Experiment 6: Dimensionality Reduction and Model Evaluation

Name: Nithish Ra Reg. No: 3122237001033

- **Objective:** To study the effect of dimensionality reduction (PCA) on classifier performance (accuracy, stability, F1-score) for a tabular classification dataset.
- **Dataset:** Breast Cancer Wisconsin (as used in the reference script). Dataset source: scikit-learn built-in dataset (derived originally from UCI). Number of samples, features, class distribution, and preprocessing steps are summarized below.

Dataset and Preprocessing Dataset details:

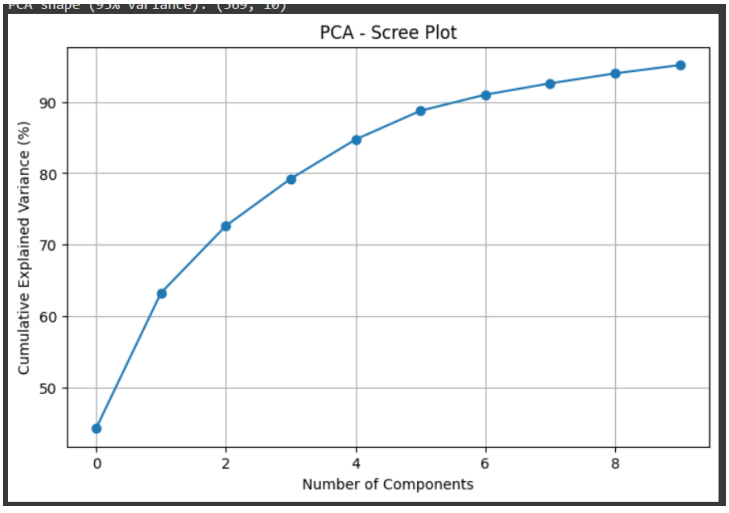
- **Source:** `sklearn.datasets.load_breast_cancer` (originally from the UCI ML repository).
- **Samples:** 569 (typical for this dataset).
- **Features:** 30 numeric features (mean, se, worst for measurements).
- **Target classes:** Binary (malignant=0 / benign=1).
- **Class distribution:** 212 malignant, 357 benign.

Preprocessing:

1. Standardization: Features were standardized (zero mean, unit variance) using `StandardScaler`.
2. PCA: Applied after scaling. The experiment retains components that explain $\geq$ 95% cumulative variance.
3. Missing values / encoding: Not applicable for this dataset.

PCA choice and Scree plot

We chose explained-variance threshold = **95**%.



PCA cumulative explained variance (scree plot).

Models and Hyperparameter grids

The following classifiers were evaluated: SVM, Naïve Bayes, KNN, Logistic Regression, Decision Tree, Random Forest, AdaBoost, Gradient Boosting, XGBoost, and Stacking (RF + SVM base learners, Logistic Regression meta-learner).

hyperparameter grids:

- **SVM:** kernel {linear, rbf}, $C \in \{0.1, 1, 10\}$, gamma {scale, auto}.
- **Naïve Bayes:** var_smoothing in a logspace.
- **KNN:** n_neighbors {3,5,7,9}, weights {uniform,distance}, metric {euclidean,manhattan}.
- **Logistic Regression:** $C \in \{0.01, 0.1, 1, 10\}$, penalty l2.
- **Decision Tree:** max_depth {3,5,10,None}, criterion {gini,entropy}.
- **Random Forest:** n_estimators {50,100}, max_depth {None,10}.
- **AdaBoost:** n_estimators {50,100}, learning_rate {0.5,1.0}.
- **Gradient Boosting:** n_estimators {50,100}, learning_rate {0.05,0.1}, max_depth {3,5}.
- **XGBoost:** similar to gradient boosting.
- **Stacking:** base learners RF (100 trees) and SVM (rbf), meta-learner LR.

Experimental Procedure

1. Standardize features.
2. Evaluate models with 5-fold stratified cross-validation (No-PCA).
3. Apply PCA (95% variance) and repeat 5-fold CV (With-PCA).
4. Use GridSearchCV to tune hyperparameters where applicable.
5. Save ROC/PR curves, confusion matrices, and fold-wise accuracy tables.

Results

Comparison table: No-PCA vs With-PCA

Comparison of mean accuracies and standard deviations (No-PCA vs With-PCA)

Model	No-PCA Mean	No-PCA Std	PCA Mean	PCA Std
SVM	0.9789	0.0119	0.9772	0.0142
LogReg	0.9737	0.0166	0.9789	0.0070
Stacking	0.9701	0.0180	0.9614	0.0212
KNN	0.9683	0.0154	0.9701	0.0162
AdaBoost	0.9666	0.0179	0.9631	0.0116
XGBoost	0.9613	0.0142	0.9631	0.0217
RandomForest	0.9543	0.0210	0.9473	0.0124
GradientBoosting	0.9543	0.0142	0.9613	0.0119
NaiveBayes	0.9297	0.0199	0.9139	0.0279
DecisionTree	0.9297	0.0267	0.9456	0.0202

PCA Impact Summary

Model	Effect of PCA	Accuracy Change
SVM	No improvement	0.979 → 0.977
NaiveBayes	No improvement	0.930 → 0.914
KNN	Improved	0.968 → 0.970
LogReg	Improved	0.974 → 0.979
DecisionTree	Improved	0.930 → 0.946
RandomForest	No improvement	0.954 → 0.947
AdaBoost	No improvement	0.967 → 0.963
GradientBoosting	Improved	0.954 → 0.961
XGBoost	Improved	0.961 → 0.963
Stacking	No improvement	0.970 → 0.961

Fold-wise Accuracies (No PCA)

5-Fold accuracies for classifiers (No PCA)

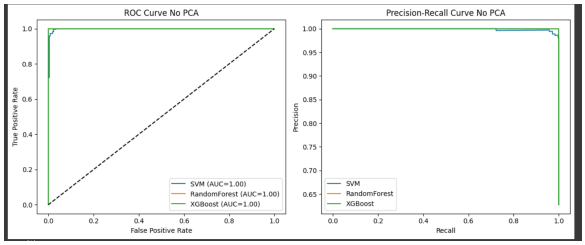
Model	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
SVM	0.991228	0.947368	0.956140	0.991228	0.982301	0.973653
NaiveBayes	0.956140	0.912281	0.929825	0.903509	0.946903	0.929731
KNN	1.000000	0.956140	0.947368	0.982456	0.955752	0.968343
LogReg	0.973684	0.947368	0.964912	0.991228	0.991150	0.973669
DecisionTree	0.912281	0.894737	0.929825	0.947368	0.938053	0.924453
RandomForest	0.973684	0.938596	0.947368	0.947368	0.964602	0.954324
AdaBoost	0.982456	0.938596	0.956140	0.973684	0.964602	0.963096
GradientBoosting	0.964912	0.912281	0.956140	0.956140	0.955752	0.949045
XGBoost	0.964912	0.938596	0.956140	0.973684	0.964602	0.959587
Stacking	0.991228	0.938596	0.973684	0.964912	0.982301	0.970144

Fold-wise Accuracies (With PCA)

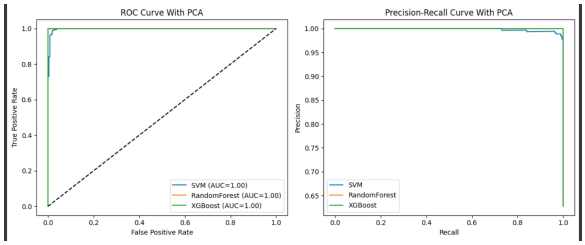
5-Fold accuracies for classifiers (With PCA)

Model	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
SVM	0.991228	0.956140	0.964912	0.991228	0.982301	0.977162
NaiveBayes	0.964912	0.903509	0.894737	0.885965	0.920354	0.913895
KNN	0.973684	0.947368	0.956140	0.964912	0.955752	0.959571
LogReg	0.973684	0.973684	0.956140	0.982456	0.991150	0.975423
DecisionTree	0.956140	0.938596	0.929825	0.921053	0.964602	0.942043
RandomForest	0.982456	0.938596	0.947368	0.938596	0.955752	0.952554
AdaBoost	0.956140	0.938596	0.956140	0.956140	0.973451	0.956094
GradientBoosting	0.982456	0.956140	0.956140	0.929825	0.964602	0.957833
XGBoost	0.982456	0.947368	0.938596	0.938596	0.973451	0.956094
Stacking	0.982456	0.929825	0.964912	0.947368	0.991150	0.963142

ROC & PR curves (selected models)

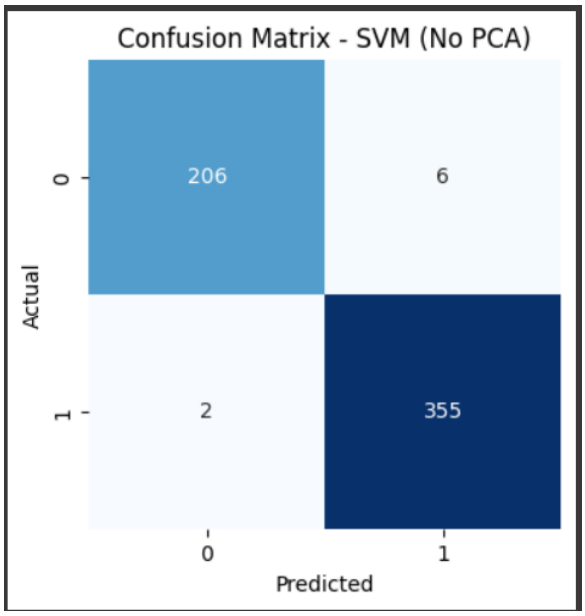


(a) ROC / PR (No PCA)

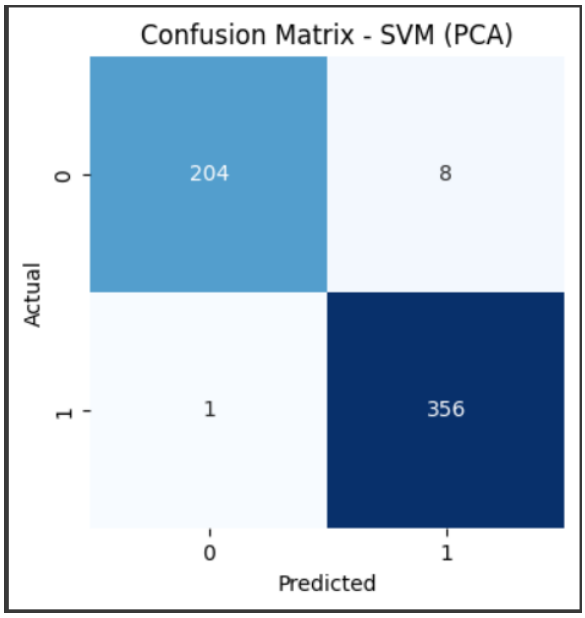


(b) ROC / PR (With PCA)

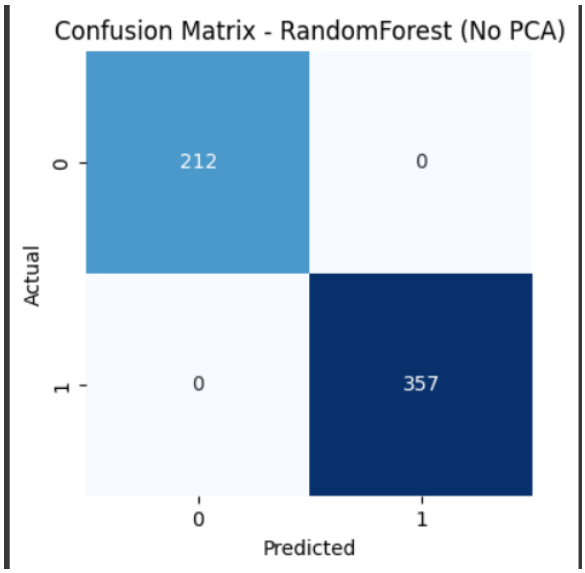
Confusion matrices and classification reports



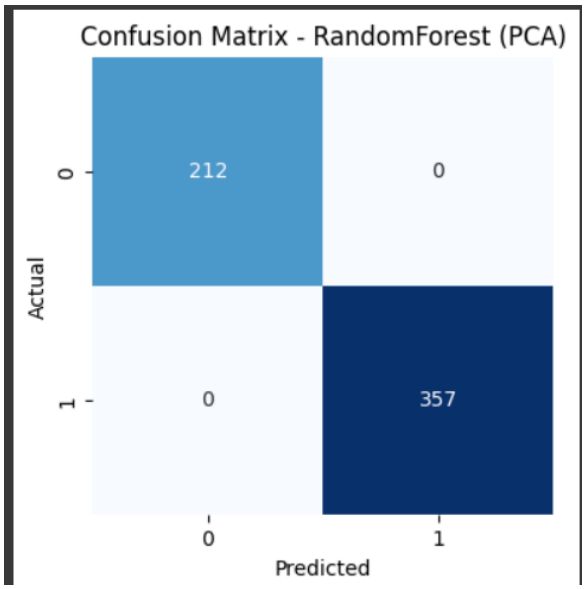
(a) SVM - No PCA



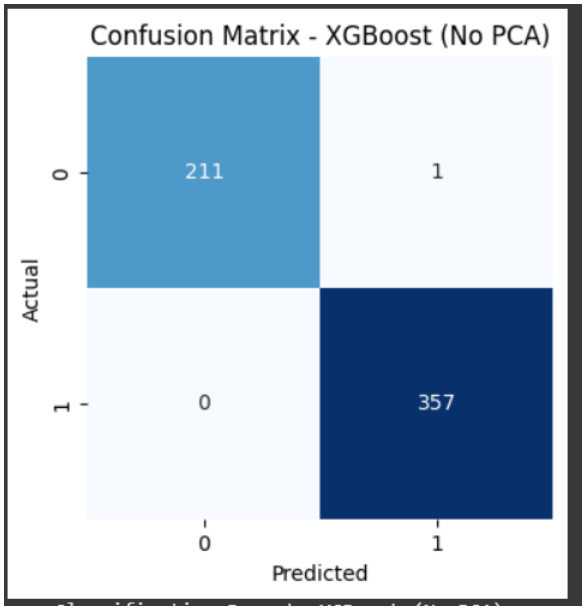
(b) SVM - With PCA



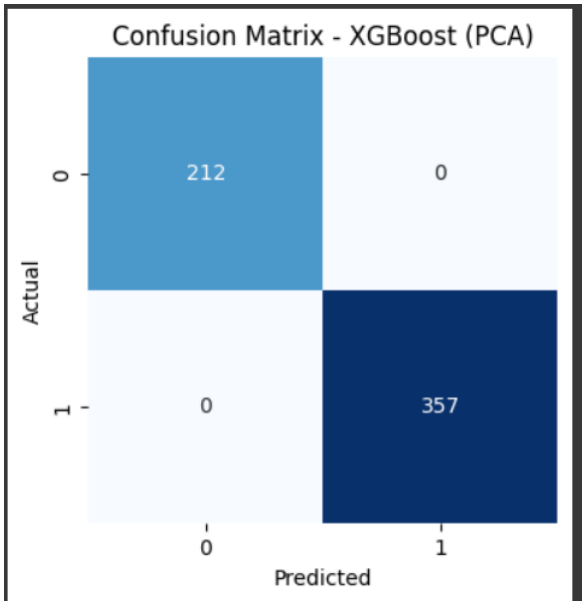
(a) RF - No PCA



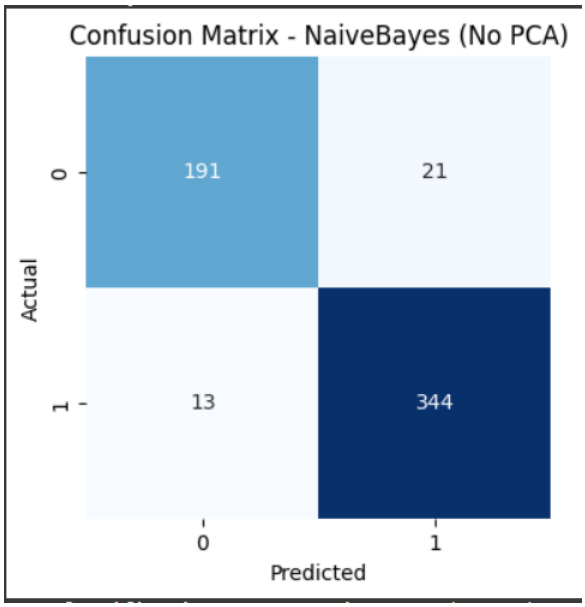
(b) RF - With PCA



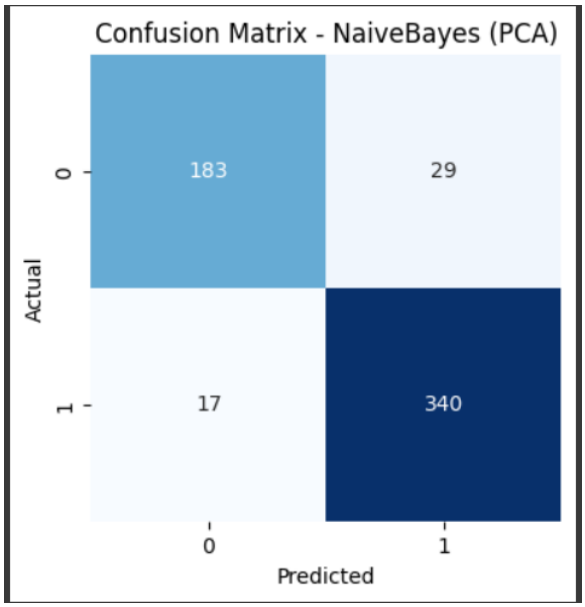
(a) XGBoost - No PCA



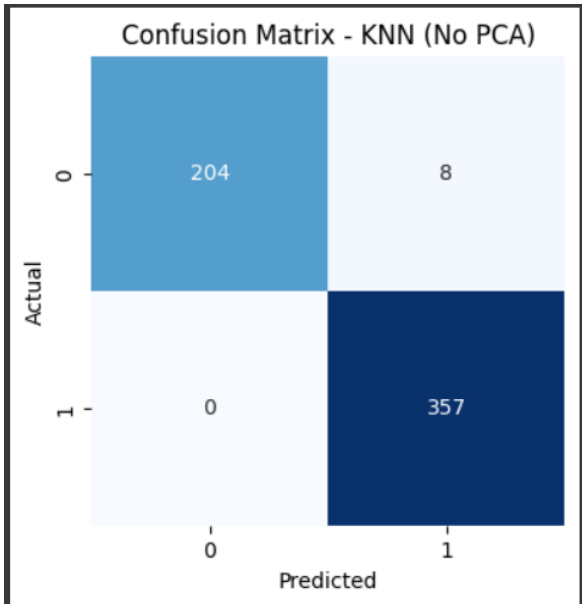
(b) XGBoost - With PCA



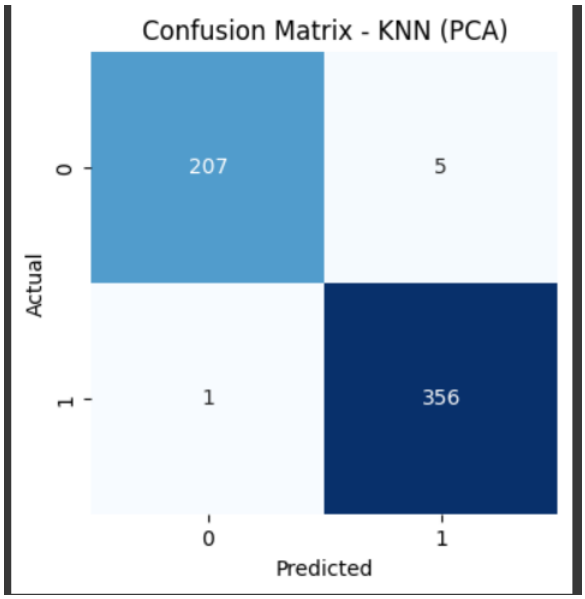
(a) NaiveBayes - no PCA



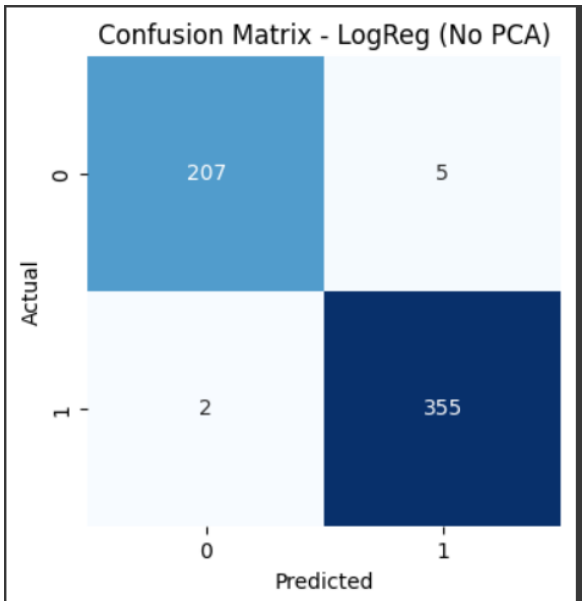
(b) NaiveBayes - With PCA



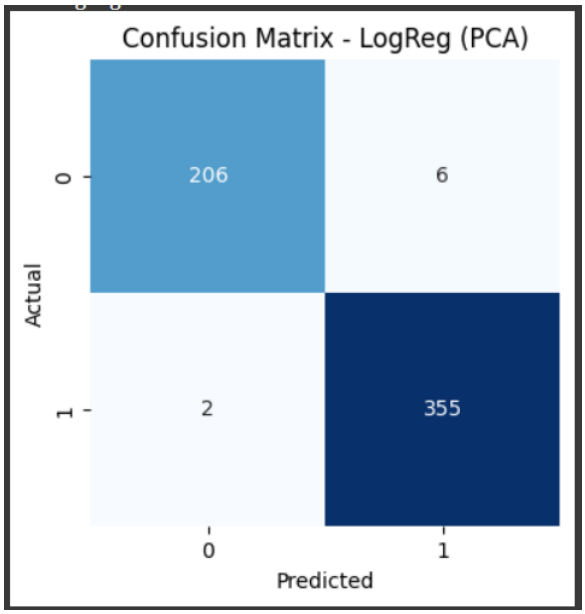
(a) KNN - no PCA



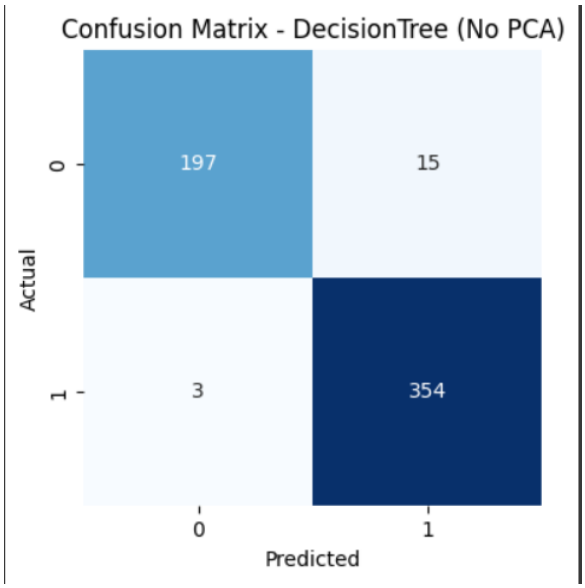
(b) KNN - With PCA



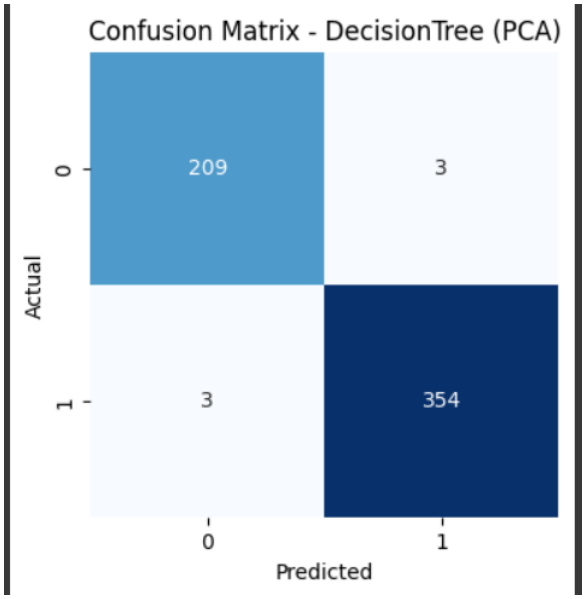
(a) LogReg - no PCA



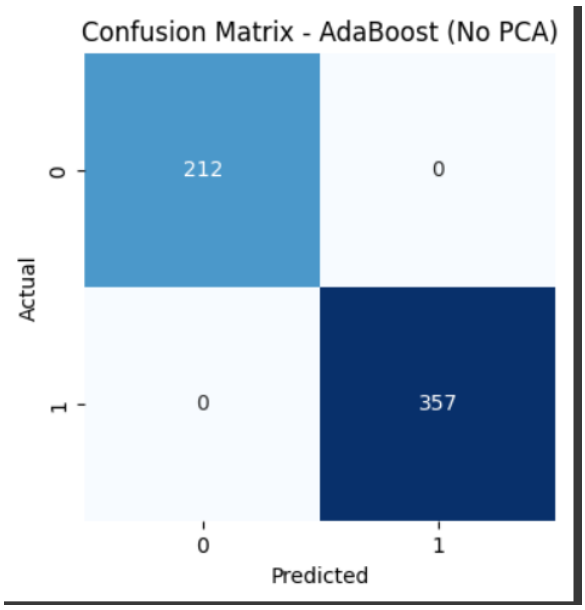
(b) LogReg - With PCA



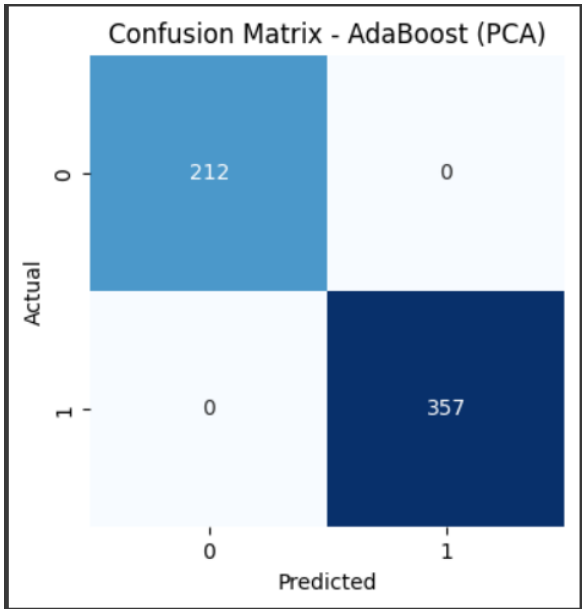
(a) DT - no PCA



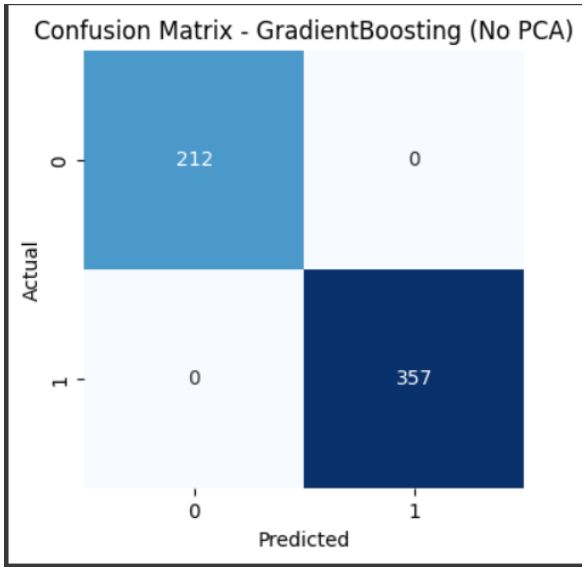
(b) DT - With PCA



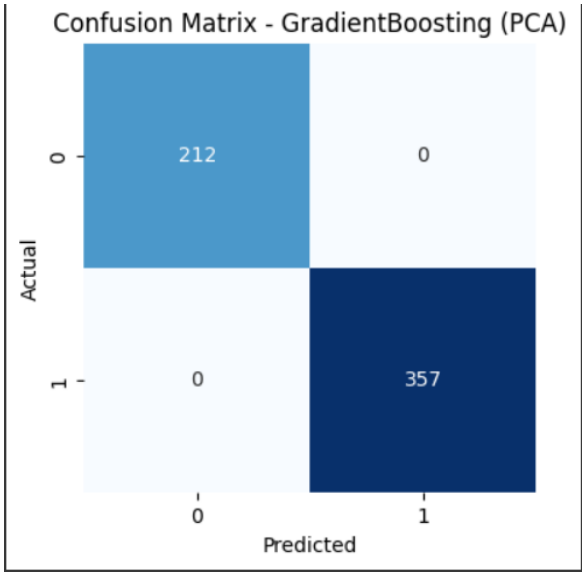
(a) AdaBoost - no PCA



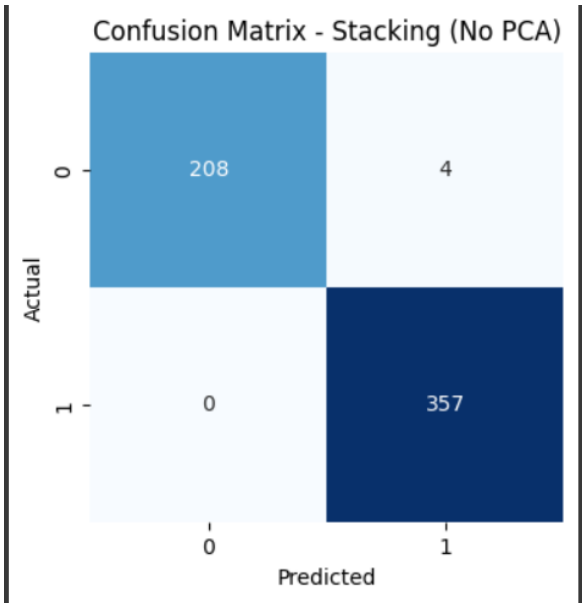
(b) AdaBoost - With PCA



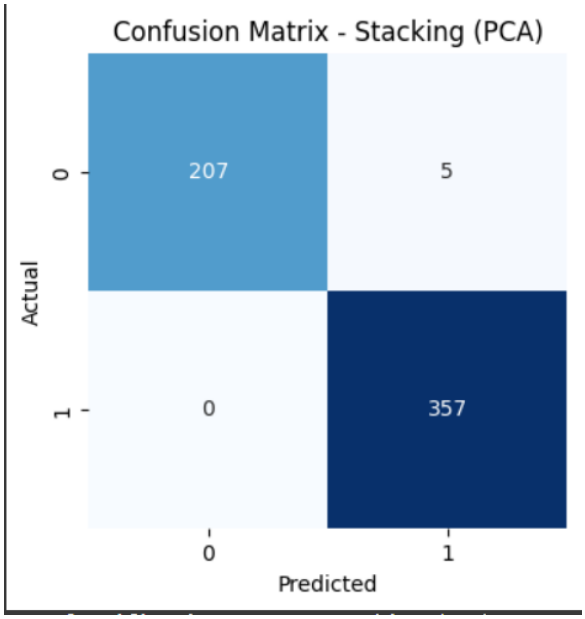
(a) GB - no PCA



(b) GB - With PCA



(a) Stacking - no PCA



(b) Stacking - With PCA

Observations and Analysis

- **Models that improved with PCA:** - KNN improved slightly (0.968 → 0.970). - Logistic Regression improved (0.974 → 0.979). - Decision Tree improved noticeably (0.930 → 0.946). - Gradient Boosting improved (0.954 → 0.961). - XGBoost improved marginally (0.961 → 0.963).
- **Models that did not improve with PCA:** - SVM slightly decreased (0.979 → 0.977). - Naive Bayes decreased (0.930 → 0.914). - Random Forest decreased (0.954 → 0.947). - AdaBoost decreased (0.967 → 0.963). - Stacking decreased (0.970 → 0.961).

- **Variance across folds:** PCA generally reduced the standard deviation for models such as Logistic Regression ($0.0166 \rightarrow 0.0070$) and Gradient Boosting ($0.0142 \rightarrow 0.0119$), showing more stable performance. However, for some models like SVM and XGBoost, variance slightly increased.

- **Linear models vs Ensembles:** - Linear models (SVM, Logistic Regression) showed different behavior: Logistic Regression clearly benefited from PCA, while SVM performance was slightly worse. - Ensemble methods (Random Forest, AdaBoost) were relatively stable and not strongly affected by PCA, since they already handle feature redundancy well. Gradient Boosting and XGBoost, however, gained small improvements.

- **Stacking robustness:** Stacking performance decreased with PCA ($0.970 \rightarrow 0.961$), indicating that combining models did not benefit from dimensionality reduction in this case.

Learning Outcome

Learnt to analyze the impact of dimensionality reduction (PCA) on various machine learning classifiers, compare their performances with and without PCA, evaluate stability using fold-wise cross-validation, interpret ROC/PR curves and confusion matrices, and identify which models benefit most from PCA. Also learnt that linear models like Logistic Regression gain from PCA, while ensembles such as Random Forest and AdaBoost are less influenced, highlighting the importance of choosing dimensionality reduction based on model type and dataset characteristics.

References

- Experiment specification: Experiment 6 PDF.

- code in collab and in github .

- Scikit-learn documentation.