SECTION A - True/False

1. Bagging reduces the chance of overfitting making the model more adaptable to unseen data. (True)
2. Averaging predictions reduces fluctuation in data (True)
3. In Boosting, the weights of data points change at every step to make the next gradient descent one 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. ~~(True)~~ (False)
7. Increasing the depth of a decision tree always prevents overfitting. ~~(True)~~ (False)
8. Every Classifier makes assumptions about the data. (True)
9. Random forests are more accurate than Single decision tree because the 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 EVM

1. fill the following table

Model	Loss function	Regularizer
SVM	Hinge Loss	L2
LASSO	Squared Loss	~~Abil~~ L1
RIDGE	Squared Loss	L2

2. Identify the Loss-function A-E

A:

B:

C:

D:

E:

3. Short Answer

(a) which loss function can be optimized using gradient descent ? why?

→ Squared, logistic Loss function can be optimized using gradient descent because it is differentiable function. (smooth ~~fn~~ loss function)

(b) which loss functions can be optimized using Newton's method ? why?

→ Squared loss and Logistic Loss functions can be optimized using Newton's method because these functions are twice differentiable (smooth).

SECTION - C Bias and Variance

1. Explain one major reason why underfitting occurs?

→ underfitting occurs when our model is too linear or simple ~~for complex~~ to capture the complex patterns in the data. it means our model has not learned anything.

2. If Both training and test error remains high, what does this imply ~~to~~ about the data?

→ It means our model fails the ~~fixed~~ training set and test set. ~~and has~~ our model has not learned anything. So if our model is not in underfitting then there is a chance that no pattern in the data to be found.

3. Explain how bagging reduces variance.

→ Bagging creates many copies of the same model. Each copy is trained on a random subset of the data. So average of all their results is usually very accurate and drastically reduces variance.

Date __/__/____
Page _____

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

→ Boosting Primarily reduces Bias by Sequentially training models to Correct the errors of previous ones. ~~It also reduces variance as~~ It can also reduce variance.

SECTION D - KNN & Curse of Dimensionality

1. what step can be taken to reduce KNN Computation time?

→ we can ~~speed up~~ reduce KNN Computation time by reducing the number of features (using dimensional reduction or shrinking the dataset. Instead of checking every single point, using ~~Spatial data str.~~ KD-Trees or Ball trees allows the KNN algorithm to efficiently skip distant neighbours.

2. (a) Does Squared Euclidean distance Change predictions? Explain.

No, it doesn't change predictions because the nearest neighbours remain the same regardless of whether we compare raw distances or Squared distances.

b) Does it affect the previous dimensionality conclusion?

→ No, it doesn't affect the previous dimensionality conclusion - because high dimensional data remains sparse and distances still lose meaning, so we have still problem of high dimensionality.

3. Why does KNN perform poorly in high dimensions with few data points?

→ In high dimensional space available data points are scattered far apart. This breaks distance based algorithm KNN because 'nearest' neighbours are no longer actually close in very high dimensions. So every point roughly the same distance away from every other point.

4. How do bias and variance change with K?

→ With a small K, we get low bias but high variance, meaning the model captures local details perfectly but is very sensitive to noise. And as we increase K bias increases and variance decreases.

5. When is KNN preferred over linear SVM?

→ KNN is preferred when decision boundary is highly non-linear or irregular, as a linear SVM is restricted to drawing straight lines to separate classes if our data has complex clusters rather than a clear linear split, KNN* is much more effective.

SECTION E - Decision Trees

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

→ actual target values ⇒ $y_1, y_2, y_3 \dots y_n$

n = number of points in the leaf

Single value $= C$

$$J(C) = \sum_{i=1}^{n} (y_i - C)^2$$

Diff. w.r.t. to C

$$\frac{dJ}{dC} = \frac{d}{dC} \sum_{i=1}^{n} (y_i - C)^2$$

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

Set the derivative to zero.

$$-2 \sum_{i=1}^{n} (y_i - C) = 0$$

$$\sum_{i=1}^{n} (y_i - C) = 0$$

$$\Rightarrow nc = \sum_{i=1}^{n} y_i$$

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

So the value c that minimizes the squared loss is exactly the airthmatic mean of the target values in that leaf.

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

→ minimum Gini impurity = 0

node is completely pure, meaning all data points belong to a single class. So if class 1 has prob. 1 and other 0.

Gini = $1 - (1^2 + 0^2 + 0^2) = 0$.

maximum Gini impurity = $\frac{2}{3}$

here data is ~~evidy~~ evenly split among all 3 classes. Each class has a prob. of $1/3$.

$$\text{Gini} = 1 - \left(\left(\frac{1}{3}\right)^2 + \left(\frac{1}{3}\right)^2 + \left(\frac{1}{3}\right)^2\right)$$

$$= \boxed{\frac{2}{3}}$$

3. Why are decision trees myopic learners?

→ decision trees are considered myopic learners. because they use a greedy algorithm. At each step they choose the split that maximize the immediate gain based on the current state without caring about future values.

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

→ (1) pre-Pruning ⇒ Stop the tree from growing before it become too complex by setting limits like maximum depths or minimum sampling.

(2) Post-Pruning ⇒ Allow the tree to grow fully to fit the training data then trim back branches that do not improve accuracy on a validation set.

Section F:- Boosting and Bagging.

1. Can Random forests use the Same data for training and testing? Justify.

→ No, we can't use the same data for training and testing. If we test a random forest on the exact Same data it was trained on, the model will likely achieve 100% accuracy. This is because the model has memorized the patterns in the dataset. So we can't generalize our model to new unseen data.

2. Explain the key difference between bagging and boosting.

→ bagging:- bagging creates many Copies of the Same model. Each Copy is trained on a random Subset of the data

boosting:- it trains model sequentially The second model focuses heavily on the data points the first model got wrong.

So key difference is working principle.