Indian Institute of Technology Delhi

Department of Chemical Engineering

Machine Learning Models for Heart Failure Prediction

A Report on Predictive Modeling for Heart Failure Detection

Author

Shashi Kumar
Undergraduate Student
Department of Chemical Engineering
Indian Institute of Technology Delhi

Research Mentors

Neha Singh, Ramya Srinivasan

Faculty Supervisor

Prof. Anurag Singh Rathore
Head Of Department
Department of Chemical Engineering
Indian Institute of Technology Delhi

Abstract

This research report evaluates multiple machine learning models for predicting heart failure using a dataset of 139 patient records with 11 clinical features. Models include Random Forest Classifier (RFC), K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), Gradient Boosting Classifier (GBC), Linear Discriminant Analysis (LDA), AdaBoost Classifier (ABC), Extra Trees Classifier (ETC), XGBoost Classifier (XGB), Logistic Regression (LR), Decision Tree Classifier (DTC), Bagging Classifier, and Gaussian Naive Bayes (GNB). Feature selection is attempted using the Boruta algorithm, which unexpectedly identified no significant features. The dataset is preprocessed by handling missing values and deriving a binary target variable (`TARGET`) based on final diagnosis and medication requirements. The SVC model achieved the highest accuracy (82.14%), but all models struggled with minority class prediction due to class imbalance. Recommendations for addressing these issues are provided.

1 Introduction

Heart failure prediction is critical for improving patient outcomes. This study uses a clinically generated dataset from **AIIMS Delhi** (JSS.csv), comprising 139 patient records and 11 clinical features, to develop predictive models. The target variable (TARGET) indicates the presence of heart failure (binary: 0 or 1), derived from final diagnoses and medication requirements. Multiple classifiers are evaluated, and feature selection is performed using the Boruta algorithm to identify key predictors. This report presents the methodology, results, challenges, and recommendations for enhancing model performance.

2 Dataset Description

The dataset contains 139 rows and 11 columns with the following input features:

- **Age (in years)**: Integer, range: 25–88.
- **Weight (in kg)**: Float, range: 39.0–126.0.
- **Orthopnoea**: Binary (0 = Absent, 1 = Present), breathing difficulty when lying down.
- **T2DM**: Binary (0 = Absent, 1 = Present), Type 2 Diabetes Mellitus.
- **HTN**: Binary (0 = Absent, 1 = Present), Hypertension.
- **IHD**: Binary (0 = Absent, 1 = Present), Ischemic Heart Disease.
- **PVD / PAD**: Binary (0 = Absent, 1 = Present), Peripheral Vascular Disease/Peripheral Artery Disease.
- **Obesity**: Binary (0 = Absent, 1 = Present), obesity indicator.
- **ECHO**: Binary (0 = Normal, 1 = Abnormal, e.g., IHD, LV Dilated, Mild LV Dysfunction).

- **Ejection Fraction**: Float, range: 0.22–0.66, heart pumping efficiency (converted from percentage x% to x/100).

The output feature is `TARGET`, a binary variable (0 or 1) derived from the final diagnosis and medication requirements. The dataset is loaded using pandas, with a shape of (139, 11).

2.1 Target Variable Derivation

The original dataset included multiple output features:

- **Final Diagnosis**: Indicates heart failure status.
- **Medication Features**: ACEI/ARB, ARNI, Beta Blockers, SGLT2i, MRA, Ivabradine, Digoxin, Diuretics, Nitrates/Hydralazine, Statins, Antiplatelets (all binary: 0 = not prescribed, 1 = prescribed).

These were reduced to a single `TARGET` variable:

- `TARGET = 0`: No heart failure diagnosis or no medications prescribed (all medication features are 0).
- `TARGET = 1`: Heart failure diagnosed or at least one medication prescribed.

This derivation simplifies the output to a binary classification problem, focusing on the presence of heart failure or the need for treatment.

3 Methodology

3.1 Data Preprocessing

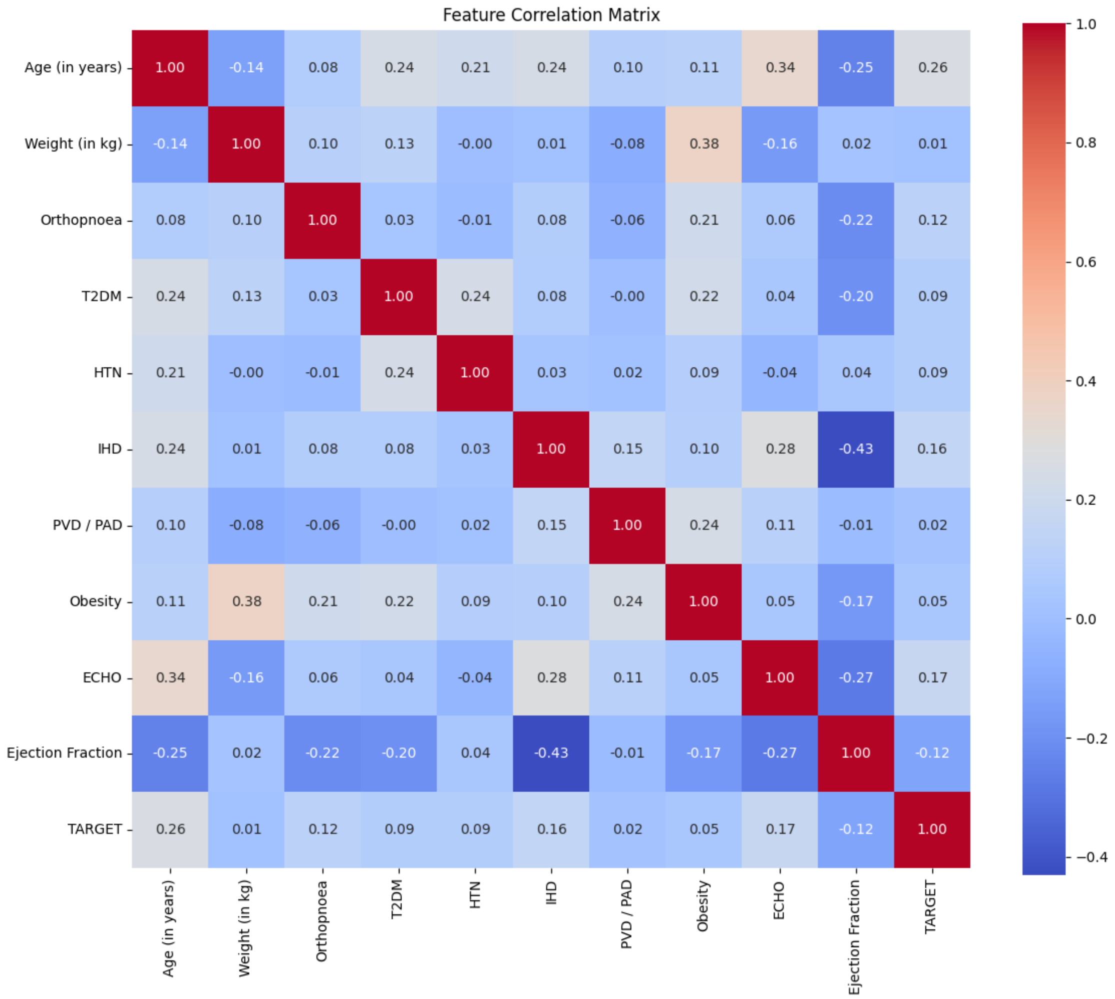
3.1.1 Feature Selection

The input features (Age, Weight, Orthopnoea, T2DM, HTN, IHD, PVD/PAD, Obesity, ECHO, Ejection Fraction) are retained as they provide relevant clinical information. The output features are consolidated into the `TARGET` variable, reducing the original 12 output features to one. No features were discarded based on the following criteria:

- **Low Variance**: All features exhibit sufficient variance.
- **High Correlation**: Correlation analysis shows all feature correlations are below ±0.9, with moderate correlations (e.g., Weight–Obesity, IHD–Ejection Fraction) indicating no multicollinearity.
- **Missing Values**: Missing values constitute only 2.08% (28/1390), below the 30–50% threshold.
- **Irrelevance**: No features are irrelevant (e.g., IDs, timestamps).
- **Low Model Importance**: All features are retained pending model evaluation.

No rows were discarded as none had greater than 50% missing values, outliers were not clearly erroneous, labels were correct, and no duplicates were found.

3.1.2 Correlation Matrix



	Age (in years)	Weight (in kg)	Orthopnoea	T2DM	HTN	IHD	PVD / PAD	Obesity	ECHO	Ejection Fraction	TARGET
Age (in years)	1.00	-0.14	0.08	0.24	0.21	0.24	0.10	0.11	0.34	-0.25	0.26
Weight (in kg)	-0.14	1.00	0.10	0.13	-0.00	0.01	-0.08	0.38	-0.16	0.02	0.01
Orthopnoea	0.08	0.10	1.00	0.03	-0.01	0.08	-0.06	0.21	0.06	-0.22	0.12
T2DM	0.24	0.13	0.03	1.00	0.24	0.08	-0.00	0.22	0.04	-0.20	0.09
HTN	0.21	-0.00	-0.01	0.24	1.00	0.03	0.02	0.09	-0.04	0.04	0.09
IHD	0.24	0.01	0.08	0.08	0.03	1.00	0.15	0.10	0.28	-0.43	0.16
PVD / PAD	0.10	-0.08	-0.06	-0.00	0.02	0.15	1.00	0.24	0.11	-0.01	0.02
Obesity	0.11	0.38	0.21	0.22	0.09	0.10	0.24	1.00	0.05	-0.17	0.05
ECHO	0.34	-0.16	0.06	0.04	-0.04	0.28	0.11	0.05	1.00	-0.27	0.17
Ejection Fraction	-0.25	0.02	-0.22	-0.20	0.04	-0.43	-0.01	-0.17	-0.27	1.00	-0.12
TARGET	0.26	0.01	0.12	0.09	0.09	0.16	0.02	0.05	0.17	-0.12	1.00

Figure 1: Correlation heatmap of input features, showing moderate correlations (e.g., Weight–Obesity, IHD–Ejection Fraction) with no multicollinearity (all correlations $< \pm 0.9$).

3.1.3 Handling Missing Values

Missing values (28/1390, 2.08%) were imputed using a model-based approach considering feature correlations:

- For each column with missing values, a separate predictive model was trained using correlated columns as predictors.

- **ECHO**: Imputed as 0 (Normal) or 1 (Abnormal) based on correlated features (e.g., IHD, Ejection Fraction).
- **Other Binary Columns** (Orthopnoea, T2DM, HTN, IHD, PVD/PAD, Obesity): Imputed as 0 (Absent) or 1 (Present).
- **Ejection Fraction**: Originally recorded as x%, converted to x/100 (range: 0 to 1) and imputed using correlated features (e.g., IHD, ECHO).

Correlation analysis confirmed moderate relationships (e.g., Weight–Obesity, IHD–Ejection Fraction), supporting imputation without introducing bias. All features were retained for model training.

3.1.4 Train-Test Split and Scaling

- **Train-Test Split**: 80% training, 20% testing using `train_test_split` with `random_state=42`.
- **Feature Scaling**: Applied for SVC using `StandardScaler`.

3.2 Model Training and Evaluation

The following models are trained and evaluated:

- **Random Forest Classifier (RFC)**: Two models—default and tuned (`max_samples=0.75`, `random_state =42`).
- **K-Nearest Neighbors (KNN)**: Default parameters.
- **Support Vector Classifier (SVC)**: With `probability=True`; `RandomizedSearchCV` attempted but interrupted.
- **Gradient Boosting Classifier (GBC)**: `n_estimators=1000`, `learning_rate=0.1`, `max_depth=3`.
- **Linear Discriminant Analysis (LDA)**: Default parameters.
- **AdaBoost Classifier (ABC)**: `n_estimators=1000`, `learning_rate=1.0`.
- **Extra Trees Classifier (ETC)**: `n_estimators=2000`, `max_features=0.5`.
- **XGBoost Classifier (XGB)**: `n_estimators=1000`, `max_depth=3`, `eval_metric='logloss'`.
- **Logistic Regression (LR)**: `max_iter=1000`.
- **Decision Tree Classifier (DTC)**: `criterion='gini'`, `max_depth=None`.
- **Bagging Classifier**: `n_estimators=1000`, `max_samples=0.8`, `max_features=1.0`.
- **Gaussian Naive Bayes (GNB)**: Default parameters.

Models are evaluated using accuracy, precision, F1-score, and specificity. ROC AUC scores are computed for some models but not all.

3.3 Feature Selection with Boruta

The Boruta algorithm is used with a Random Forest Classifier (`n_estimators='auto'`, `verbose=2`, `random_state=1`). It is fitted on the entire dataset, but no significant features are identified (`feat_selector.n_features_ = 0`).

3.4 Evaluation Metrics

- **Accuracy**: Proportion of correct predictions.
- **Precision**: Proportion of positive predictions that are correct.
- **F1-Score**: Harmonic mean of precision and recall.
- **Specificity**: Proportion of actual negatives correctly identified.

4 Results

4.1 Model Performance

Table 1: Performance Metrics of Classification Models

Model	Accuracy	Precision	F1-Score	Specificity
Random Forest Classifier (RFC)	0.7500	0.8077	0.8571	0.0000
K-Nearest Neighbors (KNN)	0.7500	0.8077	0.8571	0.0000
Support Vector Classifier (SVC)	**0.8214**	0.8214	**0.9020**	0.0000
Gradient Boosting Classifier (GBC)	0.7143	**0.8261**	0.8261	0.2000
Linear Discriminant Analysis (LDA)	0.6786	0.7917	0.8085	0.0000
AdaBoost Classifier (ABC)	0.6786	0.7917	0.8085	0.0000
Extra Trees Classifier (ETC)	0.7857	0.8148	0.8800	0.0000
XGBoost Classifier (XGB)	0.6071	0.7727	0.7556	0.0000
Logistic Regression (LR)	0.7143	0.8000	0.8333	0.0000
Decision Tree Classifier (DTC)	0.6071	0.8000	0.7442	**0.2000**
Bagging Classifier	0.7500	0.8077	0.8571	0.0000
Gaussian Naive Bayes (GNB)	0.5714	0.7619	0.7273	0.0000

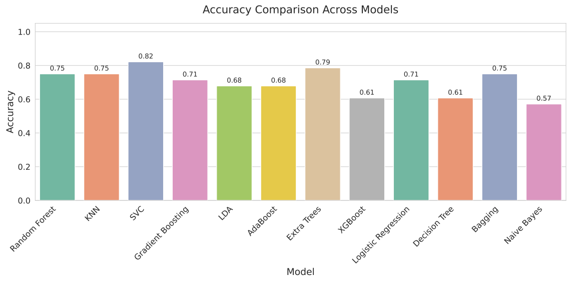


Figure 2: Accuracy comparison across models.

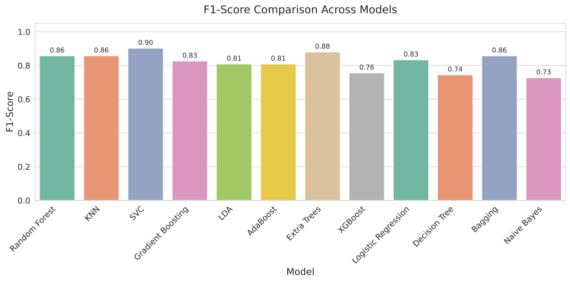


Figure 3: F1-Score comparison across models.

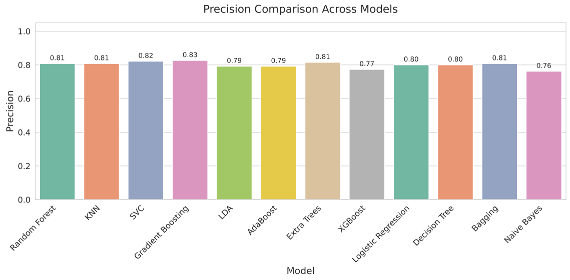


Figure 4: Precision comparison across models.

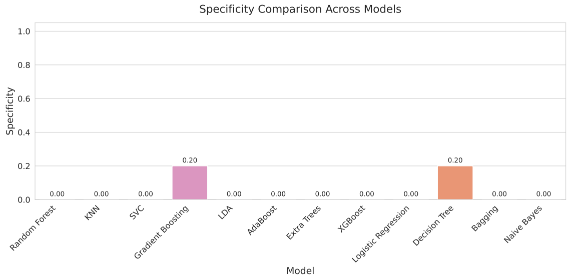


Figure 5: Specificity comparison across models.

4.2 Boruta Feature Selection

The Boruta algorithm identified no significant features (`feat_selector.n_features_ = 0`). The resulting `best_features` array had shape (139, 0), preventing model retraining with selected features.

5 Discussion

5.1 Model Performance

The SVC model achieved the highest accuracy (0.8214), followed by ETC (0.7857). The additional metrics (precision, F1-score, specificity) highlight the models' performance on the positive class (class 1) but reveal poor specificity (0.0000 for most models), indicating no correct predictions for the minority class (class 0). This is likely due to class imbalance (5 class 0 vs. 23 class 1 instances in the test set). GBC and DTC showed slightly better specificity (0.2000), but their overall accuracy was lower. GNB had the lowest accuracy (0.5714), suggesting limited suitability for this dataset.

5.2 Challenges with Boruta

The Boruta algorithm's failure to select features is a significant issue, potentially due to:

- **Small Sample Size**: 139 samples may be insufficient for reliable feature selection.

- **Class Imbalance**: May skew feature importance calculations.

5.3 Limitations

- **Class Imbalance**: Severely impacts minority class prediction, as seen in low specificity.
- **Small Dataset**: 139 samples limit model training and generalization.
- **Boruta Failure**: Prevents evaluation with optimized features.
- **Interrupted SVC Tuning**: Limits SVC optimization.
- **Missing Value Imputation**: While minimal (2.08%), model-based imputation may introduce minor biases.

6 Recommendations

- **Address Class Imbalance**: Use SMOTE or class-weighted models to improve minority class performance.
- **Fix Boruta Implementation**: Verify dataset alignment and test alternative feature selection methods (e.g., Recursive Feature Elimination).
- **Expand Dataset**: Collect more samples to enhance model robustness.
- **Complete Hyperparameter Tuning**: Re-run `RandomizedSearchCV` for SVC and explore tuning for GBC and ETC.
- **Ensemble Methods**: Combine top-performing models (SVC, ETC) using a `VotingClassifier`.

7 Conclusion

The study evaluated multiple machine learning models for heart failure prediction, with SVC achieving the highest accuracy (0.8214). The dataset was preprocessed by deriving a binary `TARGET` variable and imputing minimal missing values (2.08%) using correlation-based models. However, class imbalance and small dataset size hindered minority class prediction, and the Boruta algorithm failed to select features due to a likely small dataset. Future work should focus on addressing class imbalance, improving feature selection, and expanding the dataset to enhance predictive performance.