Task 3

1. What is the difference between overfitting and underfitting?

Overfitting: the model performance is extremely good at the data set but when it see a new instances it would performance badly meaning it can not generalize to new instances We know that it overfits when the error on the validation set is extremely bigger than the Training set error

Underfitting: the model performance is bad on either the training set and validation set Due to bad or weak algorithms like using linear regression on data that seems to be not linear, and we solve this problem by increase the polynomials of the algorithms

2. What do you know about ensemble learning?

Rather than using a single predictor (regressor or classifier) it's better to train multiple Predictors and get the aggregate of them and the result would be better than the best individual predictor so this group of predictors is called ensemble

This is done using some methods like:

Bootstrap(tree ensembles) : if we have a training set of size m we use sampling with replacement to generate a new training sets with size m and train the trees on this training sets but the if we only do that the issue would be the trees would be almost identical so we fix it : when choosing a feature to split at each node if n features are available , we just pick a random subset of k<n features and only choose from this subset of features ***"and that's typically what happens when using random forest algorithm"***

Boosting: use sampling with replacement to create a new training set of size m, but instead of picking from all examples of equal probability, make it more likely to pick examples that the previous trained tree misclassify so we can see that the trees are trained sequentially ***"XGBoost"***

For prediction :

In classification it takes the most frequently class "the majority vote"

For regression it averages the predictions for all trees

Tree ensembles

"using multiple decision tree"

The problem with the trees is that it's extremly sensitive to the small change

we overcome this problem by using multiple trees and pick the most promising one

Sampling with replacment

- picking object from the set and return it to the set before picking another one
- by this criteria we can make multiple dataset "similar to the original one but not identical & an object can appear multiple time"

Generating a tree sample

- Giving training set of size m
  use sampling with replacment to create a new training set of size m and train a decision tree on the new data set
  "the issue that the trees generated by that method is almost identical" Bagged decision tree"
  we can overcome this problem by
  "Randomizing the feature choice"
- At each node when choosing a feature to use to split, if n features are available, pick a random subset of $k < n$ features and allow the algorithm to only choose from the subset of features $K = \sqrt{n}$ "Random forest algorithm"

These information I gained from Anrew NG course and that is a part from my study