Heart Disease Prediction Using UCI Dataset

Team 16	Name	ID	Level	Department
	هاجر علي محمد محمود	2023170684	2	general
	ياسمين محمد شاكر الحسيني	2023170697	2	general
	وسام خيري عبد الحميد احمد	2023170691	2	general
	لوجينا ايمن محمد مصطفى اسماعيل النويهي	2023170457	2	general
	شيماء شعبان أحمد مرسي نصر	2023170306	2	general

Documentation: Heart Disease Prediction App (Streamlit)

Overview

This Streamlit app is a web-based interface that allows users to:

- Upload a heart disease dataset (CSV format)
- Preprocess the dataset
- Perform feature selection
- Train a machine learning model
- Evaluate the model's performance
- Predict heart disease using new input values

Imports

```
import streamlit as st
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.feature_selection import SelectKBest, f_classif
```

- Standard data processing and visualization libraries are imported.
- Various scikit-learn tools are used for model training, evaluation, and preprocessing.

1. Setup and Title

```
st.set_page_config(page_title="Heart Disease Classifier", layout="centered")
st.title("\ud83d\udc93 Heart Disease Prediction App")
```

- Configures the Streamlit page layout and sets the app title.

2. Upload Dataset

```
file = st.file_uploader("Upload your heart.csv file", type=["csv"])
```

- Allows users to upload a CSV file.
- Displays the first few rows and optional summary statistics.

3. Preprocessing

- **Missing Values Handling:**
    - Categorical columns: filled with mode
    - Numeric columns: filled with median
- **Column Fixes:**
    - Converts invalid entries in chol column
    - Drops duplicates
- **Heatmap:**
    - Optional correlation heatmap for data exploration

4. Feature Selection

```
selector = SelectKBest(score_func=f_classif, k=10)
```

- SelectKBest is used with f_classif to select the top 10 features that best correlate with the target variable.
- The user selects the target column (typically binary indicating heart disease presence).

5. Train-Test Split & Scaling

```
X_train, X_test, y_train, y_test = train_test_split(...)
scaler = StandardScaler()
```

- Data is split (80% train, 20% test).
- StandardScaler is used to normalize features.

6. Model Training & Evaluation

- Models Supported:
    - Logistic Regression
    - Support Vector Machine (Linear Kernel)
    - Decision Tree (Entropy)
    - K-Nearest Neighbors (k=7)
- Evaluation:
    - Accuracy
    - Confusion matrix
    - Optional full classification report

Heart Disease Prediction Using UCI Dataset

7. Prediction Section

```
input_vals = {}
input_df = pd.DataFrame([input_vals])
scaled_input = scaler.transform(input_df)
pred = model.predict(scaled_input)
```

- Users manually enter input values for the selected features.
- The inputs are scaled and passed into the trained model for prediction.
- The result (heart disease or not) is displayed with appropriate styling (success/error).

Session State

```
st.session_state.model = model
st.session_state.scaler = scaler
st.session_state.selected_features = selected_features
```

- Stores the trained model and preprocessing objects for use in prediction without re-training.