Appendix A: Detailed Machine Learning Models for Mortgage Prepayment Risk Assessment

1. Logistic Regression (LR)

Objective Function: The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable) and a set of independent (predictor or explanatory) variables. Logistic regression does this by estimating probabilities using a logistic function, which is a cumulative logistic distribution.

$$L(\beta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

Here, $L(\beta)$is the logistic loss function, N is the number of observations, y_i is the actual outcome, and p_i is the predicted probability of the outcome being 1 for the ith observation. The coefficients β are estimated during the training process.

Hyperparameters:

- **Regularization type** (L1, L2, ElasticNet): Determines the type of regularization applied to the model to prevent overfitting by penalizing large coefficients.
- **Regularization strength**: Controls the magnitude of the regularization term. A larger value specifies stronger regularization.
- **Solver**: The algorithm used for optimization (e.g., **liblinear**, **sag**, **saga**, **newton-cg**). Different solvers are suitable for different types of data and different regularization methods.

2. Linear Discriminant Analysis (LDA)

Objective Function: LDA aims to model differences between classes by assuming that different classes generate data based on different Gaussian distributions. The objective is to find a linear combination of features that separates two or more classes of objects or events. The resulting combination may be used as a linear classifier or for dimensionality reduction before classification.

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

Here, $J(w)$ is the objective function to be maximized, w is the vector of linear coefficients, S_B is the between-class scatter matrix, and S_W is the within-class scatter matrix.

Hyperparameters:

- **Solver**: Algorithm used for computing the LDA. Options include **svd**, **lsqr**, and **eigen**.
- **Shrinkage**: Regularization parameter to improve classification accuracy, especially in situations where the number of predictors is relatively high compared to the number of observations.

3. K-Nearest Neighbors (KNN)

KNN does not have an explicit objective function in the traditional sense as it is a non-parametric method. It classifies new cases based on a similarity measure (e.g., distance functions). Essentially, an object is

classified by a majority vote of its neighbors, with the object being assigned to the class most common among its K nearest neighbors.

Hyperparameters:

- **Number of neighbors (k)**: Determines how many neighbors will vote on the classification. A smaller K makes the model more sensitive to noise in the training data.
- **Distance metric**: The method used to calculate the distance between points. Common metrics include Euclidean, Manhattan, and Minkowski distances.

Note: KNN itself does not have a formula for learning from the training data. It classifies each test instance based on its *K* nearest neighbors determined by the distance metric.

These models, Logistic Regression, Linear Discriminant Analysis, and K-Nearest Neighbors, represent foundational techniques in machine learning for classification and prediction tasks. Each has its unique strengths and considerations, making them suitable for various scenarios in credit risk assessment.

4. Decision Trees

Objective Function: Decision trees aim to model decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is a predictive modeling approach used in statistics, data mining, and machine learning. It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves).

Splitting Criteria (Gini Impurity):

Decision trees make decisions by splitting nodes into sub-nodes. The Gini impurity measure is often used for classification tasks to choose the best split. It represents the probability of a randomly chosen sample being incorrectly classified if it was randomly labeled according to the distribution of labels in the subset.

$$G = 1 - \sum_{i=1}^{k} p_i^2$$

Here, G is the Gini impurity, $p_i$is the proportion of samples that belong to class i at a given node, and K is the number of classes.

Hyperparameters:

- **Max depth**: The maximum depth of the tree. Deeper trees can capture more complex patterns but may lead to overfitting.
- **Min samples split**: The minimum number of samples required to split an internal node. Higher values prevent creating nodes that contain too few samples.
- **Min samples leaf**: The minimum number of samples required to be at a leaf node. Similar to **min_samples_split**, this parameter helps to control over-fitting.

5. Multilayer Perceptron (MLP)

Objective Function: MLPs, a class of feedforward artificial neural networks, aim to approximate some function $f(*)$by considering a set of inputs and a known set of outputs. MLPs adjust weights based on the

error rate produced in the previous epoch (iterative process), essentially minimizing the difference between the actual and predicted outputs across all observations, typically using backpropagation.

Loss Function (Cross-Entropy Loss for Classification): For classification tasks, MLPs often use the cross-entropy loss function, which measures the performance of a classification model whose output is a probability value between 0 and 1.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{M} y_{ic}\log(p_{ic})$$

In this formula, L denotes the loss, N is the number of samples, M is the number of classes, $y_ic$is a binary indicator of whether class c is the correct classification for observation i, and p_ic is the predicted probability that observation i is of class c.

Hyperparameters:

- **Number of layers and number of neurons per layer**: Determines the depth and width of the MLP. More layers and neurons can model more complex functions but increase the risk of overfitting.
- **Activation function**: Non-linear functions like ReLU, sigmoid, or tanh that help the network learn complex patterns in the data.
- **Learning rate**: Determines the step size at each iteration while moving toward a minimum of the loss function. Too large can overshoot the minimum; too small can result in a long convergence.
- **Regularization** (e.g., L1, L2): Techniques to prevent overfitting by penalizing large weights.

6. Random Forest (RF)

Objective Function: Random Forest aims to reduce overfitting in decision trees by averaging multiple decision trees' predictions, trained on different parts of the same training set, with the goal of improving the overall accuracy. Random Forest algorithm does not have a single formula due to its ensemble nature, but it operates by building multiple decision trees and merging their predictions. The decision of the majority of trees is chosen by the random forest as the final prediction.

Hyperparameters:

Number of trees: The number of trees in the forest. More trees increase prediction stability but also computational complexity.

Max depth: The maximum depth of the trees.

Min samples split: The minimum number of samples required to split an internal node.

Min samples leaf: The minimum number of samples required to be at a leaf node.

7. Gradient Boosting Machines (GBM)

Objective Function: GBM aims to minimize a loss function by sequentially adding weak learners using a gradient descent algorithm. Each new model incrementally decreases the loss function (e.g., mean squared error for regression tasks) of the entire system.

$$F_m(x) = F_{m-1}(x) + \rho_m h_m(x)$$

Here, $F_m(x)$ is the boosted model's prediction at iteration m, $F_{m-1}(x)$ is the prediction from the previous iteration, $h_m(x)$ is the weak learner added at iteration m, and ρ_m is the learning rate.

Hyperparameters:

- **Learning rate**: Determines how corrections are made to the model with each added tree.
- **Number of learners**: The total number of trees to be built.
- **Max depth of trees**: The depth limit for each tree, controlling overfitting.

8. LightGBM

Objective Function: Similar to GBM, LightGBM also focuses on minimizing a loss function but does so more efficiently for large datasets by using a gradient-based one-side sampling and exclusive feature bundling.

Hyperparameters:

- **Number of leaves**: The maximum number of leaves in one tree.
- **Learning rate**: Speed of model learning.
- **Min data in leaf**: The minimum number of the records a leaf may have.
- **Feature fraction**: The fraction of features to be used for each tree, preventing overfitting.
- **Reg_alpha:** L1 regularization adds a penalty term to the objective function that encourages the model to minimize the absolute values of feature weights, promoting feature selection by driving some weights to zero.

9. XGBoost

Objective Function: XGBoost aims to minimize a regularized loss function that includes both the loss term and a regularization term, which helps in controlling overfitting more effectively than traditional GBM.

$$F_m(x) = F_{m-1}(x) + \rho_m h_m(x) + \lambda\Omega(h_m)$$

In this formula, $\Omega(h_m)$ represents the regularization term applied to the model h_m, adding a penalty for complexity to improve model generalization.

Hyperparameters:

- **Learning rate (eta)**: Determines the step size at each iteration to prevent overfitting.
- **Max depth**: Maximum depth of a tree, increasing this value will make the model more complex and more likely to overfit.
- **Subsample**: The fraction of samples to be used for fitting the individual base learners.

- **Colsample_bytree**: The fraction of features to be used for each tree.
- **Reg_alpha:** L1 regularization term on weights. Increasing this value will make model more conservative

10. SVM

Support Vector Machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. The objective function is as below:

$$\min_{w,b} \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i$$

Subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \quad \forall i$$

$$\xi_i \geq 0 \quad \forall i$$

Where:

- w, is the weight vector perpendicular to the hyperplane.
- b is the bias term, affecting the position of the hyperplane.
- C is the regularization parameter, which balances the margin maximization and loss.
- ξi are the slack variables that allow for the misclassification of difficult examples.
- *yi* are the class labels.
- **x***i* are the feature vectors.

Hyperparameters:

- **Kernel:** The main function of the kernel is to transform the given dataset input data into the required form. There are various types of functions such as linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.
- **Regularization (C):** Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty.
- **Gamma:** Kernel coefficient for 'rbf', 'poly', and 'sigmoid'. If gamma='scale' (default) is passed then it uses 1 / (n_features * X.var()) as value of gamma. If 'auto', uses 1 / n_features.

The tables below include both the default and final tuned parameters for machine learning models used in our study, as well as the default parameters for models where no tuning was performed.

XGBoost

Parameter	Default Value	Final/Tuned Value
n_estimators	100	200
learning_rate	0.3	0.1
max_depth	6	10
min_child_weight	1	10
colsample_bytree	1	0.7
reg_alpha	0	1

LightGBM (LGBM)

Parameter	Default Value	Final/Tuned Value
n_estimators	100	300
learning_rate	0.1	0.1
num_leaves	31	100
max_depth	-1 (no limit)	15
reg_alpha	0	1

Gradient Boosting Machines (GBM)

Parameter	Default Value	Final/Tuned Value
n_estimators	100	300
learning_rate	0.1	0.1
max_depth	3	7

Random Forest

Parameter	Default Value	Final/Tuned Value
n_estimators	100	300
max_depth	None (unlimited)	20
min_samples_split	2	5

Linear Discriminant Analysis (LDA)

Parameter	Default Value	Final/Tuned Value
Solver	svd	svd
Shrinkage	None (not applicable to `svd`)	None

Logistic Regression

Parameter	Default Value
Regularization type	L2 (Ridge)
Regularization strength (C)	1.0
Solver	lbfgs

K-Nearest Neighbors (KNN)

Parameter	Default Value
Number of neighbors	5
Distance metric	Minkowski with p=2

Decision Trees

Parameter	Default Value
Max depth	None (unlimited)
Min samples split	2
Min samples leaf	1

Multilayer Perceptron (MLP)

Parameter	Default Value
Number of layers	1 hidden layer
Number of neurons per layer	100
Activation function	ReLU
Learning rate	Constant at 0.001
Regularization (Alpha)	0.0001

Support Vector Machines (LinearSVC)

Kernel	Linear
Regularization (C)	1.0
Loss	Square Hinge
Penalty	L2

Appendix B: Performance Metrics Definitions

In this appendix, we provide detailed descriptions and the mathematical formulations of the performance metrics used to evaluate the models in our study. These metrics are essential for understanding the effectiveness of different classification algorithms in predicting mortgage prepayment risks. All metrics are derived from the confusion matrix, which includes True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The formulas for each metric offer a clear method for assessing the models' performance and ensuring the reproducibility of our results.

1. **Accuracy**: This is the proportion of true results (both true positives and true negatives) among the total number of cases examined.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Area Under the Curve (AUC)**: AUC represents the area under the ROC curve. This metric measures the ability of the classifier to discriminate between classes.

$$\text{AUC} = \int_0^1 \text{TPR}(t)\ dt$$

Here, TPR(t) is the true positive rate at threshold t.

3. **Recall (Sensitivity)**: This measures the proportion of actual positives that are correctly identified.

$$\text{Recall} = \frac{TP}{TP + FN}$$

4. **Precision**: This measures the proportion of positive identifications that were actually correct.

$$\text{Precision} = \frac{TP}{TP + FP}$$

5. **F1 Score**: The F1 Score is the harmonic mean of precision and recall.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

6. **Cohen's Kappa**: This measures the agreement between predicted and actual classifications while accounting for agreement that occurs by chance.

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

p_o: Observed agreement (accuracy)

p_e: Expected agreement based on chance

7. **Matthews Correlation Coefficient (MCC)**: This metric provides a balanced measure of classification quality, even for imbalanced datasets.

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

MCC produces a value between -1 and +1, where +1 indicates perfect classification, 0 indicates random prediction, and -1 indicates complete disagreement.

Appendix C. Variable definition

Variable	Definition
ORIG_RATE	The original interest rate on a mortgage loan as identified in the original mortgage note
ORIG_UPB	The dollar amount of the loan as stated on the note at the time the loan was originated
OLTV	Loan amount divided by the value of property at origination
OCLTV	The amount of all known outstanding loans (including home equity) at origination divided by the value of property
DTI	Loan amount divided by borrower income at origination
MIN_FICO	The minimum FICO score among all borrowers associated with the mortgage loan at origination.
NEWPURPOSE_CASH_OUT	Indicator variables for whether the loan is a cash-out refinance or not
NEWPURPOSE_NON_CASH_OUT	Indicator variables for whether the loan is a non cash-out (rate) refinance or not
NEWPURPOSE_PURCHASE	Indicator variables for whether the loan is a home purchase or not
FIRST_FLAG	An indicator that denotes if the borrower or co-borrower qualifies as a first-time homebuyer
NUM_BO	The number of individuals obligated to repay the mortgage loan
NO_UNITS	The number of housing units associated with the property securing the mortgage loan
Mortgage_Insurance	Indicator variables for whether the loan has mortgage insurance or not
MI_PCT	The percentage of the loan amount covered by mortgage insurance at origination
OCC_STAT_PRINCIPAL	An indicator that denotes whether the property occupancy status is for primary residence or not
OCC_STAT_INVESTMENT	An indicator that denotes whether the property occupancy status is for investment purpose or not
OCC_STAT_SECONDARY	An indicator that denotes whether the property occupancy status is for secondary home or not
CHANNEL_CORRESPONDENT	Indicator variables for whether the loan is originated through the correspondent channel or not
CHANNEL_RETAIL	Indicator variables for whether the loan is originated through the retail channel or not
CHANNEL_BROKER	Indicator variables for whether the loan is originated through the broker channel or not
SATO	The spread at origination, calculated as the difference between the mortgage interest rate and the market rate at the time of loan origination. |
CSCORE_B	The primary borrower's credit score (FICO) at the time of loan origination.
PMMS30	The average 30-year fixed mortgage rate from the Freddie Mac Primary Mortgage Market Survey (PMMS).
PREPAY_12 (24/36)	Indicator variables for whether the loan is prepaid within 12 (24/36) months after origination
FINTECH	Indicator variable for whether the loan is originated by a fintech lender.
NONFINTECH	Indicator variable for whether the loan is originated by a nonbank, nonfintech lender.
BANK	Indicator variable for whether the loan is originated by a traditional bank lender.