**Machine Learning Worksheets-3 Solutions**

1. SVM is one od the supervised machine learning algorithm that has the ability to solve regression as well as classification problems. It uses the kernels having different different functionalities:

**Linear Kernel** : **Linear Kernel** is used when the data is Linearly separable, that is, it can be separated using a single Line. It is one of the most common kernels to be used. It is mostly used when there are a Large number of Features in a particular Data Set. One of the examples where there are a lot of features, is **Text Classification**, as each alphabet is a new feature. So we mostly use Linear Kernel in Text Classification.

When training a SVM with a Linear Kernel, only the optimisation of the **C Regularisation** parameter is required. On the other hand, when training with other kernels, there is a need to optimise the **γ** parameter which means that performing a grid search will usually take more time.

**RBF Kernel**: Gaussian RBF(Radial Basis Function) is another popular Kernel method used in SVM models for more. RBF kernel is a function whose value depends on the distance from the origin or from some point. RBF Kernel is of the following format:

***K(X1,X2) = exponent (-y || X1 – X2||2 )***

**Polynomial Kernel:** It represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models.

Intuitively, the polynomial kernel looks not only at the given features of input samples to determine their similarity, but also combinations of these. In the context of regression analysis, such combinations are known as interaction features. The (implicit) feature space of a polynomial kernel is equivalent to that of polynomial regression, but without the combinatorial blowup in the number of parameters to be learned. When the input features are binary-valued (booleans), then the features correspond to logical conjunctions of input features.

1. R-squared is a better measure of goodness of fit of model in regression. Because Residual sum of squares is the sum of squares between the actual versus predicted values which clearly don’t give us the info regarding goodness of fit.
2. Let us assume Yi is the actual observed value of the dependent variable, y-hat is the value of the dependent variable and y-bar be the mean value of y according to the regression line, as predicted by our regression model. So lets define the statistics square term one by one.

**ESS**: It is abbreviated as explained sum of squares. The ESS is the total variation of that measures how well the regression equation explains the relationship between x and y.

ESS= summation(y-hat – y-bar)2

**RSS** : Abbreviated as residual sum of squares. is the portion of total variation that measures discrepancies (errors) between the actual values of *Y* and those estimated by the regression equation.

**RSS**=summation(Yi – y-hat)2

TSS: It is total of explained sum as well as residual sum computed from the equation.

**TSS**= summation(Yi-y-bar)2

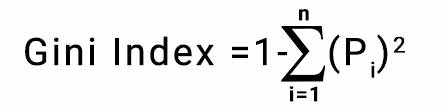
**Relation:**

TSS= ESS + RSS

R2 = ESS/TSS

1. **Gini Index**, also known as Gini impurity, calculates the amount of probability of a specific feature that is classified incorrectly when selected randomly.If all the elements are linked with a single class then it can be called pure.

Let’s perceive the criterion of the Gini Index, like the properties of entropy, the Gini index varies between values 0 and 1,where 0 expresses the purity of classification, i.e. All the elements belong to a specified class or only one class exists there. And 1 indicates the random distribution of elements across various classes. The value of 0.5 of the Gini Index shows an equal distribution of elements over some classes.

The Gini Index is determined by deducting the sum of squared of probabilities of each class from one, mathematically, Gini Index can be expressed as

1. Yes, Unregularized Decision Tree prone to overfitting.  Decision trees are prone to overfitting, especially when a tree is particularly deep. This is due to the amount of specificity we look at leading to smaller sample of events that meet the previous assumptions. This small sample could lead to unsound conclusions.
2. Ensemble techniques are the techniques used in machine learning which combines several machine learning algorithms into one predictive model in order to decrease variance(**Bagging** techniques),Bias(**Boosting** Techniques), or improve techniques.

Bagging Techniques: Random Forest

Boosting Techniques: Gradient Boosting, AdaBoost etc

1. **Bagging Techniques**:

* Various training data subsets are randomly drawn with replacement from the whole training dataset.
* Bagging attempts to tackle the over-fitting issue.
* If the classifier is unstable (high variance), then we need to apply bagging.
* Every model receives an equal weight.
* Objective to decrease variance, not bias.
* It is the easiest way of connecting predictions that belong to the same type.

**Boosting Techniques**:

* Each new subset contains the components that were misclassified by previous models.
* Boosting tries to reduce bias.
* If the classifier is steady and straightforward (high bias), then we need to apply boosting.
* Models are weighted by their performance.
* Objective to decrease bias, not variance.
* It is a way of connecting predictions that belong to the different types.

1. The Random Forest Classifier is trained using bootstrap aggregation, where each new tree is fit from a bootstrap sample of the training observations Zi=(xi,yi). **The out-of-bag (OOB) error** is the average error for each Zi calculated using predictions from the trees that do not contain Zi in their respective bootstrap sample. This allows the model to be fit and validated whilst being trained.
2. K-Fold Cross Validation is where a given data set is split into a K number of sections/folds where each fold is used as a testing set at some point.

Lets take the scenario of 5-Fold cross validation(K=5). Here, the data set is split into 5 folds. In the first iteration, the first fold is used to test the model and the rest are used to train the model. In the second iteration, 2nd fold is used as the testing set while the rest serve as the training set. This process is repeated until each fold of the 5 folds have been used as the testing set.

1. When creating a machine learning model, we will be presented with design choices as to how to define the model architecture. Often times, we don't immediately know what the optimal model architecture should be for a given model, and thus we'd like to be able to explore a range of possibilities. In true machine learning fashion, we'll ideally ask the machine to perform this exploration and select the optimal model architecture automatically. Parameters which define the model architecture are referred to as **hyperparameters** and thus this process of searching for the ideal model architecture is referred to as **hyperparameter tuning.**
2. If we have large learning rate in gradient descent then our data bounces here and there but not able to reach to the global minima that is the actual destination of it. The learning rate basically corresponds to the size of the step you take on the error surface at each iteration.

Having too large a learning rate, we risk overstepping the local minima, and therefore not converging.

1. **Bias**: Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data.

**Variance:** Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn’t seen before. As a result, such models perform very well on training data but has high error rates on test data.

**Tradeoff**: If our model is too simple and has very few parameters then it may have high bias and low variance. On the other hand if our model has large number of parameters then it’s going to have high variance and low bias. So we need to find the right/good balance without overfitting and underfitting the data.

This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can’t be more complex and less complex at the same time.

1. 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:

* Lasso Regularisation
* Ridge Regularisation

1. **Gradient boosting** generates learners during the learning process. It build first learner to predict the values/labels of samples, and calculate the loss (the difference between the outcome of the first learner and the real value). It will build a second learner to predict the loss after the first step**.**The step continues to learn the third, forth… until certain threshold.

**Adaboost** requires users specify a set of weak learners (alternatively, it will randomly generate a set of weak learner before the real learning process). It will learn the weights of how to add these learners to be a strong learner. The weight of each learner is learned by whether it predicts a sample correctly or not. If a learner is mispredict a sample, the weight of the learner is reduced a bit. It will repeat such process until converge.

1. No,we can’t use logistic regression to compute classification problem having non-linear data as an input.

Logistic Regression is one of the supervised machine learning techniques used for the binary problems.It comes under the linear model and compute the data properly if we have the linearity in our data. If we attempt to solve it raise error during processing due to ambiguity and don’t able to comes to the result.Some of the statistics process from scioy library ease the process but even results gets so differ.