Faculty of Engineering
Cairo University

Machine learning

Team 7

Name	SEC	BN	Workload
كريم علاء عبد الله	2	3	Cleaning and visualization (steps 1 - 4)
أحمد أسامة فوزي زهران	1	6	Models training and evaluation (steps 8 - 10)
محمود أسامة محمود خطاب	2	14	Report and statistics analysis (steps 11 - 14)
محمد عادل محمد عز الدين	2	9	Feature selection and generalization (steps 5 - 7)

Under Supervision of:
Eng. Mohamed Shawky

Problem Definition

Develop a machine learning model that classifies tumors as malignant or benign using clinical and imaging features.

Problem Motivation

Early and accurate detection of breast cancer is critical for effective treatment, reducing mortality, and improving patient outcomes.

Evaluation Metrics

Accuracy, Precision, Recall, F1 Score, and ROC-AUC.

Dataset Link

Kaggle Breast Cancer Dataset

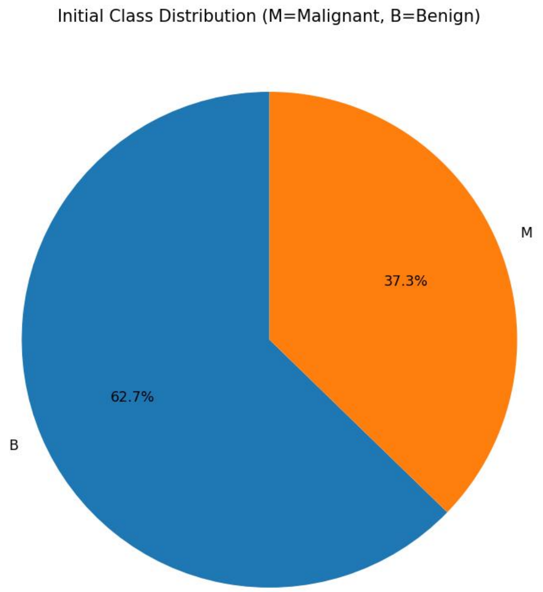
Models used

- Logistic regression
- SGD classifier
- Ensemble
- SVM
- Perceptron
- Gradient boosting
- Bagging
- Adaboost

Project Pipeline

Data loading

To kickstart our process, we acquire the dataset necessary for predicting breast cancer. Our approach accommodates both local and remote data sources, such as files and URLs, ensuring versatility in data acquisition. Upon successful loading, we examine the dataset's structure, including dimensions and critical identifiers like column titles. Visual analysis aids us in understanding class distributions (malignant vs. benign), allowing us to set the stage for further exploration and validation. The class distributions were as follows:



The pie chart clearly indicates a sort of imbalance towards the presence of breast cancer.

Data cleaning

We prioritize data integrity before diving into analysis, necessitating rigorous cleaning protocols.

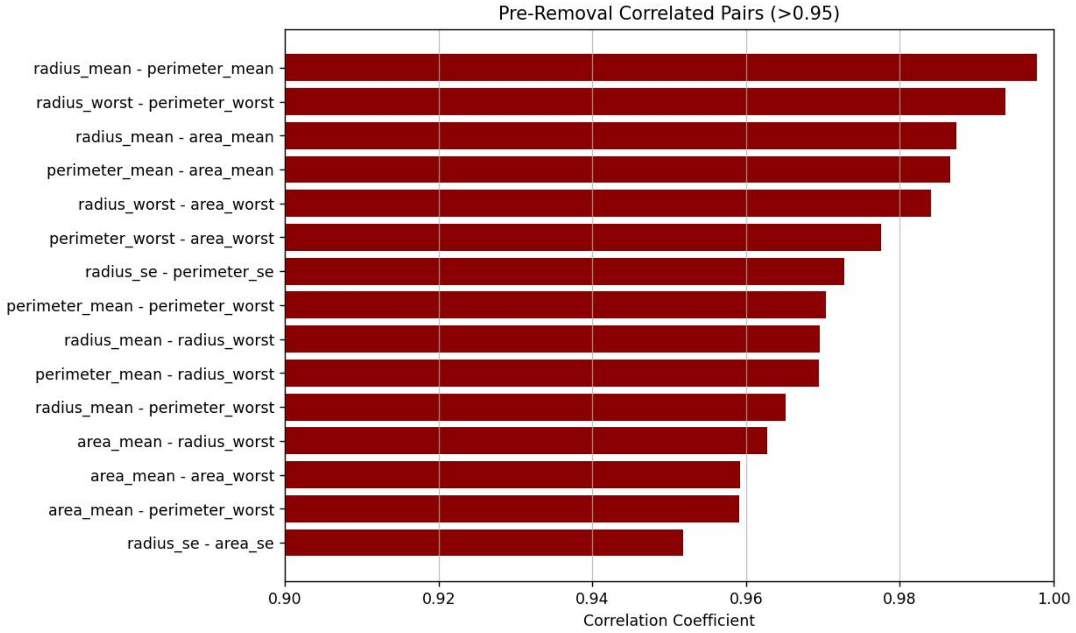
- By removing irrelevant identifiers (**id**), we streamline the dataset.
- Our conversion of target labels into numerical formats prepares them for binary classification tasks.
- Moreover, we ensure that all input features are numeric.
- We address missing values with mean imputation, bolstering the dataset against stochastic variance influences.
- By identifying and removing duplicate entries, we uphold analytic integrity, keeping insights reliable.

Feature selection

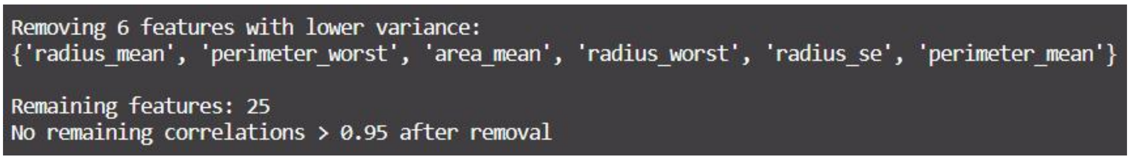
Our feature selection process is designed to simplify model complexity while preserving or enhancing informational relevance.

- We use correlation-based filtering to remove excessively co-linear features, enhancing computational performance without dimming prediction capabilities.
- Additionally, **SelectKBest** helps us isolate the most statistically meaningful features, utilizing an ANOVA F-value strategy to pinpoint features with strong class differentiation. This methodological pruning means our subsequent models interact efficiently, unhindered by redundant data.

The figure illustrates the most correlated pairs in our data:



Filtering the features that have the least variance among the ones in the above figure, we are left with 25 features:



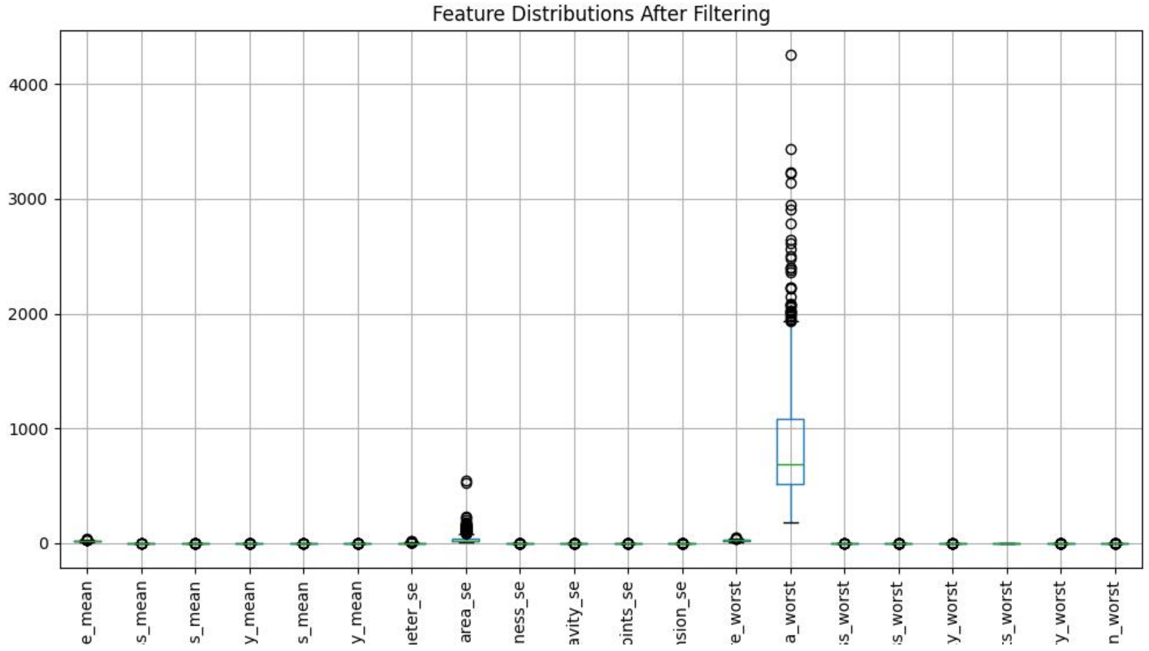
```
Removing 6 features with lower variance:
{'radius_mean', 'perimeter_worst', 'area_mean', 'radius_worst', 'radius_se', 'perimeter_mean'}

Remaining features: 25
No remaining correlations > 0.95 after removal
```

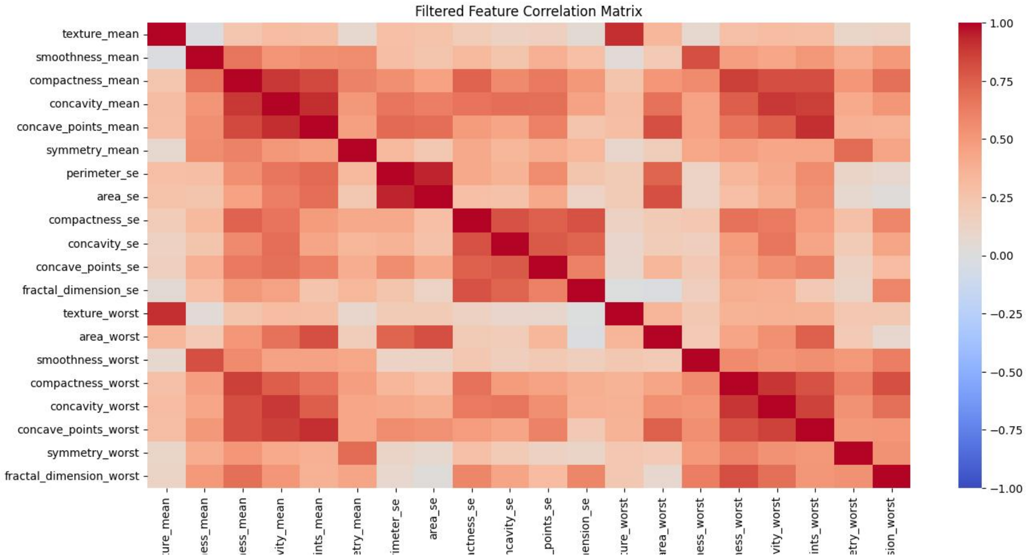
Data visualization

We view visualization as a crucial interpretation pillar, transforming raw numeric insights into visual narratives that are easier to digest. Through techniques such as box plots and heatmaps, we provide intuitive displays of feature distributions and inter-feature correlations. These graphics enable us and stakeholders to discover trends, potential outliers, and dependencies, completing the comprehensive understanding required before deeper modelling efforts.

The box plot for the features' data distribution is as follows:



As it stands till this point, non-diagonal correlations now are less than the threshold(95%):



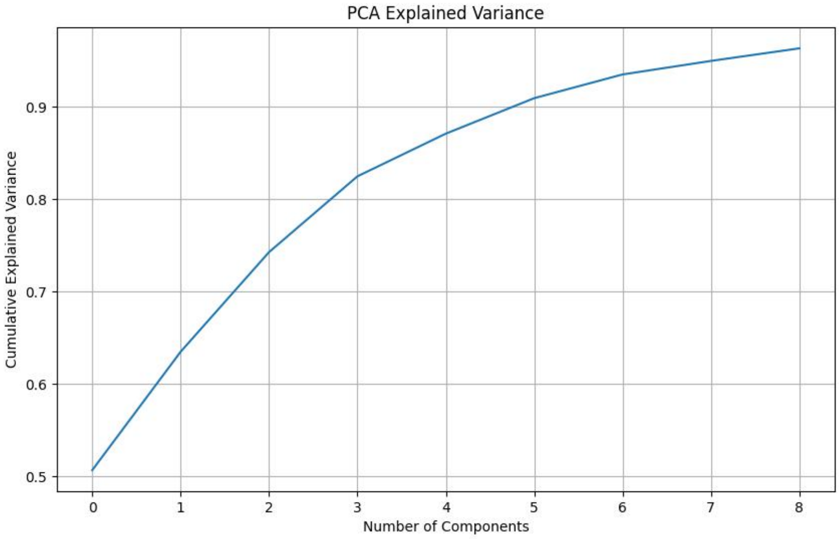
Data splitting

Preparation for model training involves splitting the dataset into training, validation, and testing segments. Our use of stratified sampling ensures each split maintains the original class distribution, reducing sampling bias and preserving predictive fidelity. This structured division forms a bedrock for predictive success and ensures consistent model comparison. The test data represents 20% of the total dataset, and the validation set represents 25% of the training set.

Preprocessing

Feature preprocessing optimizes data inputs for model consumption, utilizing transformations like standard scaling to equalize feature influence within gradient-based algorithms. Additionally, Principal Component Analysis (PCA) compresses data dimensions, addressing computational demands while safeguarding essential variance. Graphical elucidation of explained variance reinforces stakeholder confidence in our component selection logic.

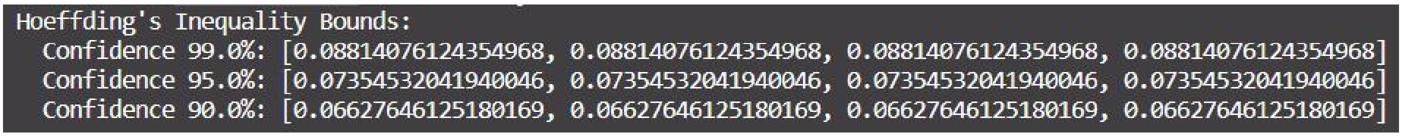
A variance of 95% was chosen for our project, leading to taking 9 principal components:



Generalization Bounds Analysis

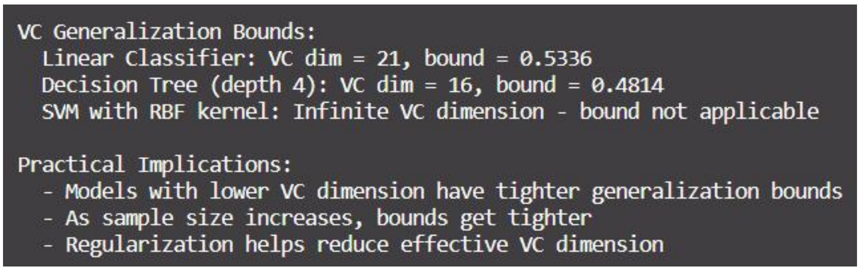
Theoretical exploration of generalization bounds offers insurance against analytic overconfidence. We examine predictive reliability through mathematical constructs like Hoeffding's inequality and VC dimension paradigms, providing bounds for model performance relative to sample complexities. This analysis alerts us to potential risks, guiding model adjustments that align realizable outcomes with conceptual expectations.

For 3 different confidence levels, the difference between the Ein and Eout should be:



```
Hoeffding's Inequality Bounds:
  Confidence 99.0%: [0.08814076124354968, 0.08814076124354968, 0.08814076124354968, 0.08814076124354968]
  Confidence 95.0%: [0.07354532041940046, 0.07354532041940046, 0.07354532041940046, 0.07354532041940046]
  Confidence 90.0%: [0.06627646125180169, 0.06627646125180169, 0.06627646125180169, 0.06627646125180169]
```

For the different model types in our project, the VC generalization bounds are:



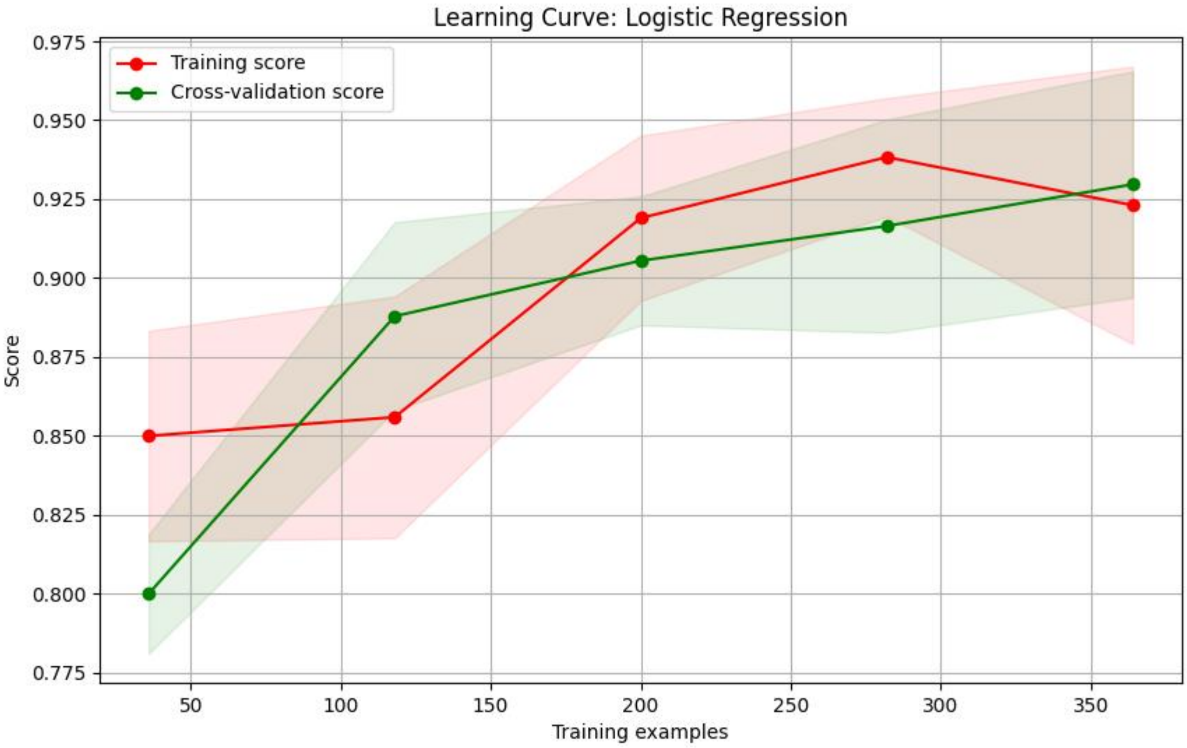
```
VC Generalization Bounds:
  Linear Classifier: VC dim = 21, bound = 0.5336
  Decision Tree (depth 4): VC dim = 16, bound = 0.4814
  SVM with RBF kernel: Infinite VC dimension - bound not applicable

Practical Implications:
  - Models with lower VC dimension have tighter generalization bounds
  - As sample size increases, bounds get tighter
  - Regularization helps reduce effective VC dimension
```

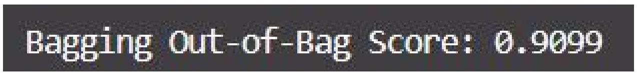
Model training

Model training is an exhaustive exercise in configuration and behaviour optimization. Various algorithms receive dedicated tuning, employing advanced techniques such as grid search to refine parameters through iterative learning and validation. Visualizing learning curves allows us to dissect bias and variance, drawing attention to potential adjustments designed to yield stronger predictive frameworks.

Taking an example for a learning curve, the following represents the logistic regression's with an increasing number of samples:



For the bagging model, an out-of-bag error is calculated:



```
Bagging Out-of-Bag Score: 0.9099
```

Ensemble training

This function constructs an ensemble model from the list of successful models, excluding the Perceptron.

Key Steps:

Model Selection: Identify models with successful training completion and exclude non-desirable ones (Perceptron in this context).

Voting Classifier Creation: Aggregate predictions using a VotingClassifier with soft voting to leverage probability estimates, enhancing accuracy by combining model strengths.

Fit and Analyze: Fit the ensemble with training data and assess individual models' contributions via accuracy metrics, ensuring the ensemble complements predictive precision with aggregate merit.

Model Evaluation

Evaluates the performance of both individual models and ensemble models using a test dataset to ensure robust predictive capabilities.

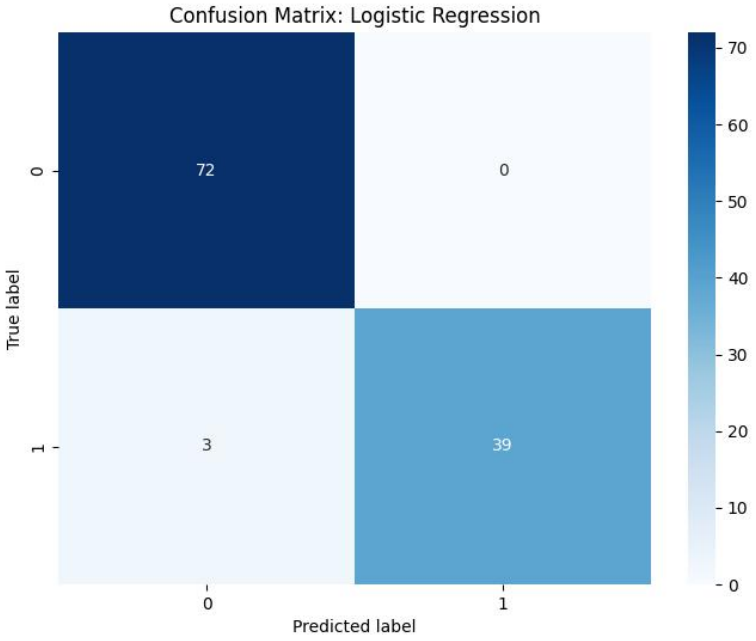
Key Steps:

Metric Collection: Calculate key performance metrics like accuracy, precision, recall, F1-score, ROC AUC, and MCC. These metrics quantify the model's ability to correctly classify instances.

Confusion Matrix Visualization: Visualize confusion matrices for individual models, highlighting error types and distribution patterns.

Ensemble Evaluation: Specifically check the ensemble's decision outputs, ensuring consensus aligns with better prediction outcomes showcased in its performance matrix.

For instance, the confusion matrix for logistic regression is as follows:



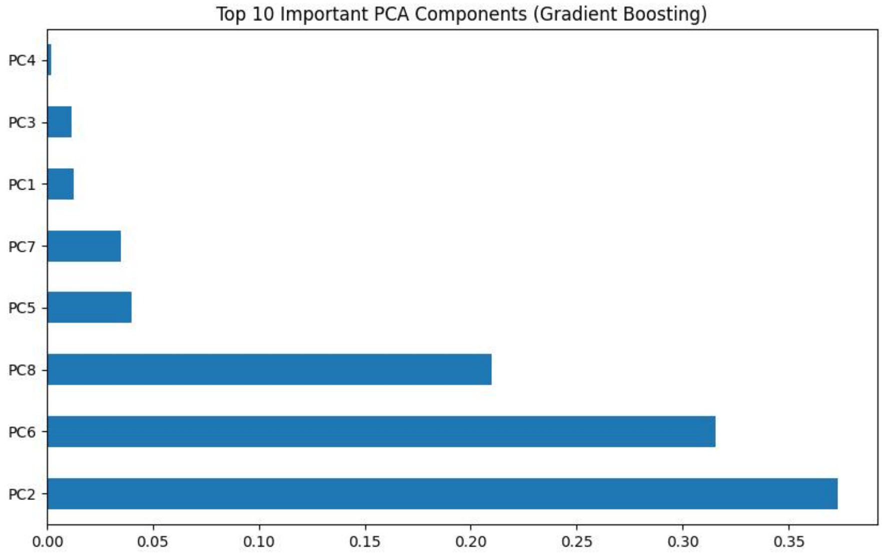
Feature importance analysis

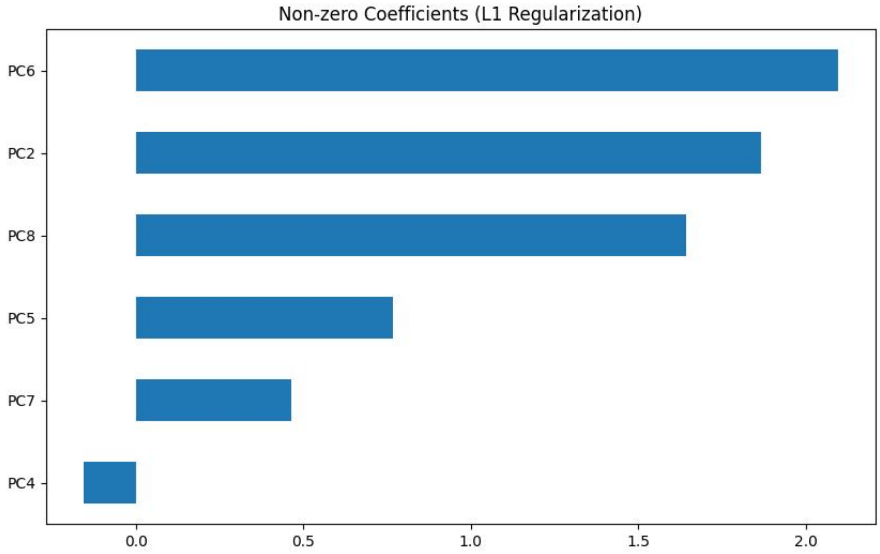
This function dissects feature importance in context with PCA components, primarily using Gradient Boosting and Logistic Regression models.

Key Steps:

Gradient Boosting Analysis: Calculate and plot feature importances derived from PCA components to capture significant model drivers in dimensional space.

Logistic Regression Feature Selection: Evaluate L1 regularization impact within logistic regression, isolating and visualizing non-zero coefficients to emphasize feature selection and influence.





Feature reduction reporting

Provides an assessment of feature reduction achieved during preprocessing, reflecting dimensionality simplification efficiencies.

Key Steps:

Quantitative Reduction Summary: Enumerates the transformation from initial features down to reduced sets post-PCA and selection phases, offering insight into transformation efficacy while retaining relevance.

Logistic regression interpretation

Evaluates logistic regression coefficients and regularization effects, facilitating interpretive insights into the model's decision mechanics.

Key Steps:

Coefficients Examination: Examine Model coefficients, visualize their relative impacts, and assert the model's path in predictive implications.

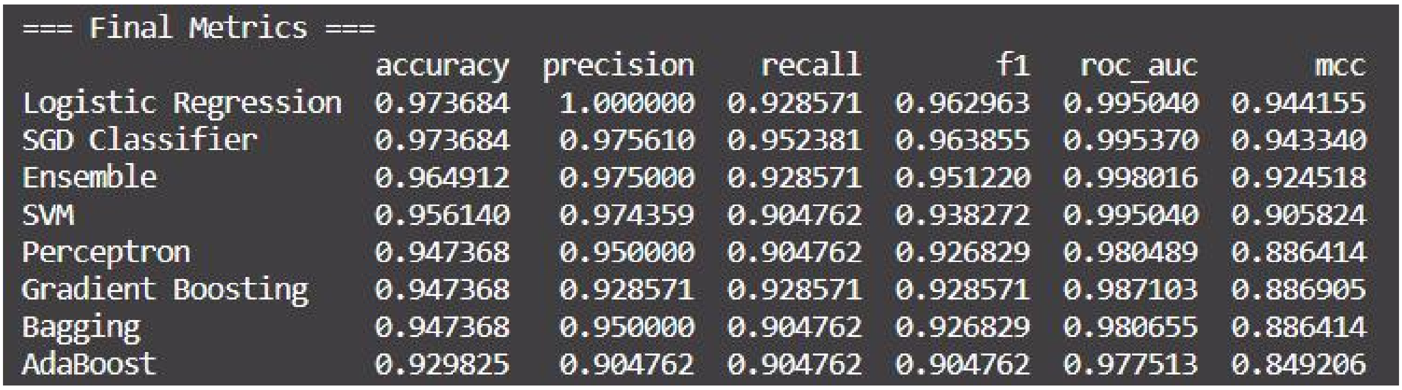
Regularization Exploration: Delve into regularization insights by calculating L1 and L2 norms, correlating regularization strength (C parameter) to model behaviour fine-tuning.

Performance comparison

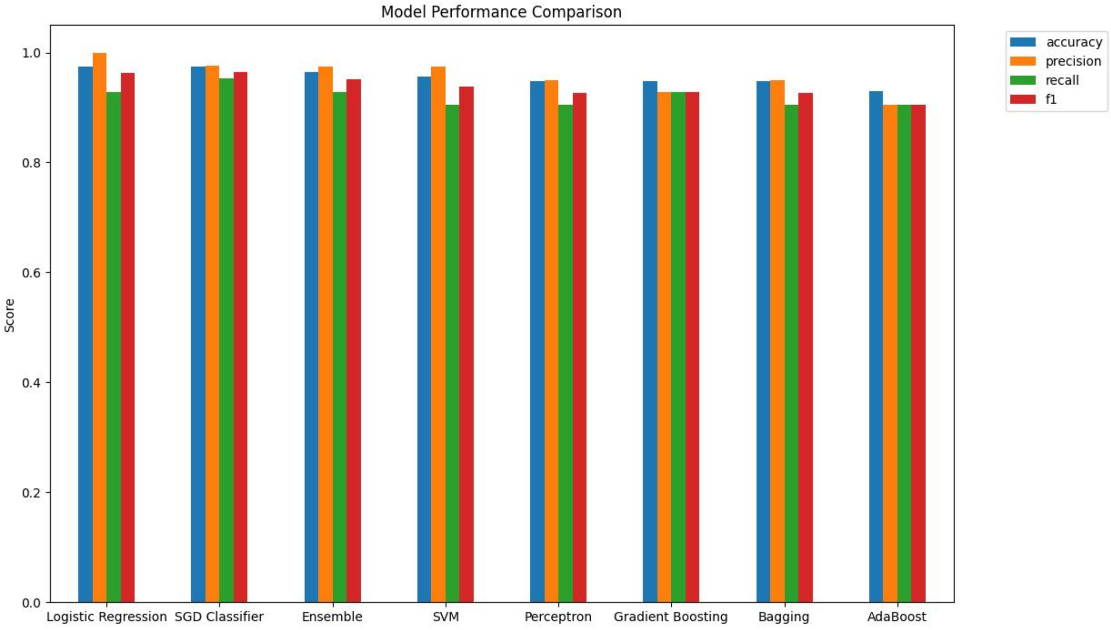
Delivers comparative insights across models concerning evaluated performance metrics, positioning stakeholder focus on promising candidates.

Key Steps:

Metric Visualization: Use bar plots to position models based on accuracy, precision, recall, and F1-score, forming a clear hierarchy of efficacy ensuring optimal understanding of model competencies.



```
=== Final Metrics ===
                     accuracy  precision    recall        f1   roc_auc       mcc
Logistic Regression  0.973684   1.000000  0.928571  0.962963  0.995040  0.944155
SGD Classifier       0.973684   0.975610  0.952381  0.963855  0.995370  0.943340
Ensemble             0.964912   0.975000  0.928571  0.951220  0.998016  0.924518
SVM                  0.956140   0.974359  0.904762  0.938272  0.995040  0.905824
Perceptron           0.947368   0.950000  0.904762  0.926829  0.980489  0.886414
Gradient Boosting    0.947368   0.928571  0.928571  0.928571  0.987103  0.886905
Bagging              0.947368   0.950000  0.904762  0.926829  0.980655  0.886414
AdaBoost             0.929825   0.904762  0.904762  0.904762  0.977513  0.849206
```



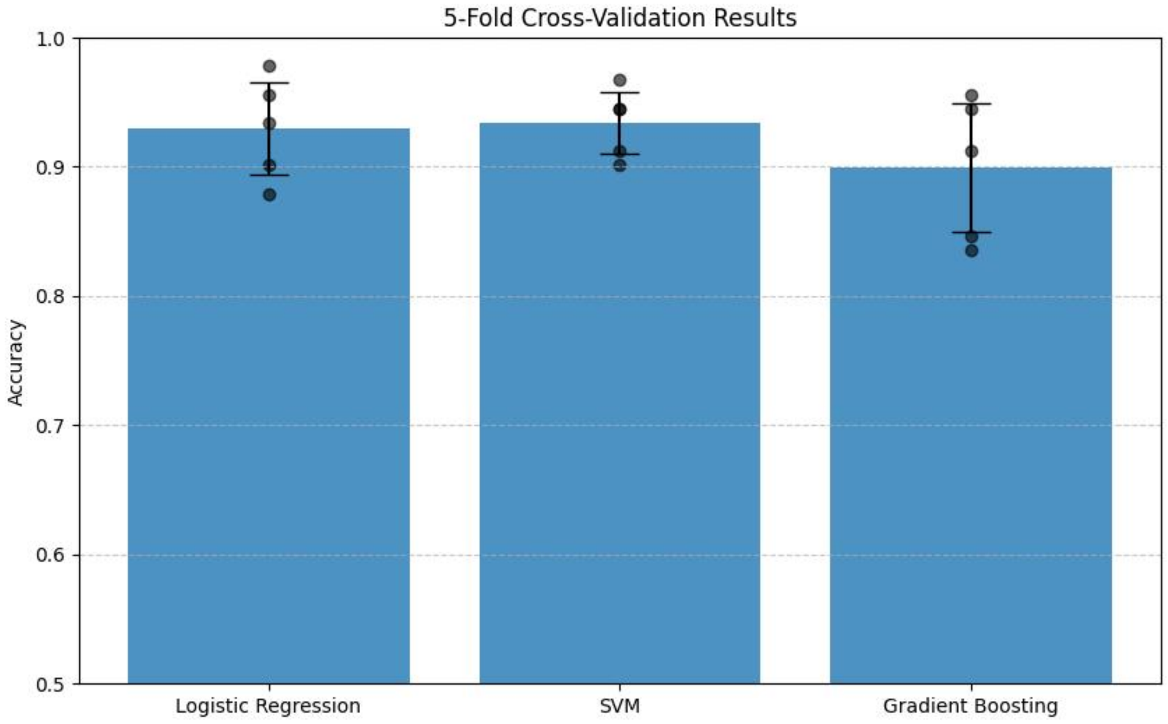
Cross-Validation analysis

Executes cross-validation analysis to authenticate top-performing models, ensuring consistency in outcomes across data partitions.

Key Steps:

K-fold Cross Validation Execution: Perform stratified K-fold validation on top models, providing averaged metrics scores and deviations to outline reliability.

Scatter Visualization: Utilize scatter plots to show individual fold scores, accentuating consistency and revealing variability trends across model rounds.



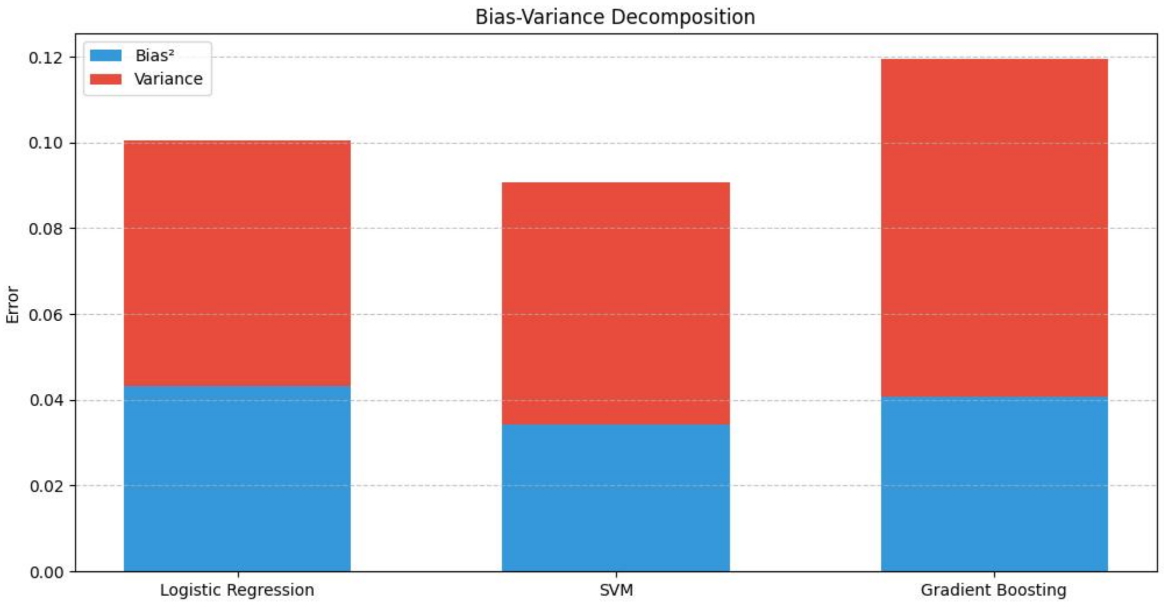
Bias-variance analysis

Analyzes bias-variance dynamics, yielding insights into how models balance complexity against predictability within bootstrap-sampled tests.

Key Steps:

Bootstrap Sampling: Implement bootstrap sampling techniques to predict bias and variance aspects from model outcomes.

Bias and Variance Decomposition: Calculate bias, variance, and irreducible errors, representing these within stacked bar charts to highlight manageable and inherent model error profiles.



Conclusion

- While it could've been argued that the ensemble would give the highest accuracy, the data set was small enough that complexities won't make that much of a difference.

- The logistic regression benefited a lot from its L2 regularization(ridge), which prevented over-fitting as much as possible.

- All models exceeded the baseline of 62.7% (Zero R base model), which was just the frequency of the highest class(B) over the total.