Assignment No. 8

Problem Statement:

Implement Ensemble Learning techniques for classification tasks using methods like Voting, Bagging, Boosting, and Stacking.

Objective:

To understand and implement Ensemble Learning techniques by combining multiple base learners to improve prediction accuracy, reduce overfitting, and increase model robustness in classification problems.

Prerequisite:

1. Python environment with libraries like numpy, pandas, matplotlib, seaborn, scikit-learn.
2. Basic knowledge of classification algorithms (e.g., Logistic Regression, Decision Trees, SVM).
3. Understanding of ensemble techniques: voting, bagging, boosting, and stacking.

Theory

What is Ensemble Learning?

Ensemble Learning is a machine learning technique where multiple models (often called "weak learners") are combined to produce a stronger, more accurate model. The idea is that by aggregating predictions from different models, the ensemble can generalise better than any individual model.

It is especially effective in reducing bias, variance, and overfitting, which helps improve predictive performance.

Types of Ensemble Learning Techniques

1. Voting Classifier

What It Does:

Combines predictions from multiple models and uses majority voting (hard voting) or average probability (soft voting) to decide the final prediction.

Use Case:

Best for combining models that perform similarly but make different types of errors.

Example:

In this assignment, Logistic Regression, SVM, and Decision Tree were combined using soft voting for multi-class classification.

2. Bagging (Bootstrap Aggregating)

What It Does:

Trains multiple models (e.g., Decision Trees) on random subsets of the training data and aggregates predictions using majority voting.

Use Case:

Reduces variance and helps avoid overfitting, especially useful for high-variance models like decision trees.

Example:

Used BaggingClassifier with 100 Decision Trees trained on different bootstrapped samples of the training data.

3. Boosting

What It Does:

Builds models sequentially, where each new model corrects the mistakes made by the previous ones.

Use Case:

Great for reducing bias and building powerful models from weak learners.

Example:

- AdaBoost adjusts the weights of misclassified samples.
- Gradient Boosting uses gradients to minimise the loss function.

4. Stacking

What It Does:

Trains multiple base models and then uses a meta-model to combine their outputs into a final prediction.

Use Case:

Takes advantage of the strengths of different base models by training a second-level model to make the final prediction.

Example:

Used Logistic Regression, SVM, and Decision Tree as base models, and Gaussian Naive Bayes as the meta-model.

Controlling Overfitting

While ensemble methods are generally good at handling overfitting, the following help even more:

- Bagging reduces overfitting by averaging over many models trained on varied data subsets.
- Boosting uses weighted training and error corrections for more focused learning.
- Stacking benefits from a well-tuned meta-model.

Conclusion

In this assignment, we successfully implemented and compared several Ensemble Learning methods:

- Voting, Bagging, Boosting, and Stacking Each method brought unique strengths in terms of stability, accuracy, and robustness.

Key Takeaways:

- Ensemble learning improves model performance by leveraging the strengths of multiple learners.
- Voting is simple and useful for combining equally strong classifiers.
- Bagging reduces variance and is effective against overfitting.
- Boosting increases model accuracy by focusing on hard-to-learn instances.
- Stacking uses a layered approach to learn from model predictions.

Overall, Ensemble Learning proved to be a powerful technique for classification problems.