Assignment-1 :

Section-A :

True : (1), (2), (4), (5), (7), (8), (9), (10).

False : (3), (6)

Section-B :

1.

Model	Loss function	Regularizer.
SVM	$\max(0, 1-y_i \hat{y}_i)$	$\frac{1}{2}\|\|W\|\|^2$
LASSO	$(y_i - \hat{y}_i)^2$	$\|\|W\|\|_1 = \sum_{j=1}^{d} \|w_j\|$
RIDGE	$(y_i - \hat{y}_i)^2$	$\|\|W\|\|^2$

2. a) Loss functions which are continuous, differentiable can be used for gradient descent. This is because while calculating the next step in gradient descent, we just use first derivative i.e: $w_{new} = w_{old} - \alpha l(w_{old})$.

b) In addition to the above conditions, the loss function should also be twice differentiable for newton method. This is because we use second derivative while calculating the next step.

Section-C :

1. Lack of number of datapoints for training is a major reason for underfitting.
2. The Model is not able to train/understand the pattern, basically underfitting is occuring.
3. Bagging reduces variance by training multiple models on different samples of training dataset and taking average of every models. Randomness in sampling process and averaging the models reduces the variance by cancelling out the prominent fluctuations.
4. Boosting can reduce Bias and can reduce the Variance.

Section-D

1. • Reducing the dimension of feature vector or using simple distances can reduce the computational time for calculating the distances.
   • Reducing k also speeds up the computation.
2. (a) Squaring of Euclidean distance doesn't change the predictions because squares and linear are ordered in same way i.e $x < y \Rightarrow x^2 < y^2$, So it doesn't really matter if take square or not.

   (b) No this doesn't affect the dimensionality's conclusion, there would still be curse of dimensionality if you increase the dimensions.

2. This happens because of curse of dimensionality. When dimension of feature vector increases, the datapoints tend to accumulate too closer which makes the distance between the points meaningless and less informative.

4. Small K results in low bias and high variance
Large K results in high bias and low variance.

5. KNN is prefered when the data has low dimensions and small in size. KNN's are also good at finding locally varying characters and have great non-linear classification. SVM's are mostly used for linear classification.

Section-E :

1. For optimal prediction, we try to minimize the loss E :

$$E = \sum_{i=0}^{n} (y^{(i)} - \hat{y})^2 \Rightarrow \frac{\partial E}{\partial \hat{y}} = \sum_{i=0}^{n} 2(y^{(i)} - \hat{y})(-1) = 0.$$

$$\Rightarrow \sum_{i=0}^{n} y^{(i)} - \sum_{i=0}^{n} \hat{y} = 0 \Rightarrow n\hat{y} = \sum_{i=0}^{n} y^{(i)}$$

$$\Rightarrow \boxed{\hat{y} = \frac{1}{n} \sum_{i=0}^{n} y^{(i)} = \text{mean}}$$

2. Gini impurity : $G = 1 - \sum_{i=1}^{k} p_i^2$, given $k = 3$

for G_{min} the probabilities would be : $(1, 0, 0)$

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

for G_{max} the probabilities would be : $(1/3, 1/3, 1/3)$

$\therefore G_{max} = 1 - \{(\frac{1}{3})^2 + (\frac{1}{3})^2 + (\frac{1}{3})^2\} = 1 - \frac{3}{9} = \frac{2}{3}$

3. Decision trees are myopic learners / greedy leaners because they tend to make optimal decision at each node without considering the effect on entire global tree.

4. Overfitting can be avoided by two methods :

   1) Pruning : Removing the nodes that are unnecessary or doesn't provide any valuable information.

   2) Controlling tree growth : tree growth can be controlled by setting constraints like maximum depth or minimum samples per leaf.

Section- F :

1. Random Forests should not use the same data for training and testing because this introduces biased performance estimates.

2. The Major difference between the boosting and bagging is that bagging is a parallel process whereas boosting is sequential process.