DIAGNOSIS OF CORONARY ARTERY DISEASE

UCI – heart disease dataset

Kacper Kozik

Mikolaj Pniak

1. INTRODUCTION

The goal of this project was to predict heart disease.

We used the UCI Heart Disease dataset to compare various machine learning models in a binary classification task - detecting whether a patient has heart disease or not.

The focus was on thorough data preprocessing, model performance comparison, and understanding how different techniques affect results.

2. UCI – HEART DISEASE DATASET

UCI Heart Disease dataset from the UCI Machine Learning Repository
Combines data from **Cleveland**, **Hungary**, **Switzerland**, and **VA Long Beach**

Focus on 13 key features related to heart disease to predict target feature

Target:
Originally ranged from 0 (no disease) to 4 (severe).
Transformed into binary:

- 0 (no disease)
- 1 (Heart disease present)

Age: Patient's age (range: 28–77)

Sex: 0 = female, 1 = male

Cp: Chest pain type (0-typical, 1-atypical, 2-non-anginal, 3-asymptomatic)

Trestbps: Resting blood pressure (mm Hg)

Chol: Serum cholesterol (mg/dl)

Fbs: fasting blood sugar > 120mg/dl = 1 (yes)

Restecg: Resting ECG results (0-normal, 1-ST-T abnormalities, 3-LV hypertrophy)

Thalach: Max heart rate achieved during excercise (bpm)

Exang: Exercise-induced angina (1 = yes)

Oldpeak: ST depression during exercise (compared to rest)

Slope: Slope of ST segment (2-upsloping, 1-flat, 0-downsloping)

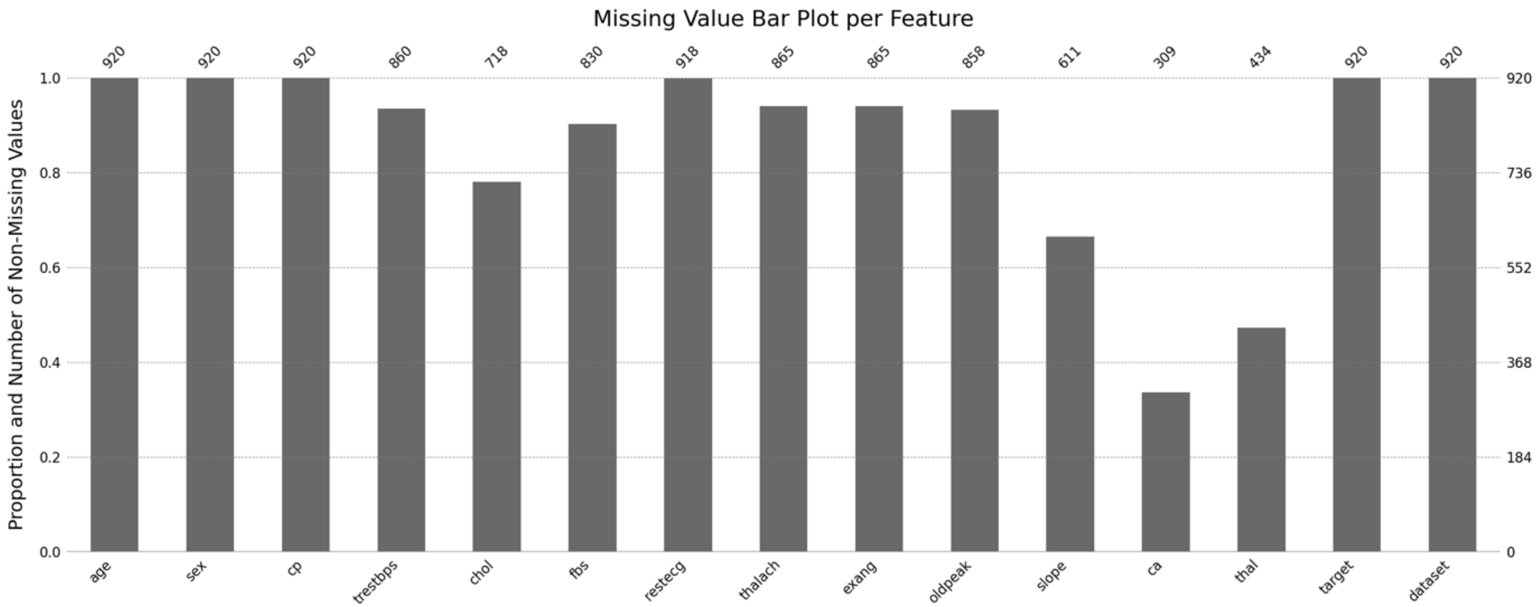
Ca: Number of major vessels with narrowing (0–3)

Thal: Thallium Stress Test Result(0–3)

UCI – HEART DISEASE DATASET

ca, thal - high number of missing values

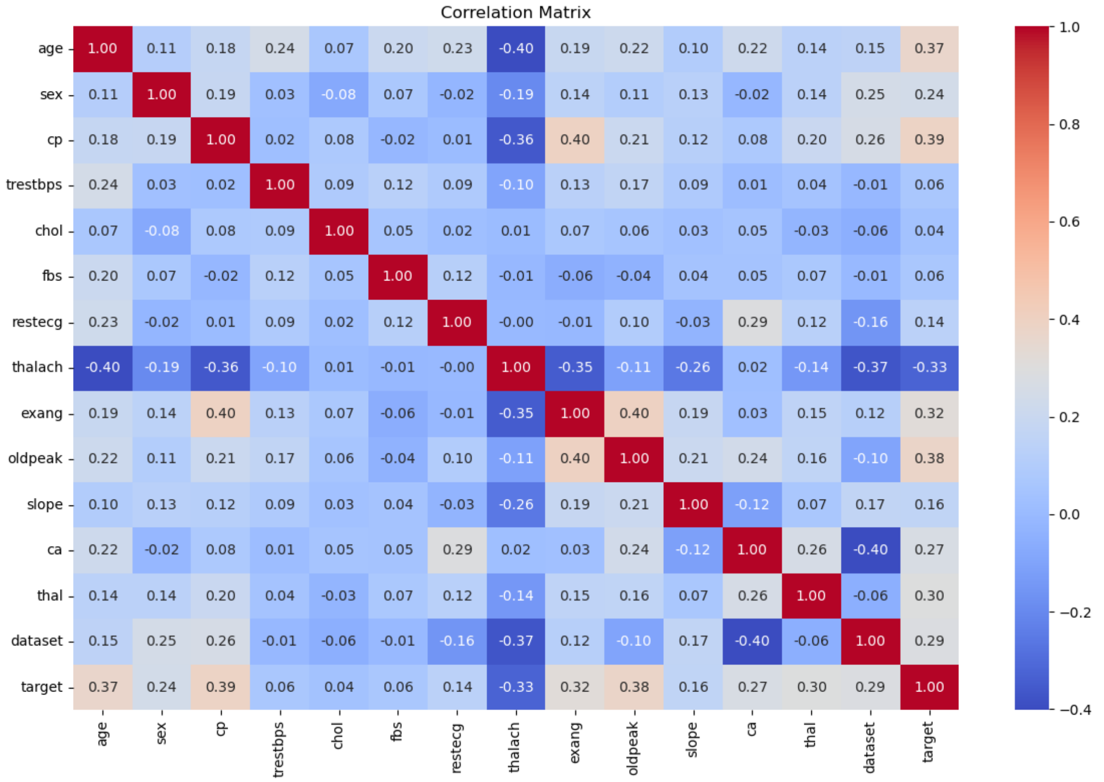
slope, **chol** - moderate missingness



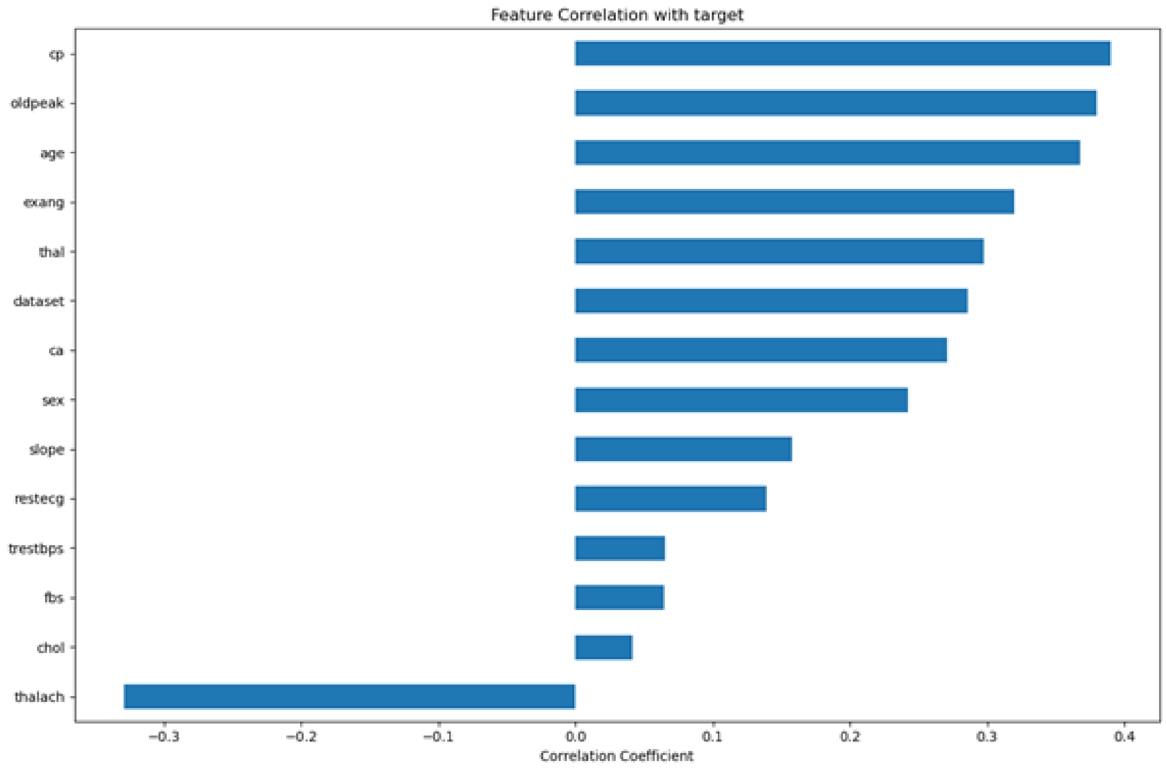
CORRELATIONS

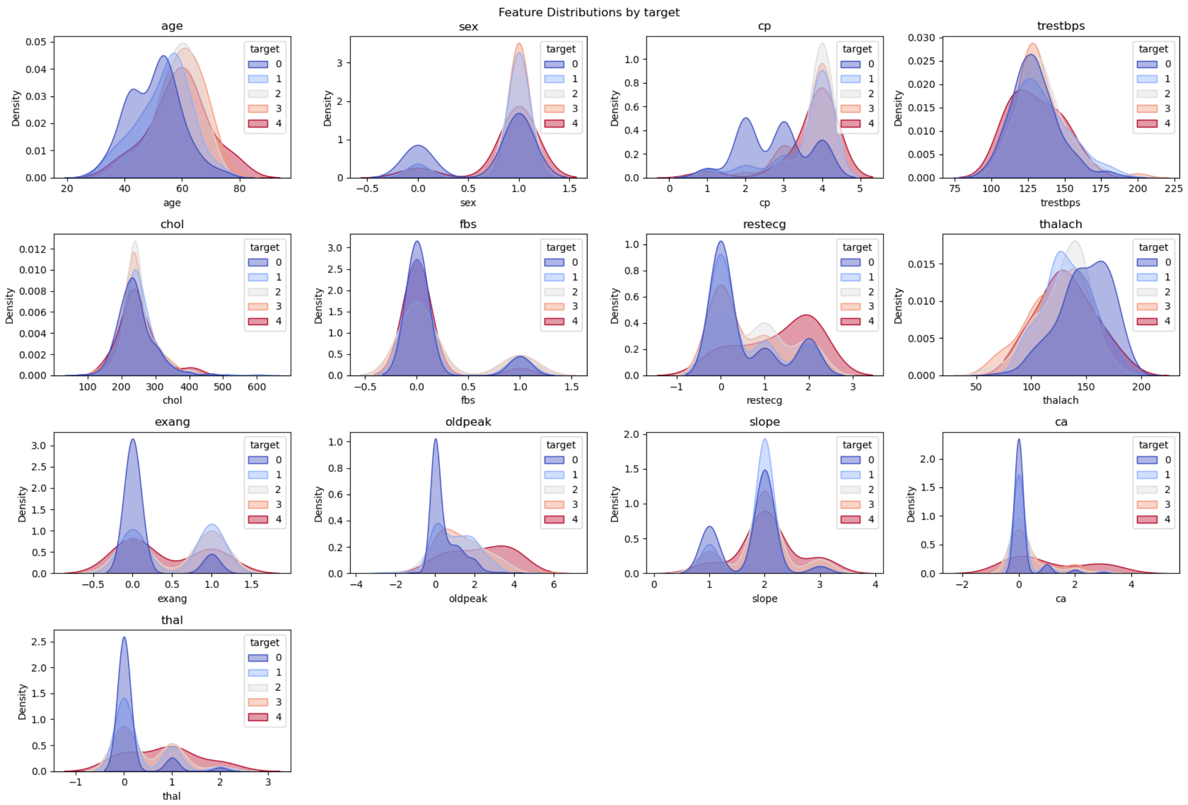
Strong feature correlations:

- age vs thalach (max. heart rate during ex.)
- oldpeak (ST depression during ex.) vs. exang (Ex. induced angina)
- thalach vs. cp. (chest pain type)
- thalach vs. exang



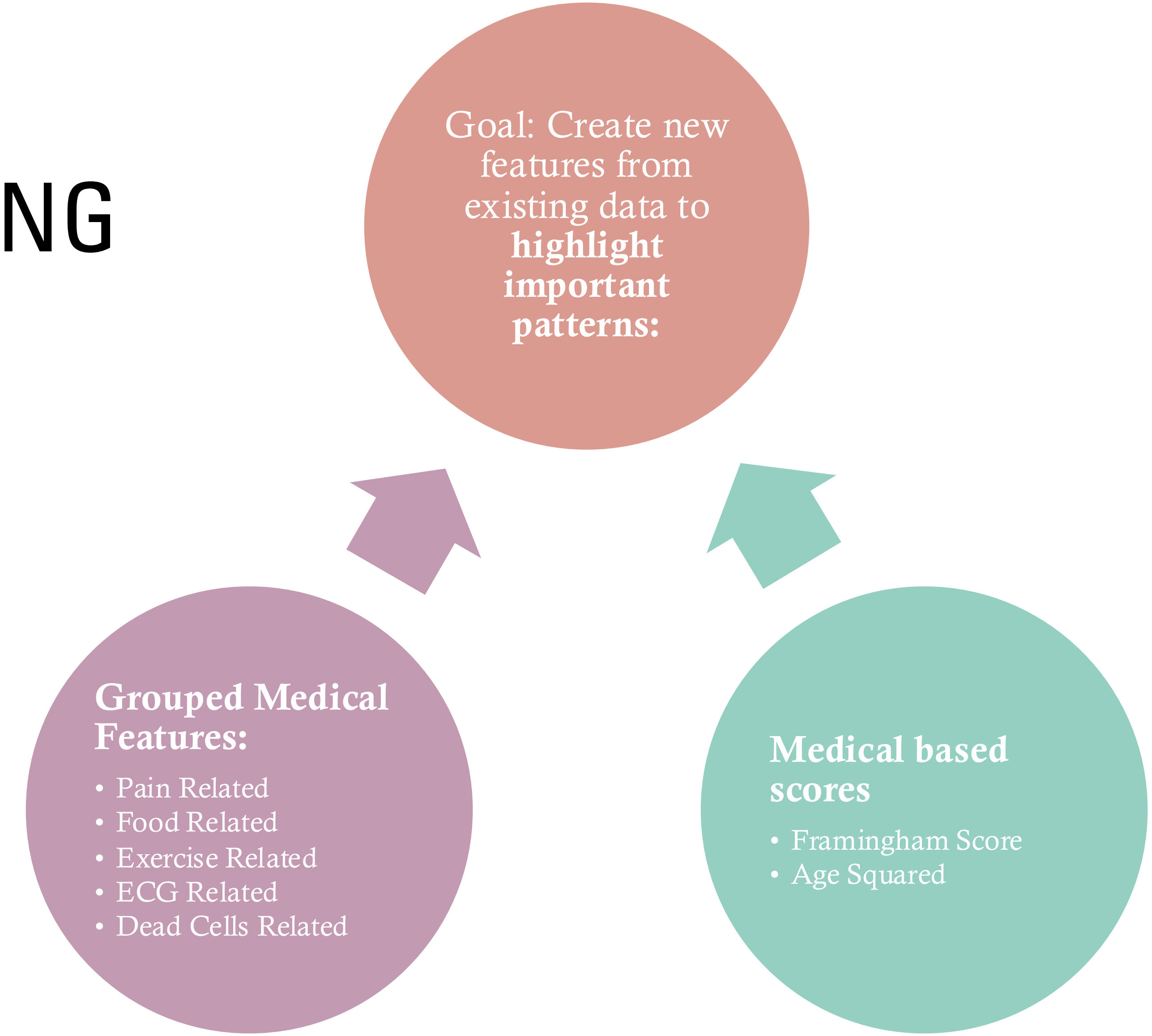
CORRELATIONS WITH TARGET





3. PREPROCESSING - FEATURE ENGINEERING

Framingham Score: Combines age, sex, cholesterol, blood pressure, and blood sugar
Captures **non-linear** effects of age on heart risk



IMPUTATION STRATEGIES

For numerical features **we used**:

- **Mean/median**
- **Knn imputation**
- **Mice**

For categorical features we used:

- **Mode imputation**

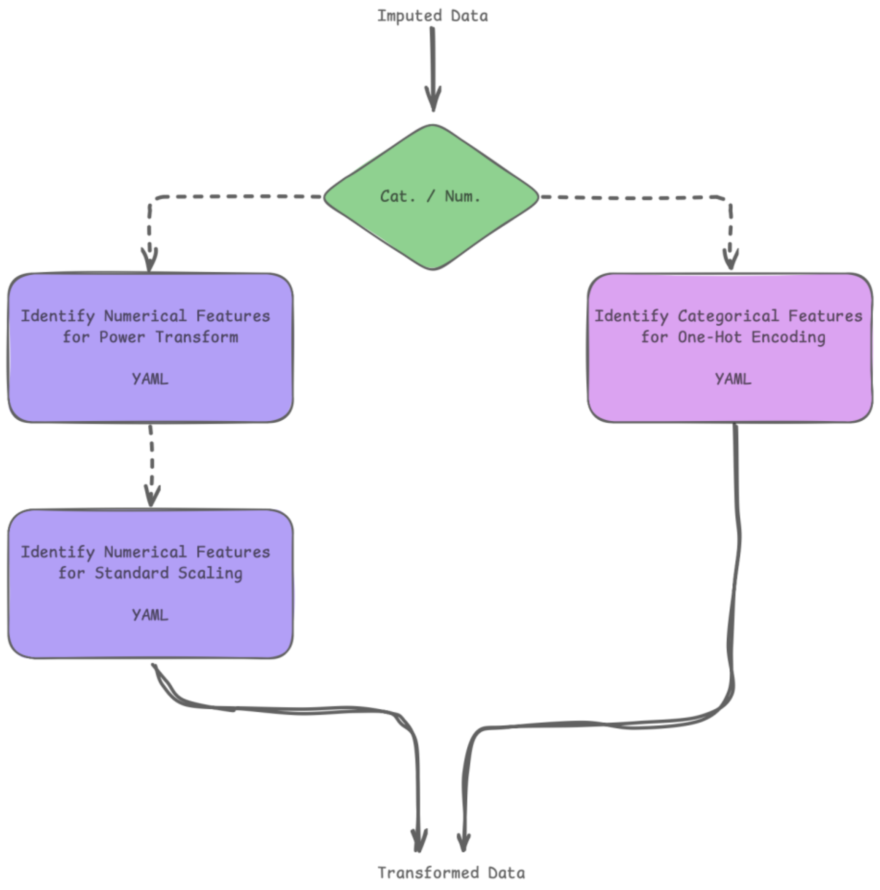
Numerical Imputation Methods

- Mean Imputation
- Median Imputation
- Regression Imputation
- K-Nearest Neighbors Imputation
- Multivariate Imputation by Chained Equations (MICE)
- Interpolation (eg. linear, spline)

Categorical Imputation Methods

- Mode Imputation
- K-Nearest Neighbors Imputation
- Logistic Regression Imputation
- Multivariate Imputation by Chained Equations (MICE)

Early mistake: fitting on full dataset, which lead to artificially high scores



SCALING AND ENCODING

Data Transformation:

Critical for scale-sensitive alghoritms (e.g. SVM)

Categorical features:

One-Hot Encoding for **cp, thal, sex, dataset**

Numerical features:

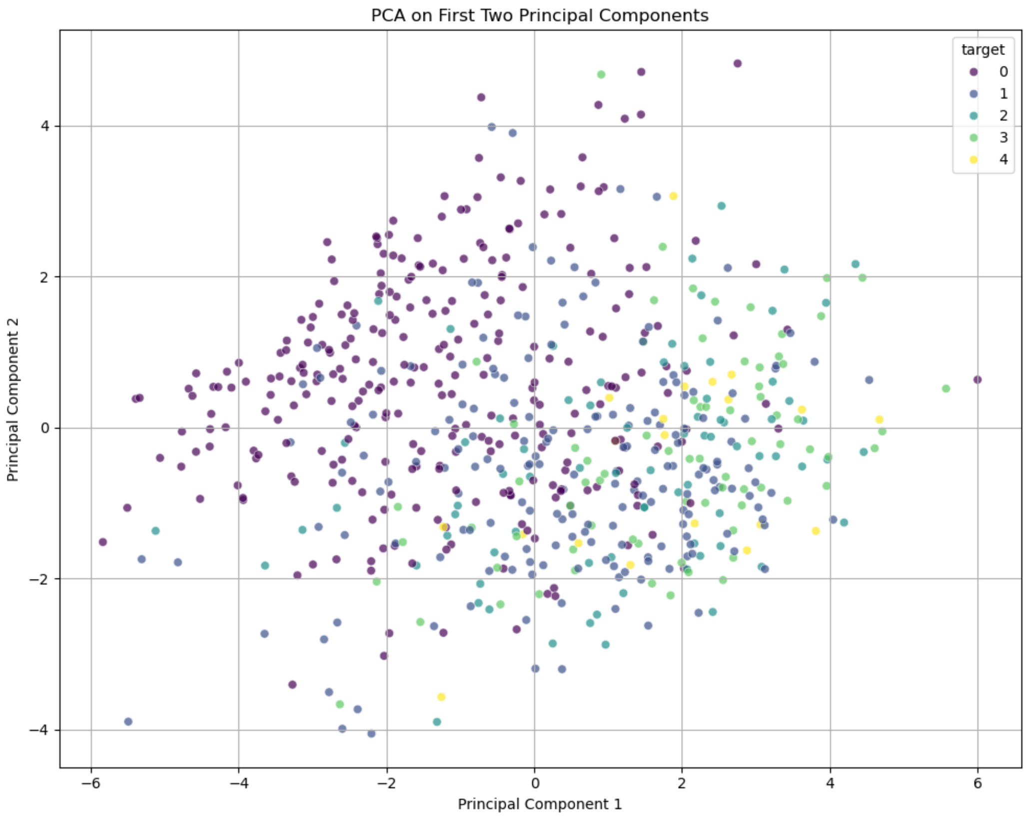
Power Transformation (Yeo-Johnson) **applied to oldpeak, trestbps, and feature engineered features.**

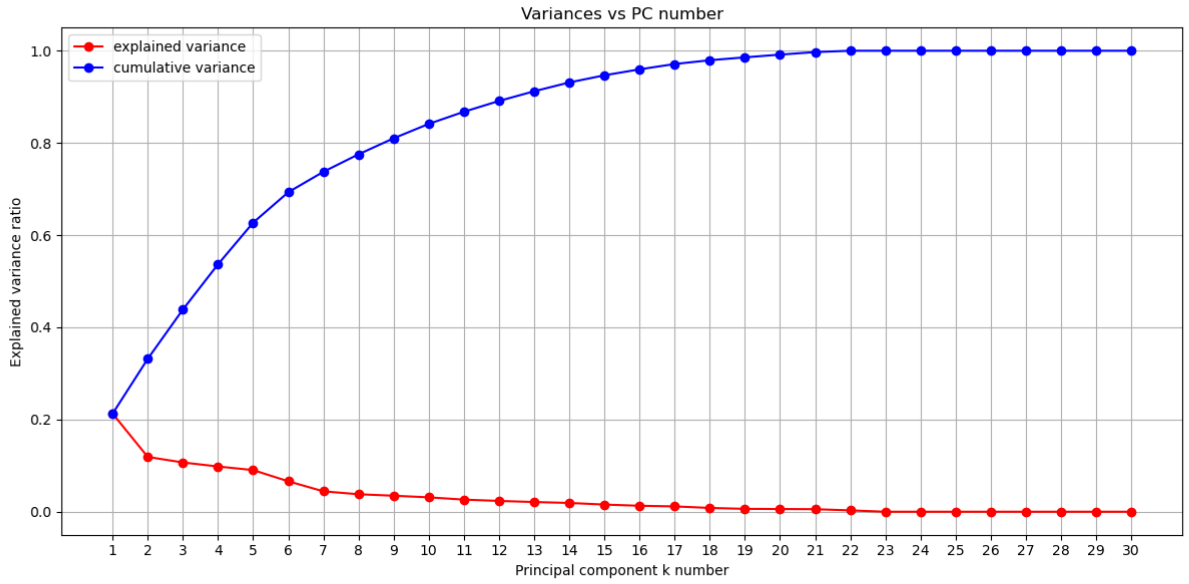
Standarization : **all features**

DIMENSIONALITY REDUCTION - PCA

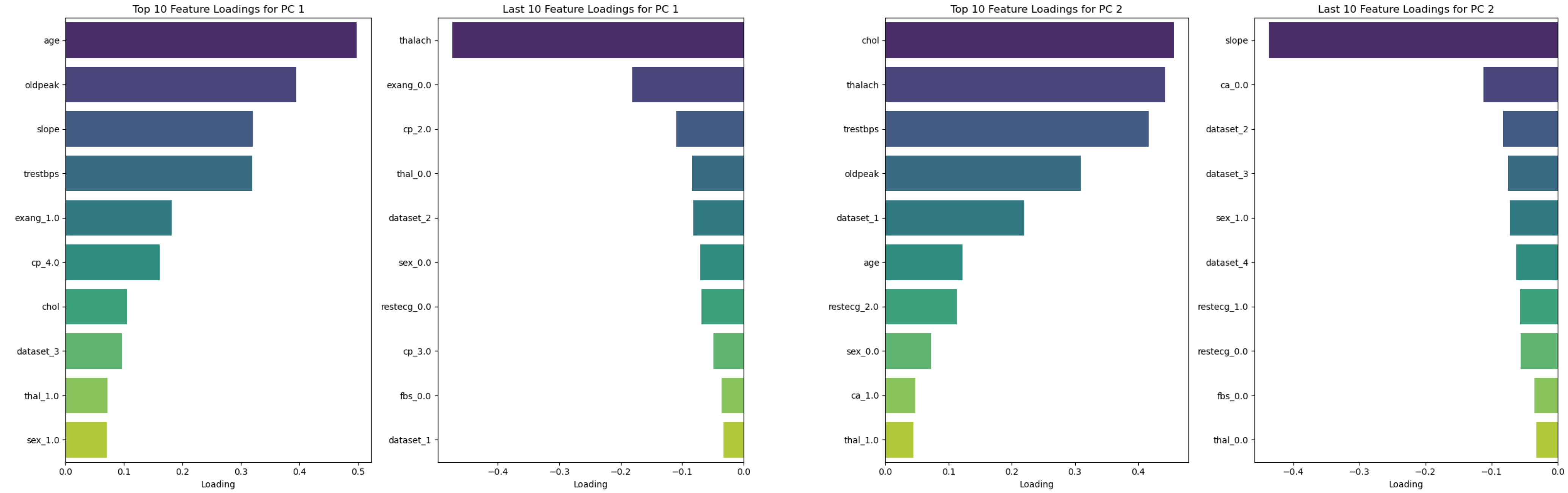
Our PCA Strategy:

Retained enough components to explain **90% of total variance**





DIMENSIONALITY REDUCTION - PCA



4. BUILDING AND TUNING PREDICTIVE MODELS

Models:

- **SVM:** Support Vector Machines
- **Random Forest**
- **LightGBM**: Light Gradient Boosting Machine

Metrics:

- Accuracy
- Precision
- Recall
- F1-score
- AUC-ROC

Baseline Models Performance:
Before optimization we trained each model using a set of defined, static hyperparameters

Hyperparameter Optimization with Optuna, **optimize F1-score**:

SVM:

C, kernel, gamma, degree

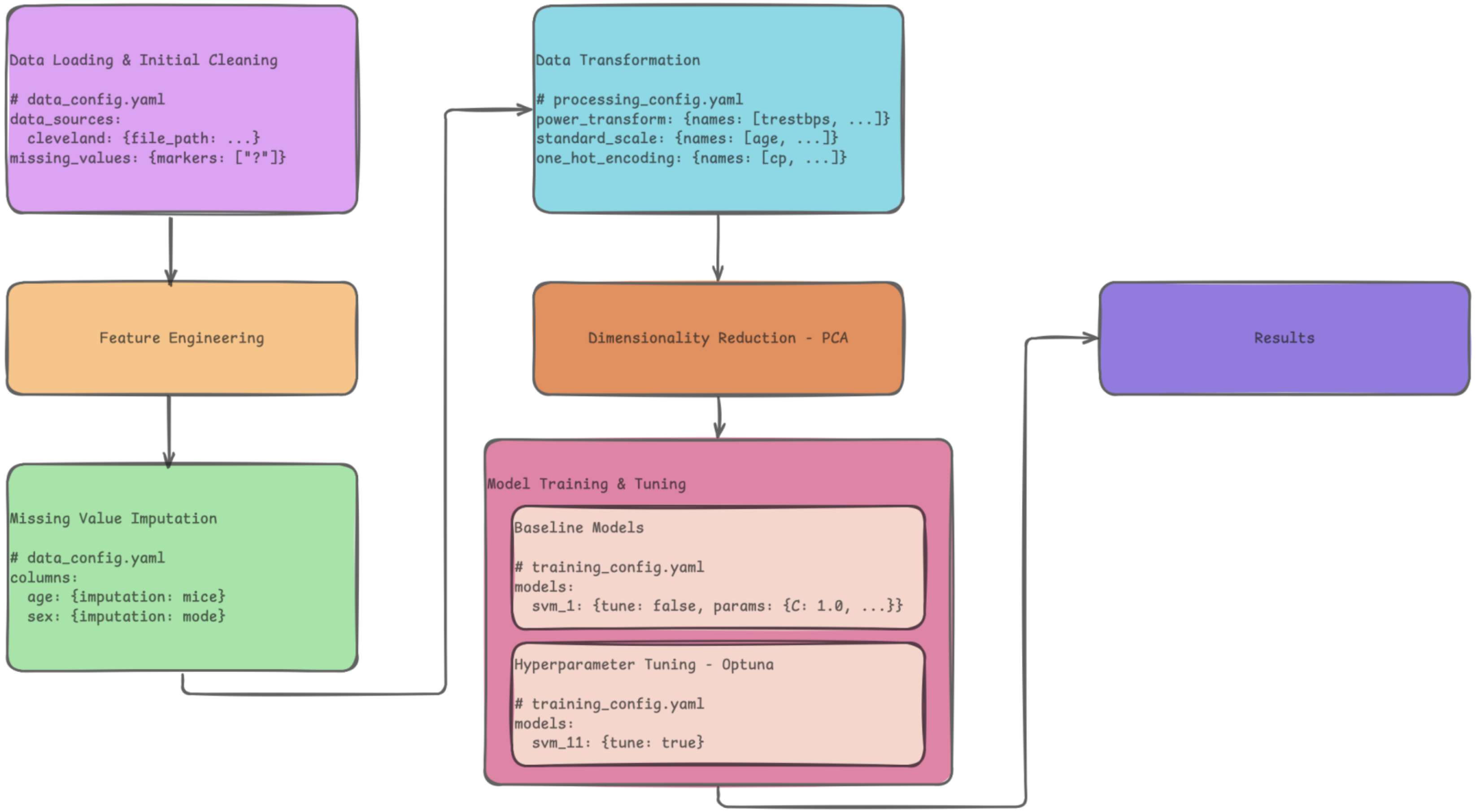
Random forest:

n_estimators, max_depth, min_samples_split, min_samples_leaf, max features

LightGBM:

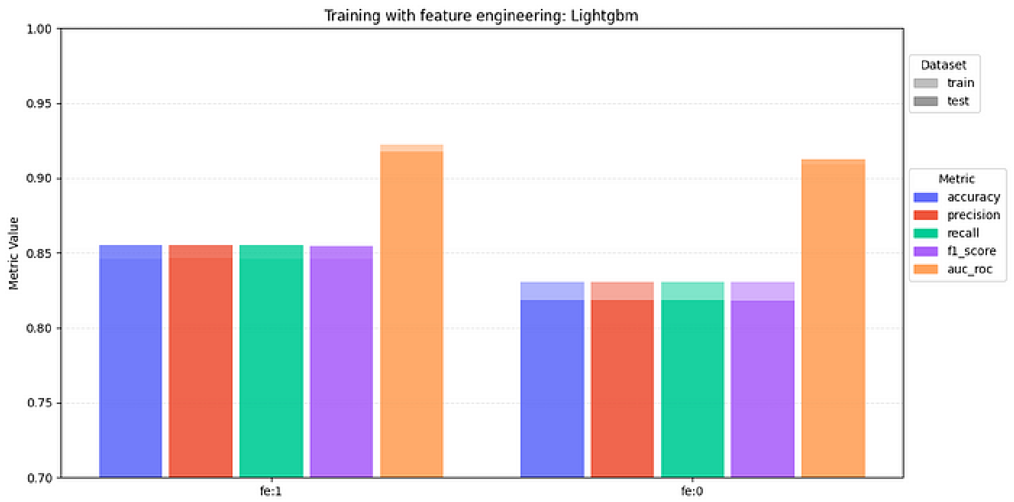
n_estimators, learning_rate, num_leaves, max_depth, reg_alpha, reg lambda, colsample_bytree, subsample, min child samples

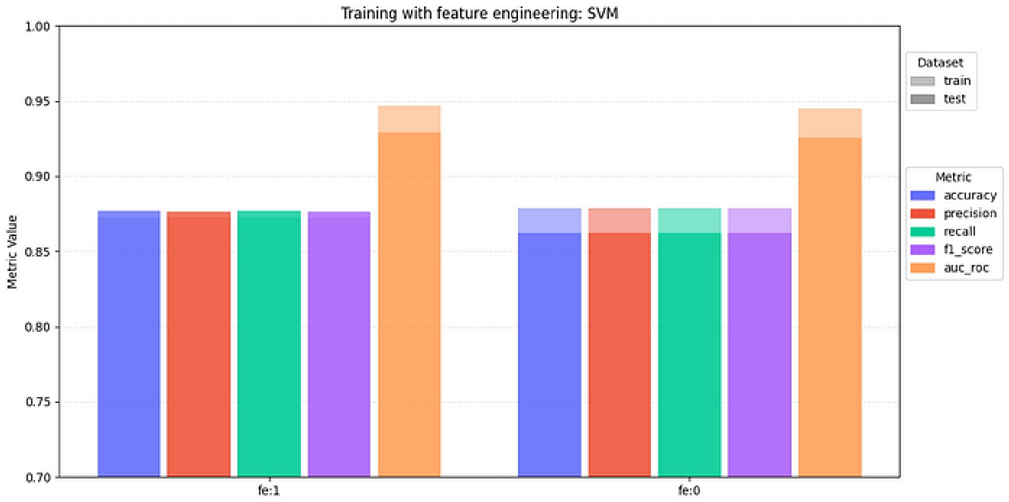
5. PIPELINE OVERVIEW

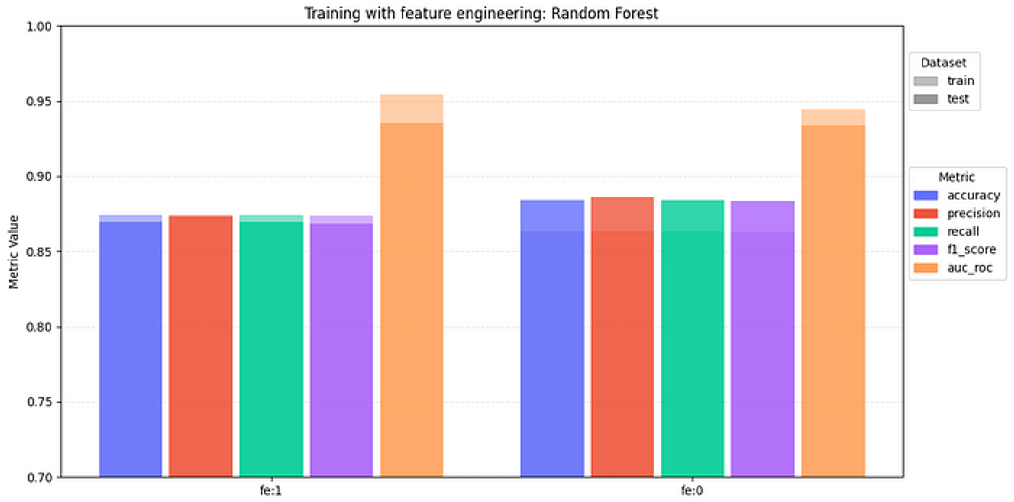


6. EVALUATING MODEL PERFORMANCE

FEATURE ENGINEERING IMPACT

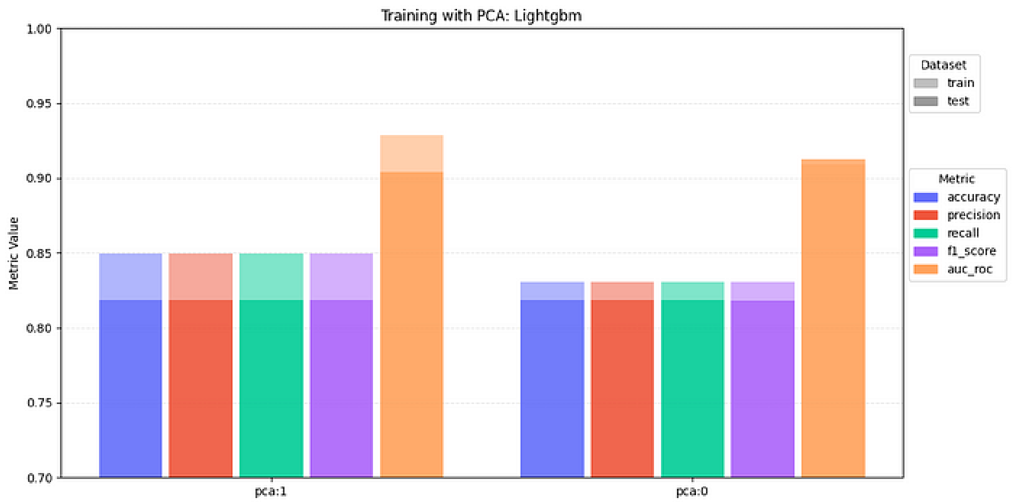


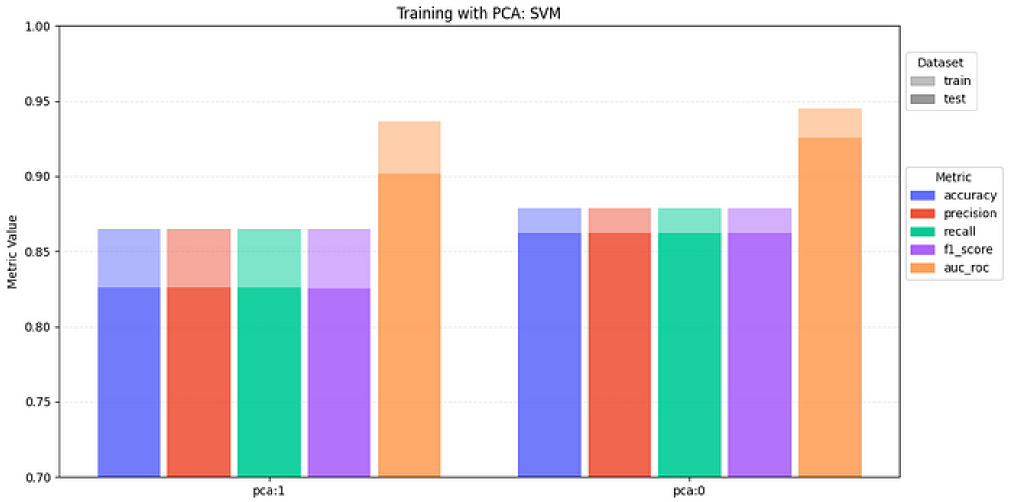


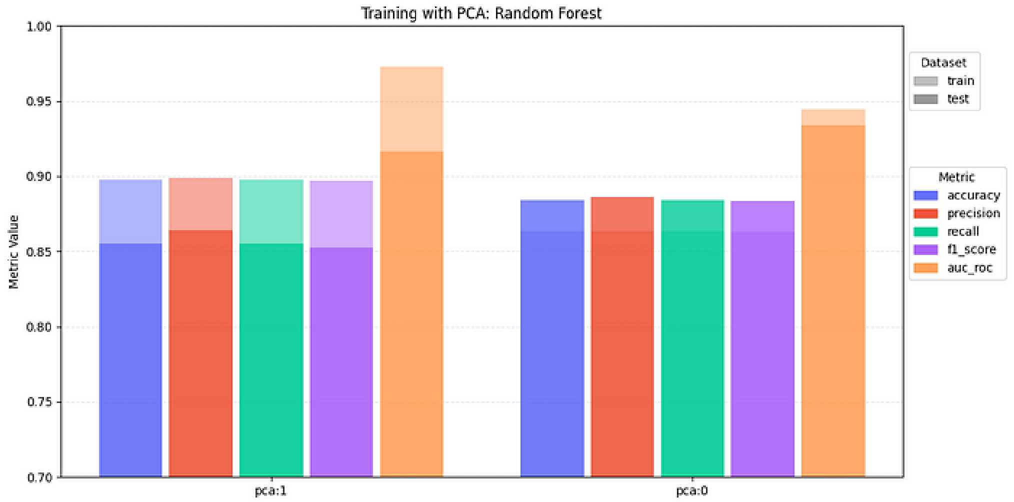


6. EVALUATING MODEL PERFORMANCE

IMPACT OF DIMENSIONALITY REDUCTION

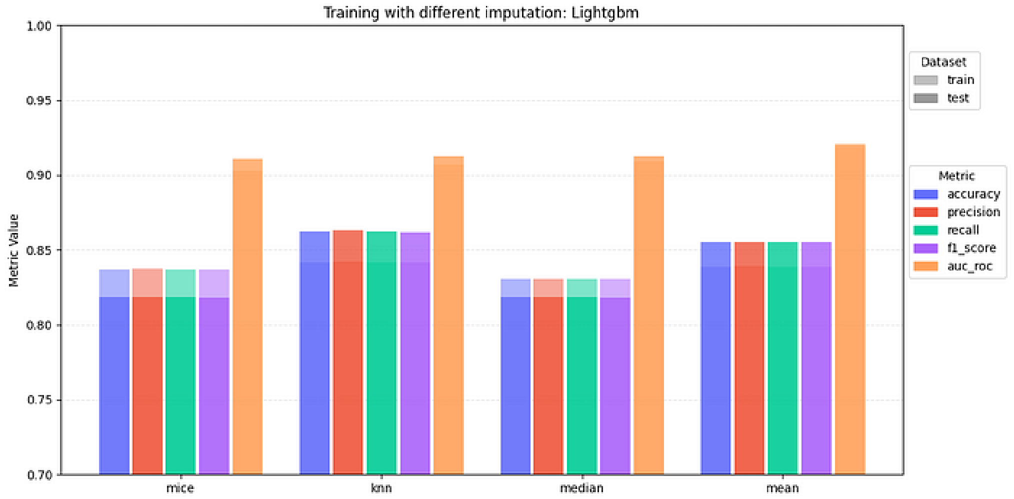


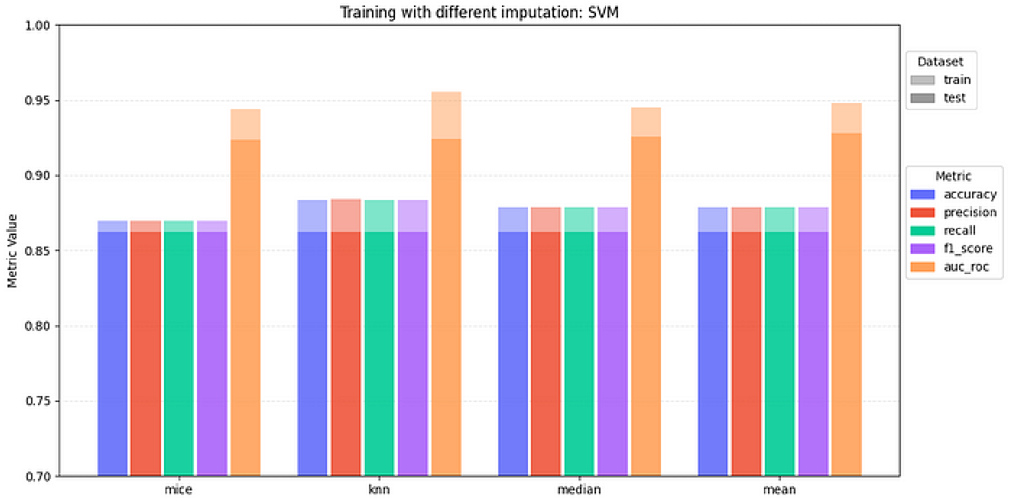


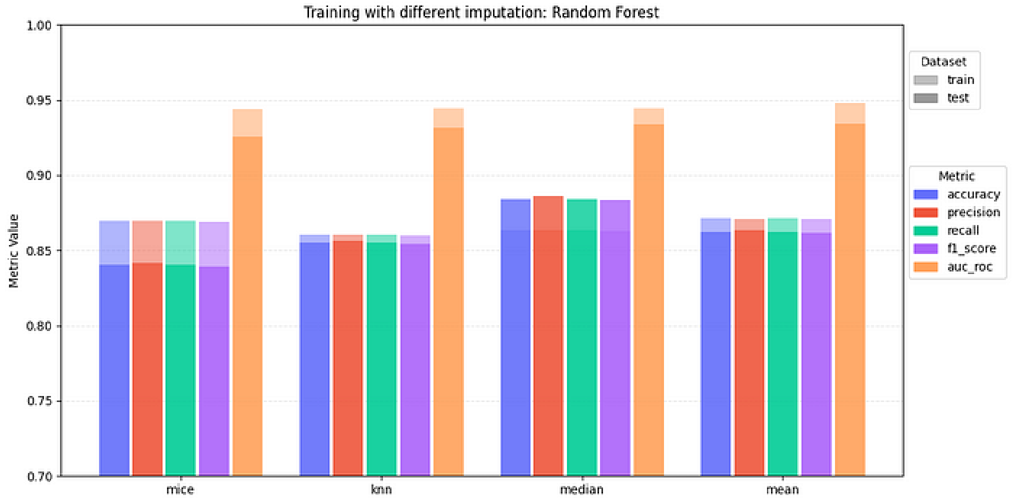


6. EVALUATING MODEL PERFORMANCE

IMPACT OF IMPUTATION METHODS

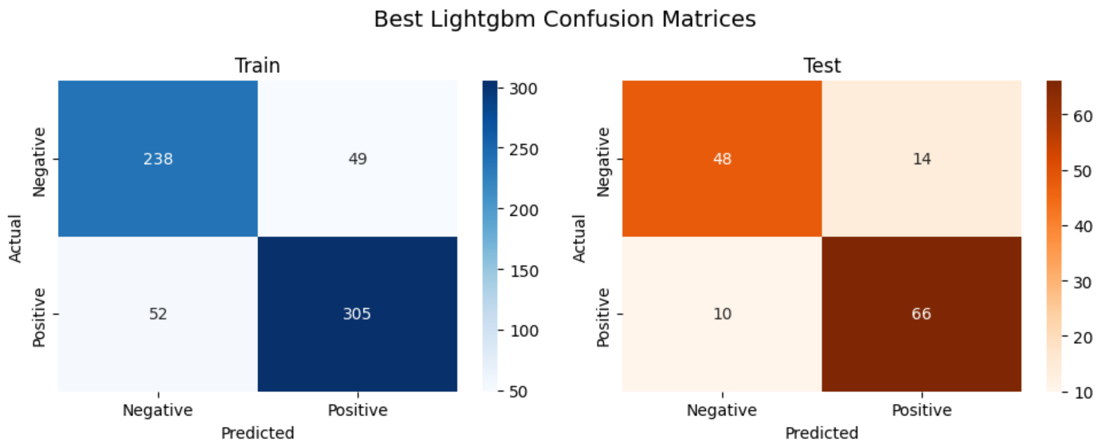


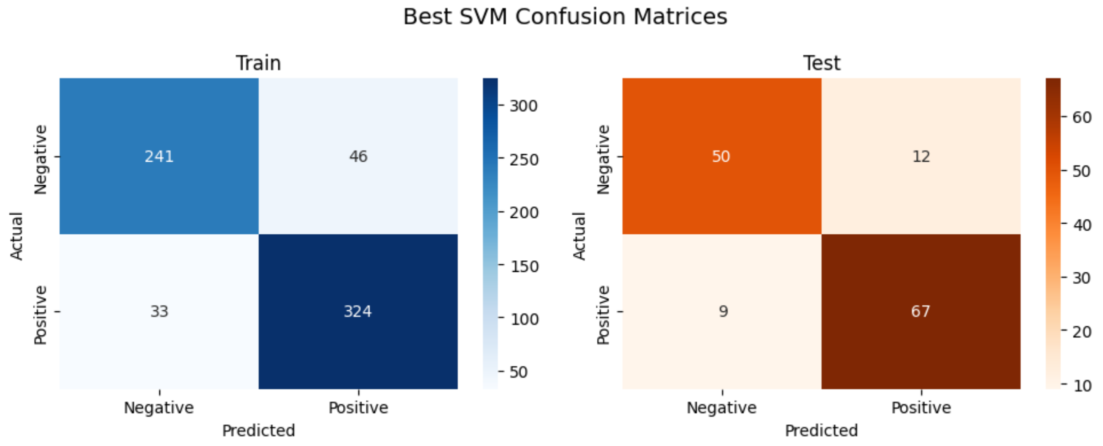


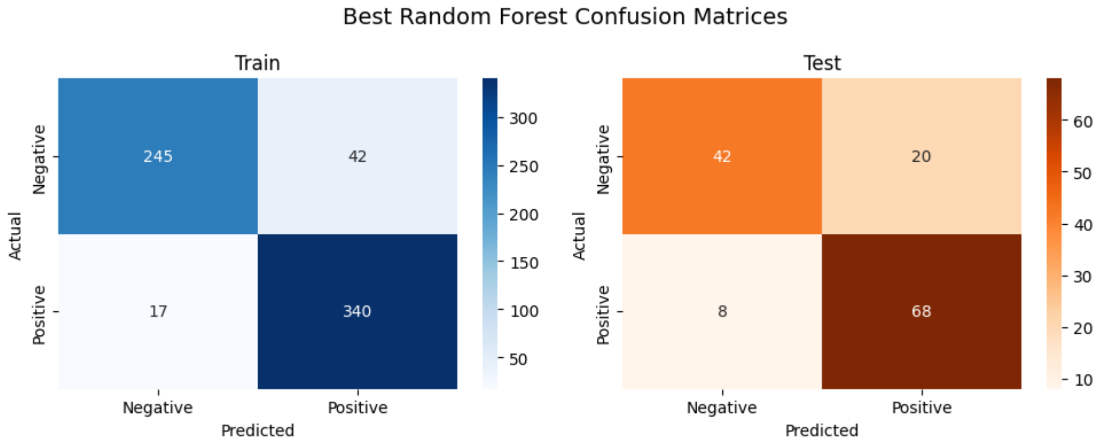


6. EVALUATING MODEL PERFORMANCE

PERFORMANCE OF BEST CLASSIFICATION MODELS







6. EVALUATING MODEL PERFORMANCE

PARAMETERS OF BEST CLASSIFICATION MODELS

Best SVM:

- **C: 1.5** — regularization parameter, smaller values specify stronger regularization, larger values less
- **class_weight: balanced** — adjusts weights inversely proportional to class frequencies to handle imbalanced data

Best Random Forest:

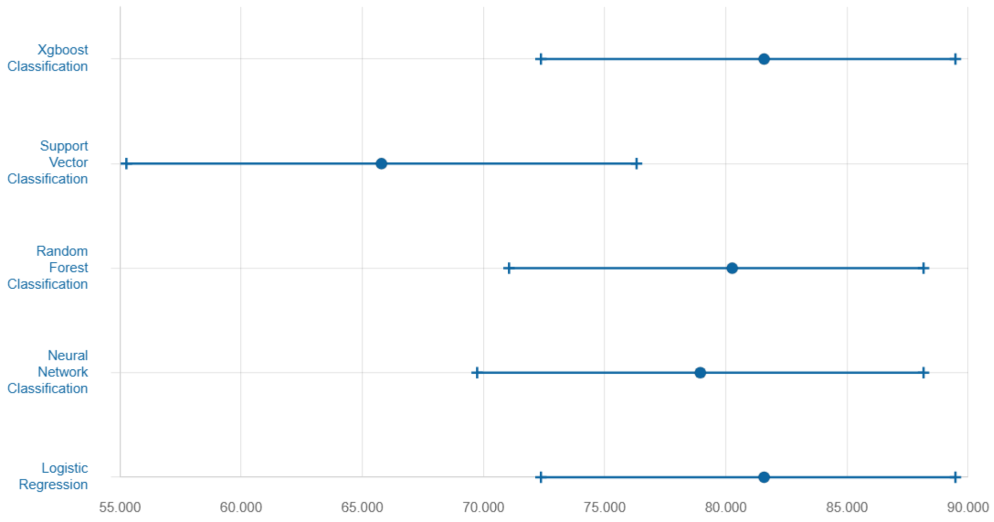
- **n_estimators: 500** — total number of decision trees to build in the forest
- **max_depth: 7** — maximum depth allowed for any individual tree in the forest
- **min_samples_split: 5** — minimum number of samples required to split an internal node of a tree
- **min_samples_leaf: 5** — minimum number of samples required to be at a leaf node of a tree
- **max_features: sqrt** — number of features to consider when looking for the best split, 'sqrt' uses the square root of total number of features

Best LightGBM:

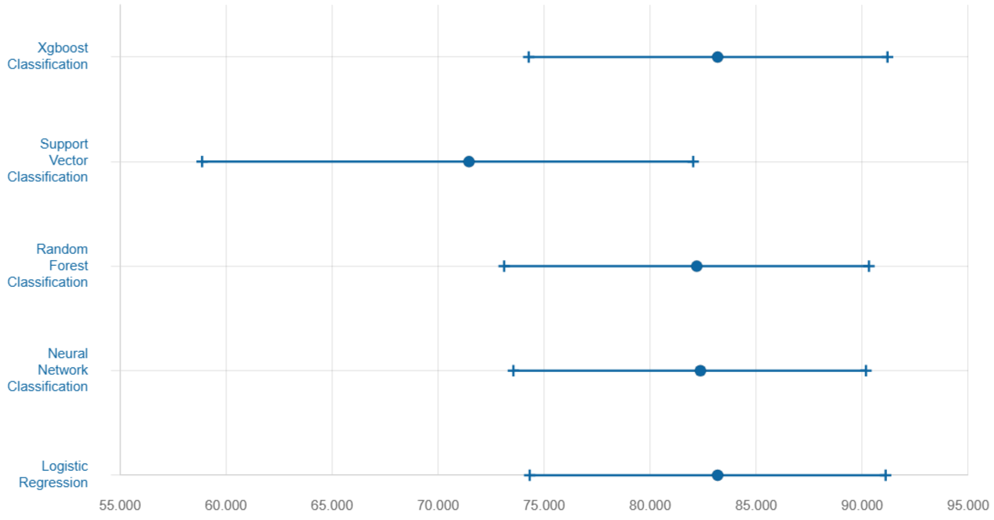
- **n_estimators: 300** — number of boosting trees to build
- **num_leaves: 15** — maximum number of leaves in one tree, a key parameter for controlling model complexity
- **min_child_samples: 100** — minimum number of data points needed in a child or leaf node
- **subsample: 0.8** — fraction of training instances to be randomly sampled for each tree
- **colsample_bytree: 0.8** — fraction of features to be randomly sampled for each tree
- **learning_rate: 0.05** — shrinks the contribution of each tree, lower values usually require more trees
- **max_depth: 3** — maximum depth of individual trees in the boosting process
- **reg_alpha: 5** — L1 regularization term on weights, encourages sparsity
- **reg_lambda: 5** — L2 regularization term on weights, helps prevent overfitting
- **class_weight: balanced** — adjusts weights to give more importance to minority classes

6. EVALUATING MODEL PERFORMANCE

Accuracy



Precision



Our best SVM:
Accuracy: 87.7%,
Precision: 87.5%

https://archive.ics.uci.edu/dataset/45/heart+disease