Assingment -1

SECTION A – True / False

1. Bagging reduces the chance of overfitting, making the model more adaptable to unseen data. True
2. Averaging predictions reduces fluctuations in data. True
3. In boosting, the weights of data points change at every step to make the next gradient descent more accurate. True
4. Sampling without randomness can introduce sampling bias. True
5. If a model is overfit, training error can be 0% while testing error can be 100%. True
6. Regularization simplifies the model, which decreases bias. False
7. Increasing the depth of a decision tree always prevents overfitting. False
8. Every classifier makes assumptions about the data. True
9. Random Forests are more accurate than single decision trees because they combine bagged trees. True
10. Irreducible error is the lower bound on error due to inherent noise in the data. True

SECTION B – ERM and SVM

1. Fill the following table:

Model	Loss Function	Regularizer
SVM	Hinge Loss : $\max(0, 1 - y(\omega \cdot x + b))$	$L_2 \|\|\omega^2\|\|$: loss func. $+ \lambda \sum_{i=1}^{m} \omega_i^2$
LASSO	Mean square Error (MSE) : $(y - \hat{y})^2$	$L_1 \|\|\omega_1\|\|$: loss func. $+ \lambda \sum_{i=1}^{m} \|\omega_i\|$
RIDGE	Mean Square Error (MSE) : $(y - \hat{y})^2$	$L_2 \|\|\omega^2\|\|$: loss func. $+ \lambda \sum_{i=1}^{m} \omega_i^2$

Here; λ = regulization Const.

3. Short Answer:

(a) Which loss functions can be optimized using gradient descent? Why?

(b) Which loss functions can be optimized using Newton's method? Why?

→ (a) Gradient descent works with first derivative of the loss function. So every differentiable (at least one) loss function can be optimized by gradient descent method. like – MSE (Convexe Upward), logistic loss can be optimized by gradient descent.

b) While Newton's method deal with gradient (first order derivative) as well as hessian

Created by Notein

(second order derivative). So loss like that are twice differentiable can be optimized by Newton's method.
like : MSE; as it is twice differentiable.

SECTION C – Bias and Variance

1. Explain one major reason why underfitting occurs.

→ Underfitting occurs when the model is **too simple** to capture the underlying pattern in the data, leading to **high bias** and poor performance on both training and test data.

2. If both training and test error is high, what does this imply about the data?

→ If both the training & test error is high, it simply implies that the model suffer from **high bias**. Hence it is a **underfitted** model and unable to learn the structure of the data.

3. Explain how bagging reduces variance.

→ Bagging first divide the total dataset into several random samples or **bootstrap**; then it make prediction on the samples and finally take **average** of these predictions. By this method, it can cancel out individual model fluctuations and stabilize the final prediction.

4. Explain the effect of boosting on bias and variance.

→ Boosting also divide the dataset into some

NOTE IN Created by Notein

random samples. Then it start from assumed initial parameters, go through each sample, calculate loss func^n, update parameters accordingly. By this sequential learning method; it can **reduces bias**. It may **slightly increase varience**, but overall improves model accuracy.

SECTION E – Decision Trees

1. Derive that the optimal prediction at a leaf (with square loss) is the mean.

→ Let a leaf contain targets $y_i, \forall i, i = \{1, \dots, n\}$

Let the optimal prediction of the leaf be C; where C is a constant.

Square loss:

$$L(c) = \sum_{i=1}^{n} (y_i - c)^2$$

$$\Rightarrow \frac{\partial L(c)}{\partial c} = -2\sum_{i=1}^{n} (y_i - c)$$

(differentiate wrt c)

As a C is the optimal prediction;

then; $L(c)\Big|_{c=C} = \text{minimum}$

$$\Rightarrow \frac{\partial L(c)}{\partial c}\Big|_{c=C} = 0 \quad \Rightarrow \sum_{i=1}^{n} y_i - nC = 0$$

$$\Rightarrow \boxed{C = \frac{1}{n}\sum_{i=1}^{n} y_i}$$

∴ Hence; the optimal prediction is the mean of the targets

Created by Notein

in the leaf.

2. What are the max/min Gini impurity values for 3 classes?

→ Gini impurity –

$$G = 1 - \sum_{K=1}^{3} p_K^2$$

P_K = Probibility for each K

Now we know that-
A lower gini indexe indicates more homogenous or pure distribution.
While a higher gini indexe indicates a more heterogenous or impure distribution.

Hence; minimum gini possible for a pure node:

→ Pure node : $P = (1, 0, 0)$

$$G_{min} = 1 - (1^2 + 0^2 + 0^2) = 0$$

Similarly maximum gini possible for a perfect impure node. A perfeck impure or heterogenous node is a node with uniform distribution.

→ Uniformly distributed node : $P = (1, 1, 1)$

$$G_{max} = 1 - ((1/3)^2 + (1/3)^2 + (1/3)^2)$$

$$= 1 - (\cancel{3} \times 1/\cancel{9}_{3}) = 2/3$$

∴ Hence $G_{min} = 0$ & $G_{max} = 2/3$ for 3 classes.

3. Why are decision trees myopic learners?

→ Myopic means short-sighted or greedy.

Decision tree acts myopic or greedy alike because at each node it choose the best split to get immediate reduction in impurity independently **without considering any previous or future split.**

It means as other algorithm try to reach global minima in loss function, decision tree does not follow that.

4. Explain two methods to avoid overfitting in decision trees.

→ 1. Limiting maximum depth; we can prevent over-complexity of model.

2. Remove features, branches; we simplify the model and improve generalization.

SECTION F – Boosting & Bagging

1. Can Random Forest use the same data for training and testing? Justify.

→ YES – Random Forest can use the same dataset for training and testing.

Because, Random Forest first do a sampling (random) over the dataset for training. So a chunck amount of data remain untouched after the training. So we can use the rest dataset as a test set. Hence Random Forests obtain an **unbiased estimate of test error** without requiring a seperate test dataset.

Created by Notein

2. Explain the key differences between bagging and boosting.

→ 1. Bagging learn parallely over all the random sampled dataset. While, boosting learns sequentially.

2. Bagging aim to reduce varience productively. While Boosting aim to reduce bias.

3. Hence, bagging deal with overfitting & boosting deal with underfitting.

4. Bagging; each model lead to diff. optima; finally take average of them. While Boosting update parameter sequentially.

NOTE IN Created by Notein