UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

Machine Learning- DS
M1

# COMPARATIVE STUDY OF CLASSIFICATION ALGORITHMS FOR MEDICAL DECISION SUPPORT ON TABULAR CLINICAL DATA

Hamza Ouba

# TABLE OF CONTENTS

# INTRODUCTION AND OBJECTIVES

- **Goal:** Build and compare supervised classifiers for medical risk detection where false negatives are costly.

- **DATASET**

| Metric | Breast Cancer | Stroke Prediction |
|---|---|---|
| **Samples** | 569 | 5,110 |
| **Features** | 30 (continuous) | 15 (mixed) |
| **Majority Class** | 357 Benign (63%) | 4,861 No-Stroke (95.1%) |
| **Minority Class** | 212 Malignant (37%) | 249 Stroke (4.9%) |
| **Imbalance Ratio** | 1.7:1 (moderate)<br>1.7 benign cases for every 1 malignant case. | 19.5:1 (extreme)<br>19.5 no-stroke cases for every 1 stroke case |
| **Clinical Priority** | Minimize FN (missed cancers) | Minimize FN (missed strokes) |

# EXISTING SOLUTIONS

## UNIFIED ALGORITHM SELECTION

| Algorithm | Type | Why Selected | Expected Strengths |
|---|---|---|---|
| **Logistic Regression** | Linear, probabilistic | Interpretable baseline, clinical transparency | Fast, explainable coefficients |
| **SVM (RBF kernel)** | Non-linear, margin-based | Captures complex boundaries | Robust to correlations |
| **Random Forest** | Ensemble, tree-based | Handles interactions, no scaling needed | Feature importance, flexibility |

# PROPOSED APPROACH AND IMPLEMENTATION

## Unified pipeline design (both datasets)

1. Data loading (ARFF / CSV)
2. Data cleaning & encoding
3. Exploratory Data Analysis
4. Train-test split (80-20, stratified)
5. SMOTE (training only)
6. Feature scaling (LR & SVM only)
7. Model training with 5-fold CV
8. Hyperparameter tuning (GridSearchCV)
9. Test evaluation
10. Threshold calibration

# DATA ANALYSIS AND VISUALIZATION

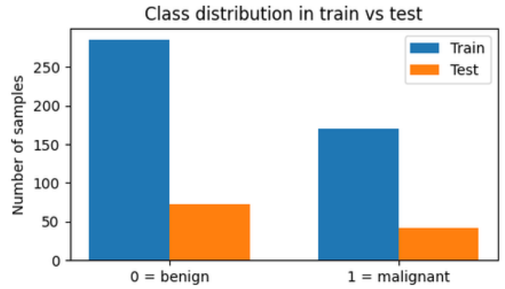
## Breast Cancer (Small dataset)

### Dataset sanity check (raw preview)

- The ARFF file loads correctly: 569 rows, 30 numeric features + 1 target column (diagnosis).
- The target arrives as byte/object values (b'1', b'2') → needs decoding and recoding to 0/1 (benign/malignant).

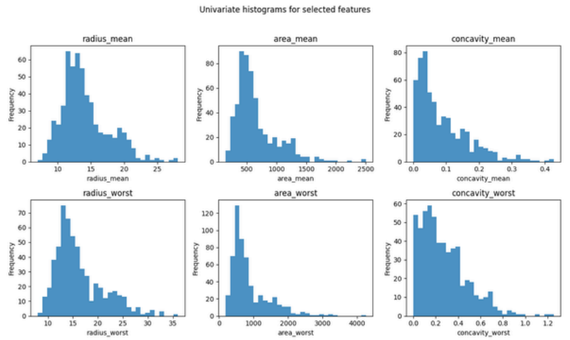
### Target distribution (bar plot)

- Class distribution is moderately imbalanced: 357 benign (≈63%) vs 212 malignant (≈37%).
- Because false negatives are critical clinically, we report recall/F1/ROC-AUC, not only accuracy.



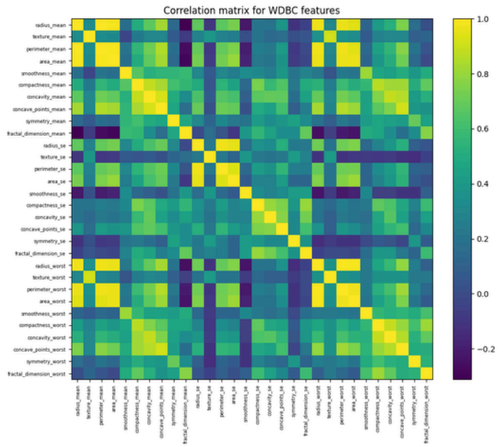

### Univariate histograms (radius/area/concavity + “worst” versions)

- Many key variables (e.g., area_mean, concavity_mean, and *_worst) are right-skewed with long tails.
- Feature scales vary massively (e.g., area features have much larger ranges than texture/shape ratios), which motivates standardization for scale-sensitive models (LR, SVM).




### Correlation heatmap (30×30)

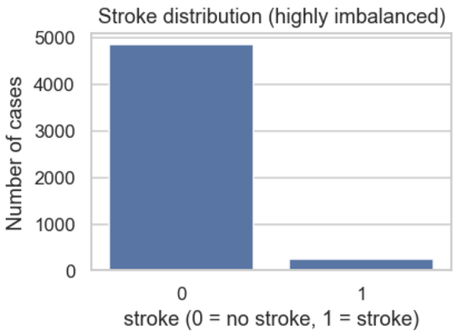
- Strong correlation blocks appear among size-related features (radius, perimeter, area, including “worst” variants).
- Implication: redundancy is fine for tree models, but for linear models we rely on regularization and careful interpretation of coefficients.

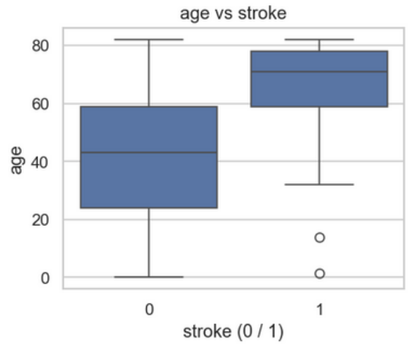


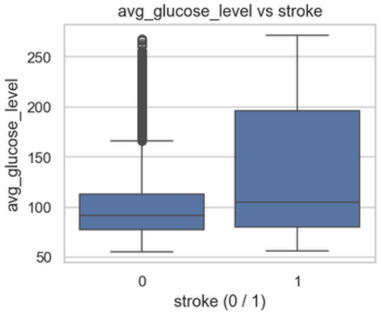
# DATA ANALYSIS AND VISUALIZATION

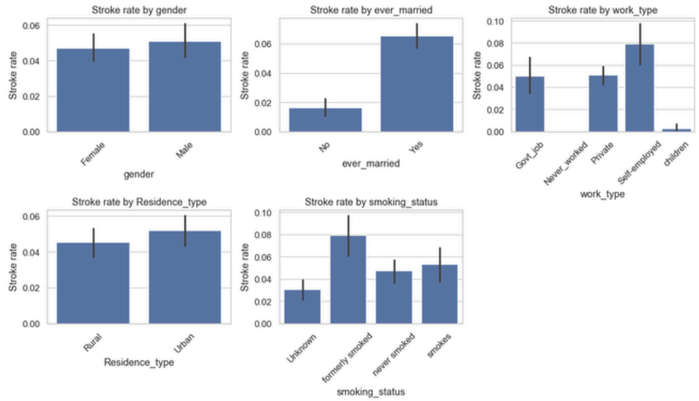
## Stroke Prediction (Large dataset)

- **Dataset overview :** 5110 rows × 15 columns, mixed numeric + categorical features
- **Severe class imbalance:** 95.1% no-stroke (4861) vs 4.9% stroke (249) → accuracy is misleading; focus on recall/precision/PR.
- **Age is the strongest signal:** stroke patients are dramatically older (clear shift + higher median).
- **Glucose is a moderate signal:** stroke patients show higher avg glucose and a heavier right tail.
- **Categorical patterns:** higher stroke rate in ever_married=Yes (age proxy) and formerly smoked; Residence type shows minimal difference.



# PREPROCESSING

## Small data (WDBC Breast Cancer)

- **Load ARFF** → DataFrame with scipy.io.arff.loadarff (ARFF often stores categories as bytes).
- **Decode target bytes (b'1', b'2')** → strings, then recode label: 0 = benign, 1 = malignant.
- **Rename V1…V30** → official WDBC feature names (30 real-valued features, 569 samples).
- **Train/test split (80/20)** with stratification to preserve class proportions.
- SMOTE on training only to balance classes (avoid leakage).
- Standardization (StandardScaler): fit on training, transform on test (critical for LR/SVM stability).
- Use an imbalanced-learn Pipeline so SMOTE + scaling happen correctly inside CV.

## Large data (Stroke prediction)

- **Clean categories:** remove the single invalid gender="Other" row (too rare to learn).
- **Handle missing values:** impute BMI using the median (robust to outliers).
- **Encode categorical variables:** One-Hot Encoding (often with drop='first' to reduce collinearity).
- Stratified train/test split to keep the real-world ~95/5 stroke ratio in both sets.
- Standardize numeric features (important for LR/SVM and distance-based steps).
- SMOTE on training only to learn from more minority examples without contaminating test evaluation.
- Prevent data leakage via Pipeline (preprocessing learned only from train folds during CV/tuning).

# LEARNING ALGORITHMS

## 1) Logistic Regression (L2-regularized)

**Math background (binary classification)**

- Model: $p(y=1 \mid x) = \sigma(w^\top x + b)$, with $\sigma(z) = \frac{1}{1+e^{-z}}$.
- Training objective (regularized negative log-likelihood / log-loss):

$$\min_{w,b} \frac{1}{n} \sum_{i=1}^{n} \Big[ -y_i \log p_i - (1-y_i) \log(1-p_i) \Big] + \lambda \|w\|_2^2$$

where $p_i = \sigma(w^\top x_i + b)$. Scikit-learn +1

- Role of **C** in scikit-learn: $C \propto 1/\lambda$ (larger $C$ = weaker regularization). Scikit-learn

### How it learns (semi-code)

Initialize w, b
repeat until convergence:
  for each sample (x_i, y_i):
    p_i = sigmoid(w·x_i + b)
  Compute gradients:
    grad_w = (1/n) Σ (p_i - y_i) x_i  + 2λ w
    grad_b = (1/n) Σ (p_i - y_i)
  Update parameters (LBFGS):
    (w, b) ← optimizer_step(w, b, grad_w, grad_b)
return w, b

With **L2**, we add a penalty proportional to the **sum of squared weights**:

$$\mathcal{L}_{L2}(w) = \mathcal{L}(w) + \lambda \sum_{j=1}^{p} w_j^2$$

- $\lambda$ controls the strength of regularization
- The bias (intercept) is usually **not regularized**

**Effect of L2 Regularization in Logistic Regression**

- Penalizes large coefficients
- Makes the model more stable
- Reduces variance
- Improves generalization to unseen data

How we implemented it in this project

- Pipeline: SMOTE → StandardScaler → LogisticRegression(penalty="l2", solver="lbfgs")
- Why: scaling stabilizes optimization; SMOTE balances training folds; L2 handles correlated features.

# LEARNING ALGORITHMS

## 2) SVM with RBF kernel (non-linear max-margin)

**Decision function**

$$f(x) = \sum_{i \in SV} \alpha_i y_i K(x_i, x) + b$$

- Prediction is a **weighted sum of similarities**
- Only **support vectors** influence the decision
- $\alpha_i$ are learned coefficients
- $b$ is the bias term

**RBF (Gaussian) kernel**

$$K(x, x') = \exp(-\gamma \|x - x'\|^2)$$

- Measures **similarity** between two points
- Close points → value near **1**
- Far points → value near **0**
- $\gamma$ controls boundary complexity

**Optimization objective (soft margin)**

$$\min_{w,\xi} \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

- Maximizes the **margin** (via $\|w\|^2$)
- Penalizes classification errors
- $C$ controls the trade-off:
  - Large $C$: less tolerance to errors
  - Small $C$: smoother boundary

### How it learns (semi-code)

Given training data (x_i, y_i)

Compute kernel matrix K_ij = exp(-γ ||x_i - x_j||^2)

Solve the convex QP (dual) to find α_i:

maximize Σ α_i - 0.5 ΣΣ α_i α_j y_i y_j K_ij

subject to 0 ≤ α_i ≤ C and Σ α_i y_i = 0

Support vectors = points with α_i > 0

Return f(x) = Σ_{SV} α_i y_i K(x_i, x) + b

How we implemented it in this project

- Pipeline: SMOTE → StandardScaler → SVC(kernel="rbf", probability=True)
- Hyperparameters tuned: C and gamma via GridSearchCV(scoring="roc_auc")

# LEARNING ALGORITHMS

## 3) Random Forest (bagging + random feature selection)

### Math/background intuition

- Train many decision trees on bootstrap samples; each split considers only a random subset of features (max_features).
- Predict by majority vote; randomness reduces variance and controls overfitting.

### How it learns (semi-code)

```
Input: training set D, number of trees T
for t = 1..T:
   D_t = bootstrap_sample(D)          # sample with replacement
   Train a decision tree:
      at each node:
         choose m features at random
         pick best split among those m (impurity decrease)
Store all trees

Predict(x):
   votes = [tree_t.predict(x) for t in 1..T]
   return majority_vote(votes)
```

How we implemented it in this project

- Pipeline: SMOTE → RandomForestClassifier(...) (no scaling needed for trees).
- Tuned: n_estimators, max_depth, max_features with GridSearchCV on ROC-AUC

# HYPER–PARAMETER TUNING

- **Goal**: find the best model configuration that generalizes (not just fits the training data).
- **Method**: Stratified 5-fold Cross-Validation to keep class proportions stable in every fold.
- **Tool**: GridSearchCV to test a grid of hyper-parameters and select the best one using the mean CV score.
- **Main selection metric:** ROC-AUC (threshold-independent, good for imbalanced data).
- **Secondary metrics (reported):** accuracy, precision, recall, F1 → because FN vs FP matters clinically.
- **Leakage-safe tuning:** SMOTE + scaling are placed inside the Pipeline, so they are applied only on the training fold during CV.

**Parameter grids used**

- **Logistic Regression:** tune C (regularization strength), keep L2 penalty.
- **SVM RBF:** tune C and gamma (controls margin vs errors and kernel flexibility).
- **Random Forest:** tune n_estimators, max_depth, max_features (controls variance, complexity, and feature randomness).

# EVALUATION METHODOLOGY

- **Train/Test split**: 80/20 stratified split to preserve class ratios in both sets.
- **Cross-validation**: 5-fold StratifiedKFold on the training set to estimate generalization.
- **Metrics reported:** Accuracy, Precision, Recall, F1 + ROC-AUC (ranking quality).
- **Imbalanced data focus (Stroke):** use Precision–Recall curve + Average Precision because ROC can look optimistic when positives are rare.
- **Threshold tuning:** sweep probability thresholds to pick an operating point (reduce FN vs reduce FP)

# TEST & RESULTS

## LARGE DATASET (STROKE)

| **Model (Large Stroke)** | **Test @ thr = 0.50** (Acc / Prec / Rec / F1 / AUC) | **Confusion Matrix** (TN FP / FN TP) | **Direct Interpretation** |
|---|---|---|---|
| **Logistic Regression** | 0.7368 / 0.1338 / 0.8000 / 0.2292 / 0.8406 | 713 259 / 10 40 | Good recall (detects 40/50 strokes) but huge FP (259) → many false alarms |
| **SVM RBF (best)** | 0.7417 / 0.1361 / 0.8000 / 0.2326 / 0.8372 | 718 254 / 10 40 | Very similar to LR: same recall (0.80), slightly fewer FP (254 vs 259) |
| **Random Forest (tuned)** | 0.9200 / 0.2000 / 0.2200 / 0.2090 / 0.7795 | 928 44 / 39 11 | High accuracy but misleading: misses majority of strokes (FN = 39/50) → medically unacceptable |

## THRESHOLD TUNING

### LOGISTIC REGRESSION (BEST AVAILABLE MODEL)

| **Threshold** | **Precision** | **Recall** | **F1** | **FN** | **FP** | **Clinical Interpretation** |
|---|---|---|---|---|---|---|
| **0.50 (default)** | 0.134 | 0.8 | 0.229 | 10 | 259 | High sensitivity, too many false alarms (259) |
| **0.82 (F1-optimal)** | 0.269 | 0.56 | 0.364 | 22 | 76 | Best F1, but misses 44% of strokes |
| **0.56 (Youden)** | 0.158 | 0.8 | 0.264 | 10 | 213 | Best compromise: same recall, fewer FP |
| **0.002 (Recall ≥90%)** | 0.049 | 1 | 0.093 | 0 | 972 | Flags everyone – unusable |

# TEST & RESULTS

## SMALL DATASET (BREAST CANCER)

| Model | Test Metrics (thr = 0.50) | Confusion Matrix (TN FP / FN TP) | Interpretation |
|---|---|---|---|
| **Logistic Regression** | Acc: 0.9737 Prec: 0.9756<br>Rec: 0.9524 F1: 0.9639 AUC: 0.9944 | [71, 1][2, 40] | Very strong separation (ROC near top-left). Only 2 missed cancers (FN) and 1 false alarm (FP) → already clinically solid. ROC/PR stable; threshold changes give small gains. |
| **SVM RBF** | Acc: 0.9912 Prec: 1.0000<br>Rec: 0.9762 F1: 0.9880 AUC: 0.9964 | [72, 0][1, 41] | Best clinical behavior: 0 false positives and only 1 false negative. ROC extremely high; threshold 0.50 already near F1-optimal → tuning improved robustness more than needing threshold tricks. |
| **Random Forest (tuned)** | Acc: 0.9825 Prec: 1.0000<br>Rec: 0.9524 F1: 0.9756 AUC: 0.9983 | [72, 0][2, 40] | Highest AUC (ranking is great), but at thr = 0.50 it still misses 2 cancers. ROC looks excellent; small threshold adjustment can trade slightly better recall vs stability. |

## THRESHOLD TUNING

### SVM RBF (BEST MODEL)

| Threshold | Precision | Recall | F1-Score | Accuracy | TN | FP | FN | TP | Interpretation |
|---|---|---|---|---|---|---|---|---|---|
| **0.500 (Default)** | 1 | 0.9762 | 0.988 | 0.9912 | 72 | 0 | 1 | 41 | **OPTIMAL** – Already at peak performance |
| **0.447** | 0.9756 | 0.9762 | 0.9759 | 0.9825 | 71 | 1 | 1 | 41 | Slight precision drop, no recall gain |
| **0.515** | 1 | 0.9524 | 0.9756 | 0.9825 | 72 | 0 | 2 | 40 | Maintains precision, loses 1 TP |
| **0.26** | 0.9535 | 0.9762 | 0.9647 | 0.9737 | 70 | 2 | 1 | 41 | Lower threshold: +2 FP, no benefit |
| **0.614** | 1 | 0.9286 | 0.963 | 0.9737 | 72 | 0 | 3 | 39 | Higher threshold: −2 TP, worse recall |

# PREPROCESSING CHALLENGES

## CHALLENGE : HANDLING MISSING VALUES

| Dataset | Challenge | Impact | Solution | Outcome |
|---|---|---|---|---|
| **Breast Cancer** | No missing values | None | Direct modeling | Clean dataset ready |
| **Stroke** | • 201 missing BMI values<br>• “Unknown” smoking status<br>• 1 “Other” gender | • Could bias models<br>• Loss of information | • **BMI**: Median imputation<br>• **Smoking**: Keep “Unknown” as category<br>• **Gender**: Remove 1 row | Preserved 99.98% of data<br>Maintained clinical meaning |

## CHALLENGE 2.1: DECIDING WHEN TO USE SMOTE

| Dataset | Imbalance Ratio | SMOTE Decision | Result |
|---|---|---|---|
| **Breast Cancer** | 1.7:1 (moderate) | Apply (285:170 → 285:285) | Success: CV matches test performance |
| **Stroke** | 19.5:1 (extreme) | Apply (3889:198 → 3889:3889) | Failure: CV overoptimistic, test poor |

# CONCLUSIONS

**Overall Findings**

**Dataset Characteristics Determine Success**

**Breast Cancer (Success Story):**

- Strong feature signals (multiple predictors with r > 0.7 with target)
- Moderate imbalance (1.7:1) manageable with standard techniques
- High feature quality (30 correlated morphological measurements)
- Result: All three models achieve clinical-grade performance (>97% metrics)

**Stroke Prediction (Challenging Case):**

- Weak feature signals (strongest predictor r = 0.25)
- Extreme imbalance (19.5:1) defeats standard balancing techniques
- Single dominant feature (age) with weak secondary predictors
- Result: No model achieves clinical-grade performance (<85% AUC, <30% precision)

**Meta-Lesson: Data quality >> Algorithm choice**

- Good data + simple algorithm > Bad data + complex algorithm

UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

# THANK YOU