**Theoretical Assignment – 1**

**True/False:**

1. True – Reduces the chance of overfitting and makes the model more adaptable to new unseen data
2. True – Averaging reduces the fluctuations in data
3. True – At every step the weights for the data are modified to make sure the next gradient descent is more accurate
4. True – If not done randomly, it might cause sampling bias.
5. False – If the model is overfit, the testing error can be 100% while training error is 0%
6. False – Regularization makes the model simpler, due to which the bias will increase.
7. False – It will overfit and make an overly complex decision tree
8. False – Every classifier makes some assumptions about the data
9. True – Bagging subsections and taking an average makes RandomForests more accurate than decision trees.
10. True – This is known as the asymptotic property of KNN’s.
11. False – Sorting has a worst time complexity of O(nlogn)
12. True – It is the error due to inherent noise in the data, and is the lower bound on the error.

**ERM and SVM**

1.

|  |  |  |
| --- | --- | --- |
| Model | Loss function | Regularizer |
| SVM | Hinge Loss  L(y) = max(0,1-wxy) where  y = +-1, w = weight | L2 regularizer:  ||W2 ||/ 2 where W = weight |
| LASSO | Mean Squared Error (MSE):    W - Weight | L1 regularizer:    W - Weight |
| RIDGE | MSE (Same as LASSO) | L2 regularizer:(same as SVM) |

2.

A – Hinge

B – MSE (Mean Squared Loss)

C – Logistic/Logarithmic

D – Exponential

E – 0-1

3.

Gradient Descent is based on the differentiability of the function. Therefore, loss functions that can be minimized by this are:

Hinge, MSE, Logistic, Exponential

* 1. being non differentiable cannot be minimized by gradient descent

Newton’s method relies on the double differentiability of the function. Therefore, loss functions that can be minimized by this are:

Logistic, Exponential

The rest are not doubly differentiable functions, and hence Newton’s method cannot be applied.

**Bias and Variance:**

1.

* Underfitting
* Low quality of data (a smaller number of features or a high amount of noise in the data)

2.

If train and test error remain high, even after high amounts of training, then it could mean that the data has a very high amount of noise i.e. wrong data or data with very high variations and no exact relation. This error cannot be removed and is an inherent quality of the data, due to which no amount of training will reduce the error.

3.

Bagging creates several versions of a model by:

* Creating Multiple Datasets: Randomly picking samples from the original dataset with replacement to make new datasets.
* Training Multiple Models: Training a model (like a decision tree) on each of these new datasets.
* Combining Predictions: Combining the predictions from all the models by averaging (for regression) or majority voting (for classification).

Averaging different predictions from different models and datasets tends to reduce the fluctuations present in the data. This helps in making the data more uniform and thus helps to make the predictions more accurate with lesser variance.

4.

Bias:

* Initially, the model makes large errors because the shallow trees are too simple.
* As boosting adds more trees, each new tree corrects the mistakes of the previous ones.
* This process reduces errors because the combined model can learn more complex patterns.

Variance:

* Boosting can sometimes lead to overfitting, where the model fits the training data too closely and doesn’t generalize well to new data.
* However, because each tree is very simple (depth = 2), the risk of overfitting is lower.
* Therefore, the overall variance of the model remains manageable, resulting in a model with lower errors and balanced complexity.

**KNN/Curse of Dimensionality:**

1.

To reduce the time taken some steps that Kim can take is:

* Reducing Dimensions: By using dimensionality reduction techniques, the value of da can be reduced which reduces the processing time.
* Instead of processing one at a time, try using the CPU to its maximum and perform parallel computing (multiple tasks at the same time) to reduce the time taken.
* Choose the right k, since this is very crucial in KNN’s and can make a huge difference in time taken.

2.

a) Yes, this will affect the KNN accuracy as it will give different outputs now. For example, let us assume we are working with only two nearest neighbours and we have the Euclidean distances of two different cases as (6,3) and (7,1). Now 6+3 = 9 > 8 = 7+1. But if we take the squared Euclidean distance, we have 62 + 32 < 72 + 12. Hence, it gives different outputs.

b) The answer to the first question is generalized and does not depend on how we calculate the distance, hence it will remain the same even if we are using the squared Euclidean distance

3.

The data has a massive dimension and very few datapoints. In higher dimensions, it becomes more difficult to distinguish between different classes as the distances between the points becomes more uniform. This coupled with the low number of datapoints can cause the KNN model to perform very poorly. Hence Kanye W has to be careful about the curse of dimensionality.

4.

As the value of k increases, the bias increases while the variance decreases. Small k will closely follow the training data, due to which it will have low bias, but it will be highly affected by noise (increasing variance). For large k, the opposite happens, since large k smoothens the data making variance less but increasing the bias.

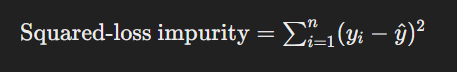
5.

Smaller datasets with no clear linear boundary is where KNN’s will be preferred over linear SVM’s, while when the dataset is large with a linear relationship, linear SVM’s perform much better than KNN’s.

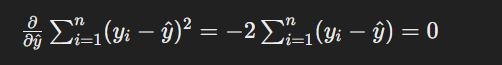
**Decision Trees:**

1.

The prediction value at a leaf node of a regression tree with squared-loss impurity is the mean of the target values of the data points in that leaf.



Now taking the derivative and equating to zero for minimizing we have



Solving we get

 hence the optimal value is the average.

2.

G(S) is maximized when all the three classes have equal probability (p1 = p2 = p3 = 1/3). G(S) = 2/3 in this case

Minimum arises when one of them is 1 and other two are zero (assume p1 = 1). G(S) = 0 in this case.

3.

Decision trees are myopic in the sense that they only look at the current best decision and not at the overall best decision. This approach might not guarantee the best decision since the tree does not look at future decisions.

4.

Pruning: Reducing the complexity by cutting the branches of the tree, lowering the chances of overfitting.

Bagging: Taking the average of various different branches of the tree, and averaging it out (creating a RandomForest in essence. This reduces the fluctuations and complexity thus again minimizing chances of overfitting.

**Boosting and Bagging**

1.

Yes, we can use the training data as the testing data for RandomForests. This is because, in RandomForests, the full data is never used for training and only parts of the data are used at each step. Then the average of all these steps is taken. Thus, since the full data is never seen by the computer at a single time, it can be used as a testing dataset without any problems of having 100% accuracy (but the accuracy will generally be higher than when used with a separate testing dataset independent of the training dataset).

2.