Ensemble Learning – Detailed Solutions (200 Marks)

Assignment Code: DA-AG-014

Question 1: What is Ensemble Learning in Machine Learning?

Ensemble Learning is a machine learning technique in which multiple individual models are trained and combined to solve the same problem. Instead of relying on a single model, ensemble methods aggregate predictions from several models to achieve higher accuracy and better generalization.

The core idea behind ensemble learning is diversity. Different models make different errors, and by combining them, these errors can cancel out. As a result, ensemble models are usually more robust and stable than individual models.

Question 2: Difference between Bagging and Boosting

Bagging (Bootstrap Aggregating) trains models independently using different bootstrap samples of the data and combines their predictions. Its primary goal is to reduce variance.

Boosting trains models sequentially, where each new model focuses on correcting the errors made by previous models. It primarily reduces bias and improves performance on difficult cases.

Question 3: Bootstrap Sampling and its Role in Bagging

Bootstrap sampling involves randomly sampling the dataset with replacement to create multiple training sets. Each sample is used to train a separate model.

In Random Forests, bootstrap sampling ensures that individual trees are diverse, which reduces overfitting and improves prediction accuracy.

Question 4: Out-of-Bag (OOB) Samples and OOB Score

Out-of-Bag samples are the data points not selected in a bootstrap sample. They are used as a validation set for estimating model performance.

Question 5: Feature Importance in Decision Tree vs Random Forest

Decision Trees calculate feature importance based on impurity reduction at each split. However, they can be unstable and sensitive to data variations.

Random Forests average feature importance across many trees, resulting in more stable and reliable importance measures.

Question 6: Random Forest on Breast Cancer Dataset

```
from sklearn.datasets import load_breast_cancer
from sklearn.ensemble import RandomForestClassifier
import pandas as pd

data = load_breast_cancer()
X, y = data.data, data.target

rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X, y)
```

```
importances = pd.Series(rf.feature_importances_, index=data.feature_names)
print(importances.sort_values(ascending=False).head(5))
```

Question 7: Bagging Classifier vs Decision Tree

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X, y = load_iris(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)

dt = DecisionTreeClassifier(random_state=42)
bag = BaggingClassifier(estimator=DecisionTreeClassifier(),
                        n_estimators=50, random_state=42)

dt.fit(X_train, y_train)
bag.fit(X_train, y_train)

print("Decision Tree Accuracy:", accuracy_score(y_test, dt.predict(X_test)))
print("Bagging Accuracy:", accuracy_score(y_test, bag.predict(X_test)))
```

Question 8: Hyperparameter Tuning using GridSearchCV

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.datasets import load_breast_cancer

X, y = load_breast_cancer(return_X_y=True)

param_grid = {'n_estimators': [50, 100], 'max_depth': [None, 5, 10]}

grid = GridSearchCV(RandomForestClassifier(random_state=42),
                    param_grid, cv=5)
grid.fit(X, y)

print("Best Parameters:", grid.best_params_)
print("Best Accuracy:", grid.best_score_)
```

Question 9: Bagging vs Random Forest Regressor

```
from sklearn.datasets import fetch_california_housing
from sklearn.ensemble import BaggingRegressor, RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

X, y = fetch_california_housing(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)

bag = BaggingRegressor(n_estimators=50, random_state=42)
rf = RandomForestRegressor(n_estimators=100, random_state=42)

bag.fit(X_train, y_train)
rf.fit(X_train, y_train)

print("Bagging MSE:", mean_squared_error(y_test, bag.predict(X_test)))
print("Random Forest MSE:", mean_squared_error(y_test, rf.predict(X_test)))
```

Question 10: Ensemble Learning for Loan Default Prediction

In loan default prediction, ensemble learning improves reliability by combining multiple models. Bagging helps reduce variance, while Boosting improves predictive power on complex patterns. Cross-validation ensures robustness, and ensemble decisions reduce financial risk.