# ECG Classification

## Table of Contents

- [Introduction](#_Introduction)  
- [Features](#_Features)  
- [Installation](#_Installation)  
- [Usage](#_Usage)  
- [Technologies Used](#_Technologies_Used)  
- [Dataset](#_Dataset)  
- [Results](#_Results)  
- [Contributing](#_Contributing)  
- [License](#_License)

## Introduction

ECG signals are critical for diagnosing various heart conditions. This project builds a classification model to analyze ECG data and predict cardiac anomalies.   
The solution is designed to be robust and interpretable, targeting both medical researchers and AI enthusiasts.

## Features

- Preprocessing and normalization of ECG data.  
- Exploratory Data Analysis (EDA) with Matplotlib and Seaborn.  
- Deep learning-based classification using TensorFlow and Keras.  
- Model evaluation using scikit-learn metrics.  
- Visualizations of results and model performance.

## Installation

Follow these steps to set up the project locally:  
  
1. Clone the repository:  
 git clone

<https://github.com/Ayush-yadav11/ECG-Classification.git>  
  
  
2. Create a virtual environment:  
 python -m venv env  
 source env/bin/activate # On Windows: env\Scripts\activate  
  
3. Install the required dependencies:  
 pip install -r requirements.txt

## Usage

1. Add your ECG dataset in the `data/` directory.

2. Run the preprocessing script:  
 python preprocess.py

3. Train the model:  
 python train.py  
  
4. Evaluate the model:  
 python evaluate.py  
  
5. View results and visualizations:  
 - Accuracy and loss curves.  
 - Confusion matrix for classification performance.

## Technologies Used

- TensorFlow and Keras: For building and training the deep learning model.  
- scikit-learn: For metrics and evaluation.  
- NumPy and Pandas: For data manipulation.  
- Matplotlib and Seaborn: For data visualization.

## Dataset

The project requires ECG signal data in a tabular or time-series format. The data set used is :-  
  
- MIT-BIH Arrhythmia Dataset

## Results

The model achieves high accuracy in classifying ECG signals. Below are some highlights:  
  
- Training accuracy: 70.43%  
- Validation accuracy: 65.63%  
- Confusion matrix and ROC curves are available for analysis in the `results/` folder.

## Contributing

Contributions are welcome! Please follow these steps:  
  
1. Fork the repository.  
2. Create a feature branch:  
  
 git checkout -b feature-name  
  
3. Commit your changes and push to the branch:  
  
 git commit -m "Added feature-name"  
 git push origin feature-name

4. Open a pull request.

## License

This project is licensed under the MIT License. See the `LICENSE` file for more details.