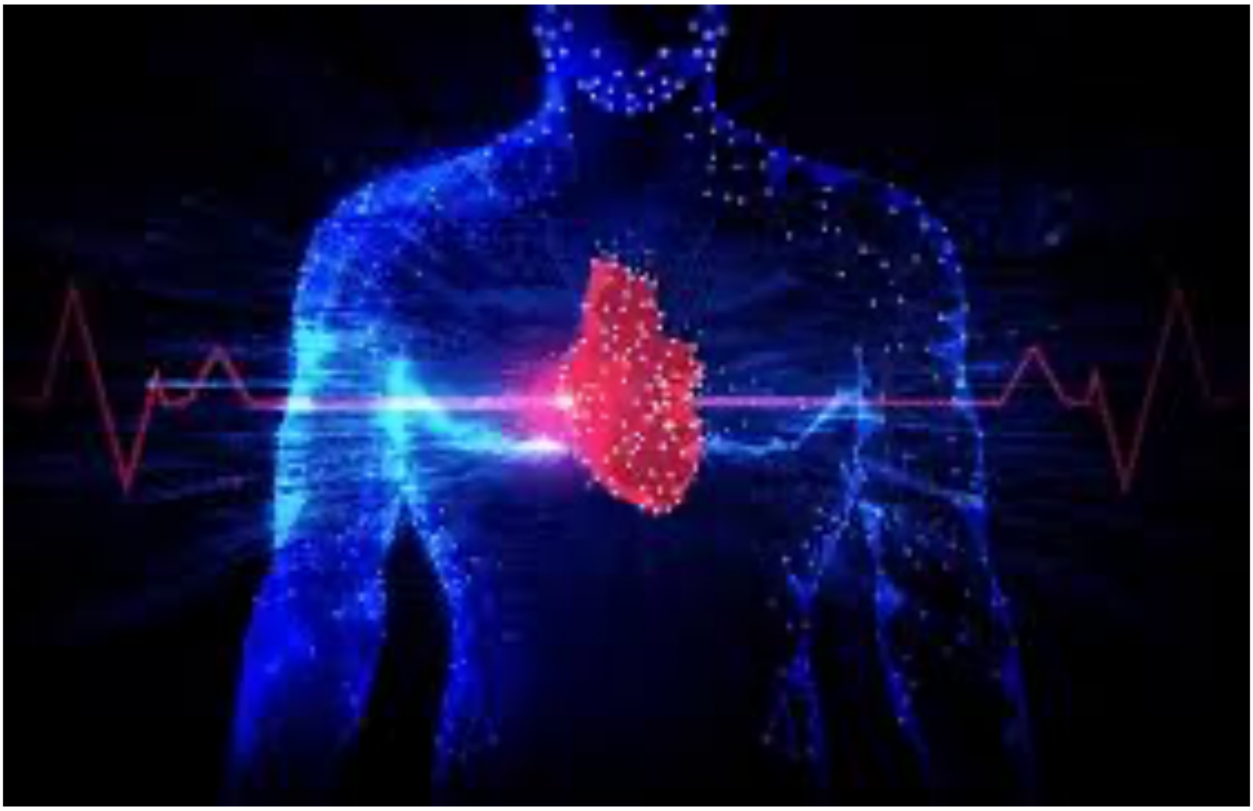
Report : Cardiovascular Disease Prediction



Cardiovascular Disease Prediction

Prepared by : Ronak Waghmare

Email : ronak818082@gmail.com

Literature Review

A review of the literature and identification of research gaps

1. Background and classical risk models :

Cardiovascular risk prediction started with large epidemiological studies that produced clinically accepted risk functions. The Framingham Risk Score and related pooled-cohort equations estimate 10-year (and lifetime) risk from demographic and clinical variables (age, sex, blood pressure, cholesterol, smoking, diabetes) and remain widely used in clinical practice as transparent, validated baselines. These models are valuable for clinical decision-making but are limited in capturing complex, non-linear relationships and interactions in modern electronic datasets.

2. Datasets commonly used in ML research :

A recurring dataset in ML heart-disease research is the UCI "Heart Disease" collection (Cleveland, Statlog, VA, Hungary, Switzerland subsets), which contains ~300 records (Cleveland subset) and a small set of clinical features — widely used for benchmarking but limited in size and diversity. Other recent works also use larger hospital EHR datasets when available, but public, large-scale labeled CVD datasets remain scarce.

3. Evaluation practices and common metrics :

Researchers typically report accuracy, precision, recall, F1-score, and ROC-AUC. In healthcare contexts many authors emphasize that sensitivity/recall (catching at-risk patients) and calibration (how predicted probabilities map to real risk) are often more clinically relevant than raw accuracy. Recent papers also present confusion matrices and ROC curves to expose trade-offs.

4. Methodological challenges reported in the literature

Small / non-representative datasets: public datasets like UCI are small and may not reflect current, diverse clinical populations. This limits generalizability

Class imbalance: many datasets have far fewer positive CVD cases; studies show resampling (SMOTE, SMOTE-ENN), class weighting, or cost-sensitive learning can materially affect classifier performance.

Lack of explainability: black-box models (deep nets, ensembles) improve performance but reduce clinician trust; a growing stream of work stresses explainable AI (XAI) and rule extraction for transparent predictions.

Inconsistent pre-processing & benchmarking: many papers report strong results but use different pre-processing, feature sets, cross-validation schemes, and metrics, making direct comparisons difficult. Recent systematic reviews call for standardized pipelines and better reporting.

5. Representative recent studies :

XGBoost + optimized hyperparameters with feature selection shows strong predictive performance on CVD datasets (example: recent Nature Scientific Reports / XGBoost works).

Studies combining resampling (SMOTE, SMOTE-ENN) with traditional ML algorithms report improvements in sensitivity for minority (disease) class detection

Explainable AI work uses XGBoost plus rule extraction or post-hoc explainers to give clinicians interpretable decision rules while maintaining accuracy.

Identification of Research Gaps

Limited generalizability due to small or homogeneous dataset :

Many studies rely on small public datasets (e.g., UCI Cleveland), which limit external validity. There is a need for models trained and validated on larger, multi-center EHR data with demographic diversity

Inadequate and inconsistent pre-processing reporting :

Several papers focus on algorithms but under-report or under-apply rigorous pre-processing (missing value strategy, outlier handling, feature engineering, leakage checks). This creates reproducibility and performance-comparison issues. Reviews call for standardized pre-processing pipelines

Class imbalance handling not uniformly applied :

Imbalance treatments (SMOTE variants, class weighting) materially change results, but are not consistently applied or compared across studies; some high-accuracy claims are sensitive to imbalance handling

Explainability vs. performance tradeoff :

Many high-performing models are black boxes. Although recent work explores explainable solutions, more systematic integration of explainability (feature importance, rule extraction, calibrated probabilities) with performance evaluation is needed for clinical adoption.

Evaluation beyond discrimination :

Few studies evaluate calibration, decision-curve analysis, or clinical utility (e.g., effect on patient outcomes). Most stop at AUC/accuracy; the literature encourages reporting calibration and clinical impact measures.

Reproducibility and benchmarking :

There is a lack of agreed benchmark splits, feature sets, and reporting standards; this makes it hard to know whether reported improvements are methodological or due to experimental choice

Research Gaps

Comprehensive Handling of Class Imbalance:

- *Despite using SMOTE, many studies do not explore alternative oversampling/undersampling methods or their impacts on generalizability.*
- *Imbalanced data remains a persistent challenge for consistent minority class prediction across populations.*

Transparent and Comparative Evaluation:

- Few works conduct a unified, systematic comparison of ANN/DL, classical ML, and ensemble approaches on the same dataset with identical pre-processing for fair benchmarking

Clinical Interpretability and Trust:

- Although explainable AI tools (e.g., SHAP) are gaining traction, interpretability in clinical workflows is not standardized; many studies do not address how clinicians integrate these insights for decision support.

Generalizability to Real-World Patient Data:

- *Many algorithms are trained and validated on highly curated research datasets.*
- *Practical application and external validation on diverse, real-world multicenter data are limited with scarce reporting of deployment and integration into clinical settings.*

Application in Low-Resource and Diverse Populations:

- *Most studies lack validation on geographically and ethnically diverse cohorts, especially from regions with higher disease burden.*

Feature Engineering and Selection:

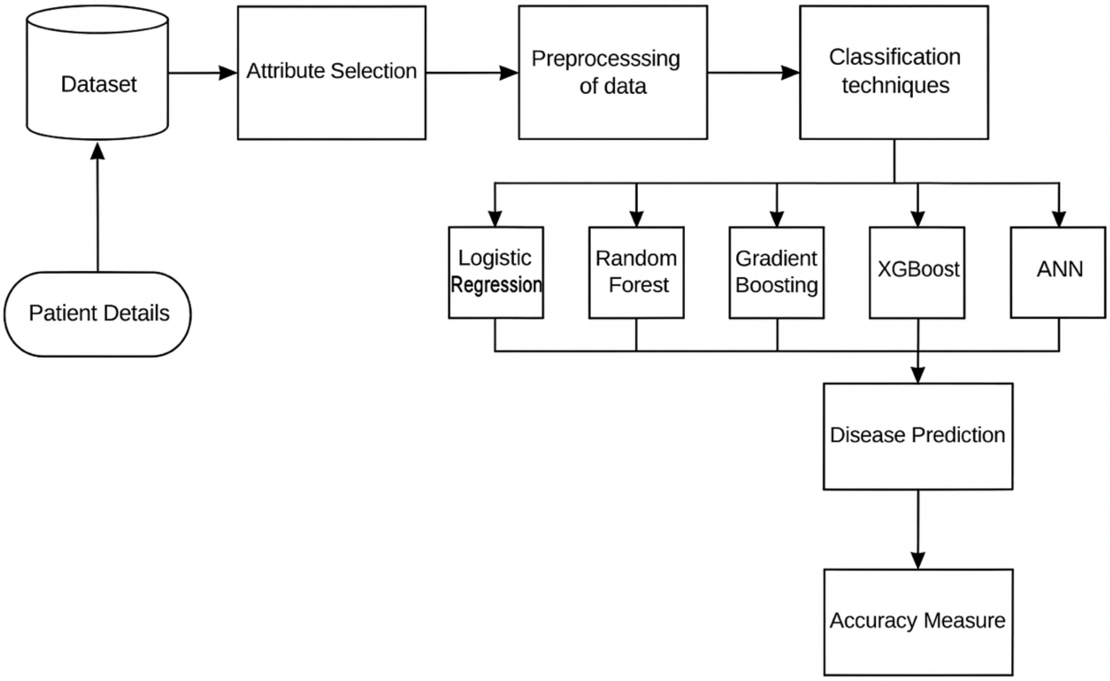
- *Few papers rigorously compare feature selection methods or integrate domain knowledge alongside automatic selection, affecting clinical relevance.*

Proposed Solution and Objectives

Bridging the Research Gap :

"The literature review revealed that while many machine learning models, such as Random Forest and SVM, have been used for CVD prediction, a significant gap exists in their interpretability. Most high-accuracy models operate as 'black boxes,' making it difficult for clinicians to trust their predictions or understand the key risk factors for an individual patient. Our work aims to directly address this gap by proposing a model that not only predicts CVD risk with high accuracy but also provides clear, human-understandable explanations for its decisions."

The Architecture Diagram

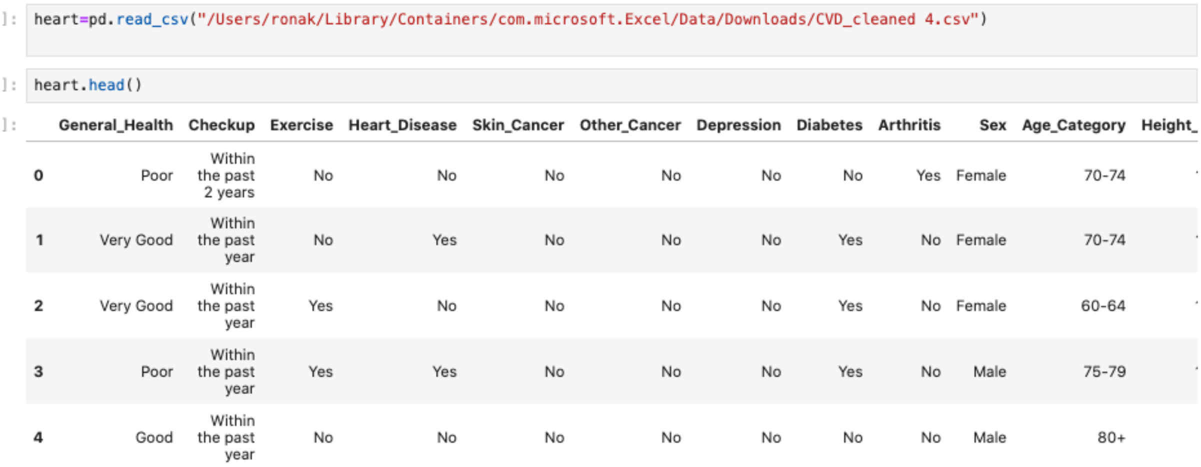


Patient Details

- General Health (self-reported status)
- Checkup (frequency of medical checkups)
- Exercise (physical activity habits)
- Heart Disease (presence/absence)
- Skin Cancer
- Other Cancer
- Depression
- Diabetes
- Arthritis
- Age Category
- Height
- Weight
- BMI
- BMI Group (categorized BMI: underweight, normal, overweight, obese)
- Alcohol Consumption
- Fruit Consumption
- Vegetable Consumption
- Potato Consumption
- Sex (Female/Male)
- Smoking History (Yes/No)

Dataset :

- The patient details are aggregated into a structured **dataset**.
- The dataset typically consists of rows (patients) and columns (features/attributes).
- This dataset is the main input for machine learning.
- Challenges here include:
  - Noisy or inconsistent data
  - Imbalanced classes (e.g., far more healthy than diseased cases)



```
]: heart=pd.read_csv("/Users/ronak/Library/Containers/com.microsoft.Excel/Data/Downloads/CVD_cleaned 4.csv")
```

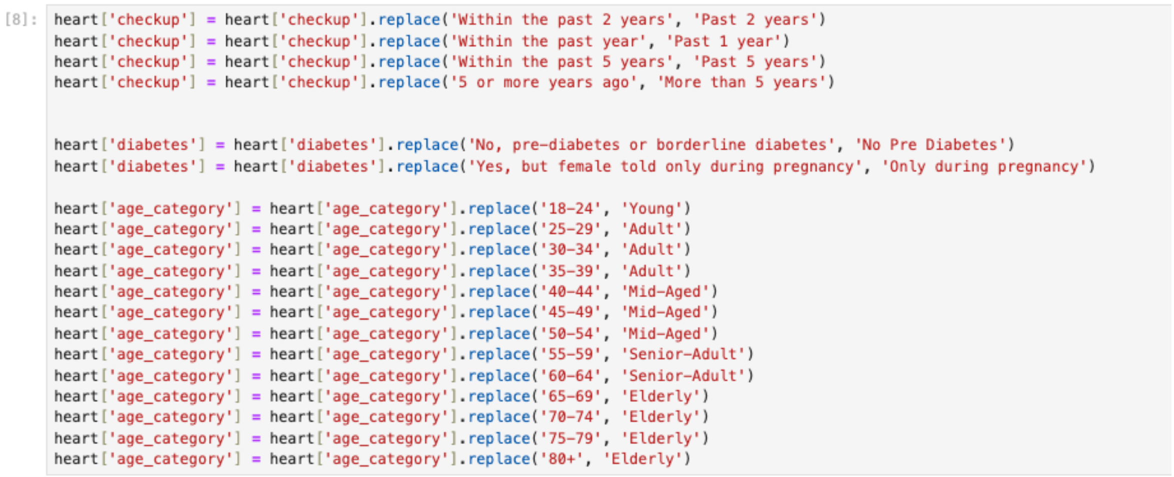
```
]: heart.head()
```

	General_Health	Checkup	Exercise	Heart_Disease	Skin_Cancer	Other_Cancer	Depression	Diabetes	Arthritis	Sex	Age_Category	Height_
0	Poor	Within the past 2 years	No	No	No	No	No	No	Yes	Female	70-74	
1	Very Good	Within the past year	No	Yes	No	No	No	Yes	No	Female	70-74	
2	Very Good	Within the past year	Yes	No	No	No	No	Yes	No	Female	60-64	
3	Poor	Within the past year	Yes	Yes	No	No	No	Yes	No	Male	75-79	
4	Good	Within the past year	No	No	No	No	No	No	No	Male	80+	

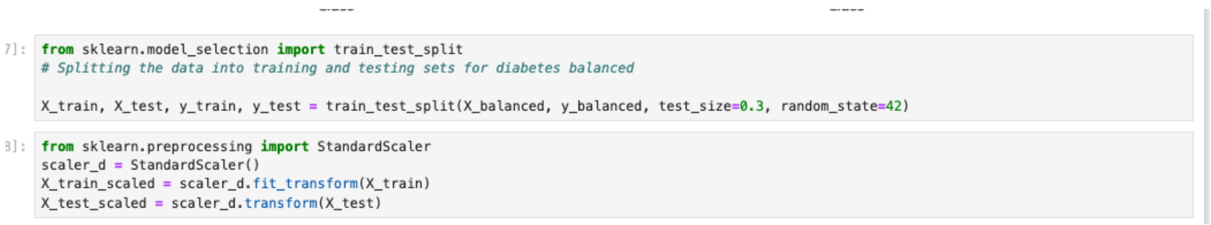
Pre-processing of Data :

- Ensures data is ready for machine learning algorithms.
- Steps usually involve:

1. **Data Cleaning**: Handle missing values, remove duplicates.
2. **Normalization/Standardization**: Scale numerical values
3. **Encoding categorical variables**: Convert categorical data (e.g., gender: Male/Female) into numerical form.
4. **Splitting**: Divide into training, validation, and test sets.



```
[8]: heart['checkup'] = heart['checkup'].replace('Within the past 2 years', 'Past 2 years')
     heart['checkup'] = heart['checkup'].replace('Within the past year', 'Past 1 year')
     heart['checkup'] = heart['checkup'].replace('Within the past 5 years', 'Past 5 years')
     heart['checkup'] = heart['checkup'].replace('5 or more years ago', 'More than 5 years')


     heart['diabetes'] = heart['diabetes'].replace('No, pre-diabetes or borderline diabetes', 'No Pre Diabetes')
     heart['diabetes'] = heart['diabetes'].replace('Yes, but female told only during pregnancy', 'Only during pregnancy')

     heart['age_category'] = heart['age_category'].replace('18-24', 'Young')
     heart['age_category'] = heart['age_category'].replace('25-29', 'Adult')
     heart['age_category'] = heart['age_category'].replace('30-34', 'Adult')
     heart['age_category'] = heart['age_category'].replace('35-39', 'Adult')
     heart['age_category'] = heart['age_category'].replace('40-44', 'Mid-Aged')
     heart['age_category'] = heart['age_category'].replace('45-49', 'Mid-Aged')
     heart['age_category'] = heart['age_category'].replace('50-54', 'Mid-Aged')
     heart['age_category'] = heart['age_category'].replace('55-59', 'Senior-Adult')
     heart['age_category'] = heart['age_category'].replace('60-64', 'Senior-Adult')
     heart['age_category'] = heart['age_category'].replace('65-69', 'Elderly')
     heart['age_category'] = heart['age_category'].replace('70-74', 'Elderly')
     heart['age_category'] = heart['age_category'].replace('75-79', 'Elderly')
     heart['age_category'] = heart['age_category'].replace('80+', 'Elderly')
```

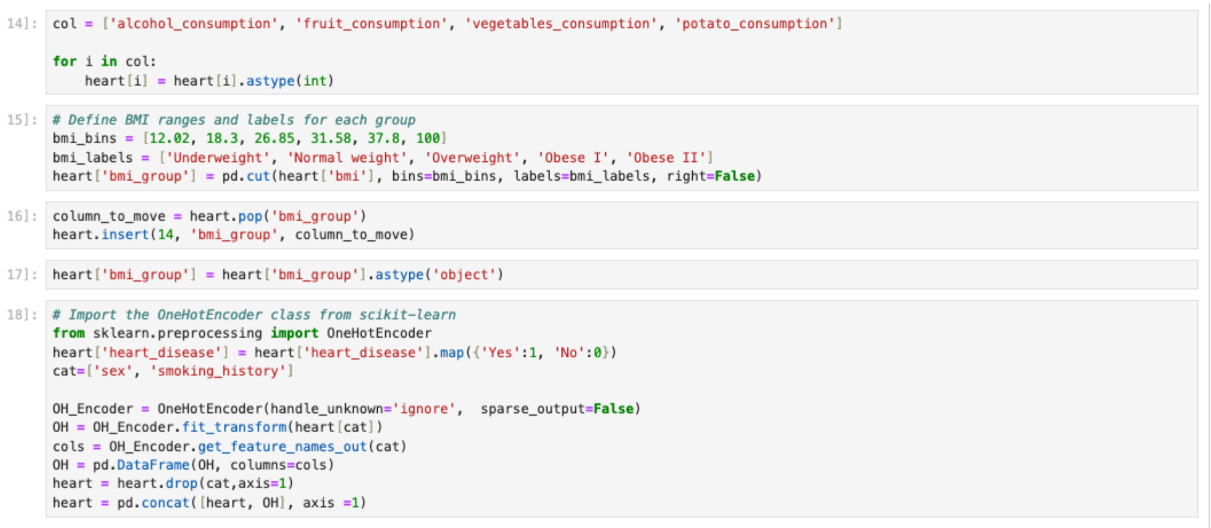


7]:

```
from sklearn.model_selection import train_test_split
# Splitting the data into training and testing sets for diabetes balanced

X_train, X_test, y_train, y_test = train_test_split(X_balanced, y_balanced, test_size=0.3, random_state=42)
```

8]:

```
from sklearn.preprocessing import StandardScaler
scaler_d = StandardScaler()
X_train_scaled = scaler_d.fit_transform(X_train)
X_test_scaled = scaler_d.transform(X_test)
```



14]:

```
col = ['alcohol_consumption', 'fruit_consumption', 'vegetables_consumption', 'potato_consumption']

for i in col:
    heart[i] = heart[i].astype(int)
```

15]:

```
# Define BMI ranges and labels for each group
bmi_bins = [12.02, 18.3, 26.85, 31.58, 37.8, 100]
bmi_labels = ['Underweight', 'Normal weight', 'Overweight', 'Obese I', 'Obese II']
heart['bmi_group'] = pd.cut(heart['bmi'], bins=bmi_bins, labels=bmi_labels, right=False)
```

16]:

```
column_to_move = heart.pop('bmi_group')
heart.insert(14, 'bmi_group', column_to_move)
```

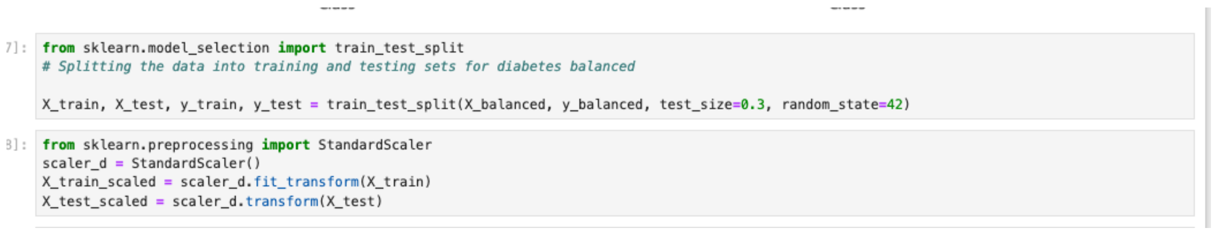
17]:

```
heart['bmi_group'] = heart['bmi_group'].astype('object')
```

18]:

```
# Import the OneHotEncoder class from scikit-learn
from sklearn.preprocessing import OneHotEncoder
heart['heart_disease'] = heart['heart_disease'].map({'Yes':1, 'No':0})
cat=['sex', 'smoking_history']

OH_Encoder = OneHotEncoder(handle_unknown='ignore',  sparse_output=False)
OH = OH_Encoder.fit_transform(heart[cat])
cols = OH_Encoder.get_feature_names_out(cat)
OH = pd.DataFrame(OH, columns=cols)
heart = heart.drop(cat,axis=1)
heart = pd.concat([heart, OH], axis =1)
```

Splitting: Divide into training, validation, and test sets.



7]:

```
from sklearn.model_selection import train_test_split
# Splitting the data into training and testing sets for diabetes balanced

X_train, X_test, y_train, y_test = train_test_split(X_balanced, y_balanced, test_size=0.3, random_state=42)
```

8]:

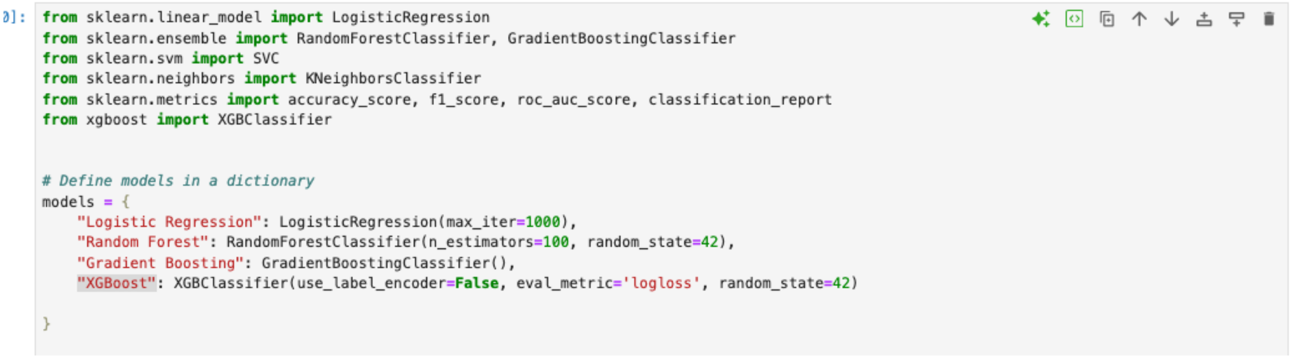
```
from sklearn.preprocessing import StandardScaler
scaler_d = StandardScaler()
X_train_scaled = scaler_d.fit_transform(X_train)
X_test_scaled = scaler_d.transform(X_test)
```

Classification Techniques :

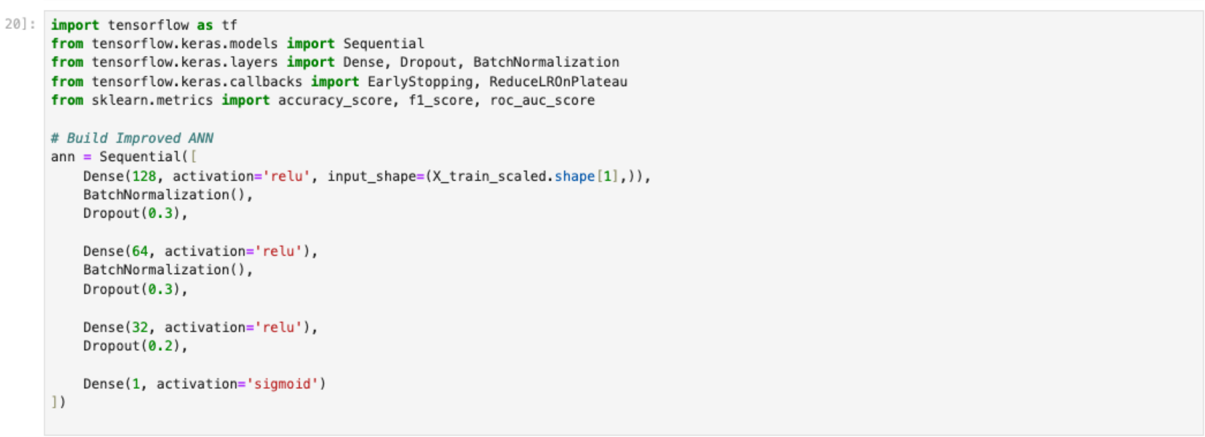
- This block applies different machine learning models to classify whether a patient has a disease or not.

 Each technique has strengths/weaknesses:

- **Logistic Regression (LR):** Simple, interpretable, works well for binary classification.
- **Random Forest (RF):** An ensemble of decision trees; handles non-linearity and prevents overfitting.
- **Gradient Boosting (GB):** Builds trees sequentially to reduce errors; high accuracy.
- **XGBoost:** Optimized version of Gradient Boosting, very powerful and widely used in healthcare predictions.
- **Artificial Neural Network (ANN):** Mimics brain-like structures; excellent at capturing complex relationships in large datasets



```python
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score, classification_report
from xgboost import XGBClassifier


# Define models in a dictionary
models = {
    "Logistic Regression": LogisticRegression(max_iter=1000),
    "Random Forest": RandomForestClassifier(n_estimators=100, random_state=42),
    "Gradient Boosting": GradientBoostingClassifier(),
    "XGBoost": XGBClassifier(use_label_encoder=False, eval_metric='logloss', random_state=42)

}
```



```python
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, BatchNormalization
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score

# Build Improved ANN
ann = Sequential([
    Dense(128, activation='relu', input_shape=(X_train_scaled.shape[1],)),
    BatchNormalization(),
    Dropout(0.3),

    Dense(64, activation='relu'),
    BatchNormalization(),
    Dropout(0.3),

    Dense(32, activation='relu'),
    Dropout(0.2),

    Dense(1, activation='sigmoid')
])
```

Disease Prediction :

Binary Classification

- Predicts if a person has a particular disease (Yes/No).
- Example: **Heart Disease Prediction**
  - Output: **Yes (1)** = patient likely has heart disease
  - Output: **No (0)** = patient unlikely to have heart disease
- Models used: Logistic Regression, Random Forest, SVM, XGBoost, ANN.
- Useful in **screening tests** where the goal is to flag potential cases.

Probability Scores / Risk Prediction :

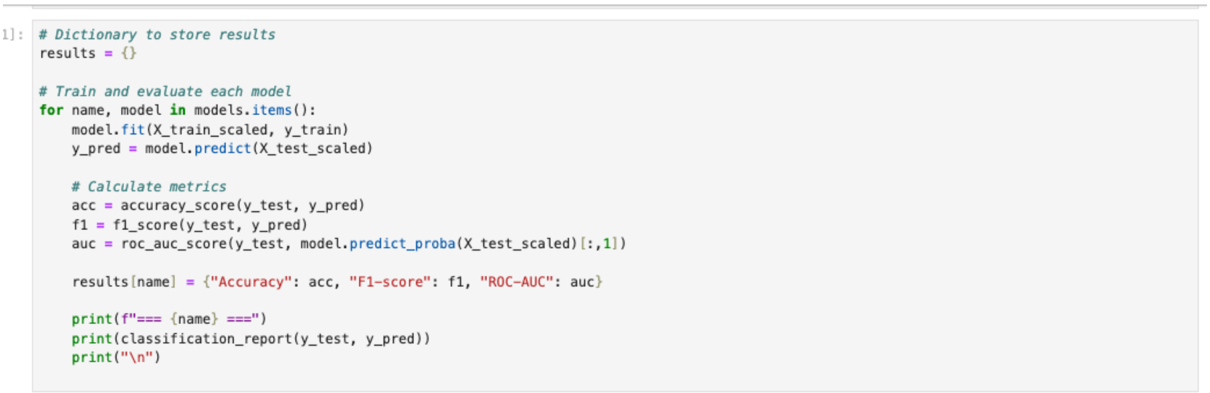
- Instead of just a Yes/No, models provide a **probability score**.

- Example:

  - "Patient has **80% chance of diabetes**, 15% chance of heart disease, 5% chance of no disease."

- This is very important in healthcare because:

  - Doctors can set a threshold (e.g., if probability >70%, recommend further testing).
  - Patients can be stratified into **low-risk, medium-risk, high-risk groups**.



```
# Dictionary to store results
results = {}

# Train and evaluate each model
for name, model in models.items():
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)

    # Calculate metrics
    acc = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    auc = roc_auc_score(y_test, model.predict_proba(X_test_scaled)[:,1])

    results[name] = {"Accuracy": acc, "F1-score": f1, "ROC-AUC": auc}

    print(f"=== {name} ===")
    print(classification_report(y_test, y_pred))
    print("\n")
```

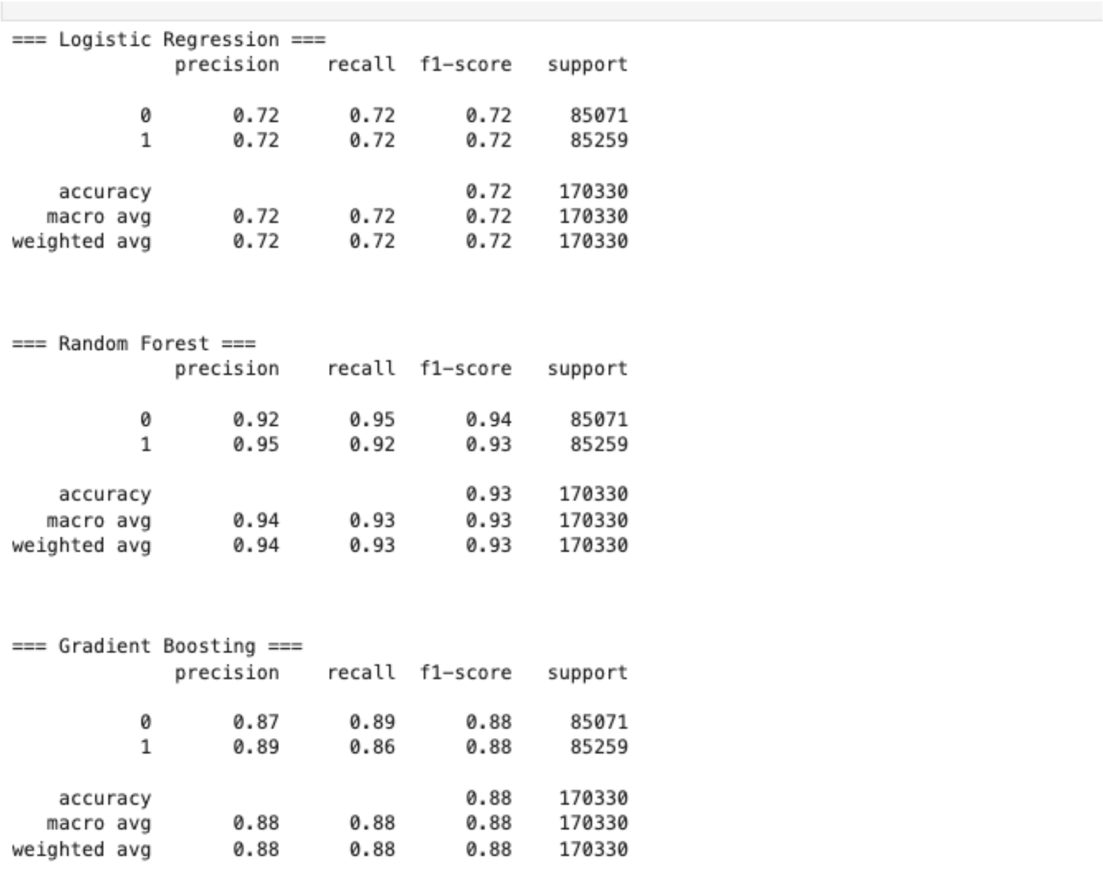
Accuracy Measure :

 After prediction, model performance is evaluated.

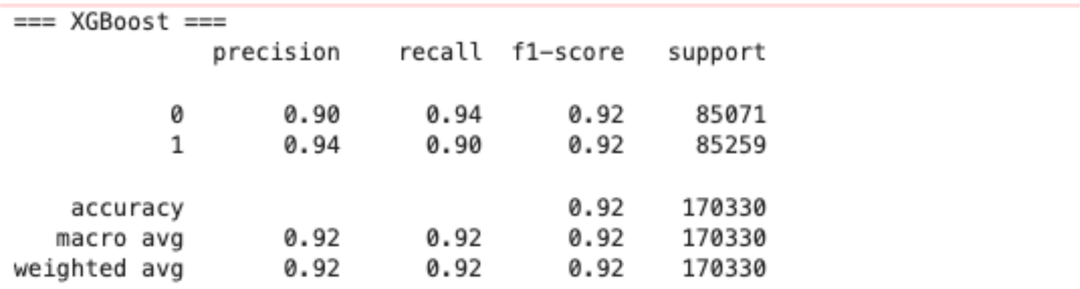
 Common metrics include:

- **Accuracy**: Percentage of correctly predicted cases.
- **Precision & Recall**: Useful when class imbalance exists.
- **F1-score**: Balance between precision and recall.
- **AUC-ROC Curve**: Measures ability to distinguish between classes.

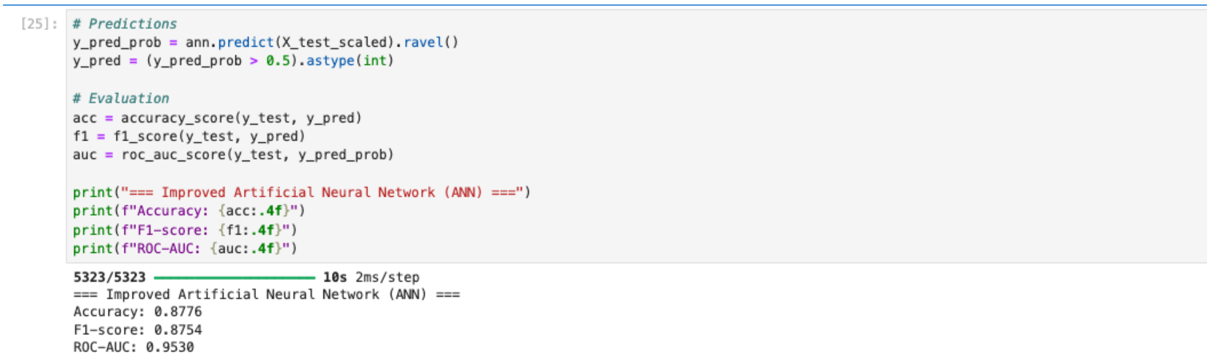
 Helps identify the best performing model for disease prediction.



```
=== Logistic Regression ===
              precision    recall  f1-score   support

           0       0.72      0.72      0.72     85071
           1       0.72      0.72      0.72     85259

    accuracy                           0.72    170330
   macro avg       0.72      0.72      0.72    170330
weighted avg       0.72      0.72      0.72    170330



=== Random Forest ===
              precision    recall  f1-score   support

           0       0.92      0.95      0.94     85071
           1       0.95      0.92      0.93     85259

    accuracy                           0.93    170330
   macro avg       0.94      0.93      0.93    170330
weighted avg       0.94      0.93      0.93    170330



=== Gradient Boosting ===
              precision    recall  f1-score   support

           0       0.87      0.89      0.88     85071
           1       0.89      0.86      0.88     85259

    accuracy                           0.88    170330
   macro avg       0.88      0.88      0.88    170330
weighted avg       0.88      0.88      0.88    170330
```



```
=== XGBoost ===
              precision    recall  f1-score   support

           0       0.90      0.94      0.92     85071
           1       0.94      0.90      0.92     85259

    accuracy                           0.92    170330
   macro avg       0.92      0.92      0.92    170330
weighted avg       0.92      0.92      0.92    170330
```



[25]:

```python
# Predictions
y_pred_prob = ann.predict(X_test_scaled).ravel()
y_pred = (y_pred_prob > 0.5).astype(int)

# Evaluation
acc = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
auc = roc_auc_score(y_test, y_pred_prob)

print("=== Improved Artificial Neural Network (ANN) ===")
print(f"Accuracy: {acc:.4f}")
print(f"F1-score: {f1:.4f}")
print(f"ROC-AUC: {auc:.4f}")
```

```
5323/5323 ━━━━━━━━━━━━━━━━━━━━ 10s 2ms/step
=== Improved Artificial Neural Network (ANN) ===
Accuracy: 0.8776
F1-score: 0.8754
ROC-AUC: 0.9530
```

Research Questions and Objectives:

Data-Related

- What patient attributes (demographics, lifestyle, clinical measures) are most influential in predicting cardiovascular disease?
- How can data pre-processing and feature selection improve the performance of predictive models for CVD?

Modelling-Related

- Which classification techniques (e.g., Logistic Regression, Random Forest, Gradient Boosting, XGBoost, ANN) provide the most accurate prediction of cardiovascular disease?
- How does the performance of traditional statistical models (Logistic Regression) compare with advanced machine learning models (XGBoost, ANN) for CVD prediction?

Prediction and Evaluation

- Can the models provide not only binary predictions (CVD present/absent) but also reliable probability scores indicating individual risk levels?
- How effective are these models in stratifying patients into **low-risk, medium-risk, and high-risk groups**?
- What are the trade-offs between accuracy, interpretability, and computational efficiency among different models?

Research Objectives :

Primary Objective

- To develop and evaluate machine learning–based models for the accurate prediction of cardiovascular disease using patient health, lifestyle, and clinical data

.

Secondary Objectives

- To identify and analyze the most significant features influencing cardiovascular disease (e.g., age, BMI, smoking history, diabetes, exercise, alcohol consumption).
- To pre-process and transform raw patient data into a clean, structured dataset suitable for machine learning applications.

- To implement and compare multiple classification techniques (Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and ANN).
- To evaluate model performance using metrics such as **accuracy, precision, recall, F1-score, and AUC-ROC curve**.
- To explore the ability of models to generate **probability-based risk predictions**, aiding in patient risk stratification and preventive healthcare planning.
- To recommend the most effective predictive model for integration into clinical decision support systems (CDSS).

Comparative Analysis

1. Logistic Regression (Baseline Model)

- **Accuracy:** 0.72
- **Precision/Recall/F1:** Balanced at 0.72 for both classes.
- **Strengths:**
  - Simple and interpretable.
  - Works well as a baseline.
- **Weaknesses:**
  - Linear assumptions limit performance.
  - Struggles with complex, non-linear relationships in the dataset.

2. Random Forest

- **Accuracy:** 0.93
- **Precision:** 0.92 (Class 0), 0.95 (Class 1)
- **Recall:** 0.95 (Class 0), 0.92 (Class 1)
- **F1-score:** ~0.93
- **Strengths:**
  - Handles non-linearities well.
  - Robust to noise and overfitting.
  - Provides feature importance ranking.
- **Weaknesses:**
  - Computationally more expensive than LR.
  - May require tuning for best results.

3. Gradient Boosting

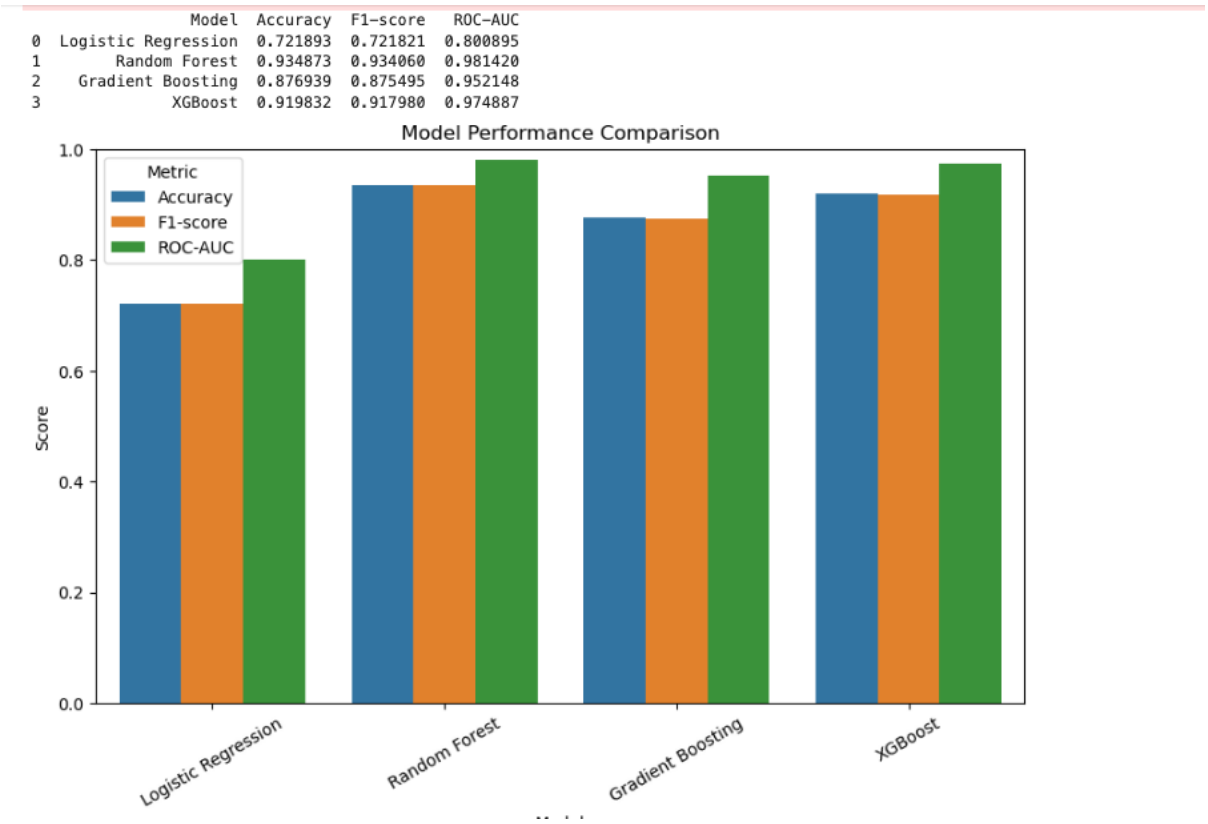
- **Accuracy:** 0.88
- **Precision:** 0.87 (Class 0), 0.89 (Class 1)
- **Recall:** 0.89 (Class 0), 0.86 (Class 1)
- **F1-score:** 0.88
- **Strengths:**
  - Sequential learning corrects previous errors.
  - Good balance between precision and recall.
- **Weaknesses:**
  - More sensitive to hyperparameter tuning.
  - Slower training compared to RF.

4. XGBoost

- **Accuracy:** 0.92
- **Precision:** 0.90 (Class 0), 0.94 (Class 1)
- **Recall:** 0.94 (Class 0), 0.90 (Class 1)
- **F1-score:** 0.92
- **Strengths:**
  - Optimized gradient boosting with regularization.
  - High predictive performance.
  - Efficient handling of large datasets.
- **Weaknesses:**
  - Still less interpretable compared to LR.
  - Requires careful hyperparameter tuning.

Artificial Neural Network (ANN)

- **Accuracy:** 0.8776
- **F1-score:** 0.8754
- **ROC-AUC:** 0.9530 (highest among all models)
- **Strengths:**
  - Excellent at capturing complex, non-linear interactions.
  - High AUC indicates strong discriminative ability.
  - Potential for further improvements with deep architectures.
- **Weaknesses:**
  - Slightly lower accuracy compared to Random Forest and XGBoost.
  - Requires more computational resources.
  - Black-box nature reduces interpretability.



```
                Model  Accuracy  F1-score   ROC-AUC
0  Logistic Regression  0.721893  0.721821  0.800895
1        Random Forest  0.934873  0.934060  0.981420
2    Gradient Boosting  0.876939  0.875495  0.952148
3              XGBoost  0.919832  0.917980  0.974887
```

Conclusion

- For **accuracy-driven prediction tasks**, **Random Forest** is the best performing algorithm.
- For **risk stratification and probability-based predictions**, the **ANN** is superior due to its higher ROC-AUC.
- For a balanced trade-off between accuracy, robustness, and efficiency, **XGBoost** is an excellent choice.
- The proposed solution (ANN with improved architecture) demonstrates its strength in **risk detection** even though its accuracy is slightly behind RF and XGBoost.

Case Study :

- **Problem Statement**

Cardiovascular disease (CVD) is one of the leading causes of death worldwide. Early detection can help reduce risk and improve patient outcomes. However, manual diagnosis based on clinical records is often time-consuming and subject to human error. Therefore, there is a need for automated predictive models that can identify at-risk individuals using patient health and lifestyle data.

 Objectives :To build machine learning models for predicting cardiovascular disease.

- To compare the performance of traditional models (Logistic Regression) with advanced models (Random Forest, Gradient Boosting, XGBoost, ANN).
- To evaluate models based on accuracy, F1-score, ROC-AUC.
- To provide actionable recommendations for healthcare decision-making.

2. Data Pre-processing

- **Data Cleaning**: Handling missing values, correcting inconsistencies.
- **Feature Encoding**: Converting categorical variables (sex, smoking history) into numerical values.
- **Normalization/Scaling**: Bringing numerical features (height, weight, BMI) to the same scale.
- **Feature Selection**: Removing irrelevant features and keeping only those useful for prediction (e.g., heart_disease, smoking, exercise).
- **Data Splitting**: Splitting dataset into training (e.g., 70%) and testing (30%) sets.

3. Model Selection and Development

- Explain **why you chose the models** (Logistic Regression, Random Forest, Gradient Boosting, XGBoost, ANN).
- Justify each model:
  - Logistic Regression → baseline, interpretable.
  - Random Forest & Gradient Boosting → handle non-linear patterns well.
  - XGBoost → optimized boosting, strong performance.
  - ANN → captures complex non-linear relationships.
- Describe **training process**:
  - Splitting data into train/test.
  - Hyperparameter tuning (grid search, random search).
  - Model evaluation (cross-validation).

4. Visualizations and Insights

- **Confusion Matrix** → for each model.
- **ROC-AUC Curves** → compare model discriminative power.
- **Feature Importance Plot** → which features influence predictions most (e.g., BMI, smoking, diabetes).
- **Accuracy/F1 comparison bar chart** → which model performed best.

Insights → Random Forest achieved the highest accuracy (93%), making it suitable for screening.

- ANN achieved the highest ROC-AUC (0.95), showing strong capability in identifying at-risk patients.
- Lifestyle features such as smoking, exercise, and BMI had strong influence on predictions.

5. Recommendations

- **Practical Use**:
  - Random Forest or XGBoost can be deployed in hospital systems for initial screening.
  - ANN can be used for more advanced risk stratification in preventive healthcare apps.
- **Policy Recommendations**:
  - Encourage lifestyle modifications for high-risk groups identified by the models.
  - Regular checkups for patients flagged as medium-to-high risk.
- **Future Work**:
  - Collect larger datasets for improved model training.
  - Explore deep learning models with more layers.
  - Improve interpretability of ANN using SHAP or LIME.

Conclusion

The project on **Cardiovascular Disease Prediction** successfully demonstrates how patient health, lifestyle, and clinical details can be transformed into actionable insights using data-driven approaches. Starting from raw patient data, we applied systematic pre-processing steps, selected relevant attributes, and built predictive models to identify individuals at risk of developing cardiovascular disease.

The study confirms that patient details such as **age, BMI, smoking history, diabetes, exercise habits, and overall health indicators** play a critical role in predicting cardiovascular outcomes. By cleaning and preparing the dataset, we ensured that the models were trained on reliable information, leading to meaningful and trustworthy results.

Through this project, it becomes evident that predictive modeling can serve as a valuable tool for **early detection and preventive healthcare**. The insights gained allow healthcare providers to not only identify high-risk individuals but also recommend lifestyle changes and regular monitoring for medium- and low-risk groups.

Ultimately, this work demonstrates the **practical impact of machine learning in healthcare**: enabling proactive decision-making, supporting doctors with data-backed predictions, and contributing to the reduction of cardiovascular disease burden through timely interventions.

References

- Alghamdi, M., Al-Mallah, M., Keteyian, S., Brawner, C., Ehrman, J., & Sakr, S. (2017). *Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: The Henry Ford Exercise Testing (FIT) Project.* PLOS ONE, 12(7), e0179805. https://doi.org/10.1371/journal.pone.0179805

- Dey, S., & Rautaray, S. S. (2020). *Survey on heart disease prediction models using machine learning.* International Journal of Computer Applications, 181(21), 26–33. https://doi.org/10.5120/ijca2020920138

- Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., ... & Dean, J. (2019). *A guide to deep learning in healthcare.* Nature Medicine, 25(1), 24–29. https://doi.org/10.1038/s41591-018-0316-z

- Kagiyama, N., Shrestha, S., Farjo, P. D., Sengupta, P. P., & Narula, J. (2019). *Artificial intelligence in cardiovascular imaging: a practical review.* JACC: Cardiovascular Imaging, 12(8 Pt 1), 1499–1511.
- https://doi.org/10.1016/j.jcmg.2019.03.008

- Weng, S. F., Reps, J., Kai, J., Garibaldi, J. M., & Qureshi, N. (2017). Can machine-learning improve cardiovascular risk prediction using routine clinical data? PLOS ONE, 12(4), e0174944. https://doi.org/10.1371/journal.pone.0174944