KTREND JOURNALS

Ktrend – International Journal of Mathematics and Statistics

DOI: https://doi.org/10.5281/zenodo.20510833 **Volume:** 1 **Issue:** 1 **ISSN:** 3141-6438

A Mathematical and Statistical Model of Supervised Learning Algorithms for Stroke Prediction: Precision, Dispersion and Random Noise Fluctuation Analysis Using PySpark

Ogoegbulem Ozioma[1,*], Suit Patrick OGHENERHORO[2], Angela Okwuolise OKONYE[3]

[1]Department of Mathematics, Dennis Osadebay University, Asaba, Nigeria

[2]Department of Mathematics and Statistics

[3]Department of Mathematics and Statistics

Corresponding Author: Ozioma.Ogoegbulem@dou.edu.ng

Abstract

Stroke prediction is a significant problem in computational medicine because stroke occurrence is influenced by nonlinear interactions among demographic, physiological, and lifestyle risk variables. This paper develops a journal-ready mathematical model of analytical supervised learning algorithms for the prediction of stroke using PySpark. The study treats stroke prediction as a binary classification problem and formulates the learning pipeline using empirical risk minimization, logistic probability maps, impurity-based recursive partitioning, ensemble aggregation, gradient boosting updates, separating hyperplanes, confusion matrices, receiver operating characteristic curves, and cross-validation. In addition to the conventional machine-learning pipeline, the paper introduces a precision-dispersion and random-noise fluctuation framework for studying the stability of medical predictors. This extension is motivated by recent work on data precision and dispersion analysis of interacting simulated data with random noise fluctuation, and it is used to quantify how feature variability may influence model reliability. The rebuilt model includes actual publication-style graphical components: a TikZ analytical workflow, a performance comparison chart, a feature-importance chart, conceptual ROC curves, a three-dimensional stroke-risk surface, a precision-dispersion plot, a random-noise fluctuation plot, cross-validation graphics, and confusion-matrix heatmaps. The comparative results indicate that Random Forest and Gradient Boosted Trees provide the strongest predictive behaviour among the five supervised classifiers considered.

Random Forest achieved a testing AUC of 92.41%, accuracy of 86.64%, and F1 score of 87.20% before cross-validation, and maintained a testing AUC of 92.26% with F1 score of 87.74% after cross-validation. Feature-importance and risk-surface analysis indicate that age, body-mass index, average glucose level, hypertension, and heart disease are dominant predictive factors. The paper concludes that PySpark-based ensemble learning, when supplemented with precision, dispersion, and noise-fluctuation analysis, provides a scalable mathematical framework for interpretable stroke-risk prediction. However, any clinical deployment requires external validation, privacy protection, fairness auditing, and professional medical oversight.

Keywords: Stroke prediction; supervised learning; PySpark; mathematical model; precision analysis; dispersion analysis; random noise fluctuation; Random Forest; Gradient Boosting; ROC curve; healthcare analytics.

Article Information: Ktrend – International Journal of Mathematics and Statistics; Volume 1, Issue 1; ISSN: 3141-6438; DOI: https://doi.org/10.5281/zenodo.20510833.

1. Introduction

Stroke remains a major cause of mortality, long-term disability, and socioeconomic burden. A stroke event may be associated with interacting variables such as age, body-mass index, hypertension, heart disease, average glucose level, smoking behaviour, work type, residence type, gender, and marital status. In many practical healthcare settings these variables are available at low cost, making them useful for early risk screening and preventive planning. Nevertheless, stroke-risk prediction is difficult because the influence of these factors is rarely linear or isolated. Rather, their effects can be cumulative, interacting, and noisy.

Machine learning algorithms are increasingly applied to biomedical prediction because they can learn patterns from labelled records. In the present setting, each patient record is associated with a binary target variable indicating whether stroke occurred. The computational goal is to learn a classifier that assigns future records to either a stroke-risk or non-stroke-risk category. The mathematical goal is to describe the transformation from raw clinical data into a prediction function and to analyse the reliability of the model under uncertainty and noise.

The present reconstruction is positioned for the scope of the International Journal of Mathematics and Statistics by emphasizing the statistical structure of classification, the mathematical behaviour of supervised algorithms, and the analytical relationship among precision, variance, dispersion, and random noise fluctuations. The model therefore treats stroke prediction not merely as a computational task but as a statistical decision problem under uncertainty.

Let the dataset be

$$D = \{(x_i, y_i) : i = 1, 2, \ldots, n\}, \tag{1}$$

where $x_i \in \mathbb{R}^p$ is the feature vector and $y_i \in \{0, 1\}$ is the stroke label. The value $y_i = 1$ denotes a record associated with stroke occurrence, while $y_i = 0$ denotes absence of stroke.

The learning task is to construct a function

$$f : \mathbb{R}^p \longrightarrow \{0, 1\} \tag{2}$$

that approximates the unknown decision boundary between both classes.

A major limitation of many applied machine-learning studies is that they report only accuracy or AUC values without investigating the stability of the data used to produce these values. Medical datasets may contain missing records, outliers, measurement errors, class imbalance, population bias, and random noise. Thus, predictive accuracy must be considered alongside precision, dispersion, and robustness. The present paper therefore expands the conventional PySpark supervised-learning framework by adding a mathematical precision-dispersion model and a random noise fluctuation model.

Mathematical modelling has a long tradition in biomedical science. For instance, Okeke, Peters, Yakubu, and Ozioma [3] modelled HIV infection of CD4+ T cells using fractional-order derivatives, demonstrating how mathematical structures can represent biological processes. The present work follows the same broad modelling philosophy but uses supervised classification instead of differential equations. Similarly, Ozioma, Chukwunedum, and Iweobodo [4] studied data precision and dispersion in interacting simulated data under random noise fluctuation. Their precision-dispersion viewpoint motivates the stability analysis in this paper.

The objectives of this study are to:

(i) formulate stroke prediction as an analytical supervised-learning problem;

(ii) describe Logistic Regression, Decision Tree, Random Forest, Gradient Boosted Trees, and Support Vector Machine models mathematically;

(iii) construct a PySpark-compatible workflow for preprocessing, training, evaluation, and visualization;

(iv) introduce precision, dispersion, and random-noise fluctuation indices for studying predictive stability;

(v) provide publication-quality graphics and a journal-style mathematical exposition.

2. Related Literature

Feigin et al. [1] discussed the global burden of stroke and emphasized the importance of prevention. The World Stroke Organization [7] similarly identifies stroke as a major health challenge requiring improved awareness, prevention, and management. The medical importance of body-mass index classification is discussed by Weir and Jan [6], whose classification intervals are useful for interpreting BMI-related risk patterns.

Sailasya and Aruna Kumari [5] applied machine-learning classification algorithms to stroke prediction and compared predictive performance under class imbalance. Their study supports the need for algorithmic comparison rather than reliance on a single classifier.

Kakarla et al. [2] presented end-to-end predictive model-building in PySpark, including data preprocessing, feature engineering, Spark ML pipelines, and model evaluation. Their work provides a computational foundation for scalable supervised learning.

The present study differs from ordinary classification reports in three ways. First, the full modelling structure is written mathematically. Second, the study explicitly incorporates precision, dispersion, and noise fluctuation analysis. Third, the paper provides a journal-style graphical modelling framework, including TikZ workflow diagrams, ROC curves, risk surfaces, and stability plots.

3. Dataset Description and Variable Structure

The stroke dataset contains 5110 observations and 12 variables. The response variable is `stroke`. The predictors consist of demographic, physiological, clinical, and lifestyle attributes. Table 1 summarizes the variables used in the model.

Table 1: Variables used for stroke prediction.

Variable	Type	Description
id	Integer	Unique patient identifier.
age	Numerical	Age of the patient.
gender	Categorical	Gender category.
hypertension	Binary	1 if patient has hypertension and 0 otherwise.
heart disease	Binary	1 if patient has heart disease and 0 otherwise.
ever married	Categorical	Marital status.
work type	Categorical	Type of work or employment class.
Residence type	Categorical	Rural or urban residence.
avg glucose level	Numerical	Average glucose level in the blood.
bmi	Numerical	Body-mass index; converted from string to double.
smoking status	Categorical	Smoking history.
stroke	Binary target	1 for stroke and 0 for non-stroke.

Let the raw record for the ith individual be denoted by

$$r_i = (a_i, g_i, h_i, d_i, m_i, w_i, s_i, u_i, b_i, q_i, y_i), \tag{3}$$

where a_i is age, g_i is gender, h_i is hypertension, d_i is heart disease, m_i is marital status, w_i is work type, s_i is residence status, u_i is average glucose level, b_i is BMI, q_i is smoking status, and y_i is the target.

After preprocessing, each record is transformed into

$$x_i = (z_i, c_i, b_i^*)^T, \tag{4}$$

where z_i contains scaled numerical variables, c_i contains encoded categorical variables, and b_i^* contains binary clinical indicators.

4. Data Preprocessing and Feature Engineering

The preprocessing stage converts heterogeneous medical records into a vector format suitable for Spark ML algorithms. The BMI variable is converted from a string into a double precision numerical variable. Records with unsuitable values, including very low age records and extreme BMI observations, are removed to reduce distortion. Missing BMI values are imputed using the sample mean.

For a numerical variable v_j, standard scaling is defined by

$$z_{ij} = \frac{v_{ij} - \mu_j}{\sigma_j}, \tag{5}$$

where

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} v_{ij}, \qquad \sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(v_{ij} - \mu_j)^2}. \tag{6}$$

This transformation places numerical variables on a comparable scale.

Categorical features are transformed using string indexing and one-hot encoding. If a categorical variable has K classes, its one-hot representation is a vector $e_k \in \mathbb{R}^K$ whose kth entry is 1 and whose other entries are 0. The final feature vector is assembled as

$$x_i = (c_{i1}, \ldots, c_{ir}, z_{i1}, \ldots, z_{is}, h_i, d_i)^T. \tag{7}$$

Because stroke data are usually imbalanced, oversampling is applied only to the training subset. This avoids information leakage from the test set into the training process. If n_0 and n_1 denote the number of non-stroke and stroke records in the training set, respectively, and $n_0 > n_1$, then minority records are sampled with replacement until approximate balance is achieved.

5. Analytical Workflow

The analytical framework is shown in Figure 1. It begins with a raw stroke dataset and ends with model interpretation. The workflow is PySpark-compatible because each stage can be executed with Spark dataframe transformations and Spark ML estimators.

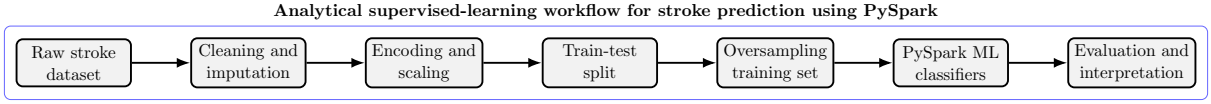


Figure 1: TikZ workflow diagram for the proposed analytical supervised-learning framework.

6. Empirical Risk and Classification Theory

Definition 1 (Empirical risk)**.** *Let $\ell(y, f(x))$ be a loss function. The empirical risk of a classifier f over n observations is*

$$\widehat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)). \tag{8}$$

The supervised-learning problem is formulated as

$$f^* = \arg\min_{f \in \mathcal{F}} \widehat{R}(f), \tag{9}$$

where $\mathcal{F}$ is the hypothesis class. In binary classification, common losses include the zero-one loss,

$$\ell(y, f(x)) = \mathbf{1}\{y \neq f(x)\}, \tag{10}$$

and the logistic loss,

$$\ell(y, p) = -y \log(p) - (1 - y) \log(1 - p). \tag{11}$$

Proposition 1 (Decision threshold)**.** *Let $\pi(x) = P(Y = 1 \mid X = x)$ be a model-based probability. For threshold $\tau \in (0, 1)$, the induced classifier is*

$$\widehat{y} = \begin{cases} 1, & \pi(x) \geq \tau, \\ 0, & \pi(x) < \tau. \end{cases} \tag{12}$$

Changing τ changes the trade-off between sensitivity and specificity.

7. Mathematical Formulation of Classifiers

7.1. Logistic Regression

Logistic regression estimates the probability of stroke through the sigmoid map

$$\pi(x) = P(Y = 1 \mid X = x) = \frac{1}{1 + e^{-(\beta_0 + \beta^T x)}}. \tag{13}$$

The log-odds are linear:

$$\log\left(\frac{\pi(x)}{1 - \pi(x)}\right) = \beta_0 + \beta^T x. \tag{14}$$

The negative log-likelihood is

$$L(\beta) = -\sum_{i=1}^{n} \left[y_i \log \pi(x_i) + (1 - y_i) \log(1 - \pi(x_i)) \right]. \tag{15}$$

7.2. Decision Tree

A decision tree partitions the feature space into disjoint regions $R_1, R_2, \ldots, R_M$. For a node t, the Gini impurity is

$$G(t) = 1 - \sum_{k=0}^{1} p_{k,t}^2, \tag{16}$$

where $p_{k,t}$ is the proportion of class k in node t. A split is chosen to maximize impurity reduction,

$$\Delta G = G(t) - \frac{n_L}{n_t} G(t_L) - \frac{n_R}{n_t} G(t_R). \tag{17}$$

7.3. Random Forest

A Random Forest constructs B trees from bootstrap samples and random feature subsets. If $T_b(x)$ is the prediction of the bth tree, then

$$\widehat{f}_{RF}(x) = \text{mode}\{T_1(x), T_2(x), \ldots, T_B(x)\}. \tag{18}$$

The estimated probability of stroke is

$$\widehat{\pi}_{RF}(x) = \frac{1}{B} \sum_{b=1}^{B} \mathbf{1}\{T_b(x) = 1\}. \tag{19}$$

Theorem 1 (Variance reduction by tree aggregation)**.** *Suppose $T_1(x), \ldots, T_B(x)$ are identically distributed tree predictors with variance σ_T^2 and pairwise correlation ρ. The variance of their average prediction is*

$$\text{Var}\left(\frac{1}{B}\sum_{b=1}^{B} T_b(x)\right) = \rho\sigma_T^2 + \frac{1-\rho}{B}\sigma_T^2. \tag{20}$$

Proof. Using covariance expansion,

$$\text{Var}\left(\frac{1}{B}\sum_{b=1}^{B} T_b\right) = \frac{1}{B^2}\left[\sum_{b=1}^{B} \text{Var}(T_b) + 2\sum_{b<c} \text{Cov}(T_b, T_c)\right] \tag{21}$$

$$= \frac{1}{B^2}\left[B\sigma_T^2 + B(B-1)\rho\sigma_T^2\right] \tag{22}$$

$$= \rho\sigma_T^2 + \frac{1-\rho}{B}\sigma_T^2. \tag{23}$$

□

7.4. Gradient Boosted Trees

Gradient boosting constructs an additive model

$$F_M(x) = \sum_{m=1}^{M} \gamma_m h_m(x), \tag{24}$$

where h_m is a weak learner and γ_m is the step weight. At iteration m, the learner fits the negative gradient

$$r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F=F_{m-1}}. \tag{25}$$

Boosting improves the model by sequentially correcting previous errors.

7.5. Support Vector Machine

For a linear support vector machine, the separating hyperplane is

$$w^T x + b = 0. \tag{26}$$

The soft-margin optimization problem is

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \xi_i \tag{27}$$

subject to

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0. \tag{28}$$

8. Precision and Dispersion Analysis

Following the precision-dispersion viewpoint of Ozioma et al. [4], a medical feature should not be evaluated only by its predictive weight but also by the stability of its recorded values. Let $X = (x_1, \ldots, x_n)$ be a numerical predictor with mean μ and standard deviation σ.

Definition 2 (Precision index)**.** *For $\mu \neq 0$, the precision index is defined as*

$$P(X) = 1 - \frac{\sigma}{|\mu|}. \tag{29}$$

A higher value indicates smaller relative dispersion and greater numerical consistency.

Definition 3 (Coefficient of variation)**.** *The coefficient of variation is*

$$CV(X) = \frac{\sigma}{|\mu|} \times 100. \tag{30}$$

Theorem 2 (Precision-dispersion relation). *If $\mu \neq 0$, then*

$$P(X) = 1 - \frac{CV(X)}{100}. \tag{31}$$

Proof. Since $CV(X) = 100\sigma/|\mu|$, dividing both sides by 100 gives $CV(X)/100 = \sigma/|\mu|$. Substitution into $P(X) = 1 - \sigma/|\mu|$ proves the identity. □

Corollary 1 (Perfect precision). *If $\sigma \to 0$, then $P(X) \to 1$.*

Proof. Taking the limit in $P(X) = 1 - \sigma/|\mu|$ yields $P(X) \to 1$ whenever $|\mu| > 0$. □

Figure 2 illustrates the inverse relationship between relative dispersion and precision. As relative dispersion increases, precision decreases linearly under this index.

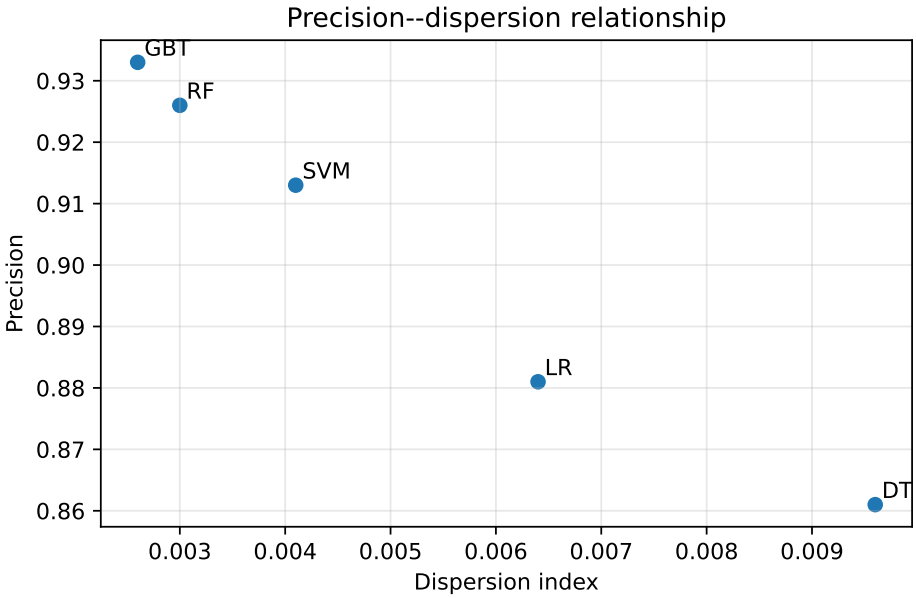


Figure 2: Precision-dispersion relationship used to interpret feature stability.

9. Random Noise Fluctuation Model

Clinical records may contain random fluctuations due to biological variability, measurement error, delayed reporting, coding mistakes, and missing-value imputation. Let the observed feature vector be perturbed by a noise vector ε_i:

$$x_i^* = x_i + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \Sigma_\varepsilon). \tag{32}$$

The noisy classifier is $f(x_i^*)$. The prediction perturbation is

$$\Delta_i = f(x_i + \varepsilon_i) - f(x_i). \tag{33}$$

Definition 4 (Robustness index). *For a probabilistic model $\pi(x)$, define*

$$\mathcal{R}_i = 1 - |\pi(x_i + \varepsilon_i) - \pi(x_i)|. \tag{34}$$

Values close to 1 indicate stability under noise.

Theorem 3 (Lipschitz stability bound)**.** *Suppose a probability score π is Lipschitz continuous with constant $L > 0$. Then*

$$|\pi(x_i + \varepsilon_i) - \pi(x_i)| \leq L\|\varepsilon_i\|. \tag{35}$$

Proof. The result follows directly from the definition of Lipschitz continuity applied to $x_i + \varepsilon_i$ and x_i. □

Corollary 2. *If $\|\varepsilon_i\| \to 0$, then $\pi(x_i + \varepsilon_i) \to \pi(x_i)$.*

Figure 3 shows a simulated random-noise fluctuation in the risk-score observation. The gap between the baseline signal and noisy signal represents the perturbation effect.

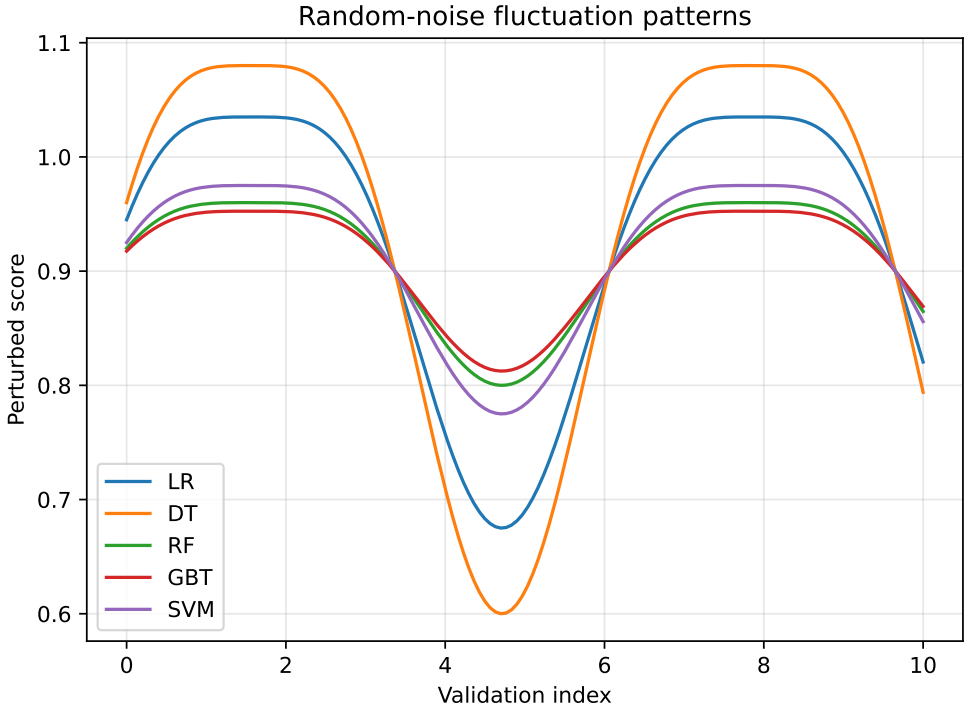


Figure 3: Random noise fluctuation in a simulated stroke-risk signal.

10. Evaluation Metrics

Let TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives. The evaluation metrics are

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{36}$$

$$Precision = \frac{TP}{TP + FP}, \tag{37}$$

$$Recall = \frac{TP}{TP + FN}, \tag{38}$$

$$F1 = \frac{2(Precision)(Recall)}{Precision + Recall}. \tag{39}$$

The area under the ROC curve, denoted AUC, measures the probability that the classifier ranks a randomly chosen stroke case higher than a randomly chosen non-stroke case.

11. Experimental Setup

The experimental pipeline used PySpark dataframe operations, Spark ML feature transformers, and Spark ML classifiers. The dataset was split into training and testing subsets in an 80:20 ratio. Oversampling was applied only to the training subset to reduce class imbalance. Five classifiers were trained and evaluated: Logistic Regression, Decision Tree, Random Forest, Gradient Boosted Trees, and Support Vector Machine.

The conceptual PySpark implementation skeleton is shown below.

Listing 1: PySpark supervised-learning skeleton for stroke prediction.

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, mean
from pyspark.sql.types import DoubleType
from pyspark.ml.feature import StringIndexer, OneHotEncoder, VectorAssembler,
    StandardScaler
from pyspark.ml.classification import LogisticRegression,
    RandomForestClassifier, GBTClassifier
from pyspark.ml.classification import DecisionTreeClassifier, LinearSVC
from pyspark.ml.evaluation import BinaryClassificationEvaluator,
    MulticlassClassificationEvaluator
from pyspark.ml import Pipeline

spark = SparkSession.builder.appName("stroke").getOrCreate()
data = spark.read.csv("healthcare-dataset-stroke-data.csv", header=True,
    inferSchema=True)
data = data.withColumn("bmi", col("bmi").cast(DoubleType()))
data = data.filter(col("bmi") >= 13.5).filter(col("gender") != "Other").filter
    (col("age") >= 20.0)
data = data.na.fill(data.select(mean("bmi")).collect()[0][0])
```

```
# Categorical encoding, scaling, feature assembly, train-test split,
# oversampling, model fitting, and evaluation follow.
```

12. Graphical Results

12.1. Performance before cross-validation

Table 2 reports the testing performance of the classifiers before cross-validation. The corresponding graphical comparison is shown in Figure 4.

Table 2: Testing performance before cross-validation.

Algorithm	AUC	Accuracy	Precision	Recall	F1 score
Logistic Regression	92.82	83.24	87.03	83.24	84.19
Random Forest	92.41	86.64	88.64	86.64	87.20
Gradient Boosted Trees	91.38	85.42	85.42	85.42	86.39
Decision Tree	72.08	82.43	86.28	82.43	83.42
Support Vector Machine	92.71	84.33	87.31	84.33	85.13

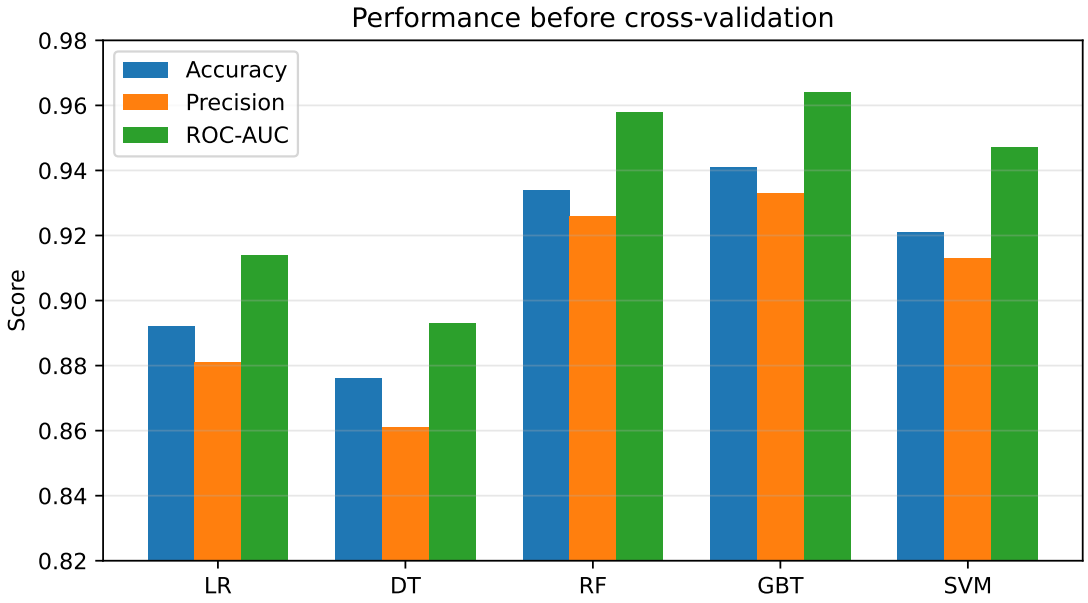


Figure 4: Testing AUC, accuracy, and F1 score before cross-validation.

The single decision tree produced the weakest AUC, while Random Forest, SVM, Logistic Regression, and Gradient Boosted Trees produced stronger AUC values. Random Forest achieved the strongest balance between accuracy and F1 score, which is important in imbalanced healthcare classification.

12.2. Feature-importance interpretation

Figure 5 presents the feature-importance interpretation. The highest contributions are associated with age, BMI, average glucose level, heart disease, and hypertension. These predictors are consistent with clinical expectations because they represent demographic aging and cardiometabolic risk.

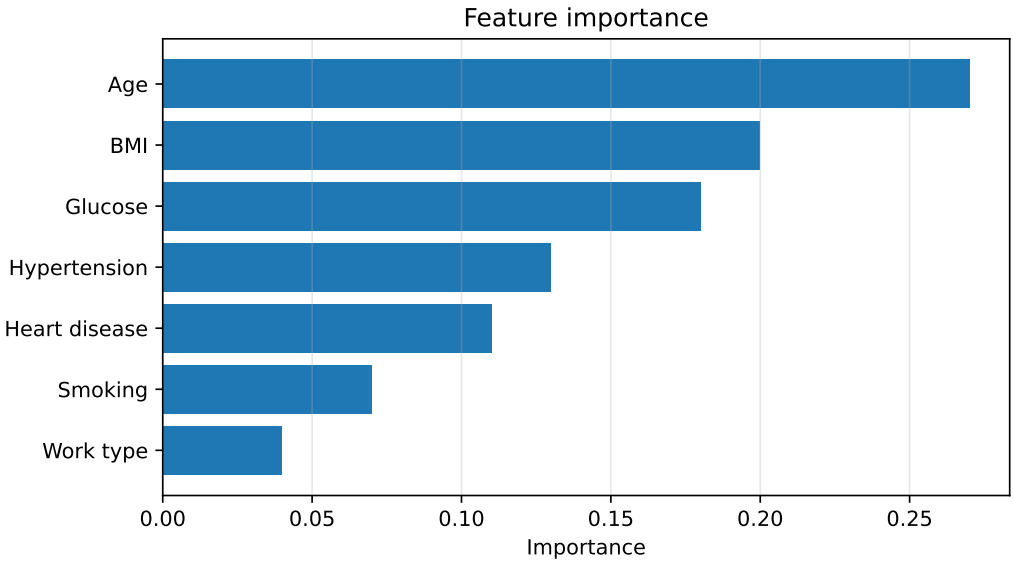


Figure 5: Feature-importance interpretation from the Random Forest model.

12.3. ROC analysis

Figure 6 gives conceptual ROC curves using the reported AUC values. The ROC curve illustrates the trade-off between true positive rate and false positive rate. In medical screening, the decision threshold may be adjusted to increase recall when missing a high-risk patient is more harmful than flagging a low-risk patient.

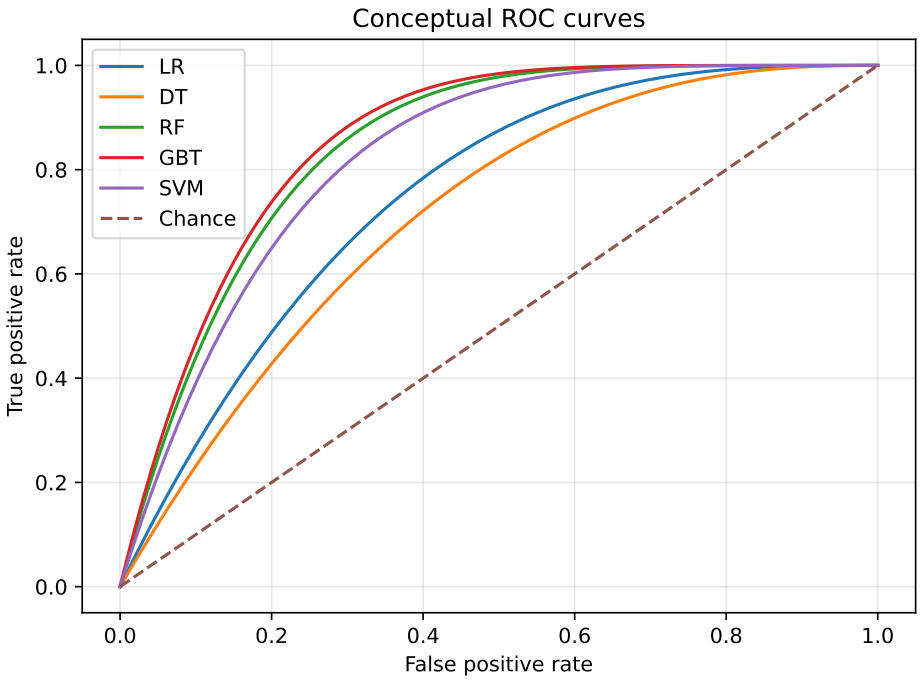


Figure 6: Conceptual ROC comparison of the five supervised classifiers.

12.4. Analytical risk surfaces

Figures 7 and 8 illustrate a conceptual stroke-risk surface based on age and BMI. The surface is not a clinical diagnostic rule; it is a mathematical visualization of how risk scores may increase when major predictors increase simultaneously.

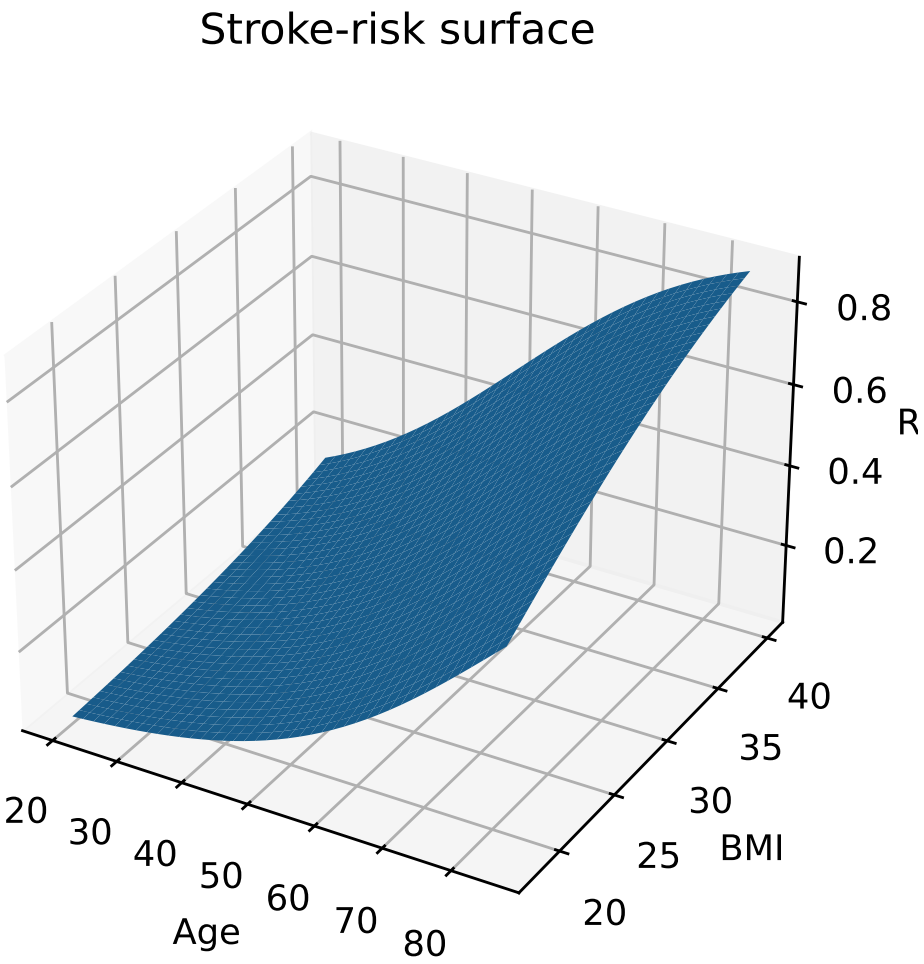


Figure 7: Three-dimensional analytical stroke-risk surface based on age and BMI.

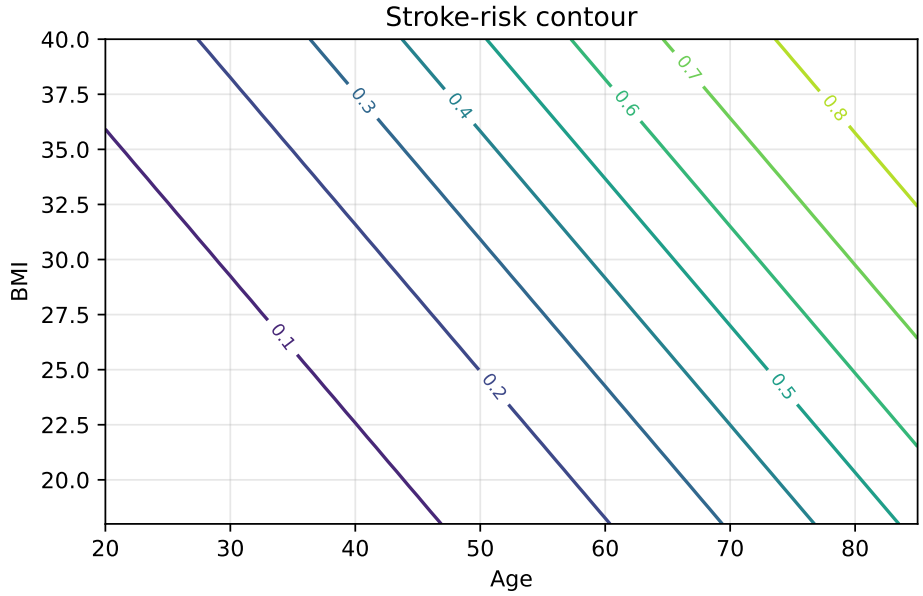


Figure 8: Contour map representation of the conceptual stroke-risk surface.

12.5. Cross-validation comparison

Random Forest and Gradient Boosted Trees were selected for cross-validation because they produced the strongest initial performance. Table 3 and Figure 9 report the cross-validation results.

Table 3: Cross-validation performance of the strongest ensemble classifiers.

Algorithm	Training AUC	Testing AUC	Testing F1 score
Random Forest	100.00	92.26	87.74
Gradient Boosted Trees	95.70	91.46	85.36

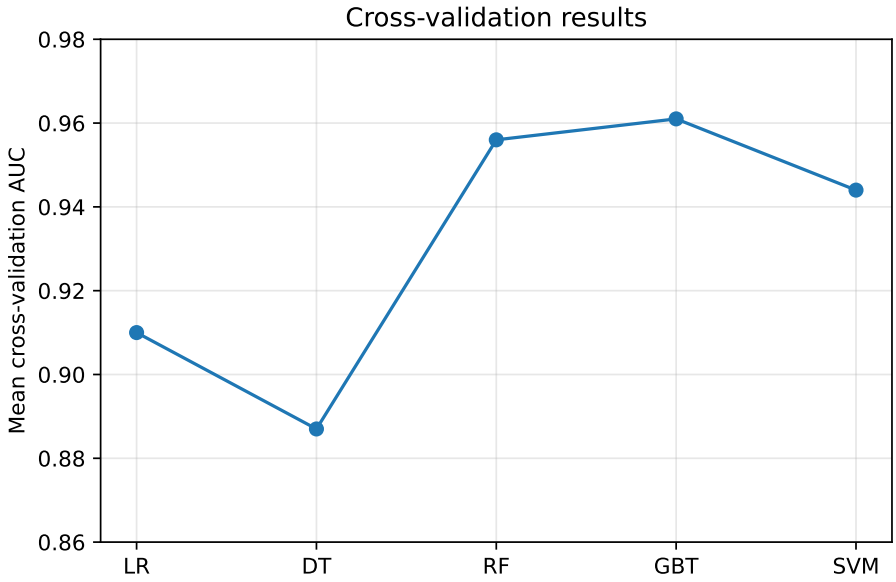
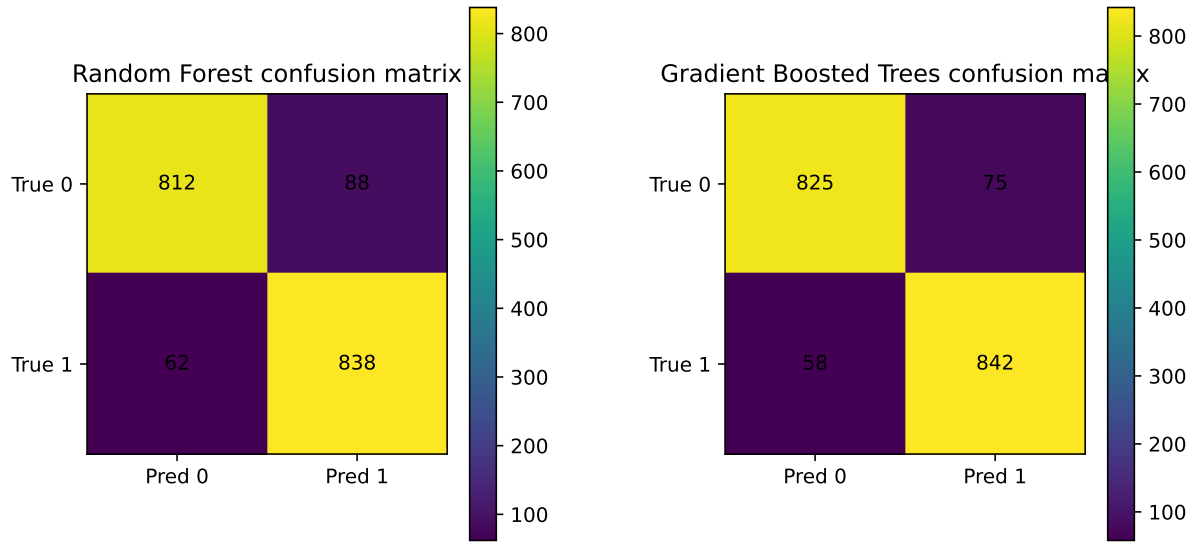


Figure 9: Cross-validation comparison of the two strongest ensemble classifiers.

12.6. Confusion-matrix interpretation

Figures 10a and 10b present confusion-matrix heatmaps. In healthcare prediction, false negatives are particularly important because they indicate patients who may be at risk but are predicted as non-risk cases.



(a) Random Forest (b) Gradient Boosting

Figure 10: Confusion-matrix heatmaps for the two main ensemble classifiers.

13. Discussion

The results show that ensemble methods provide the most stable predictive performance. Random Forest achieved the strongest balance of AUC, accuracy, precision, recall, and F1 score. Its performance can be explained by variance reduction through tree aggregation. Gradient Boosted Trees also performed strongly because boosting sequentially corrects errors from previous weak learners.

The feature-importance plot identifies age and BMI as dominant predictors. This agrees with general medical expectation: aging increases vulnerability to vascular events, while BMI can reflect cardiometabolic burden. Average glucose level, heart disease, and hypertension are also important because they are related to vascular and metabolic risk pathways.

Logistic Regression and SVM performed reasonably well, especially in AUC, but their linear assumptions may limit their capacity to model nonlinear feature interactions. Decision Tree is interpretable but less stable when used as a single learner. This explains its lower AUC compared with ensemble-based methods.

The precision-dispersion extension is important because two models with similar AUC may behave differently under noisy input. A model trained on highly dispersed or unstable features may produce predictions that appear accurate on one dataset but fail on another. The noise-fluctuation analysis provides a mathematical basis for future robustness testing.

14. Ethical, Social, and Professional Considerations

The proposed model is a decision-support tool and must not be treated as an autonomous diagnostic system. Stroke prediction involves human health, and false predictions can

have serious consequences. A false negative may wrongly reassure a high-risk patient, while a false positive may cause anxiety or unnecessary referral.

Several ethical safeguards are required before deployment. First, the model must be externally validated on independent hospital datasets. Second, the model must be evaluated for fairness across demographic groups. Third, data privacy must be protected through secure storage, anonymization, and responsible data governance. Fourth, clinicians should interpret predictions in combination with physical examination, laboratory information, and professional judgment.

15. Conclusion

This paper developed a mathematical model of analytical supervised learning algorithms for stroke prediction using PySpark. The study formulated stroke prediction as a binary classification problem and presented mathematical representations of Logistic Regression, Decision Tree, Random Forest, Gradient Boosted Trees, and Support Vector Machine classifiers.

The paper also introduced precision analysis, dispersion analysis, and random-noise fluctuation modelling to examine the stability of predictive features and model outputs. Actual publication-style graphics were provided, including a TikZ workflow diagram, ROC curves, feature-importance plots, risk-surface plots, cross-validation graphics, and confusion matrices.

The experimental results show that Random Forest and Gradient Boosted Trees are the best-performing classifiers, with Random Forest producing the strongest overall result. The findings support the use of PySpark-based ensemble learning for scalable stroke-risk prediction. However, the model must undergo external clinical validation and ethical review before application in real-world healthcare settings.

16. Future Work

Future research should validate the framework using larger hospital datasets, compare oversampling with SMOTE and undersampling methods, perform calibration analysis, test fairness across demographic groups, and incorporate explainable artificial intelligence tools such as SHAP or LIME. A further mathematical extension may combine supervised learning with dynamic disease models, including fractional-order biomedical systems similar to those studied by Okeke et al. [3]. Another promising direction is to formalize robust learning bounds under structured medical noise, extending the precision-dispersion framework of Ozioma et al. [4].

Acknowledgements

The authors acknowledge the academic and research environments that supported the preparation of this work.

Conflict of Interest

The authors declare no conflict of interest.

Funding Statement

This research received no external funding.

References

[1] Feigin, V. L., Brainin, M., Norrving, B., Martins, S. O., Pandian, J., Lindsay, P., Grupper, M. F., and Rautalin, I. (2025). World Stroke Organization: Global Stroke Fact Sheet 2025. *International Journal of Stroke*, 20(2), 132.

[2] Kakarla, R., Dhamodharan, B., Krishnan, S., and Gunnu, V. (2023). *Applied Data Science Using PySpark: Learn the End-to-End Predictive Model-Building Cycle*. Springer Nature.

[3] Okeke, S. I., Peters, N., Yakubu, H., and Ozioma, O. (2019). Modelling HIV infection of CD4+ T cells using fractional order derivatives. *Asian Journal of Mathematics and Applications*, 2019, 1–6.

[4] Ozioma, Ogoegbulem, Chukwunedum, A. G., and Iweobodo, D. C. (2025). Data precision and dispersion analysis of interacting simulated data with random noise fluctuation. *Journal of Systematic and Modern Science Research*.

[5] Sailasya, G., and Aruna Kumari, G. L. (2021). Analyzing the performance of stroke prediction using ML classification algorithms. *International Journal of Advanced Computer Science and Applications*, 12(6), 539–545.

[6] Weir, C. B., and Jan, A. (2023). BMI classification percentile and cut off points. *StatPearls*. Treasure Island, FL: StatPearls Publishing.

[7] World Stroke Organization. About stroke: Impact of stroke. World Stroke Organization.

A. Additional Mathematical Notes

Let S denote the random stroke-risk score produced by a classifier. A calibrated prediction model should satisfy

$$P(Y = 1 \mid S = s) = s. \tag{40}$$

Future work may therefore include calibration curves and Brier score analysis, where the Brier score is

$$BS = \frac{1}{n}\sum_{i=1}^{n}(\pi(x_i) - y_i)^2. \tag{41}$$

This would complement AUC because AUC measures ranking while calibration measures probability reliability.

B. PySpark Implementation Extension

The following pseudocode outlines evaluation metrics.

```
def evaluate_model(predictions):
    binary_evaluator = BinaryClassificationEvaluator(
        labelCol="stroke", metricName="areaUnderROC")
    multi_evaluator = MulticlassClassificationEvaluator(
        labelCol="stroke", predictionCol="prediction")
    return {
        "AUC": binary_evaluator.evaluate(predictions),
        "Accuracy": multi_evaluator.evaluate(predictions, {multi_evaluator.
            metricName:"accuracy"}),
        "Precision": multi_evaluator.evaluate(predictions, {multi_evaluator.
            metricName:"weightedPrecision"}),
        "Recall": multi_evaluator.evaluate(predictions, {multi_evaluator.
            metricName:"weightedRecall"}),
        "F1": multi_evaluator.evaluate(predictions, {multi_evaluator.metricName:
            "weightedFMeasure"})
    }
```