MSA-Management science journal
ISSN 2974-3036
Volume: 4, Issue:3, Year: 2025 pp.96-113

Feature Selection and Classification in Machine Learning: Methods and Models for Peer-to-Peer Lending

Markus Atef
Faculty of Management Sciences, October University for Modern Sciences and Arts (MSA), Giza, Egypt
Department of Information Systems, Faculty of Commerce and Business Administration, Helwan University, Cairo, Egypt
Email: markatef@msa.edu.eg

Menna Ibrahim Gabr
Department of Information Systems, Faculty of Commerce and Business Administration, Helwan University, Cairo, Egypt

Wafaa Seoud
Department of Business Administration, Faculty of Commerce and Business Administration, Helwan University, Cairo, Egypt

Shimaa Ouf
Department of Information Systems, Faculty of Commerce and Business Administration, Helwan University, Cairo, Egypt

Abstract: Feature Selection (FS) is a pivotal technique in machine learning (ML) that improves predictive performance, model interpretability, and computational efficiency by reducing data dimensionality and isolating the most informative variables. In the dynamic environment of peer-to-peer (P2P) lending, FS is crucial for accurate credit risk assessment, borrower profiling, and loan default prediction. P2P platforms generate vast and heterogeneous datasets encompassing demographic, financial, behavioural, and transactional information, where redundant or irrelevant features can degrade model accuracy and scalability. This review provides a comprehensive examination of FS methodologies, including filter, wrapper, and embedded approaches, analysing their trade-offs in accuracy potential, computational cost, and interpretability. The study further explores classification models, supervised learning algorithms designed to predict borrower repayment behaviour, covering linear, non-linear, and tree-based ensembles widely applied in credit scoring. Classification methods address critical challenges in P2P lending, such as class imbalance, explainability, and the need for scalable, high-performing predictive systems. By synthesizing recent advances and practical applications, this review offers a structured guide for researchers and practitioners to select FS techniques and classification models aligned with P2P lending's requirements. Emphasis is placed on optimizing predictive accuracy, enhancing interpretability, and supporting data-driven decision-making to strengthen credit evaluation processes, mitigate default risk, and promote sustainable growth across P2P lending platforms.

Keywords: Feature selection, machine learning, classification models, peer-to-peer lending, credit risk prediction, loan default analysis

1 Introduction

The process of selecting a subset of the most relevant features from a wider set of candidate features is known as feature selection (FS), and it is an essential stage in the machine learning (ML) pipeline. In addition to improving the interpretability of machine learning models, FS increases computational efficiency and accuracy by eliminating redundant, unnecessary or noisy input. The significance of FS has increased as machine learning applications spread throughout industries, including biology, banking and artificial intelligence. Various FS approaches are being developed and improved to handle different data complexities (Dhal, et al. 2022; Xie, et al. 2023).

Feature selection techniques can be broadly classified into three categories: filter methods, wrapper methods, and embedded methods. Filter methods evaluate features independently of any learning algorithm, often using statistical measures such as correlation or mutual information to rank features (Chandrashekar & Sahin, 2014). Despite their computational efficiency, some approaches might not take feature interactions into consideration. Wrapper methods, on the other hand, evaluate subsets of features by training a machine learning model using those features and selecting the subset that optimizes model performance. While wrapper methods typically result in better performance, they are computationally expensive (Patil et al., 2024; Kozodoi et al., 2019). Embedded methods combine the strengths of both approaches by performing feature selection during the model training process itself, often through regularization techniques such as Lasso or decision tree-based models (Guyon & Elisseeff, 2003). These techniques guarantee that feature selection is in line with the model-building procedure, making them ideal for high-dimensional datasets and complex models.

FS is important since it increases model efficiency and interpretability in addition to accuracy. By reducing the number of irrelevant features, FS can lessen the curse of dimensionality, where high-dimensional data leads to overfitting and poor generalization (Theng et al., 2024). Furthermore, FS aids in removing redundant features and noise, which might impair the model's performance. Feature selection is crucial for determining the most useful variables and guaranteeing that the model is both effective and reliable in real-world applications where data may be noisy or lacking (Liu et al., 2024, Sadeghian et al., 2023). A feature selection method based on multiple feature subsets extraction and result fusion for improving classification performance. Furthermore, FS is essential for improving model interpretability, which is especially critical in delicate domains like healthcare or finance where comprehension of the rationale behind predictions is often as critical as the accuracy of the predictions themselves.

Machine learning models are algorithms or mathematical frameworks that enable computers to learn from data and make predictions or decisions without being explicitly programmed. They "learn" from historical data and use that knowledge to identify patterns, classify new data, or predict future outcomes. The three main categories of machine learning are reinforcement learning, unsupervised learning, and supervised learning. In supervised learning, the algorithm learns from labelled data to make predictions or classify new data points, with a focus on minimizing the error between the predicted and actual values. Unsupervised learning, in contrast, deals with unlabeled data, where the goal is to identify hidden patterns or structures within the data. Reinforcement learning is a type of learning where an agent interacts with an environment, taking actions and receiving feedback to maximize cumulative reward (Dunsin, et al. 2025).

The choice of learning model influences both the feature selection techniques employed and the complexity of the resulting models, as each category poses unique challenges in terms of data representation, labelling, and evaluation (Asnicar et al., 2024).

The current review explored various types of FS methods and their significance in improving model accuracy, computational efficiency, and generalization, while addressing their impact on feature relevance and interpretability. The wide range of machine learning approaches was also discussed. The significance of classification models was emphasized, particularly in fields where precise predictions are necessary. The article offers recent insights and scientific references on the integration of feature

selection (FS) in machine learning systems, presenting a comprehensive overview of its considerable potential across various fields. Moreover, this review emphasizes the pivotal role of FS and classification models in peer-to-peer (P2P) lending, where robust borrower profiling, accurate credit risk assessment, and loan default prediction are essential to maintaining platform sustainability and investor trust. By aligning FS techniques with the unique challenges of P2P lending, such as large, heterogeneous datasets, class imbalance, and the need for explainability, platforms can improve predictive accuracy, optimize decision-making, and unlock actionable insights that drive more secure and data-driven lending ecosystems.

2. Feature Selection

One of the first stages in creating a predictive model is reducing the number of input variables via a process known as feature selection. When using machine learning in the actual world, it is very uncommon for all the variables included within a dataset to be relevant in the process of developing a model. The model's capacity to generalize becomes more limited because of the addition of superfluous variables, which may also result in a decrease in the classifier's accuracy level. In addition, the total complexity of a model will rise as more and more variables are added to it. Consequently, feature selection has developed into an essential component of the process of developing machine learning models. Finding the optimal combination of features that enables one to construct accurate models of the processes being researched is the objective of the feature selection step of machine learning.
For instance, in a loan default prediction model, the features of borrowers are fed into a model that predicts whether a loan will go into default. Given D features, there are 2^D possible feature combinations. It takes an exponential amount of computer effort to test all the possible combinations; even D is a very modest number. It is a very time-consuming process that, under some conditions, may even be impossible to finish searching through all these potential combinations to discover the best feature subset. Fortunately, there are methods for feature selection that can help alleviate this situation by selecting a subset of features while maintaining the model performance (Guyon & Elisseeff, 2003; Kozodoi et al., 2019), and subsets of features are sufficient to finish the classification (Somol et al., 2005).

2. 1 Feature Selection (FS) Methods

It is possible to divide the FS methods into a variety of groups. As shown in Fig.1, FS is often broken down into three distinct categories: unsupervised, semi-supervised and supervised FS (Chandrashekar and Sahin 2014).

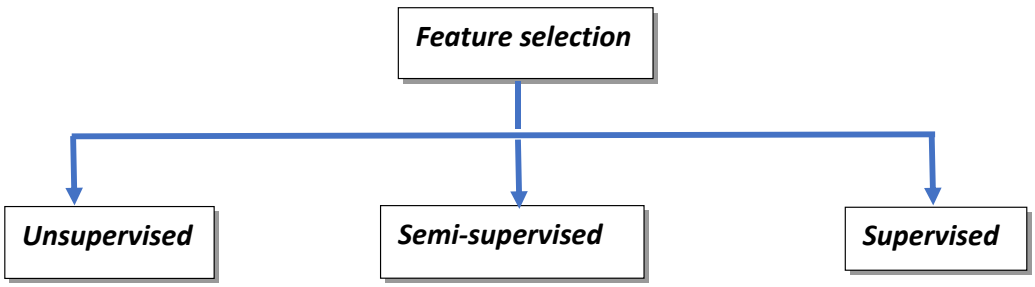


Fig.1 Different types of feature selection methods

2.1.1 Unsupervised FS Methods

When there are no known class labels but one still wants to pick a subset of the most relevant features with reference to predefined criteria, such as variance or correlation, the unsupervised FS is the feature selection method that is used. Clustering, which is a well-known example of unsupervised

learning, is one of the most prevalent applications of the unsupervised FS (Guyon & Elisseef, 2003). According to Dong and Liu (2018), the strategies for unsupervised feature selection may be broken down into three primary approaches:

• Filter techniques choose the features that are most relevant via the data itself. This means that features are assessed based on the intrinsic qualities of the data, as opposed to making use of any clustering approach that may direct the search for relevant features. The quickness of filter techniques and their ability to utilize are the primary characteristics of these approaches.

• Wrapper techniques conduct feature subset evaluations by applying the findings of a particular clustering algorithm to the data. Finding feature subsets that contribute to improving the quality of the results of the clustering algorithm used for the selection is a defining characteristic of the methodologies that are created within the context of this methodology. Wrapper techniques, on the other hand, often have a high computational cost and can only be used in combination with a certain clustering algorithm. This is the primary drawback of wrapper methods.

• Hybrid techniques attempt to make the most of the beneficial aspects of both filters and wrappers to achieve an optimal balance between the amount of computing work required and the level of success achieved (quality in the associated objective task when using the selected features).

2.1.2 Semi-supervised FS Methods

When only part of the class labels is available, the semi-supervised FS may be used to assess the significance of the features and to pick the best feature subset by using both labeled and unlabeled data. This is the case in situations when only some of the class labels are known. Methods that are semi-supervised are often used in situations in which it is difficult to obtain appropriately labeled samples from the actual world; for instance, in the fields of medical diagnosis and fraud detection (Sheikhpour et al., 2017).

A variety of semi-supervised feature selection approaches are classified into categories according to two distinct viewpoints. The first way of exploring is through the lens of the fundamental taxonomy of feature selection methods, which divides semi-supervised feature selection methods into a few different categories according to the way in which they interact with the learning algorithm. This is the foundation of the first perspective. Semi-supervised feature selection techniques may be broken down into the following three categories, according to the first taxonomy's point of view: filter (Han, 2015), wrapper (Bellal, 2012), and embedding (Ang, 2015). Then, the approaches that are employed for semi-supervised feature selection that are found in the literature are used to break each category down into other subcategories. The second viewpoint is based on the taxonomy of semi-supervised learning methods, which divides semi-supervised feature selection methods into a few different categories depending on which semi-supervised learning algorithm corresponds to the procedure used in the semi-supervised feature selection method. This is the basis for the second perspective. The structure of the hierarchy of semi-supervised feature selection techniques is shown in Fig. 2, which is based on the taxonomy of semi-supervised learning methods. Graph-based semi-supervised feature selection (Song, 2014), self-training based semi-supervised feature selection (Han, 2011), co-training based semi-supervised feature selection (Barkia, 2011), support vector machine (SVM) based semi-supervised feature selection (Ang, 2015), and other semi-supervised feature selection methods (Han, 2015) are the five categories that semi-supervised feature selection methods can be placed into, according to this taxonomy.

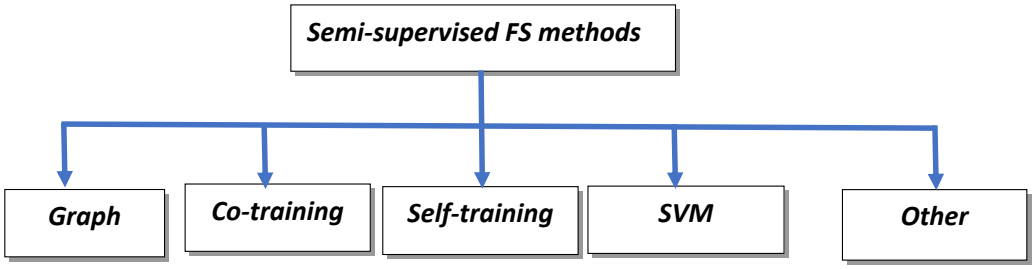


Fig. 2 Semi-supervised feature selection techniques

2.1.3. Supervised FS Methods

In the supervised feature selection methods, the class labels for all the observations are known in advance, and this knowledge is used in the process of selecting the best feature subset (Saeys et al., 2007). The supervised FS approaches are used within the framework of supervised learning, and they are versatile enough to be applied to classification as well as regression issues. The supervised binary classification is going to be one of the main topics of this thesis, and as a result, the supervised feature selection methods will be considered and discussed in detail in this research.
Under the category of supervised feature selection, there are primarily three techniques (Guyon & Elisseeff, 2003), Fig. 3:

2.1.3.1 Filter Methods

The filter method involves selecting features by basing those selections on various statistical measurements. This technique is independent on the learning algorithm and performs feature selection as a step in the pre-processing phase. Below are some of the most common filter methods:

Information Gain

The amount of loss in entropy associated with changing a dataset is directly proportional to the amount of information gained. A strategy for selecting features from a set may be derived from it by computing the information contributed by each variable in relation to the target variable (Dhal &. Azad,222).

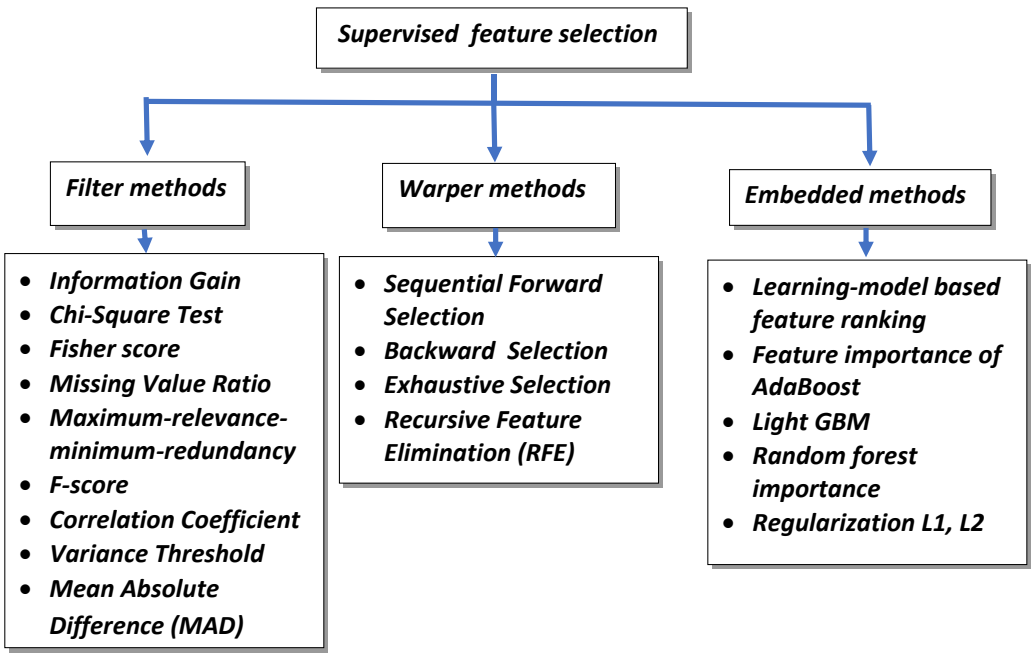


Fig. 3 Techniques of supervised feature selection

Chi-Square Test

In the Chi-Square test, a test statistic is computed that may be used to analyze the degree to which the two variables are dependent on one another. To analyze the statistical importance of the value, it is possible to compare it to the critical threshold of the Chi-square distribution (McHugh, 2013).
When doing FS based on Chi-Square, numerous Chi-Square tests are carried out to study the correlations between the features and the target variable in an individual fashion. After that, the results of the tests are used to rank the variables, and the features that have the greatest correlation with the variable of interest are the ones that are included in the final classification (Zheng et al., 2004, Liu & Setiono, 1995). The Chi-Square FS has several benefits, such that it is easy to use, straightforward, and requires little processing power.

Fisher score

The Fisher score is a useful method for reducing the number of the features in data. Its primary objective is to identify a feature subset in such a way that, within a data space spanned by the selected features, it is possible to simultaneously maximize the distances between data points belonging to different classes while simultaneously minimizing the distances between data points belonging to the same class (Baesens et al., 2016; Bishop, 1995).

Missing Value Ratio

The missing value ratio may be used to evaluate the feature set in comparison to the threshold value. The missing value ratio may be calculated using the following. The total number of observations is multiplied by the number of columns containing unaccounted-for values. If the variable in question has a value that is higher than the cutoff, it will be discarded (Dong & Liu, 2018).

Maximum-relevance-minimum-redundancy (mRMR)

It is rooted in the notion of two well-known FS techniques, which are referred to as maximum relevance FS and minimal redundancy FS. The maximal relevance FS selects the features for the final feature subset that is used for classification based on which ones have the highest relevance to the target class based on some measure (typically correlation or mutual information). This allows for the most accurate and efficient classification possible (Peng et al., 2002).

ANOVA F-value

ANOVA F-value feature selection is a method employed to identify the most significant features in a dataset for the purpose of classification. The process entails computing the F-statistic values for each feature and arranging them in order of their impact on the classification task. The utilisation of ANOVA F-value feature selection has led to improved classification performance and reduced false classification rates. This method has demonstrated its effectiveness in identifying distinguishing features from datasets that have many dimensions and contain a lot of noise, resulting in improved classification outcomes (Guyon et al., 2006).

Correlation Coefficient

The idea behind selecting features using correlation coefficient assumes that useful features will have a high degree of connection with the aim. In addition, there should be a correlation between the features and the objective, but there shouldn't be any correlation between the features themselves. When two features are correlated, predictions about one feature based on another can be made. If two features are associated, the model only will have to consider one of them because the other feature does not provide any new information. In this instance, use of the Pearson Correlation should be made. To determine the features, choice on an absolute value, such as 0.5, is needed as the threshold. If the predictor variables relate to one another, we will be able to exclude the feature that has a lower correlation coefficient value with the variable that we are interested in. It is also possible to calculate

multiple correlation coefficients to determine whether more than two features are connected to one another. The occurrence of this phenomenon is referred to as multicollinearity (Dhal &. Azad,2022).

Variance Threshold

The variance threshold is a straightforward method for selecting features that serves as a baseline. It gets rid of any features whose variance isn't high enough to fulfill certain criteria. It eliminates all features that have zero variance, or features that have the same value in every sample (Ferreira & Figueiredo, 2012)

Mean Absolute Difference (MAD)

This method is like a variance threshold method but the primary distinction between the MAD measure and the variance measure is that the MAD measure does not include the square. The mean absolute difference is a statistical measure that determines the absolute deviation from the mean value. The mean absolute deviation, or MAD, is also a scale variant, much like the variance. What this implies is that the larger the MAD, the greater the discriminating power (Ferreira & Figueiredo, 2012).

2.1.3.2 Wrapper Methods

Wrapper feature selection approaches go through several possible feature subsets in an iterative process and choose the most effective subset depending on how well the prediction model performs (Kozodoi et al., 2019). Wrapper approaches are straightforward (Guyon & Elisseeff, 2003), and they consider the feature dependencies as well as the interaction that occurs between the prediction algorithm and the feature subset search (Saeys et al., 2007). They utilize the algorithm's prediction performance to evaluate the utility of feature subsets, and they demand large amounts of computing (Liang et al., 2015). In addition, they have a significant risk of overfitting because they use the prediction performance of the algorithm. The process of evaluating all the potential feature combinations requires a lot of computing power. The following are some examples of wrapper methods:

Sequential Feature Selection (SFS)

SFS can be either a forward selection or backward selection.

The sequential forward selection method is an Iterative Wrapper-Type Forward Selection Method. This method starts with an empty feature set and, at each step of the process, adds a feature that improves the value of the selected objective function (evaluation criterion). The criteria for assessment that is most often used is one that is associated with a measure of classification performance, most frequently the classification accuracy or classification error. The first stage of the method consists of adding each feature to the feature set one at a time and then calculating the value of the objective function using the classification model that has been selected. After that, the feature that offers the highest value in relation to the evaluation criteria is added to the feature set that was previously empty. The feature that performs the best in a pair with the feature that was selected in the first phase (in terms of the evaluation criteria that was specified) is included in the list of selected features in the second stage (Chandrashekar and Sahin 2014).

The sequential backward selection method is another kind of iterative process; however, it works in the opposite direction of the forward selection method. This approach starts the process by thinking about all the features, and then it eliminates the one that is the least important. This process of elimination will continue until taking features away from the model does not result in an improvement in its overall performance (Chandrashekar & Sahin 2014; Dong & Liu, 2018).

Exhaustive Feature Selection

One of the most effective ways for selecting features is called exhaustive feature selection, and it does so by applying a brute-force evaluation to each feature set. It indicates that this technique will try to create every conceivable combination of features before returning the feature set that performs the best (Chandrashekar & Sahin 2014; Dong & Liu, 2018).

Recursive Feature Elimination (RFE)

RFE is an algorithm for backward selection that was proposed by Guyon et al. (2002) to prevent the need for refitting many models at each stage of the search. RFE is responsible for training the classifier, computing the ranking criteria for each feature, and removing the feature that has the lowest ranking criterion (Guyon et al., 2002). This process is carried out in an iterative manner until the required number of features has been achieved, and to speed up the process, more than one feature may be eliminated at a time (Guyon et al., 2006). RFE can make an accurate selection of features from a training dataset that are more or more relevant in the process of predicting the target label. In the process of selecting features for use in the loan default prediction issue, RFE has been used (Ma et al., 2018). RFE requires the setup of several critical settings, including the number of features to be used and the base algorithm that will be used to choose those features.

2.1.3.3 Embedded Methods

By considering the interrelationship of features and maintaining a low computational cost, embedded techniques incorporated the benefits that are associated with filter methods and wrapper methods. These are faster processing techniques that are more accurate than the filter method, yet the filter method is the faster of the two. Embedded feature selection approaches combine the feature selection process with the learning process and choose features concurrently with the training of the model (Kozodoi et al., 2019). They consider the interaction that occurs between the classifier and the process of feature selection, and they need less processing power than wrapper approaches do (Saeys et al., 2007). These approaches are likewise iterative, which assesses each iteration to determine the most essential features that contribute the most to training in a specific iteration in an optimum manner. The following are some examples of embedded methods.

Learning-model based Feature Ranking

A feature ranking that is based on an integrated learning model (LMBFR). Due to the method's ease of use and high level of productivity, it was used extensively for FS in many of the earlier studies (Xia et al. 2017; Zhou et al., 2019). The fundamental concept behind the procedure is straightforward and easy to grasp. To begin, the relevant classification model is trained and estimates of the relative relevance of features in the model are calculated. Then, as the final feature subset, the features that have the greatest estimated feature relevance are utilized. The suggested feature subset is only applicable to the classifier (Ishwaran 2007; Kazemitabar et al., 2017).

Feature Importance of Adaptive Boosting (AdaBoost-FS)

The feature significance that is assigned to an AdaBoost feature is derived from the feature importance that is assigned to an AdaBoost feature by its base classifier. If a decision Tree as primary classifier is used, then the feature significance that AdaBoost assigns to each feature is calculated based on the weighted average of the feature value assigned by each decision Tree. This is somewhat comparable to the traditional method of using a forest of trees to establish the significance of a feature. It takes advantage of the fact that features located at the top of the tree contribute to the final prediction decision of a larger fraction of input samples, and this expected fraction can therefore be used to estimate the relative importance of a feature. In other words, it uses the fact that features found at the top of the tree contribute to the final prediction decision. The production of variations of the basic classifier is where the AdaBoost and other methods, such as a Random Forest (also known as a "forest of trees"), form part ways. Both methods have the potential to play a role in the significance of feature determination. The first method generates variations by placing a greater emphasis on "difficult" cases, whereas the second method generates variants by including an element of randomness in the process of tree construction (Wang, 2012).

CatBoost-FS

CatBoost feature selection is an embedded method that leverages the CatBoost gradient boosting framework to determine the relative importance of input features during model training. Unlike wrapper methods that require training multiple models for evaluation, CatBoost derives feature importance internally by analyzing how much each feature contributes to reducing the chosen loss function or changing predictions. Commonly used importance types include Prediction Values Change, which measures the average effect of a feature on the model's output, and Loss Function Change, which quantifies how much the loss increases when a feature is permuted or removed. This approach benefits from CatBoost's unique strengths, including its native handling of categorical variables, resistance to overfitting via ordered boosting, and efficiency in high-dimensional and imbalanced datasets. Furthermore, it offers multiple importance estimation options, such as SHAP values, allowing for deeper interpretability of feature contributions. These advantages make CatBoost feature selection particularly suitable for credit risk and loan default prediction, fraud detection, customer churn analysis, ranking and recommendation systems, and other applications that involve large-scale tabular data with mixed feature types, where accuracy and explainability are equally critical (Hancock & Khoshgoftaar, 2020).

Light Gradient-boosting machine (LightGBM-FS)

The light GBM framework is a gradient boosting approach that makes use of a tree-based learning algorithm. The LightGBM method develops trees in a vertical direction, whereas other algorithms grow trees in a horizontal direction. This means that the LightGBM algorithm grows trees leaf-wise, while other algorithms grow level-wise (Wang, 2012; Dong & Liu, 2018).

Random Forest (RF-FS)

Random Forest is the many tree-based techniques of feature selection that are available to assist in determining the relevance of features and give a means by which features may be picked. In this context, feature importance identifies whether feature is more important in terms of the overall model-building process or has a significant bearing on the variable of interest. Random Forest is an example of this sort of tree-based approach. Random Forest is a bagging algorithm that combines a variable number of decision trees into a single output. Gini importance and permutation importance are two sophisticated approaches that may be used to quantify the relevance of features in RF data (Strobl et al., 2007). The difference in accuracy achieved before and after applying a random permutation to a feature is used to evaluate the significance of the permutation. The computation may be found in its entirety in the research carried out by Archer & Kimes (2008), Strobl et al. (2008), and Janitza et al. (2016). Gini relevance may be evaluated based on the degree to which Gini impurity is reduced. The reduction of Gini impurity is the form of RF that is usually used (Boulesteix et al., 2012).

Regularization

Regularization involves the addition of a penalty term to various features of the machine learning model to prevent the model from being overfit. Because this penalty term is applied to the coefficients, some of the coefficients are reduced until they become zero. Those features of the dataset that have 0 coefficients may be eliminated from the analysis. L1 regularization, also known as LASSO (least absolute shrinkage and selection operator) regularization and Elastic Nets are the two categories of regularization methods (L1 and L2 regularization) (Dong & Liu, 2018).

Table 1 summarizes each method's underlying concept, estimated computation time, accuracy potential, strengths, and typical applications. This comparison highlights the trade-offs between computational efficiency and predictive performance, providing a practical reference for method selection in machine learning. In the context of peer-to-peer (P2P) lending, such structured evaluation is crucial for building predictive solutions that can manage large, heterogeneous borrower datasets while ensuring accurate and timely credit risk assessment. By selecting feature selection (FS) methods that optimize dimensionality reduction without compromising predictive power, P2P platforms can

enhance the scalability of credit scoring systems, improve the reliability of borrower risk profiles, and streamline loan approval processes. Furthermore, aligning FS techniques with P2P lending priorities; such as interpretability, computational efficiency, and real-time decision-making, enables lenders and investors to extract actionable insights quickly, reduce default rates, and strengthen trust in platform operations. Ultimately, effective FS practices empower P2P lending ecosystems to transform raw borrower data into meaningful, data-driven insights that support sustainable growth and responsible credit allocation.

Table 1 Comparative analysis of supervised feature selection methods: concepts, performance metrics, and application domains

Type	Method	Concept	Computation Time	Accuracy Potential	Strengths	Typical Applications	Ref.
Filter	Information Gain	Measures reduction in entropy	Very low	Low	Simple, fast, scalable ·	Text classification, decision trees	Dhal &. Azad,222
	Chi-Square Test	Tests statistical independence	Very low	Low	Categorical data, interpretable	Feature filtering in classification tasks	McHugh, 2013; Liu et al, 2005
	Fisher Score	Class-based separability via mean/variance	Low	Medium	Good for multiclass tasks, interpretable	Bioinformatics, image recognition	Kozodoi et al.,2019
	Missing Value Ratio	Filters based on data completeness	Very low	Very low	Cleans noisy data, useful preprocessing	Early data cleaning and Exploratory Data Analysis (EDA)	Dong & Liu, 2018
	Maximum-relevance-minimum-redundancy	Maximizes relevance, minimizes redundancy	Medium	high	Reduces redundancy, captures dependencies	Genomics, microarray data	Peng et al., 2002
	ANOVA F-value	Based on variance between classes	Low	Medium	Simple binary class selection	Text mining, anomaly detection	Guyon et al., 2006
	Correlation Coefficient	Measures linear dependence	Very low	Low	Useful for linear features	Financial modelling	Dhal &. Azad,2022
	Variance Threshold	Removes low-variance features	Very low	Very low	Removes non-informative noise	Baseline dimensionality reduction	Ferreira & Figueiredo, 2012
	Mean Absolute Difference	Measures of variability in a feature	Very low	Very low	Simple statistical filter	Initial feature screening	Ferreira & Figueiredo, 2012
Wrapper	Sequential Forward Selection	Adds features iteratively for best score	High	High	Greedy yet effective	Model tuning, small datasets	Chandrasheka r & Sahin 2014; Dong & Liu, 2018
	Backward Feature Selection	Removes features one at a time	High	High	Good accuracy, more global search	Credit risk scoring, clinical diagnosis	Chandrasheka r & Sahin 2014; Dong & Liu, 2018
	Exhaustive Feature Selection	Try all combinations	Very high	Very high	Best accuracy (if feasible)	Benchmarking, research models	Chandrasheka r & Sahin 2014; Dong & Liu, 2018

	Recursive Feature Elimination	Iteratively removes weakest features	Medium	High	Model-specific, efficient	SVM, Logistic Regression, predictive analytics	Guyon et al., 2002, 2006; Peng et al., 2002
Embedded	Learning-model based Feature Ranking	Uses model coefficients/weights	Low	Medium	Efficient, interpretable	Linear/logistic regression, Generalized Linear Models (GLMs)	Ishwaran 2007; Kazemitabar et al., 2017
	AdaBoost-FS	Uses ensemble weight-based importance	Medium	High	Captures non-linear interactions	Fraud detection, ranking systems	Wang, 2012
	CatBoost	Evaluates the importance of each feature by measuring the change in predictions or loss when the feature is permuted or removed	Moderate to High	High	Handles categorical variables, performs well on high-dimensional	Credit risk and loan default prediction, fraud detection	Hancock & Khoshgoftaar, 2020
	LightGBM - FS	Gradient boosting with efficient split/gain metric	Low	Very high	Fast and accurate for structured data	Credit default prediction, time series, large-scale ML	Wang, 2012; Dong & Liu, 2018
	Random - FS	Measures importance via impurity/permutation	Medium	High	Robust to outliers, nonlinear handling	Risk modelling, interpretability tools	Archer & Kimes (2008), Strobl et al. (2008), and Janitza et al. (2016; Strobl et al., 2007
	Regularization (L1, L2)	Penalizes large coefficients; L1 induces sparsity	Low	High	Prevents overfitting, automatic feature selection (L1)	Text data, high-dimensional regression	Dong & Liu, 2018

2.2. Machine Learning

The field of research known as machine learning (ML) is a subfield of artificial intelligence (AI) that gives computers the capability to automatically learn from data and previous experiences while simultaneously recognizing patterns to generate predictions with minimum input from humans. The approaches of machine learning make it possible for computers to function independently without the need for explicit programming. Applications that use machine learning are constantly being updated with new data, which allows them to autonomously learn, grow, evolve, and adapt. Machine learning can glean meaningful information from huge volumes of data by using algorithms that are able to detect patterns and learn from experience in an iterative process. This allows machine learning to achieve its goal. Instead of depending on any preconceived equation that may act as a model, machine learning algorithms employ computing techniques to learn directly from data. This contrasts with traditional approaches. To develop a model, machine learning techniques are often used on a training dataset. The trained machine learning algorithm will generate a prediction based on the established model whenever new input data is added to the algorithm (Liu et al., 2017; Dietterich, 1997).

The emergence of big data, internet of things, and ubiquitous computing has made machine learning a crucial tool for problem solving in a wide variety of applications, including but not limited to healthcare industry, finance sector, retail sector, travel industry and social media.

There are several approaches in which machine learning algorithms may be learned, and each of these approaches has both advantages and disadvantages. Machine learning may be roughly broken down into four distinct subfields (unsupervised, semi-supervised, supervised and reinforcement), each of which is characterized by a distinct set of learning strategies and approaches (Bishop, 2006; Dietterich 1997), Fig. 4.

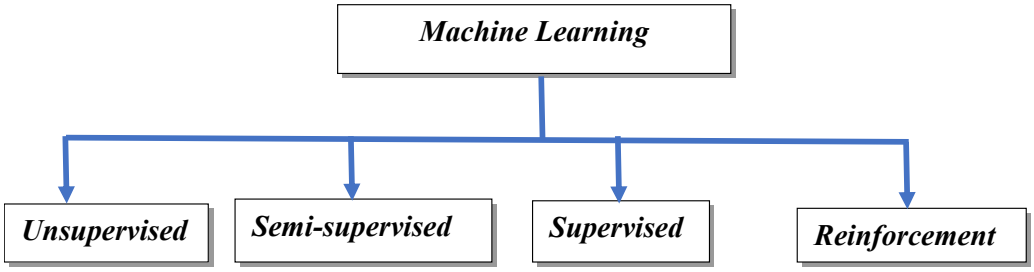


Fig.4 Subfields of machine learning

Supervised Machine Learning

This sort of machine learning requires supervision, since computers are trained on labeled datasets and then given the ability to predict output based on the training. Some input and output parameters are already mapped, according to the labeled dataset. As a result, the machine gets trained using the input and output. In later stages, a device is created to predict the result using the test dataset (Kotsiantis, 2007).

The second level of categorization in supervised machine learning differentiates between two primary types of algorithms: classification and regression. Classification algorithms are used to solve classification problems, where the goal is to predict categorical output variables such as yes/no, true/false, or male/female. These algorithms focus on assigning input data to specific classes or categories. On the other hand, regression algorithms are employed to address regression problems, where there is a linear relationship between input and output variables. These algorithms are particularly effective for predicting continuous

output variables, making them valuable for forecasting tasks (Bishop, 2006).

Unsupervised Machine Learning

Unsupervised learning is an approach of learning that does not need supervision. The computer is trained using an unlabeled dataset and can predict the output without the need for human intervention. Unsupervised learning algorithms attempt to categorize unsorted data based on similarities, differences and patterns in the input (Ghahramani, 2004).

Semi-supervised Learning

Semi-supervised learning combines both supervised and unsupervised machine learning features. It trains its algorithms using a mix of labeled and unlabeled datasets. Semi-supervised learning solves the shortcomings of the previous solutions by using both kinds of datasets (Zhu, 2009).

Reinforcement Learning

Reinforcement learning is a feedback-driven method of learning. Here, the AI component uses the hit-and-trial approach to autonomously assess its surroundings, act, learn from its experiences, and improve performance. Every positive activity is rewarded, while every negative action is punished. As a result, by doing excellent activities, the reinforcement learning component seeks to maximize the rewards (Sutton, 2015; Muddasar, 2020).

2.2.1 Classification models

Classification machine learning models are a type of supervised learning algorithm designed to predict the categorical label or class of an input based on its features. The primary goal of classification is to assign input data to predefined classes or categories. These models are trained on labelled datasets, where the class label for each input is known, enabling the model to learn patterns from this data and generalize to predict the correct class for new, unseen inputs. The classification methods may be categorized into three groups according to the kind of algorithm employed. The classifiers employ linear, non-linear or tree-based algorithms. The classification techniques are explained in the following.

2.2.1.1 Classification approach based on linear algorithms

Logistic Regression (LR)

Logistic regression, which falls under the umbrella of the supervised learning technique, is one of the most common and widely used machine learning algorithms. The categorical dependent variable may be predicted by utilizing it in conjunction with a predetermined group of independent variables. The LR is quite like the linear regression, with the primary difference being how each method is used. When trying to solve regression difficulties, linear regression is the method of choice, but LR is used when attempting to solve classification issues (Kleinbaum, 2010; Park, 2013).

In cases where the dataset is unbalanced, it has been observed that the bias in the regression intercept tends to increase with the imbalance. This bias can be corrected by incorporating a prior that accounts for the minority class, or by applying a penalized likelihood approach where the likelihood is weighted according to the proportion of ones in the target variable. Previous studies have highlighted the good performance of LR in such contexts.

LR may be broken down into three distinct subtypes based on the categories, which are as follows (Hilbe, 2009):

Binomial: The dependent variables in a binomial LR analysis may only take on one of two potential forms, either 0 or 1, Pass or Fail, etc.

Multinomial: There may be three or more potential unordered categories of the dependent variable in multinomial LR. Some examples of these categories are "cat," "dogs," and "sheep."

Ordinal: There may be three or more potential ordered kinds of dependent variables in ordinal logistic regression. These types of ordered may include "Low," "Medium," or "High," for example.

As mentioned above, the fundamental idea behind the LR is somewhat like the linear regression model that is most often used. On the other hand, in the case of LR, the model parameters are determined using maximum likelihood estimation, while in the case of linear regression, the model parameters are often estimated via the use of the ordinary least-squares estimation. This indicates that the parameters are selected such that they correspond to the values of the model parameters that are most likely to be found in the data that is being used; to put it another way, these parameters maximize the value of the likelihood function that is being utilized (Bishop, 2006, Hosmer et al. 2013, Peng et al. 2002).

2.2.1.2 Classification approach based on non-linear algorithms

Naive Bayes (NB)

It is a method of classification that is based on Bayes' Theorem and assumes that predictors are independent of one another. It is a probabilistic classifier, which means that it makes its predictions based on the likelihood that an item would be found. A NB classifier works on the assumption that the existence of one feature in a class is independent of the presence of any other feature. For instance, for some fruit to be identified as an apple, it must be red, spherical, and have a diameter of around 3 inches. Even if these features rely on one another or on the presence of the other features, each of these features independently contributes to the chance that this fruit is an apple, which is why it is said to be 'naive.' The NB model is simple to construct and is especially helpful for working with extremely big data sets. In addition to its ease of use, the NB technique is renowned for its ability to outperform even the most complex classification approaches (Provost & Fawcett 2013; Zhang 2005).

The NB models have been used well in a variety of contexts, including spam filtering and text categorization, and because they need very little CPU power, they have also been utilized often in the context of real-time prediction. Additionally, the NB classifiers are often used to benchmark the more complex classifiers in a variety of classification situations (Bishop 2006; Zhang 2005).

Because the features may be represented individually with proper models, the NB classification models are particularly helpful when the predictors contain both continuous and categorical variables. This is because the models can distinguish between the two types of variables (Bishop, 2006).

The following is a list of the different kinds of Naive Bayes models that exist (Provost & Fawcett 2013):

Gaussian: The Gaussian model works on the assumption that individual features follow a normal distribution. This indicates that if predictors take continuous values rather than discrete values, then the model assumes that these values are sampled from the Gaussian distribution. If the predictors take discrete values, then the model does not make this assumption.

The Gaussian Naive Bayes (GNB) classifier (John & Langley, 1995; Han, Wang, & Mao, 2005; Chowdhury & Alspector, 2003; Rennie, 2001) is particularly suitable for high-dimensional feature spaces where density estimation is challenging.

Multinomial: When dealing with data that follows a multinomial distribution, the Multinomial Naive Bayes classifier is the method of choice. Its primary use is in the resolution of issues pertaining to the categorization of documents; this implies determining in which category a certain document fits, such as education, politics, sports, etc. The classifier bases its predictions on the frequency with which words appear.

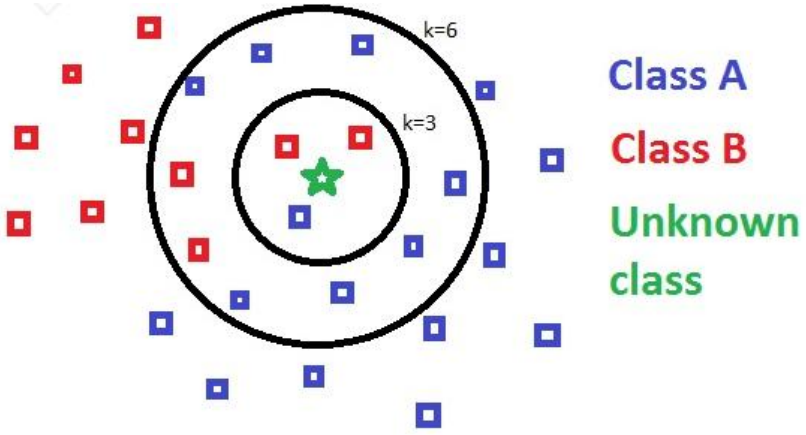
Bernoulli: The Bernoulli classifier performs its function in a manner that is like that of the multinomial classifier; however, the predictor variables in this case are the independent Boolean variables, for example, if a certain term appears or does not appear in each text. Additionally, well-known for its use in document categorization problems is this model.

K-nearest neighbors (KNN)

The KNN method is a supervised learning classifier that makes use of proximity to create classifications or predictions about the grouping of an individual data point. Although it may be used to issues involving either regression or classification, it is more often employed as a classification technique since it is based on the concept that points with similar characteristics can be found near one another. The KNN algorithm is non-parametric, which implies that it does not make any assumptions about the data it is analyzing KNN maintains all available records and guesses the class of new instances based on the probability of similarity measurements from the nearest neighbors.

The "K" in the k-NN algorithm refers to the "k" number of closest neighbors whose votes are used to predict the label for a new record in the vicinity of those neighbors. Let's suppose K is equal to three. Then a circle with the new data item as the center will be displayed as large as to include just three closest neighbour data points, and the label of the new record will be determined by the distance between the record and each of the neighbors. Each class's bounds may be created for a particular K-value. These lines may effectively divide one class from another. If K reaches a very large value and eventually approaches infinity, everything becomes one class, or the one with the entire majority.

An example of K-nearest neighbor algorithm approach is presented in Fig. 5. The KNN Algorithm is not difficult to implement, can function normally despite having noisy training data, and has the potential to be more successful when there is a large quantity of training data available. Despite its seeming ease, it can produce extremely competitive outcomes. It can deal with issues of both the classification and regression sorts that are predictive. On the other hand, it can accomplish and carry out categorization tasks in a more streamlined manner (Friedman, 2001; Murphy, 2012).



With K=3, Class B is assigned, with K=6 Class A is assigned
Fig. 5 An example of K-nearest neighbor algorithm approach

The simplicity and efficiency of k-NN make it an attractive algorithm in machine

learning. However, it has several significant drawbacks as noted in (Dubey & Pudi, 2013; Tan, 2005). These issues may include challenges with high-dimensional data, computational inefficiency in large datasets, and sensitivity to irrelevant features, among others. Furthermore, KNN technique is known as a lazy learning method because it keeps the data members stored simply in efficient data structures like hash tables, which reduces the computation cost of checking and applying the appropriate distance function between the new observation and all k number of different data points stored, and then coming to any conclusion about the label of the new data point, without having to construct a mapping function or internal model like other classification techniques.

Support Vector Machine (SVM)

SVM is one of the most well-liked supervised learning algorithms that is used to solve Classification and Regression issues. However, it is largely employed in machine learning classification issues. The Support Vector Machine (SVM) was introduced by Vapnik (1998) as a kernel-based machine learning model designed for both classification and regression tasks (Cervantes et al., 2020). Many experts choose SVM because it offers notable accuracy, while using less processing resources. The SVM algorithm's objective is to establish the optimal line or decision boundary that can divide n-dimensional space into classes, allowing us to quickly classify new data points in the future. A hyperplane is the name given to this optimal decision boundary. SVM selects the extreme vectors and points that aid in the construction of the hyperplane. Support vectors, which are used to represent these extreme instances, form the basis for the SVM method. Consider the diagram below, Fig. 6, where a decision boundary or hyperplane is used to categorize two distinct categories (Noble 2006; Provost & Fawcett 2013).

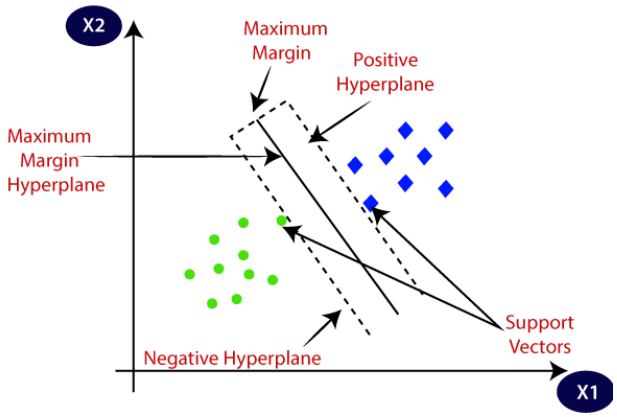


Fig. 6 Basis for the SVM method

It's important to note that the SVM only relies on the support vectors, which are the data points closest to the decision boundary. This makes SVM an attractive choice for moderately imbalanced datasets Akbani et al., 2004; Coussement et al., 2008). However, its performance tends to degrade when the class distribution is highly skewed (Tian et al., 2011).

2.2.1.3 Tree- base classifiers

The tree-based family of supervised machine learning conducts classification and regression tasks by developing a tree-like structure for selecting the target variable class or value based on the features. This structure is used to decide whether the target variable will have a high or low value. From the simplest to the most advanced, tree-based machine

learning techniques are explained in the following.

Single estimator model

Single estimator model is the simplest tree-based machine learning techniques.

Decision Tree

Decision tree is a supervised learning approach that may be used to solve both classification and regression problems; however, it is most often employed to solve classification issues. It's termed a decision tree because, like a tree, it begins with the root node and grows into a tree-like structure with additional branches. A decision tree simply asks a question and divides the tree into sub trees depending on the response (Yes/No). The decision node and the leaf node are the two nodes of a decision tree. Leaf nodes are the result of those decisions and do not include any more branches, while decision nodes are used to make any decision and have several branches, Fig. 7. Decision trees are designed to simulate human thinking abilities while making decisions, making them simple to comprehend. Because the decision tree has a tree-like form, the rationale behind it is simple to comprehend. When compared to other algorithms, there is a much-reduced need for data cleansing (Mohri et al., 2012; Song & Lu 2015; Friedman, Hastie, & Tibshirani, 2001).

In the context of imbalanced datasets, some researchers argue that decision trees may not be suitable (Branco et al., 2016; Weiss et al.,2004). However, others propose alternative splitting strategies, such as using the Hellinger distance, to address this issue and improve performance on imbalanced datasets (Branco et al., 2016; Yin et al., 2013).

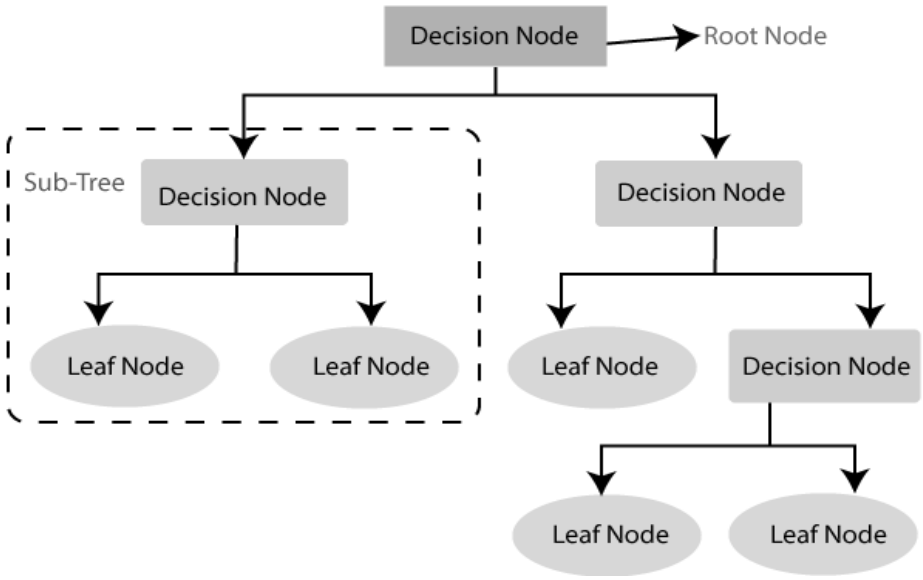


Fig. 7 Decision tree method

Ensemble models

A machine learning model called an ensemble combines predictions from two or more algorithms. In machine learning, the three most popular ensemble learning techniques are bagging, boosting, and stacking.

Bagging

Bagging is a method of ensemble modeling, which is primarily used to solve supervised machine learning problems. It is generally completed in two steps as follows. Bootstrapping step is a random sampling technique that uses the replacement procedure to generate samples from data. This technique involves feeding random data samples to the primary model, followed by running a base learning algorithm on the samples to complete the learning process. Aggregation step entails combining the output from all base models and, using their output, predicting an overall outcome with a higher degree of accuracy and reduced variance.

Random Forest (RF)

An instance of a bagging technique is RF. It is a well-known machine learning algorithm that uses supervised learning. RF is an algorithm for machine learning that was developed by (Breiman, 2001) and is frequently used. It is based on ensemble learning, which is a method of integrating numerous classifiers to solve a complicated issue and enhance the model's performance. RF is a classifier that includes several decision trees on different subsets of a given dataset and takes the average to enhance the predicted accuracy of that dataset. Instead, than depending on a single decision tree, the RF collects the forecasts from each tree and predicts the final output based on the majority votes of predictions. The bigger the number of trees in the forest, the more accurate it is and the issue of overfitting is avoided (Kotsiantis 2007; Malekipirbazari & Aksakalli, 2015). RF has shown excellent performance while having very few parameters that need to be adjusted (Genuer et al., 2017). It's possible to think about RF as a natural progression from bagging CART trees. Each tree is constructed with no observations that are chosen at random by bootstrap sampling with replacement at the split of each node. The subset of features is also chosen at random from all the features, and there is no need to conduct any pruning in RF since it is not necessary (Genuer et al., 2017). Archer & Kimes (2008) and Genuer et al. (2010) both provide detailed explanations of the RF algorithm. It has seen widespread use as a classifier in the personal loan default prediction process (Florez-Lopez & Ramon Jeronimo, 2015; Kruppa et al., 2013; Xia et al., 2017). The Random Forest method is represented in Fig. 8.

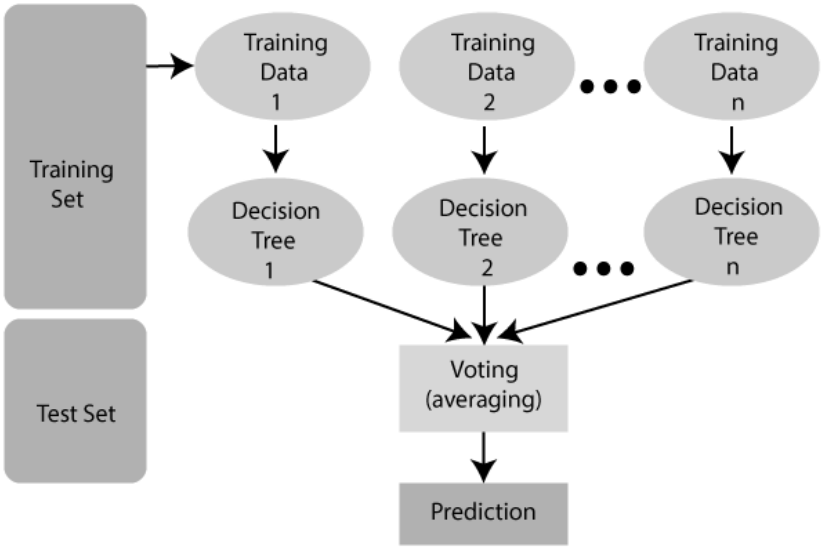


Fig. 8 Random Forest method

RF can handle both classification and regression problems and can handle huge datasets with a lot of dimensionalities. It improves the model's accuracy and eliminates the problem of overfitting. It requires less training time than other algorithms and predicts output with excellent accuracy, especially when dealing with enormous datasets. When a considerable

amount of the data is missing, it may still retain accuracy (Breiman 2001; Malekipirbazari & Aksakalli 2015).

Boosting

As mentioned above, bagging is a method of creating many models at once. Each model operates independently of the others. Boosting is used for increasing the accuracy of the model. Unlike bagging, boosting produces models one by one. The first model, for example, learns from the training data. The second model then learns from the same training dataset as well as the previous model's faults. The third model, like the previous two, learns from the same training dataset and the prior model's errors. This process is repeated until many models have been created. Because each base model/estimator learns from the prior model's faults, boosting may improve accuracy. It's often defined as transforming individual weak learners into a group of strong learners. Instead of learning separately, it's like having a group of cooperative machines (Dietterich, 2000). Adaptive boosting and gradient boosting are two types of boosting algorithms.

Adaptive Boosting (AdaBoost)

AdaBoost is a machine learning method that is known for its rapid convergence and ease of implementation. The mistakes that poor learners make are considered when AdaBoost makes its adjustments (Wang, 2012). First, there is a uniform distribution of weight across all the samples. The outcomes of the weak learner's training mistakes from the previous iteration influence the weights of the samples in a way that causes them to change. The weights of samples that were misclassified are raised to drive the less capable learner to concentrate on the hard samples that are included in the training set. In the subsequent iteration, the weak learner is provided with training using the weighted sample. Following several rounds, the final outcomes are determined by merging the outputs of ineffective learners from each iteration using a weighted majority vote (Freund & Schapire, 1997; Wen et al. 2015).

AdaBoost is a widely used technique for a variety of applications, including the detection of vehicles (Khammari et al., 2005; Liu et al., 2005; Wen et al., 2015), the classification of tumors (Huang et al., 2020), the diagnosis of heart disease (Rajesh & Dhuli, 2018), facial expression recognition (Ruan & Yin, 2009), and face detection (Jung et al., 2005). AdaBoost achieves results that are much superior to those achieved by the single most effective classifier used in credit risk assessment (Finlay, 2011; Ma et al., 2018).

Categorical Boosting (CatBoost)

CatBoost is a gradient boosting technique that was introduced by Dorogush et al. (2018). It is capable of effectively dealing with categorical information. When it chooses the tree structure, it employs a novel schema to compute leaf values to lessen the likelihood of overfitting during the permutation phase (Dorogush et al., 2018). CatBoost has accomplished remarkable results in the fields of psychology, traffic engineering, cybersecurity, biochemistry, biology, and marketing (Hancock & Khoshgoftaar, 2020). Additionally, it has been applied to the issue of predicting loan defaults, where it gets the highest performance when compared with LR, RF, and XGBoost (Xia et al., 2019)

Gradient Boosting Machine (GBM)

Like RF in the bagging approach, GBM is another boosting tree-based machine learning technique. GBM, unlike adaptive boosting, learns from the prior model's residual errors of actual and predictable values to reduce them. This technique generates a sequence of models to progressively reduce residual errors (Haykin, 2008).

Extreme Gradient Boosting (XGBoost)

As the name implies, extreme gradient boosting is the next step up from regular gradient boosting. Its accuracy is estimated to be "very" high. XGBoost, like random forest and gradient boosting, is based on a foundation of several decision tree models. Because the gradient boosting model may predict well for the training dataset but poorly for the test dataset, overfitting is an issue. Gradient boosting has the flaw of overfitting, hence XGBoost was created as a regularized gradient boosting to address this issue. Because regularization is employed to manage the overfitting issue, it allows for quick and flexible model adjustment. This is one reason why it is effective (Daoud, 2019).

In the world of machine learning contests, XGBoost has a strong reputation. It was used by the top 10 winning teams in the 2015 KDD Cup competition, as well as 17 of the 29 winning solutions that were posted on the well-known machine learning competition website Kaggle in 2015. (Chen & Guestrin, 2016). Xia et al. (2017) contribute to the area of loan default prediction by proposing a sequential ensemble loan default prediction model that is based on XGBoost. On average, it performs better than the baseline models. According to Lu et al. (2019), XGBoost performs the best among several classifiers when it comes to the prediction of microloans.

Light Gradient Boosting Machine (LightGBM)

LightGBM is an upgraded version of GBM. To increase the effectiveness of the training, LightGBM introduces gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) (Ke et al., 2017). To manage a significant number of data instances, GOSS has been implemented. When computing the information gain, GOSS retains the data instances that have big gradients and executes sampling at random on the data instances that have small gradients. EFB is used as a solution to the issue of a high number of features present in the data. This solution involves the bundling of features that are unique to one another into a single feature to minimize the number of features (Ke et al., 2017). LightGBM has the potential to considerably surpass XGBoost in terms of both the speed of computation and the amount of memory. In the evaluation of personal credit risk, Ma et al. (2018) show that LightGBM performs better than XGBoost on a dataset from Lending Club. Daoud (2019) finds that LightGBM is faster than XGBoost and CatBoost on a credit dataset, and he obtains more accurate predictions while using the same time budget for hyperparameter optimization.

2.2.1.4 Classification approach based on deep learning

Artificial neural network (ANN)

The biological neural networks in the brains of animals serve as an inspiration for the artificial neural network (ANN). An artificial neural network, which also is called multi-layer perceptron (MLP), is a networked nodes that simulate brain neurons. These are linked by connections, which stand in for the synapses of the brains. A network of artificial neurons can receive signals from other neurons, process them, and then transmit those processed signals onto other neurons in the network. The activation function is a non-linear function of the total of the inputs that each neuron uses to calculate its output, where the signal is a real number. The relative weight of neurons and edges changes throughout the learning process. The signal strength at a connection can be adjusted by adjusting the weight. Usually, layers of neurons are aggregated. The inputs to several layers may undergo distinct modifications. From the input layer to the output layer, signals may go through several hidden layers along the way. Fig. 9 depicts an artificial neural network approach, where each circular node stands for a neuron and an arrow connects the output of one neuron

to the input of another. When there are two or more hidden layers in a network, it is called a deep neural network. Artificial neural networks are frequently employed for tasks such as adaptive control, predictive modelling, and other activities that necessitate training with a dataset. Additionally, they are employed to address issues related to artificial intelligence. ANNs are capable of learning from their mistakes and drawing insights from complicated datasets that at first glance appear unrelated (Haykin,2008).

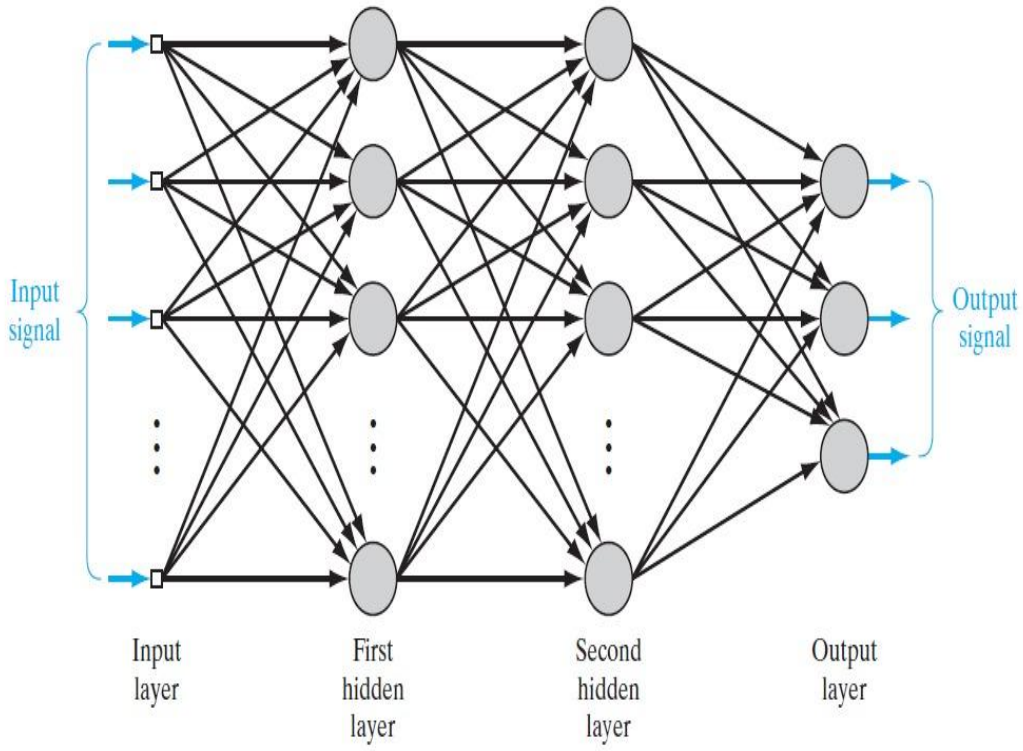


Fig.9 Artificial neural network approach

Table 2 provides an overview of linear, non-linear, tree-based, and deep learning classifiers, highlighting their conceptual foundations, expected computation time, accuracy potential, and common application areas. This comparative insight aids in aligning algorithm selection with the problem's complexity, data characteristics, and performance goals. In the context of peer-to-peer (P2P) lending, such comparisons are critical for selecting classifiers that balance predictive accuracy, interpretability, and computational efficiency to assess borrower creditworthiness and predict loan defaults. Effective classifier selection underpins the analytical framework of P2P lending platforms, enabling robust credit scoring, real-time risk evaluation, and streamlined loan approval processes. Understanding the trade-offs among different classifier families allows P2P practitioners to tailor models to the unique challenges of high-volume, heterogeneous borrower data, such as class imbalance and non-linear relationships. Moreover, aligning model capabilities with platform objectives, such as minimizing default risk, improving decision-making speed, and enhancing investor confidence, ensures that machine learning classifiers play a central role in building reliable, data-driven P2P lending ecosystems.

Table 2 Benchmark comparison of linear, non-linear, tree-based, and deep learning classifiers

Category	Algorithm	Concept	Computation Time	Accuracy Potential	Strengths	Typical Applications	Ref.
Linear Classifier	**Logistic Regression**	Models' linear relationship between features and log-odds of class membership	Very Low	Medium	Interpretable, efficient, works well with linearly separable data	Credit scoring, medical diagnosis	Kleinbaum, 2010; Park, 2013; Bishop, 2006; Hosmer et al. 2013, Peng et al. 2002; Bishop 2006
Non-linear Classifiers	**Naive Bayes**	Probabilistic model assuming feature independence	Very Low	Low	Fast, simple, performs well on small/noisy datasets	Spam filtering, text classification	Provost & Fawcett 2013; Zhang 2005
	K-Nearest Neighbours	Classifies based on majority vote of nearest neighbours	Medium (High on large data)	Medium	No training time, good for local structure	Pattern recognition, recommender systems	Friedman, 2001; Murphy, 2012; Dubey & Pudi, 2013; Tan, 2005
	Support Vector Machine	Maximizes margin between classes using kernel trick	Medium–High	High	Effective for high-dimensional and non-linear data	Bioinformatics, text mining	Vapnik, 1998; Cervantes et al., 2020; Noble 2006; Provost & Fawcett 2013;
Tree-based Classifiers	**Decision Tree**	Recursive binary splitting based on feature values	Low	Medium	Easy to interpret, fast to train	Risk modelling, customer segmentation	Mohri et al., 2012; Song & Lu 2015; Friedman, Hastie, & Tibshirani, 2001

	Random Forest	Ensemble of decision trees with bootstrapped data and random features	Medium	High	Reduces overfitting, robust to noise	Credit risk, fraud detection	Florez-Lopez & Ramon Jeronimo, 2015; Kruppa et al., 2013; Xia et al., 2017; Kotsiantis 2007; Malekipirbazari & Aksakalli, 2015; Genuer et al., 2017
	AdaBoost	Combines weak learners iteratively to reduce bias	Medium	High	Improves weak models, handles imbalance	Marketing analytics, churn prediction	Wang, 2012; Freund & Schapire, 1997; Wen et al., 2015; Khammari et al., 2005; Liu et al., 2005; Wen et al., 2015; Huang et al., 2020; Rajesh & Dhuli, 2018; Ruan & Yin, 2009; al., 2005; Finlay, 2011; Ma et al., 2018
	Gradient Boosting Machine	Sequentially builds trees minimizing loss function	Medium–High	High	High accuracy	Ranking, regression, classification	Haykin, 2008
	XGBoost	Optimized GBM with regularization and parallelism	Medium	Very High	Fast, accurate, handles missing values well	Data science competitions, credit risk	Daoud, 2019; Chen & Guestrin, 2016l Xia et al. ;2017; Lu et al., 2019
	LightGBM	GBM variant with leaf-wise tree growth and histogram binning	Low–Medium	Very High	Highly efficient, fast on large data	Credit default prediction, large, structured data	Ke et al., 2017; Ma et al., 2018, Daoud, 2019
	CatBoost	GBM with native support for categorical features	Medium	Very High	No need for one-hot encoding, robust to overfitting	Financial modelling, churn prediction	Dorogush et al., 2018; Dorogush et al., 2018; Hancock & Khoshgoftaar, 2020; Xia et al., 2019

Deep Learning	**Artificial Neural Network**	Multi-layer perceptron learning non-linear patterns	High	High	Captures complex non-linearities, adaptable	Time series forecasting, speech/image recognition	Haykin,2008

Conclusions and Future Research

Feature selection (FS) and classification remain indispensable pillars of machine learning (ML) for peer-to-peer (P2P) lending, where accurate borrower risk assessment and loan default prediction directly affect platform sustainability, investor trust, and long-term market viability. FS enhances predictive accuracy by isolating the most relevant borrower attributes, reducing computational overhead, and improving interpretability, crucial for platforms processing large, heterogeneous datasets in real time. By systematically eliminating redundant and noisy features, FS supports the development of credit scoring systems that are not only efficient but also transparent and adaptable to rapid changes in borrower behavior and market conditions. Classification models complement FS by translating optimized borrower attributes into actionable credit risk predictions. Linear classifiers offer interpretability required for regulatory compliance; non-linear methods capture complex borrower–lender interactions; and ensemble approaches, such as random forests and gradient boosting, provide robust performance for high-dimensional, imbalanced loan data.

Yet persistent barriers limit optimal deployment. Many FS techniques struggle with concept drift, data sparsity, and the integration of emerging behavioral or alternative data sources. Classification models continue to face significant challenges, including class imbalance, explainability, and the trade-off between predictive performance and fairness.

Future research should focus on creating hybrid feature selection methods that combine behavioral, transactional, and alternative borrower data, along with adaptive and explainable classification models that remain transparent and accurate as borrower and market conditions change. It is also important to build fairness-focused frameworks to detect and prevent bias in feature selection and classification processes while establishing clear benchmarking standards that address concept idea, class imbalance, regulations, and real-world performance. By tackling these challenges, next-generation P2P lending models can deliver more accurate, fair, and transparent predictions, building trust, supporting sustainable growth, and strengthening the resilience of digital lending systems.

References

Ang, J., Haron, H., & Hamed, H.N. (2015). Semi-supervised SVM-based Feature Selection for Cancer Classification using Microarray Gene Expression Data. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, 468–477. https://doi.org/10.1007/978-3-319-19066-2_45

Archer, K. J., & Kimes, R. V. (2008). Empirical characterization of random forest variable importance measures. Computational Statistics & Data Analysis, 52(4), 2249 –2260. https://doi.org/10.1016/j.csda.2007.08.015

Asnicar, F., Thomas, A. M., Passerini, A., et al. (2024). Machine learning for microbiologists. Nature Reviews Microbiology, 22, 191–205

Barkia, H., Elghazel, H., & Aussem, A. (2011). Semi-supervised feature importance evaluation with ensemble learning. Proceedings of the 2011 IEEE 11th International Conference on Data Mining, 31–40. https://doi.org/10.1109/ICDM.2011.129

Bellal, F., Elghazel, H.,& Aussem, A. (2012). A semi-supervised feature ranking method with ensemble learning. Pattern Recognition Letters, 33 1426–1433. http://dx.doi.org/10.1016/j.patrec.2012.03.001

Bishop, C.M. (2006). Pattern recognition and machine learning. New York, Springer. http://dx.doi.org/10.1117/1.2819119

Boulesteix, A. L., Bender, A., Lorenzo Bermejo, J., & Strobl, C. (2012). Random forest Gini importance favors SNPs with large minor allele frequency: Impact, sources and recommendations. Briefings in Bioinformatics,13(3),292–304. http://dx.doi.org/10.1093/bib/bbr053

Branco, P., Torgo, L., & Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. ACM Computing Surveys, 49(2), 1–50. https://doi.org/10.1145/2907070

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. Computers and Electrical Engineering, 40 (1), 16-28. https://doi.org/10.1016/j.compeleceng.2013.11.024

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., et al. (2020). A comprehensive survey on support vector machine classification: Applications, challenges, and trends. Neurocomputing, 408, 189–215. https://doi.org/10.1016/j.neucom.2020.02.091

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785 –794. http://dx.doi.org/10.1145/2939672.2939785

Chowdhury, A., & Alspector, J. (2003). Data duplication: An imbalance problem? In ICML'2003 Workshop on Learning from Imbalanced Data Sets II, Washington, DC.

Daoud, E. A. (2019). Comparison between XGBoost, LightGBM and CatBoost using a home credit dataset. International Journal of Computer and Information Engineering, 13(1), 6 –10.
https://api.semanticscholar.org/CorpusID:159037080

Dhal, P., & Azad, C. (2022). A comprehensive survey on feature selection in the various fields of machine learning. Applied Intelligence, 52(8), 4543–4581.
https://link.springer.com/article/10.1007/s10489-021-02550-9

Dietterich, T.G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization. Machine Learning, 40(2),139-157.
https://api.semanticscholar.org/CorpusID:12394453

Dietterich, T.G. (1997). Machine-learning research: four current directions. AI Magazine, 18(4), 97-136. https://doi.org/10.1609/aimag.v18i4.1324

Dong. G., & Liu, H. (2018). Feature engineering for machine learning and data analytics. http://dx.doi.org/10.1201/9781315181080

Dorogush, A. V., Ershov, V., & Gulin., A. (2018). CatBoost: Gradient boosting with categorical features support. http://dx.doi.org/10.48550/arXiv.1810.11363

Dubey, H., & Pudi, V. (2013). Class-based weighted k-nearest neighbour over imbalanced datasets. In J. Pei, V. S. Tseng, L. Cao, H. Motoda, & G. Xu (Eds.), Advances in Knowledge Discovery and Data Mining: PAKDD 2013 (Vol. 7819, pp. 323–334). Springer. https://doi.org/10.1007/978-3-642-37456-2_26

Dunsin, D., Ghanem, M. C., Ouazzane, K., & Vassilev, V. (2025). Reinforcement learning for an efficient and effective malware investigation during cyber incident response. High-Confidence Computing, 100299. https://doi.org/10.1016/j.hcc.2025.100299

Ferreira A. J. &., Figueiredo, M. A.T. (2012). Efficient feature selection filters for high-dimensional data, Pattern Recognition Letters, 33(13), 1794-1804. https://doi.org/10.1016/j.patrec.2012.05.019

Finlay, S. (2011). Multiple classifier architectures and their application to credit risk assessment. European Journal of Operational Research, 210(2), 368 –378 . https://doi.org/10.1016/j.ejor.2010.09.029

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119 –139.
https://doi.org/10.1006/jcss.1997.1504

Friedman, J., Hastie, T., & Tibshirani, R. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics) Springer, New York

Ghahramani, Z. (2004). Unsupervised Learning. In: Bousquet, O., von Luxburg, U., & Rätsch, G. (eds) Advanced Lectures on Machine Learning. ML 2003. Lecture Notes in Computer Science, 3176. Springer, Berlin, Heidelberg

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3, 1157 –1182. http://dx.doi.org/10.1162/153244303322753616

Han Y., Park K., & Lee Y.K. (2011). Confident wrapper-type semi-supervised feature selection using an ensemble classifier, in: Proceedings of the 2011 2nd Artificial Intelligence, Management Science and Electronic Commerce, AIMSEC, 4581–4586. http://dx.doi.org/10.1109/AIMSEC.2011.6010202

Han Y., Yang Y., Yan Y., Ma Z., Sebe N., & Member S. (2015). Semi supervised feature selection via spline regression for video semantic recognition, IEEE Transaction Neural Network Learning System, (26), 252–264. http://dx.doi.org/10.1109/TNNLS.2014.2314123

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In International Conference on Intelligent Computing (pp. 878–887). Springer. https://api.semanticscholar.org/CorpusID:12126950

Hancock, John & Khoshgoftaar, Taghi. (2020). CatBoost for big data: an interdisciplinary review. Journal of Big Data. 7. 10.1186/s40537-020-00369-8.

Haykin, S. (2008) Neural Networks and Learning Machines, 3rd edition, Pearson Education, Inc., Upper Saddle River, New Jersey 07458.

Hilbe, J.M. (2009). Logistic Regression Models (1st ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9781420075779

Hosmer, Jr., &, David W. (2013). Stanley Lemeshow, and Rodney X. Sturdivant. Applied logistic regression, 398. John Wiley & Sons. DOI:10.1002/9781118548387

Huang, Q., Chen, Y., Liu, L., Tao, D., & Li, X. (2020). On combining biclustering mining and AdaBoost for breast tumor classification. IEEE Transactions on Knowledge and Data Engineering, 32(4), 728 –738. http://dx.doi.org/10.1109/TKDE.2019.2891622

Ishwaran, H., (2007). Variable importance in binary regression trees and forests. Electronic Journal of Statistics, 1, 519-537. http://dx.doi.org/10.1214/07-EJS039

Janitza, S., Tutz, G., & Boulesteix, A.-L. (2016). Random forest for ordinal responses: Prediction and variable selection. Computational Statistics & Data Analysis, 96, 57 –73. http://dx.doi.org/10.1016/j.csda.2015.10.005

John, G. H., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (pp. 338–345). Morgan Kaufmann Publishers Inc.

Jung, S. U., Kim, D. H., An, K. H., & Chung, M. J. (2005). Efficient rectangle feature extraction for real-time facial expression recognition based on AdaBoost. In 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1941 – 1946. https://doi.org/10.1109/IROS.2005.1545534

Kazemitabar, S.J., Amini, A.A., Bloniarz, A., & Talwalkar; A. (2017). Variable Importance using Decision Trees. 31st Conference on Neural Information Processing Systems (NIPS)

Ke, Guolin & Meng, Qi & Finley, Thomas & Wang, Taifeng & Chen, Wei & Ma, Weidong & Ye, Qiwei & Liu, Tie-Yan. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree.

Khammari, A., Nashashibi, F., Abramson, Y., & Laurgeau, C. (2005). Vehicle detection combining gradient analysis and AdaBoost classification. In 2005 IEEE Intelligent Transportation Systems, 66 –71. https://doi.org/10.1109/ITSC.2005.1520202

Kleinbaum, D. G., & Klein, M. (2010). Logistic regression (statistics for biology and health) (3rd ed.). New York, NY: Springer-Verlag, New York Inc. http://dx.doi.org/10.1007/978-1-4419-1742-3

Kozodoi, N., Lessmann, S., Papakonstantinou, K., Gatsoulis, Y., & Baesens, B. (2019). A multi-objective approach for profit-driven feature selection in credit scoring. Decision Support Systems, 120, 106 –117. http://dx.doi.org/10.1016/j.dss.2019.03.011

Liang, D., Tsai, C.-F., & Wu, H.-T. (2015). The effect of feature selection on financial distress prediction. Knowledge-Based Systems, 73, 289 –297. http://dx.doi.org/10.1016/j.knosys.2014.10.010

Liu, J., Li, D., Shan, W., & Liu, S. (2024). A feature selection method based on multiple feature subsets extraction and result fusion for improving classification performance. Applied Soft Computing, 150, 106498. http://dx.doi.org/10.1016/j.asoc.2023.111018

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F.E. (2017). A survey of deep neural network architectures and their applications. Neurocomputing, 234, 11 -26 . https://doi.org/10.1016/j.neucom.2016.12.038

Liu, T., Zheng, N., Zhao, L., & Cheng, H. (2005). Learning based symmetric features selection for vehicle detection. In Intelligent Vehicles Symposium, 124 –129. https://doi.org/10.1109/IVS.2005.1505089

Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In Advances in Neural Information Processing Systems, 3146 –3154

Lu, T., Zhang, Y., & Li, B. (2019). The value of alternative data in credit risk prediction: evidence from a large field experiment. In Fortieth International Conference on Information Systems

Muddasar, N., Syed, R., & Antonio, C. (2020). A gentle introduction to reinforcement learning and its application in different fields. IEEE. http://dx.doi.org/10.1109/ACCESS.2020.3038605

Noble, W. S. (2006). What is a support vector machine? Nature Biotechnology, 24(12), 1565-1567. http://dx.doi.org/10.1038/nbt1206-1565

Ma, X., Sha, J., Wang, D., Yu, Y., Yang, Q., & Niu, X. (2018). Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning. Electronic Commerce Research and Applications, 31, 24 –39. https://doi.org/10.1016/j.elerap.2018.08.002

Ma, L., Zhao, X., Zhou, Z., & Liu, Y. (2018). A new aspect on P2P online lending default prediction using meta-level phone usage data in China. Decision Support Systems, 111, 60 –71. http://dx.doi.org/10.1016/j.dss.2018.05.001

Malekipirbazari, M., & Aksakalli, V. (2015). Risk assessment in social lending via random forests. Expert Systems with Applications, 42(10), 4621 -4631. https://doi.org/10.1016/j.eswa.2015.02.001

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). Foundations of Machine Learning. Cambridge, MIT Press. http://dx.doi.org/10.1007/s00362-019-01124-9

Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press

Patil, P. R., Parasar, D., & Charhate, S. (2024). Wrapper-based feature selection and optimization-enabled hybrid deep learning framework for stock market prediction. International Journal of Information Technology & Decision Making, 23(1), 475–500. http://dx.doi.org/10.1142/S0219622023500116

Park, H. (2013). An introduction to logistic regression: from basic concepts to interpretation with particular attention to nursing domain. Journal of Korean Academy of Nursing, 43, 154-164. http://dx.doi.org/10.4040/jkan.2013.43.2.154

Peng, C.Y.J., Lee, K. L., & Ingersoll, G.M. (2002). An introduction to logistic regression analysis and reporting. The Journal of Educational Research, 96(1), 3-15. http://dx.doi.org/10.1080/00220670209598786

Provost, F., & Fawcett, T. (2013). Data Science for Business: What you need to know about data mining and data-analytic thinking. O'Reilly Media, Inc., Sebastopol.

Rajesh, K. N. V. P. S., & Dhuli, R. (2018). Classification of imbalanced ECG beats using re-sampling techniques and AdaBoost ensemble classifier. Biomedical Signal Processing and Control, 41, 242 –254. http://dx.doi.org/10.1016/j.bspc.2017.12.004

Rennie, J. D. (2001). Improving multi-class text classification with Naive Bayes. Technical Report AITR, 4.

Ruan, J., & Yin, J. (2009). Multi-pose face detection using facial features and AdaBoost algorithm. In 2009 Second International Workshop on Computer Science and Engineering, 2, 31 –34. http://dx.doi.org/10.1109/WCSE.2009.760

Sadeghian, Z., Akbari, E., Nematzadeh, H., & Motameni, H. (2023). A review of feature selection methods based on meta-heuristic algorithms. Journal of Experimental & Theoretical Artificial Intelligence, 37(1), 1–51.

Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. Bioinformatics, 23(19), 2507–2517. http://dx.doi.org/10.1093/bioinformatics/btm344

Sheikhpour, R., Sarram, M.A., Gharaghani, S., & Chahooki. M.A.Z. (2017). A survey on semi-supervised feature selection methods. Pattern Recognit 64(2016):141–158. https://doi.org/10.1016/j.patcog.2016.11.003

Song X., Zhang, J., Han, Y., & Jiang, J. (2014). Semi-supervised feature selection via hierarchical regression for web image classification. Multimedia System, 22 (1), 41-49. https://doi.org/10.1007/s00530-014-0390-0

Song, Y.Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. Shanghai Arch Psychiatry, 27(2), 130-5. http://dx.doi.org/10.11919/j.issn.1002-0829.215044

Strobl, C., Boulesteix, A. L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and solutions. BMC Bioinformatics, 8, 25. http://dx.doi.org/10.1186/1471-2105-8-25

Strobl, C., Boulesteix, A. L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. BMC Bioinformatics, 9, 307. http://dx.doi.org/10.1186/1471-2105-9-307

Sutton R., & Barto A. (2015). Reinforcement learning: an introduction, vol. 2, 2nd ed. Cambridge, MA, USA: MIT Press

Tan, S. (2005). Neighbour-weighted k-nearest neighbour for unbalanced text corpus.

Expert Systems with Applications, 28(4), 667–671. https://doi.org/10.1016/j.eswa.2004.12.023

Theng, D., & Bhoyar, K. K. (2024). Feature selection techniques for machine learning: A survey of more than two decades of research. Knowledge and Information Systems, 66, 1575–1637. http://dx.doi.org/10.1007/s10115-023-02010-5

Tian, J., Gu, H., & Liu, W. (2011). Imbalanced classification using support vector machine

ensemble. Neural Computing and Applications, 20(2), 203–209. http://dx.doi.org/10.1007/s00521-010-0349-9

Vapnik, V. (1998). Statistical learning theory. Wiley.

Wang, R. (2012). AdaBoost for feature selection, classification and its relation with SVM, a review. Physics Procedia, 25, 800 –807. https://doi.org/10.1016/j.phpro.2012.03.160

Weiss, G. M. (2004). Mining with rarity: A unifying framework. ACM SIGKDD Explorations Newsletter, 6(1), 7–19. https://doi.org/10.1145/1007730.1007734

Wen, X., Shao, L., Xue, Y., & Fang, W. (2015). A rapid learning algorithm for vehicle classification. Information Sciences, 295, 395 –406. http://dx.doi.org/10.1016/j.ins.2014.10.040

Xia, Y., Liu, C., Li, Y., & Liu, N. (2017). A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. Expert Systems with Applications, 78, 225 –241. https://doi.org/10.1016/j.eswa.2017.02.017

Xia, Y., Liu, C., & Liu, N. (2017). Cost-sensitive boosted tree for loan evaluation in peer-to-peer lending. Electronic Commerce Research and Applications, 24, 30-49. https://doi.org/10.1016/j.elerap.2017.06.004

Xia, Y., He, L., Li, Y., Liu, N., & Ding, Y. (2019). Predicting loan default in peer-to-peer lending using narrative data. Journal of Forecasting, 39, 260 –280. http://dx.doi.org/10.1002/for.2625

Xie, J., Sage, M., & Zhao, Y. F. (2023). Feature selection and feature learning in machine learning applications for gas turbines: A review. Engineering Applications of Artificial Intelligence, 117(A), 105591. http://dx.doi.org/10.1016/j.engappai.2022.105591

Yin, L., Ge, Y., Xiao, K., et al. (2013). Feature selection for high-dimensional imbalanced data. Neurocomputing, 105, 3–11. http://dx.doi.org/10.1016/j.neucom.2012.04.039

Zhang, H. (2005). Exploring conditions for the optimality of Naïve Bayes. International Journal of Pattern Recognition and Artificial Intelligence, 19(2),183-198. http://dx.doi.org/10.1142/S0218001405003983

Zhou, J., Li, W., Wang, J., Ding, S., & Xia, C. (2019). Default prediction in P2P lending from high dimensional data based on machine learning. Physica A: Statistical Mechanics and its Applications, (534), 122370. https://doi.org/10.1016/j.physa.2019.122370

Zhu, X., & Goldberg, A.B. 2009. Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning, 3(1), 1-30. http://dx.doi.org/10.2200/S00196ED1V01Y200906AIM006