.002x1′=30,000−10,00010,000,000−10,000≈0.002

Just rinse and repeat such normalization for every value in your dataset. Of course if you are in a multivariate scenario remember to skip feature x0x0, since x0=1x0=1 as seen in the [previous episode](https://www.internalpointers.com/post/multivariate-linear-regression).

1. 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?

Ans: When we use accuracy, we assign equal cost to false positives and false negatives. When that data set is imbalanced - say it has 99% of instances in one class and only 1 % in the other - there is a great way to lower the cost. Predict that every instance belongs to the majority class, get accuracy of 99% and go home early.

The problem starts when the actual costs that we assign to every error are not equal. If we deal with a rare but fatal disease, the cost of failing to diagnose the disease of a sick person is much higher than the cost of sending a healthy person to more tests.

In general, there is no general best measure. The best measure is derived from your needs. In a sense, it is not a machine learning question, but a business question. It is common that two people will use the same data set but will choose different metrics due to different goals.

Accuracy is a great metric. Actually, most metrics are great and I like to evaluate many metrics. However, at some point you will need to decide between using model A or B. There you should use a single metric that best fits your need.

For extra credit, choose this metric before the analysis, so you won't be distracted when making the decision.

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

Ans: 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’.The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model’s precision and recall.The F-score is commonly used for evaluating information retrieval systems such as search engines, and also for many kinds of machine learning models, in particular in natural language processing.It is possible to adjust the F-score to give more importance to precision over recall, or vice-versa. Common adjusted F-scores are the F0.5-score and the F2-score, as well as the standard F1-score.

The formula for the standard F1-score is the harmonic mean of the precision and recall. A perfect model has an F-score of 1:

**F**-**Measure** = (2 \* Precision \* Recall) / (Precision + Recall)

1. What is the difference between fit(), transform() and fit\_transform()?

Ans: 1.Fit(): Method calculates the parameters μ and σ and saves them as internal objects.

2.Transform(): Method using these same parameters apply the transformation to a particular dataset.

3.Fit\_transform(): joins the fit() and transform() method for transformation of dataset.

For a given data set there will be only one mean and one standard deviation, which can be represented by training data set.

**μ-mean**

**σ-standard deviation**

therefore, fit\_transform(X\_train) calculated μ and σ on training data set. The same factors will be used by the transform() function.

