Deployment Report: Cardiovascular Disease Risk Prediction Streamlit Application

Overview

This report outlines the deployment of a web application developed using Streamlit for predicting cardiovascular disease risk based on user-provided health information. The application leverages a trained XGBoost model, along with preprocessing and feature selection pipelines, to make real-time predictions. Additionally, it includes a Dashboard

Key Components

1. Model and Pipeline Loading

1. **Model**: The XGBoost model, saved as a JSON file, is loaded into the application using pickle.
2. **Preprocessor**: A preprocessing pipeline is used to transform user inputs into a suitable format for the model.
3. **Feature Selector**: A feature selection mechanism is implemented to ensure that only the most relevant features are used for prediction.

2. User Interface

1. The application is designed with an intuitive user interface using Streamlit, allowing users to interact with the model easily.
2. **Tabs**: The interface is divided into two main tabs:
    1. **Predict Risk**: Users enter their health information to receive a risk prediction.
    2. **Data Visualizations**: Users can explore the dataset through various visual analysis options.

3. Input Fields for Prediction

1. Users can input their health details, including:
    1. **Demographics**: Gender, height, weight, age.
    2. **Health Metrics**: Blood pressure, cholesterol levels, glucose levels.
    3. **Lifestyle Factors**: Smoking status, alcohol consumption, physical activity.
2. The application calculates derived metrics such as BMI and pulse pressure in real-time.

4. Prediction Logic

1. Upon clicking the "Predict Risk" button, the application:
    1. Constructs a Data Frame from user inputs.
    2. Apply the preprocessing steps to format the data.
    3. Use the feature selector to retain significant features.
    4. Generates a prediction and associated probability of CVD risk.
2. The results are displayed to the user, indicating whether they are at "High Risk" or "Low Risk" of CVD.

5. Data Visualization Features

1. The application provides tools for data exploration:
    1. **Univariate Analysis**: Users can visualize distributions of numerical and categorical variables.
    2. **Bivariate Analysis**: Users can explore relationships between two variables using scatter plots, bar plots, and box plots.
    3. **Multivariate Analysis**: Users can generate correlation heatmaps to investigate relationships among multiple numerical variables.

6. Caching for Performance

1. To enhance performance, Streamlit's caching features (@st. cache_resource and @st. cache_data) are utilized. This ensures that the model and data are loaded efficiently without reloading on every interaction.

7. Error Handling

1. The application includes basic error handling to inform users if any issues occur during the prediction process, ensuring a smoother user experience.

Deployment Steps

1. **Environment Setup**: The application was developed in a Python environment with the necessary libraries (Streamlit, pandas, numpy, pickle, matplotlib, seaborn, scikit-learn).
2. **Model Training**: The XGBoost model was trained separately and saved for deployment.
3. **Web Application Development**: The Streamlit application was coded, incorporating the model, preprocessing, and visualization components.
4. **Testing**: The application was tested locally to ensure functionality and accuracy of predictions.
5. **Deployment**: The application is deployed on Streamlit Server