ASSIGNMENT – 7

# MACHINE LEARNING

1.Which of the following in sk-learn library is used for hyper parameter tuning?
Ans:
A) GridSearchCV()

2.In which of the below ensemble techniques trees are trained in parallel?
Ans:
D) All of the above

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

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

C) Both of them

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

B) model is overfitting

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.

$$Gini = 1 - \sum_{i=1}^{j} P(i)^2$$

$$Gini_{maths} = 1 - \left(\frac{4}{7}\right)^2 - \left(\frac{3}{7}\right)^2 = .4897$$

10. What are the advantages of Random Forests over Decision Tree?
Ans:
Random forest is an ensemble of many decision trees. Random forests are built using a method called **bagging** in which each decision trees are used as parallel estimators. If used for a classification problem, the result is based on majority vote of the results received from each decision tree. For regression, the prediction of a leaf node is the mean value of the target values in that leaf. Random forest regression takes mean value of the results from decision trees.

Random forests reduce the risk of overfitting and accuracy is much higher than a single decision tree. Furthermore, decision trees in a random forest run in parallel so that the time does not become a bottleneck.

The success of a random forest highly depends on using uncorrelated decision trees. If we use same or very similar trees, overall result will not be much different than the result of a single decision tree. Random forests achieve to have uncorrelated decision trees by **bootstrapping** and **feature randomness.**

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

When approaching almost any ***unsupervised learning*** problem (any problem where we are looking to cluster or segment our data points), **feature scaling is a fundamental step** in order to asure we get the expected results.
Forgetting to use a feature scaling technique before any kind of model like K-means **or** DBSCAN, can be fatal and completely bias or invalidate our results.

The main feature scaling techniques are Standardisation and Normalisation

***Normalisation-*** scales our features to a predefined range (normally the 0–1 range), independently of the statistical distribution they follow. It does this **using the minimum and maximum values** of each feature in our data set, which makes it a bit sensitive to outliers

***Standardisation-*** takes our data and makes it follow a **Normal distribution**, usually of mean 0 and standard deviation 1. It is best to use it when we know our data follows an standard distribution or if we know there are many outliers.

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

Ans:
Every time we train a deep learning model, or any neural network for that matter, we're using **gradient descent** (with backpropagation). We use it to *minimize a loss* by *updating the parameters/weights* of the model.
The parameter update depends on two values: a gradient and a learning rate. The learning rate gives you control of how big (or small) the updates are going to be. A bigger learning rate means bigger updates and, hopefully, a model that learns faster.

define a model and generate a synthetic dataset

randomly initialize the parameters

explore the loss surface and visualize the gradients understand the effects of using different learning rates

understand the effects of feature scaling

y= b+ wx+ c

In this model, we use a **feature** (***x***) to try to predict the value of a **label** (***y***). There are three elements in our model

**parameter *b***, the *bias* (or *intercept*), which tells us the expected average value of $y$ when $x$ is zero

**parameter *w***, the *weight* (or *slope*), which tells us how much $y$ increases, on average, if we increase $x$ by one unit

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?
Ans:

An imbalanced dataset is a kind of data distribution where one or more classes have the most number of samples belonging to their class than the other classes. The class distribution in an imbalanced dataset is not uniform among the classes. To simplify let's consider a binary classification problem where we need to classify if a particular sample is a Cat or not. Say, the dataset we have for this problem is an imbalanced one where only 100 samples out of 1000 samples belongs to the positive class meaning it's a Cat. The rest of the 900 samples are negative classes meaning they are not Cats. Now let's train a machine learning model for this dataset.

An imbalanced classification problem is an example of a classification problem where the distribution of examples across the known classes is biased or skewed. The distribution can vary from a slight bias to a severe imbalance where there is one example in the minority class .

An imbalanced classification problem is an example of a classification problem where the distribution of examples across the known classes is biased or skewed. The distribution can vary

from a slight bias to a severe imbalance where there is one example in the minority class for hundreds, thousands, or millions of examples in the majority class or classes.

Imbalanced classification is the problem of classification when there is an unequal distribution of classes in the training dataset.

The imbalance in the class distribution may vary, but a severe imbalance is more challenging to model and may require specialized techniques.

Many real-world classification problems have an imbalanced class distribution, such as fraud detection, spam detection, and churn prediction.

14. 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 kind of machine learnings 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.

$$\begin{aligned} F_1 &= \frac{2}{\frac{1}{\mathrm{recall}} \times \frac{1}{\mathrm{precision}}} = 2 \times \frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \\ &= \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})} \end{aligned}$$

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

Fit- means to fit the pre-processor to the data being provided. This is where the pre-processor "learns" from the data.

Transform-it  means to transform the data (produce outputs) according to the fitted pre-processor; it is normally used on the *test* data, and unseen data in general (e.g. in new data that come after deploying a model).

fit transform- means to do both - Fit the pre-processor to the data, then transform the data according to the fitted pre-processor. Calling fit_transform is a convenience to avoid needing to call fit and transform sequentially on the same input, but of course this is only applicable to the *training* data (calling again fit_transform in test or unseen data is unfortunately a common rookie mistake).

Ans: