Sprints

Comprehensive Machine Learning Full Pipeline on Heart Disease UCI Dataset

1. General Description of the Task:

This project aims to **analyze, predict, and visualize heart disease risks** using machine learning. The workflow involves **data preprocessing, feature selection, dimensionality reduction (PCA), model training, evaluation, and deployment**. Classification models like **Logistic Regression, Decision Trees, Random Forest, and SVM** will be used, alongside **K-Means and Hierarchical Clustering** for unsupervised learning. Additionally, a **Streamlit UI** will be built for user interaction, deployed via **Ngrok**, and the project will be hosted on **GitHub**.

1.1 Objectives:

- **Perform Data Preprocessing & Cleaning** (handle missing values, encoding, scaling).
- **Apply Dimensionality Reduction (PCA)** to retain essential features.
- **Implement Feature Selection** using statistical methods and ML-based techniques.
- **Train Supervised Learning Models** (Logistic Regression, Decision Trees, Random Forest, SVM) for classification.
- **Apply Unsupervised Learning** (K-Means, Hierarchical Clustering) for pattern discovery.
- **Optimize Models** using Hyperparameter Tuning (GridSearchCV, RandomizedSearchCV).
- **Deploy a Streamlit UI** for real-time user interaction. **[Bonus]**
- **Host the application using Ngrok [Bonus]** and **upload the project to GitHub** for accessibility.

1.2 Tools to be Used:

- **Programming Languages:** Python
- **Libraries:** Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, TensorFlow/Keras (optional)
- **Dimensionality Reduction & Feature Selection:** PCA, RFE, Chi-Square Test
- **Supervised Models:** Logistic Regression, Decision Trees, Random Forest, SVM
- **Unsupervised Models:** K-Means, Hierarchical Clustering
- **Model Optimization:** GridSearchCV, RandomizedSearchCV
- **Deployment Tools:** Streamlit **[Bonus]**, Ngrok **[Bonus]**, GitHub

---

2. Requirements & Steps

2.1 Data Preprocessing & Cleaning

Steps:

1. Load the **Heart Disease UCI dataset** into a Pandas DataFrame.
2. Handle **missing values** (imputation or removal).
3. Perform **data encoding** (one-hot encoding for categorical variables).
4. Standardize numerical features using **MinMaxScaler** or **StandardScaler**.
5. Conduct **Exploratory Data Analysis (EDA)** with **histograms, correlation heatmaps, and boxplots**.

Deliverable:

✔️ Cleaned dataset ready for modeling

2.2 Dimensionality Reduction - PCA (Principal Component Analysis)

Steps:

1. Apply **PCA** to reduce feature dimensionality while maintaining variance.
2. Determine the **optimal number of principal components** using the explained variance ratio.
3. Visualize PCA results using a **scatter plot** **and cumulative variance plot**.

not applicable IRL

Deliverable:

✔️ PCA-transformed dataset

✔️ Graph showing variance retained per component

2.3 Feature Selection

Steps:

1. Use **Feature Importance (Random Forest / XGBoost feature importance scores)** to rank variables.
2. Apply **Recursive Feature Elimination (RFE)** to select the best predictors.
3. Use **Chi-Square Test** to check feature significance. not applicable in our case
4. Select only the most relevant features for modeling.

Deliverable:
✔️ Reduced dataset with selected key features
✔️ Feature importance ranking visualization

2.4 Supervised Learning - Classification Models

Steps:

1. Split the dataset into **training (80%) and testing (20%)** sets.
2. Train the following models:
    - Logistic Regression
    - Decision Tree
    - Random Forest
    - Support Vector Machine (SVM)
3. Evaluate models using:
    - Accuracy, Precision, Recall, F1-score
    - ROC Curve & AUC Score

Deliverable:
✔️ Trained models with performance metrics

2.5 Unsupervised Learning - Clustering

Steps:

1. Apply **K-Means Clustering** (elbow method to determine K).
2. Perform **Hierarchical Clustering** (dendrogram analysis).
3. Compare clusters with actual disease labels.

Deliverable:
✔️ Clustering models with visualized results

2.6 Hyperparameter Tuning

Steps:

1. Use **GridSearchCV & RandomizedSearchCV** to optimize model hyperparameters.
2. Compare optimized models with baseline performance.

Deliverable:
✔️ Best performing model with optimized hyperparameters

2.7 Model Export & Deployment

Steps:

1. **Save the trained model** using **joblib or pickle (.pkl format)**.
2. Ensure reproducibility by saving **model pipeline** (preprocessing + model).

Deliverable:
✔️ Model exported as `.pkl` file

2.8 Streamlit Web UI Development [Bonus]

Steps:

1. Create a **Streamlit UI** to allow users to input health data.
2. Provide **real-time prediction output** based on user inputs.
3. Add **data visualization** for users to explore heart disease trends.

Deliverable:
✔️ Functional **Streamlit UI** for user interaction

2.9 Deployment using Ngrok [Bonus]

Steps:

Didn't use Ngrok, but used streamlit community app instead

1. Deploy the **Streamlit app** locally.
2. Use **Ngrok** to create a public access link.
3. Share the **Ngrok link** for live access to the web application.

Deliverable:
✔️ Publicly accessible Streamlit app via **Ngrok link**

2.10 Upload the Project to GitHub

Steps:

1. Create a **GitHub repository** for the project.
2. Push the following files:
    - Data preprocessing scripts
    - Trained models in `.pkl` format
    - Notebook files for each step
    - Streamlit UI source code
    - README file with instructions
3. Add **requirements.txt** for easy environment setup.
4. Include **deployment steps for Ngrok** in documentation.

Deliverable:
✔️ GitHub repository with **all project files and documentation**

---

3. Final Deliverables

✔️ **Cleaned dataset with selected features**
✔️ **Dimensionality reduction (PCA) results**
✔️ **Trained supervised and unsupervised models**
✔️ **Performance evaluation metrics**
✔️ **Hyperparameter optimized model**
✔️ **Saved model in .pkl format**
✔️ **GitHub repository with all source code**
✔️ **Streamlit UI for real-time predictions [Bonus]**
✔️ **Ngrok link to access the live app [Bonus]**

---

4. File Structure

```
Heart_Disease_Project/
|── data/
|   ├── heart_disease.csv
|── notebooks/
|   ├── 01_data_preprocessing.ipynb
|   ├── 02_pca_analysis.ipynb
|   ├── 03_feature_selection.ipynb
|   ├── 04_supervised_learning.ipynb
|   ├── 05_unsupervised_learning.ipynb
|   ├── 06_hyperparameter_tuning.ipynb
|── models/
|   ├── final_model.pkl
|── ui/
|   ├── app.py (Streamlit UI)
|── deployment/
|   ├── ngrok_setup.txt
|── results/
|   ├── evaluation_metrics.txt
|── README.md
|── requirements.txt
|── .gitignore
```

Sprints

---

5. Dataset Link

📌 [Heart Disease UCI Dataset]