Heart Disease Classification Project

Project Description

A heart disease classifier based on the CDC dataset (**319,795 samples**).
The project implements a classification pipeline to predict the occurrence of heart disease based on medical and lifestyle indicators.

Project Goal

To build an efficient classification model that predicts the occurrence of heart disease in the population using medical and demographic data.

Key Performance Indicators (KPIs)

- **F1-score (weighted) ≥ 0.75** (tracked in MLflow)
- **LogLoss** as an indicator of probabilistic classification quality
- Stability of cross-validation metrics (**std ≤ 0.05**)
- Automated hyperparameter tuning with **Optuna**
- Code and model version tracking via **MLflow + Git**

Risk Assessment

- **Data imbalance risk:** highly imbalanced classes (91.4% “No”, 8.6% “Yes”) – requires balancing (class_weight, stratified sampling)
- **Model dependency risk:** use of a specific framework (CatBoost)
- **Overfitting risk:** controlled through early stopping and cross-validation
- **Reproducibility risk:** minimized with version control (joblib, MLflow, git hash, random_state)

Dataset Description

The dataset consists of **319,795 samples and 18 columns** (17 features + 1 target).

Columns:

Column	Description
HeartDisease	Heart disease diagnosis (Yes/No) – **target variable**
BMI	Body Mass Index
Smoking	Smoking status (Yes/No)
AlcoholDrinking	Alcohol consumption (Yes/No)
Stroke	Stroke history (Yes/No)
PhysicalHealth	Days of poor physical health (0–30)
MentalHealth	Days of poor mental health (0–30)
DiffWalking	Difficulty walking (Yes/No)
Sex	Gender (Male/Female)
AgeCategory	Age category
Race	Race/ethnicity
Diabetic	Diabetes status (Yes/No/Borderline/Yes during pregnancy)
PhysicalActivity	Physical activity (Yes/No)

GenHealth	General health (Excellent/Very good/Good/Fair/Poor)
SleepTime	Hours of sleep per day
Asthma	Asthma (Yes/No)
KidneyDisease	Kidney disease (Yes/No)
SkinCancer	Skin cancer (Yes/No)

Target variable distribution:

- **No** (no heart disease): 292,422 samples (91.4%)
- **Yes** (heart disease): 27,373 samples (8.6%)

⚠ Significant class imbalance – addressed with class_weights in CatBoost and stratified sampling.

🤖 Model Description

Model: CatBoostClassifier (gradient boosting)

CatBoost features:

- Native categorical support (no one-hot encoding required)
- Ordered boosting – reduces overfitting risk
- Built-in handling of class imbalance (class_weights)

🔧 Hyperparameter Tuning

- Implemented using **Optuna** + mlflow.start_run(nested=True)
- Best parameters stored in best_params.pkl and logged in MLflow
- Key parameters optimized:
  - iterations
  - learning_rate
  - depth
  - l2_leaf_reg
  - class_weights

🔁 Cross-Validation

- Implemented with catboost.cv using 5-fold stratified shuffle
- Metrics tracked: **F1, LogLoss, AUC-ROC**
- Results visualized with error bands using **Plotly**

🗂 Project Structure

```
├── .devcontainer/        # Codespaces / Docker configuration
├── .github/workflows/    # CI/CD pipeline
├── data/                 # project data (raw, interim, processed, etc.)
├── models/               # saved models and monitoring artifacts
├── docs/                 # documents
```

```
├── reports/              # reports, visualizations
├── results/              # experiment / prediction outputs
├── notebooks/            # Jupyter notebooks (EDA, experiments)
├── ARISA_DSML/           # main source code (data prep, training, prediction, utilities)
├── tests/                # unit / integration tests
├── Makefile              # automation (lint, test, train)
├── README.md             # project description
├── pyproject.toml        # package and dependency configuration
├── setup.cfg             # tool configuration (flake8, black)
├── requirements.txt      # list of dependencies
└── .gitignore            # ignored files and folders
```

⚙ Prerequisites

- Python 3.11+
- Pandas & NumPy
- Scikit-learn
- Matplotlib & Plotly
- Jupyter Notebook
- MLflow
- Git & GitHub

🚀 Installation & Run

1. Clone the repository

   ```
   git clone https://github.com/Pawel20240101/PZ_ARISA_MLOps_Final.git

   cd PZ_ARISA_MLOps_Final
   ```

2. Create a virtual environment

   ```
   python -m venv .venv

   # Windows

   .\.venv\Scripts\activate

   # Linux/Mac

   source .venv/bin/activate
   ```

3. Install dependencies

pip install -r requirements.txt

4. Place the dataset

   skopiuj heart_2020_cleaned.csv do data/raw/

5. Start MLflow UI

   mlflow ui

   Lub

   mlflow ui --host 127.0.0.1 --port 5000

🖌️ Data Preprocessing

- Target conversion (Yes/No → 1/0)
- Train/test split (stratified)
- Validation of categorical columns
- Class balancing: **class_weight + stratified sampling**

📈 Evaluation Metrics

Tracked metrics:

- **F1-score (weighted)**
- Precision & Recall for the positive class
- AUC-ROC
- Confusion Matrix
- LogLoss

📡 Monitoring & Support

- **MLflow** – experiment tracking, metrics, artifacts, model versions
- **NannyML** – data drift detection
- **Git** – version control

🔄 CI/CD Pipeline

Automated workflows:

- lint-code.yml – linting triggered on every PR to main
- ci.yml – linting, formatting, and tests on push/PR to main and test

Safeguards:

- main branch protection
- Required code review
- Pre-commit hooks (flake8, black)

🩹 Code Quality

- Linting: flake8 + black + isort
- Unit tests: (to be implemented)

- Documentation: docstrings + README.md

📊 Experiment Results

Cross-Validation (N=5):

- Mean F1 Score: ~0.77 (stable after ~50 iterations)
- Mean LogLoss: ~0.49 (after convergence)
- Standard deviation $\ll 0.05 \rightarrow$ no signs of overfitting

SHAP Analysis:

- Most important features: AgeCategory, GenHealth, Stroke, BMI
- Less important features: Race → model is fair (non-discriminatory)

📜 Medical Insights

- Key risk factors: older age, poor general health, stroke history, high BMI
- Model achieved KPIs (F1 ≈ 0.77, LogLoss ≈ 0.49)
- Interpretability provided via SHAP values
- Results consistent with medical knowledge