MACHINE LEARNING – WORKSHEET 3

Q1 to Q15 are subjective answer type questions, Answer them briefly.

1. **Give short description each of Linear, RBF, Polynomial kernels used in SVM.**

SVM algorithms use a set of mathematical functions that are defined as the kernel.The kernels are mapping function to map the data from one space to a new space which is simpler for SVM to dealt with.The function of kernel is to take data as input and transform it into the required form. The kernel functions return the inner product between two points in a suitable feature space. Thus by defining a notion of similarity, with little computational cost even in very high-dimensional spaces. Different SVM algorithms use different types of kernel functions. These functions can be different types. For example linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.The linear, polynomial and RBF or Gaussian kernel are simply different in case of making the hyperplane decision boundary between the classes. Usually linear and polynomial kernels are less time consuming and provides less accuracy than the rbf or Gaussian kernels.

1. **R-squared or Residual Sum of Squares (RSS) which one of these two is a better measure of goodness of fit of model in regression and why??**

Ans: Regression is a measurement that helps determine the strength of the relationship between a dependent variable and a series of other changing variables or independent variables.

R2 represents the proportion of the variance in your data which is explained by your model; the closer to one, the better the fit.

The residual sum of squares (RSS) is the sum of the squared distances between your actual versus your predicted values. A residual sum of squares (RSS) is a statistical technique used to measure the amount of variance in a data set that is not explained by a regression model.

R-squared will give you an estimate of the relationship between movements of a dependent variable based on an independent variable's movements. It doesn't tell you whether your chosen model is good or bad, nor will it tell you whether the data and predictions are biased. A high or low R-square isn't necessarily good or bad, as it doesn't convey the reliability of the model, nor whether you've chosen the right regression. You can get a low R-squared for a good model, or a high R-square for a poorly fitted model, and vice versa but RSS gives you a clear idea about how good or bad your model is.So RSS is a better measure of goodness of fit of model in regression

1. **What are TSS (Total Sum of Squares), ESS (Explained Sum of Squares) and RSS (Residual Sum of Squares) in regression. Also mention the equation relating these three metrics with each other.**

Ans:

* **Explained sum of squares (ESS):** Also known as the *explained variation*, the ESS is the portion of total variation that measures how well the regression equation explains the relationship between *X* and *Y*.

You compute the ESS with the formula

image2.png

* **Residual sum of squares (RSS):** This expression is also known as *unexplained variation* and is the portion of total variation that measures discrepancies (errors) between the actual values of *Y* and those estimated by the regression equation.

You compute the RSS with the formula

image3.png

The smaller the value of RSS relative to ESS, the better the regression line fits or explains the relationship between the dependent and independent variable.

* **Total sum of squares (TSS):**

The sum of RSS and ESS equals TSS.

image4.png

*R*2 is the ratio of explained sum of squares (ESS) to total sum of squares (TSS):

image5.png

You can also use this formula:

image6.png

Based on the definition of *R*2, its value can never be negative. Also, *R*2 can’t be greater than 1, so

image7.png

1. **What is Gini –impurity index?**

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.

1. **Are unregularized decision-trees prone to overfitting? If yes, why?**

Ans: Over-fitting is the phenomenon in which the learning system tightly fits the given training data so much that it would be inaccurate in predicting the outcomes of the untrained data.

In decision trees, over-fitting occurs when the tree is designed so as to perfectly fit all samples in the training data set. Thus it ends up with branches with strict rules of sparse data. Thus this effects the accuracy when predicting samples that are not part of the training set.

One of the methods used to address over-fitting in decision tree is called pruning which is done after the initial training is complete**. In decision trees regularization can be done by pruning the tree.** If left to its own device the tree can continue to fit till each data point is a different leaf in the tree. This obviously will not generalize well so you have to put in different criteria to stop splitting the nodes beyond a point. This can be done by specifying how many minimum data points are needed at each node for splitting In pruning, you trim off the branches of the tree, i.e., remove the decision nodes starting from the leaf node such that the overall accuracy is not disturbed. This is done by segregating the actual training set into two sets: training data set, D and validation data set, V. Prepare the decision tree using the segregated training data set, D. Then continue trimming the tree accordingly to optimize the accuracy of the validation data set, V.

1. **What is an ensemble technique in machine learning?**

Ans: One of the major tasks of machine learning algorithms is to construct a fair model from a dataset. The process of generating models from data is called learning or training and the learned model can be called as hypothesis or learner. The learning algorithms which construct a set of classifiers and then classify new data points by taking a choice of their predictions are known as Ensemble methods.

It has been discovered that ensembles are often much more accurate than the individual classifiers which make them up. The ensemble methods, also known as committee-based learning or learning multiple classifier systems train multiple hypotheses to solve the same problem. One of the most common examples of ensemble modelling is the random forest trees where a number of decision trees are used to predict outcomes.

An ensemble contains a number of hypothesis or learners which are usually generated from training data with the help of a base learning algorithm. Most ensemble methods use a single base learning algorithm to produce homogenous base learners or homogenous ensembles and there are also some other methods which use multiple learning algorithms and thus produce heterogenous ensembles. Ensemble methods are well known for their ability to boost weak learners.

1. **What is the difference between Bagging and Boosting techniques?**

Ans: Bagging and Boosting are both ensemble techniques, where a set of weak learners are combined to create a strong learner that obtains better performance than a single one.

Bagging tries to implement similar learners on small sample population and then takes a mean of all the predictions. In generalised bagging, we can use different learner on different population as this can help in reducing variance error.

Boosting is an iterative technique which adjusts the weight of an observation based on the last classification. If an observation was classified incorrectly, it adjust the weight of this observation. Boosting in general decreases the bias error and build strong predictive models, however they may overfit on training data.

1. **what is out-of-bag error in random forests?**

Ans: Random forests technique involves sampling of the input data with replacement (bootstrap sampling). In this sampling, about one third of the data is not used for training and can be used to testing. These are called the out of bag samples. Error estimated on these out of bag samples is the out of bag error.

1. **What is K-fold cross-validation?**

Ans: Generally we perform training on the 70% of the given data-set and rest 30% is used for the testing purpose. The major drawback of this method is that we perform training on the 50% of the dataset, it may possible that the remaining 50% of the data contains some important information which we are leaving while training our model i.e higher bias.

This drawback can be overcome by cross validation technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set, by doing this we train our model with all the data in subsets

In **K-Fold Cross Validation**, we split the data-set into k number of subsets(known as folds) then we perform training on the all the subsets but leave one(k-1) subset for the evaluation of the trained model. In this method, we iterate k times with a different subset reserved for testing purpose each time.

1. **What is hyper parameter tuning in machine learning and why it is done?**

Ans: A Machine Learning model is defined as a mathematical model with a number of parameters that need to be learned from the data. By training a model with existing data, we are able to fit the model parameters.

However, there is another kind of parameters, known as Hyperparameters, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn.

Some examples of model hyperparameters include:

The penalty in Logistic Regression Classifier i.e. L1 or L2 regularization

The learning rate for training a neural network.

The C and sigma hyperparameters for support vector machines.

The k in k-nearest neighbors.

Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. Two best strategies for Hyperparameter tuning are:

**GridSearchCV,RandomizedSearchCV**

1. **What issues can occur if we have a large learning rate in Gradient Descent?**

Ans: Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. In machine learning, we use gradient descent to update the parameters of our model. Parameters refer to coefficients in Linear Regression and weights in neural networks.

Let's understand with an example of mountain adjoining a sea surface

Starting at the top of the mountain, we take our first step downhill in the direction specified by the negative gradient. Next we recalculate the negative gradient (passing in the coordinates of our new point) and take another step in the direction it specifies. We continue this process iteratively until we get to the bottom of our graph, or to a point where we can no longer move downhill–a local minimum

The size of these steps is called the learning rate. With a high learning rate we can cover more ground each step, but we risk overshooting the lowest point since the slope of the hill is constantly changing. With a very low learning rate, we can confidently move in the direction of the negative gradient since we are recalculating it so frequently. A low learning rate is more precise, but calculating the gradient is time-consuming, so it will take us a very long time to get to the bottom.

1. **What is bias-variance trade off in machine learning?**

Ans: We need to understand what variance and bias are in a machine learning model

**Bias:** Bias is basically how far we have predicted the value from the actual value. We say the bias is too high if the average predictions are far off from the actual values. A high bias will cause the algorithm to miss a dominant pattern or relationship between the input and output variables. When the bias is too high, it is assumed that the model is quite simple and does not fathom the complexity of the data set to determine the relationship and thus, causing underfitting.

**Variance:** On an independent, unseen data set or a validation set. When a model does not perform as well as it does with the trained data set, there is a possibility that the model has a variance. It basically tells how scattered the predicted values are from the actual values. A high variance in a data set means that the model has trained with a lot of noise and irrelevant data. Thus causing overfitting in the model. When a model has high variance, it becomes very flexible and makes wrong predictions for new data points. Because it has tuned itself to the data points of the training set.

Let us also try to understand the concept of bias-variance mathematically. Let the variable that we are predicting to be Y and the other independent variables to be X. Now let us assume there is a relationship between the two variables such that: Y = f(X) + e

In the above equation, Here e is the estimated error with a mean value 0. When we make a classifier using algorithms like linear regression, SVM, etc, the expected squared error at point x will be:

**err(x) = Bias2 + Variance + irreducible error**

We can put the relationship between bias-variance in four categories listed below:

**High Variance-High Bias – The model is inconsistent and also inaccurate on average**

**Low Variance-High Bias – Models are consistent but low on average**

**High Variance-Low Bias – Somewhat accurate but inconsistent on averages**

**Low Variance-Low Bias – It is the ideal scenario, the model is consistent and accurate on average.**

Finding the right balance between the bias and variance of the model is called the **Bias-Variance trade-off**. It is basically a way to make sure the model is neither overfitted or underfitted in any case.If the model is too simple and has very few parameters, it will suffer from high bias and low variance. On the other hand, if the model has a large number of parameters, it will have high variance and low bias. This trade-off should result in a perfectly balanced relationship between the two. **Ideally, low bias and low variance is the target for any Machine Learning model.**

1. **What is the need of regularization in machine learning?**

**Ans**: Regularisation is a technique used to reduce the errors by fitting the function appropriately on the given training set and avoid overfitting.

The commonly used regularisation techniques are :

L1 regularisation, L2 regularisation, Dropout regularisation

A regression model which uses **L1 Regularisation**technique is called **LASSO(Least Absolute Shrinkage and Selection Operator)** regression.  
A regression model that uses **L2 regularisation** technique is called **Ridge regression**.  
**Lasso Regression** adds “absolute value of magnitude” of coefficient as penalty term to the loss function(L).

https://media.geeksforgeeks.org/wp-content/uploads/20200705212608/L1regularisation1-300x23.png

**Ridge regression** adds “squared magnitude” of coefficient as penalty term to the loss function(L).

https://media.geeksforgeeks.org/wp-content/uploads/20200705212953/L2regularisation-e1593964838310-300x29.png

**NOTE** that during Regularisation the output function(y\_hat) does not change. The change is only in the loss function.

The output function:  
https://media.geeksforgeeks.org/wp-content/uploads/20200705215755/outputy_hat1-300x19.png

1. **Differentiate between Adaboost and Gradient Boosting**

**Ans**: The basic idea of boosting (an ensemble learning technique) is to combine several weak learners into a stronger one. The general idea of boosting algorithms is to try predictors sequentially, where each subsequent model attempts to fix the errors of its predecessor.

Adaboost is more about ‘voting weights’ and Gradient boosting is more about ‘adding gradient optimization’.

AdaBoost stands for Adaptive Boosting. Adaboost increases the accuracy by giving more weightage to the target which is misclassified by the model. At each iteration, Adaptive boosting algorithm changes the sample distribution by modifying the weights attached to each of the instances. It increases the weights of the wrongly predicted instances and decreases the ones of the correctly predicted instances.

Gradient boosting calculates the gradient (derivative) of the Loss Function with respect to the prediction (instead of the features). Gradient boosting increases the accuracy by minimizing the Loss Function (error which is difference of actual and predicted value) and having this loss as target for the next iteration.

Gradient boosting algorithm builds first weak learner and calculates the Loss Function. It then builds a second learner to predict the loss after the first step. The step continues for third learner and then for fourth learner and so on until a certain threshold is reached.

1. **Can we use Logistic Regression for classification of Non-Linear Data? If not, why?**

Logistic regression is considered a generalized linear model because the outcome always depends on the sum of the inputs and parameters in other words, the output cannot depend on the product (or quotient, etc.) of its parameters. Logistic regression is an algorithm that learns a model for binary classification. A nice side-effect is that it gives us the probability that a sample belongs to class 1 or vice versa: class 0. Logistic Regression has traditionally been used as a linear classifier, i.e. when the classes can be separated in the feature space by linear boundaries. Logistic regression has traditionally been used to come up with a hyper plane that separates the feature space into classes. But if we suspect that the decision boundary is nonlinear we may get better results by attempting some nonlinear functional forms for the logit function. Solving for the model parameters can be more challenging but the optimization modules in scipy can help.