Stroke Prediction Machine-Learning Project

---

1. Executive Summary

We built a supervised-learning pipeline that predicts the likelihood of a patient suffering a stroke given routinely collected demographic and clinical variables.\ Using a public healthcare dataset (5 110 records, \~5 % stroke cases), we performed end-to-end data cleaning, feature engineering, class-imbalance handling, model selection and performance evaluation.

- **Best AUROC:** 0.842 (Logistic and SVM).
- **Highest recall (stroke cases):** 0.80 at default threshold (Logistic / XGBoost).
- **Best precision-recall balance:** XGBoost with a tuned decision threshold (F1 = 0.33, precision = 0.25, recall = 0.50).

The workflow is fully reproducible (pipelines + notebooks) and demonstrates practical trade-offs between sensitivity and precision—crucial for clinical screening tools.

---

2. Dataset & Problem Statement

- **Source:** Kaggle “Healthcare Dataset – Stroke Data”.
- **Features (11):** age, gender, hypertension, heart disease, ever married, work type, residence type, average glucose level, BMI, smoking status, etc.
- **Target:** *stroke* (binary; 1 = patient previously had a stroke).
- **Challenge:** Severe class imbalance (\~4.9 % positive).

Objective: Flag high-risk patients so clinicians can prioritise follow-up tests or lifestyle interventions.

---

3. Methodology

1. **Exploratory Data Analysis**
2. Visualised distributions & relationships (e.g. stroke prevalence climbs sharply with age; hypertension & heart disease are strong risk factors).
3. **Pre-processing**
4. Median imputation (BMI), mode imputation (categoricals).
5. Robust scaling (numerics) + one-hot encoding (categoricals).
6. **SMOTE** oversampling applied *within* the CV folds to balance classes.
7. **Modelling**
8. Algorithms: Regularised Logistic Regression, Support Vector Machine (RBF), Balanced Random Forest, XGBoost.
9. Hyper-parameter tuning via 5-fold Stratified GridSearchCV (optimising F1 for the minority class).
10. **Evaluation**

11. Held-out test set (20 %).
12. Metrics: precision, recall, F1, accuracy, AUROC; confusion matrices & PR curves.
13. Explicit threshold sweeps to illustrate sensitivity-specificity trade-off.

---

4. Test-Set Results (key operating points)

Metric \ Model	**Logistic**(thr = 0.5)	**SVM**(thr = 0.7)	**BRF**(thr = 0.5)	**XGB**(thr = 0.5)	**XGB**(thr = 0.6)	**XGB**(thr = 0.7)
precision (0)	0.986	0.983	0.981	0.986	0.983	0.973
precision (1)	0.139	0.202	0.128	0.127	0.151	**0.250**
recall (0)	0.746	0.854	0.747	0.716	0.786	**0.923**
recall (1)	**0.800**	0.720	0.720	**0.800**	0.740	0.500
F1-score (0)	0.849	**0.914**	0.848	0.830	0.874	**0.947**
F1-score (1)	0.237	0.316	0.217	0.219	0.251	**0.333**
Accuracy	0.749	0.847	0.746	0.720	0.784	**0.902**
ROC-AUC	**0.842**	**0.842**	0.812	0.829	0.829	0.829

Thresholds show deliberate tuning to illustrate trade-offs; see PR curves for the full operating range.

---

5. Interpretation

- **Logistic Regression** delivers the *highest sensitivity* (80 %) with strong AUROC (0.842) out-of-the-box, making it a transparent baseline suitable for clinical settings.
- **SVM** matches AUROC but at a higher threshold balances overall accuracy (85 %) and minority F1 (0.32).
- **Balanced Random Forest** under-performs relative to simpler models—class weighting plus SMOTE was already effective.
- **XGBoost** excels when we tune the decision threshold: at 0.7 it doubles precision (25 %) while maintaining a respectable 50 % recall, demonstrating the model's flexibility.

In short, the choice of operating point depends on policy: if missing a stroke is worst-case we favour Logistic (recall = 0.80); if false alarms are costly we can shift XGBoost to precision-optimised mode.

---

6. Key Insights & Impact

- **Risk factors confirmed:** age, hypertension, and heart disease are the top contributors (SHAP & L1 coefficients).

- **Actionable model:** delivered as a scikit-learn pipeline; can be wrapped in a FastAPI endpoint for real-time scoring.
- **Cost awareness:** provided cost-curve & threshold notebook so clinicians can select the sensitivity level that meets guidelines.

---

7. Next Steps

1. **Calibration:** apply isotonic regression or Platt scaling to improve probability estimates.
2. **External Validation:** test on a different hospital's EMR data to confirm generalisation.
3. **Feature Enrichment:** include longitudinal vitals or lab results to boost recall without harming precision.
4. **Deployment:** containerise the best pipeline + monitoring to detect data drift.

---

8. Repository Structure

```
├── data/healthcare-dataset-stroke-data.csv
├── notebooks/
│   ├── 01_EDA.ipynb
│   ├── 02_Logistic.ipynb
│   ├── 03_SVM.ipynb
│   ├── 04_BRF.ipynb
│   └── 05_XGBClassifier.ipynb
├── src/
│   ├── preprocessing.py  # reusable pipeline builder
│   └── evaluate.py       # metrics + visual utilities
└── README.md             # quick-start & findings
```
