Assignment-1

(A)
1. True
2. True
3. False
4. True
5. True
6. False
7. False
8. True
9. True
10. True

(B) 1:

	Model	Loss function		Regularizer
1.	SVM -	Hinge Loss	$\max(0, (1 - y\hat{y}))$	- L2 Norm ($\|w\|^2$)
2.	LASSO -	Squared Loss	$(y - \hat{y})^2$	- L1 Norm ($\Sigma\|w\|$)
3.	RIDGE -	Squared Loss	$(y - \hat{y})^2$	- L1 Norm ($\Sigma\|w\|$)

2

(a) Loss functions that are differentiable, can be optimized using gradient descent. Examples include mean squared loss and hinge loss (due to existence of sub-gradient).

(b) Loss functions that are atleast twice differentiable, can be optimized using newtons method. Squared loss can be optimized using Newtons method.

(C)

1. One of the major reason why underfitting occurs is when the model is too simple so it can't capture the complex features in data. This can happen due to using a strong regularization or using overly simple model like linear Regression which may not be able to capture non-linear features.

2. If both training and testing errors are high, then it implies that either the data is too complex for the model to extract and learn some features or, the in the data, the important features are missing.

3. Bagging reduces variance by training multiple models on different samples of data (obtained by random sampling with repetition) and then, averaging their predictions, which may help in cancelling out individual model fluctuations and reduce noise in output.

4. Boosting reduces bias by sequentially training models such that each new model focuses more on wrong predictions made by previous models. In this way, by repeatedly correcting these wrong predictions, it finally helps build a strong model with weights that can now capture the features of our complex data. Bias decrease as model is now not simple and properly capture the features in the data.
Variance generally increases as boosting sequentially leads to overfitting.

(E)

1. Let at a leaf node, all samples reaching the leaf have target values $\{y_1, y_2 \text{ ——— } y_n\}$.

Let the predicted value by model be λ.

So, Squared loss $\Rightarrow$ $L(p) = \sum_{i=1}^{n} (y_i - p)^2$

So, optimal prediction (λ) will be such that the squared loss is minimum.

$$\text{So, } \frac{\partial L(p)}{\partial p} = -\sum_{i=1}^{n} 2(y_i - p) = 0$$

$$p = \lambda = \frac{\sum y_i}{n} \longrightarrow \underline{\text{mean}}$$

2. We know, $\text{Gini} = 1 - \sum_{i=1}^{n} p_i^2$

where, p_i = probability of class i

* $(\text{Gini})_{\min}$ $\longrightarrow$ when all the samples belong to one target class (so one of the class has $p=1$ and others 0)

For $n=3 \rightarrow$ 3 classes,

$$(Gini)_{min} = 1 - 1^2 = \boxed{0}$$

* $(Gini)_{max} \longrightarrow$ When the samples are equally likely to occur in each class (so all class has same probability)

For $n=3 \rightarrow 3$ classes,

$$(Gini)_{max} = 1 - 3 \times \left(\frac{1}{3}\right)^2 = \boxed{\frac{2}{3}}$$

3.

Decision trees work in such a way that they select the split that gives maximum immediate reduction in impurity without looking on the long-term impact of that split. And since, once a split is made, the decision becomes irreversible. So a better local split may affect the global output. Therefore, because of this greedy way the decision trees work, they are called myopic learners.

4.

Pre-pruning :- This method avoids overfitting by restricting tree growth using constraints like maximum depth (no. of levels in a tree), minimum samples per leaf (so that tree don't create leaves that have very less data points, noise) and minimum impurity decrease.

Post-pruning :- This method reduces overfitting by first letting a tree grow completely and then removing branches that do not significantly improve performance, to improve generalization. (removes noise).

(F)

1. Yes, in random forests, each tree is only trained on a subset of that data so many data points are there that are not yet seen by that tree. So, we can have an algorithm to sample the data in such a way that we make 2 datasets for each tree (by sampling), that have quite less data points common between them and hence these datasets can be used as training and testing.

2.

Bagging	Boosting
- Trains many models independently	- Trains model sequentially
- Uses randomly sampled datasets for each model.	- Improves weights of data points (in each iteration)
- Mainly tries to reduce variance	- Main role is to reduce bias