Stacking Ensemble Learning Implementation using K-Fold Blending

1. Introduction

Ensemble learning is a machine learning technique where multiple models are combined to improve overall prediction performance. Instead of relying on a single classifier, ensemble methods leverage the strengths of multiple models to achieve better generalization and reduce errors. In this project, a Stacking Ensemble Learning model was implemented. Stacking combines predictions from different base models and trains a second-level model (meta-learner) to make the final prediction.

2. Objective

The objectives of this task are:

- Implement a stacking ensemble using at least three diverse base classifiers.
- Use K-Fold cross-validated blending (Out-of-Fold predictions) to generate meta-features.
- Compare ensemble performance against individual learners using Accuracy and F1-score.
- Visualize results using an accuracy plot and confusion matrix.
- Save the best-performing model for future use.

3. Dataset

The Breast Cancer Wisconsin dataset (available in scikit-learn) was used for this classification task. It contains 569 samples with 30 numerical features computed from digitized images of breast mass. The target variable is binary: 0 (Malignant) and 1 (Benign).

4. Technologies Used

- Python 3.x
- scikit-learn
- Pandas
- NumPy
- Matplotlib
- Joblib

5. Models Used

5.1 Base Learners

1. Logistic Regression (Linear Model) – fast and interpretable baseline.
2. Random Forest (Tree-Based Ensemble) – captures non-linear patterns and reduces variance.
3. Support Vector Machine (SVM with RBF Kernel) – effective for complex non-linear decision boundaries.

5.2 Meta Learner

Logistic Regression was used as the meta-classifier as it performs well on probability-based meta-features and remains stable across folds.

6. Methodology

6.1 Individual Model Training & Evaluation

Each base model was trained independently on the training split and evaluated on the test split. Performance was measured using Accuracy and F1-score.

6.2 Stacking Ensemble with K-Fold Blending

To avoid data leakage in stacking, K-Fold blending was used to create meta-features using Out-of-Fold (OOF) predictions.

Steps followed:

- Split training data into K folds using StratifiedKFold.
- For each fold, train each base learner on K-1 folds and predict on the held-out fold to get OOF predictions.
- Store OOF predictions from all base models as meta-features for the training set.
- Generate meta-features for the test set by averaging predictions across folds.
- Train the meta-learner on training meta-features and predict final labels on test meta-features.

7. Performance Metrics

7.1 Accuracy

Accuracy is the proportion of correct predictions among all predictions.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

7.2 F1 Score

F1-score is the harmonic mean of Precision and Recall and is useful for balanced evaluation.

F1 = 2 × (Precision × Recall) / (Precision + Recall)

8. Results

The following table shows performance comparison between individual models and the stacking ensemble. Metrics are based on the test set results from the submitted notebook run.

Model	Accuracy	F1 Score
Logistic Regression	0.9824	0.9861
SVM	0.9824	0.9861
Stacking Ensemble	0.9736	0.9790
Random Forest	0.9473	0.9583

In this run, Logistic Regression and SVM achieved the best results. Stacking performed slightly lower but correctly demonstrated stacking with cross-validated blending.

9. Confusion Matrix Analysis

A confusion matrix was plotted for the best-performing model to understand classification performance. The matrix indicates the number of correct and incorrect predictions for each class.

10. Visualizations

- Accuracy comparison bar plot (Matplotlib).
- Confusion matrix visualization for the best model.

11. Model Saving

The best-performing model was saved using Joblib so that it can be reused without re-training. Saved file example: best_stacking_model.pkl (if stacking best) or best_model.pkl (if a single model best).

12. Conclusion

This project successfully implemented a stacking ensemble learning pipeline using three diverse base learners and a logistic regression meta-learner. K-Fold blending with Out-of-Fold predictions was used to generate reliable meta-features without leakage. The ensemble was compared against individual models using Accuracy and F1-score along with visual outputs.

13. Future Improvements

- Add more base learners such as Gradient Boosting or XGBoost.
- Perform hyperparameter tuning for base learners and meta learner.
- Include additional evaluation metrics like ROC-AUC.
- Test the stacking approach on larger real-world datasets.