UNIT 6 ASSIGNMENT

Improving Performance with   
Ensemble Methods

## Instructions

The questions below will prepare you for future interviews as they relate to concepts discussed throughout the week. You’ve practiced these concepts in the coding activities, exercises, and coding portion of the assignment. Now, let’s formulate your programming into well-thought responses.

Except as indicated, use this document to record all your assignment work and responses to any questions. At a minimum, you will need to turn in a digital copy of this document to your facilitator   
as part of your assignment completion. You may also have additional supporting documents that   
you will need to submit. Your facilitator will provide feedback to help you work through your findings.

**Note:** Though your work will only be seen by those grading the course and will not be used or shared outside the course, you should take care to obscure any information you feel might be of a sensitive or confidential nature.

*Begin your assignment by completing the questions below. Directions to submit your work can be found on the assignment page. Information about the grading rubric is available on any of the course assignment pages online. Do not hesitate to contact your facilitator if you have any questions about the assignment.*

Week 6 Written Portion

# Choosing Your Model

Answer the questions below about ensemble methods.

1. Explain ensemble modeling. What is the advantage of using this technique?

| **Ensemble methods** are a class of techniques that train multiple models and aggregate them into a single prediction. It combines multiple independent models that are predicting the same outcome, and this technique is applicable to both classification and regression problems. The idea is that it averages the predictions of various models to cover the deviations of error to lead to better generalization.  There are a few ensemble methods: Bagging, Boosting, and Stacking  Some advantages: better performance, less overfitting/underfitting, deals better with diverse data, can be applied to a wide range of algorithms  Disadvantages: more expensive, more time, less interpretability. |
| --- |

1. Explain what bias and variance are, along with the bias-variance tradeoff.

| You compute **Model Error =** Estimation Bias + Estimation Variance. Estimation Bias is the model’s ridgity that prevents adaptation to nuances of the data. Bias is difference between the average prediction of our machine learning model and the correct value, which we're trying to predict. If it has high bias it is too simplistic and doesn't capture the complexity of the data.With high bias, the model is not flexible enough to learn from the data and are said to have a "bias towards" a certain outcome.  Estimation Variance is a model’s flexibility that causes the estimated model to be sensitive to data nuances. Variance is the variability of a model prediction for a given data point or a value, which tells us the spread of our data. With high variance the model pays a lot of attention to the training data and overfits.  Bias-variance tradeoff is the relationship between bias and variance and how they affect a model's performance.If a model is too simple or has high bias, it may fail to capture important patterns aka underfitting. if it’s too complex or has high variance, it becomes overly sensitive to the training data aka overfitting.  Low Bias; High Variance: more flexible and captures complexity but sensitive to outliers  High Bias; Low Variance: too simple and can’t capture important patterns. |
| --- |

1. Explain the differences among the ensemble methods bagging, boosting, and stacking.

| **Stacking**: taking a weighted combination of the predictions of a total of K different models. You can weigh each prediction separately but it trains on the same dataset.   * ONLY one that doesnt have a specific algorithmic implementation * Weighted sum of the indivdual model prediction * by averaging the models, it increases the chances of canceling out any error of any individual model * captures more patterns * better with bigger data to train the separate models * MUST   + vary the algorithm   + vary the features   + vary the hyperparameters   + vary the training data   **Bagging:** generating multiple models from the same data by taking bootstrapped sameples and averaging the indvidual model predictions   * + bootstrap and aggregating   + reduces overall estimation model variance   + **Bootstrapping**: training multiple instances of the same base model on different subsets of the data     - For i in Num\_Bootstraps:       * Bootstrap data = Sameole N examples randomly with replacement       * Build a Decision Tree on the bootstrap data       * Add the ith Decision Tree to the ensemble     - use the set of models to make predictions   + **Aggregation:**     - averaging model’s outputs(regression) or majority voting (classification)   **Boosting:** sequentially training models and the new models try to correct the errors of the previous ones.   * assigns different weights to the data based on their difficulty to learn * literally builds models * most influential hyperparameters: n\_estimator, max\_depth, learning\_rate |
| --- |

1. Explain the random forest algorithm and how it relates to decision trees and bagging.

| **Random forest** is a set of decision trees varied by using different subsets of the features. To make a prediction, you input your feature vector into each tree to get a set of individual predictions. If this is a classification task, the individual trees can output either a class label or probability of belonging to a specific class labeled one. If this is a regression, the output would be the average value of the label. You figure out how to make the trees more indpendent with bagging. You test with the bootstrapping algorithm and make the prediction by getting predictions from all trees and averaging them. you can tune the hyperparameters; most common one is n\_estimators.  Classification: resulting outputs from each tree are generally aggregated using a majority vote  Regression: resulting outputs from each tree are generally aggregated using mean  TRADEOFFS: random forest tends to do a better job at generalization than a single tree alone, the training time as well as the prediction time are more costly than a single tree alone |
| --- |

1. What’s the difference between gradient boosting decision trees and random forest?

| Gradient Boosting Decision Trees (GBDT) and Random Forest both use decision trees. Random Forest: builds multiple decision trees in parallel by training each tree independently on random subsets of the data and features. the prediction is taken by averaging all the predictions of each individual tree. GBDT builds decision trees sequentially and then creates weights to learn from the mistakes.  Random Forest trains each tree independently. GBDT builds trees in a “given” manner– each new tree is trained given the previous ones.  Random forest sets the number of trees in advance. GBDT requires more trees but is more susceptible to overfitting. And, GBDT are less interpretable since the tree evaluation metrics change. |
| --- |



*To submit this assignment, please refer to the instructions in the course*.