COMP 4334 - Lab 7 - Heart Disease Prediction

Michael Ghattas - May 18, 2025

File Structure

- **heartTraining.csv** - Training data for the heart disease prediction model.
- **heartTesting.csv** - Testing data for evaluating model performance.
- **Lab7.py** - Final script for data preprocessing, feature engineering, and logistic regression model training.

Overview

This lab implements a machine learning pipeline using **PySpark** to predict heart disease based on patient data. The script handles data preprocessing, feature extraction, model training, and prediction evaluation, all within the Spark environment to leverage distributed processing.

Key Features

1. Data Loading

- Reads **heartTraining.csv** and **heartTesting.csv** directly from Databricks FileStore.
- Automatically detects headers and infers schema for efficient processing.

2. Data Cleaning

- Removes leading and trailing spaces from column names to prevent indexing errors.
- Explicitly casts **"chol"** (cholesterol) to **DoubleType** to ensure compatibility with the vector assembler.

3. Feature Engineering

- **Age Binning**:
  - Transforms continuous age values into meaningful age categories:
    - **Below 40**
    - **40-49**
    - **50-59**
    - **60-69**
    - **70 and above**
- **Label and Feature Encoding**:
  - Converts **"sex"**, **"pred"**, and **"AgeCategory"** into numerical indices for model compatibility.

4. Model Pipeline

- Constructs a complete machine learning pipeline including:
  - **StringIndexer** for categorical encoding.
  - **VectorAssembler** for feature vector construction.
  - **LogisticRegression** for binary classification.

5. Model Training and Prediction

- Trains a **Logistic Regression** model on the training data.
- Evaluates the model on the test data, providing **probability** and **prediction** for each test instance.

6. Results Display

- Prints the top 100 predictions, including:
  - **id** - Unique patient ID
  - **probability** - Probability of each class
  - **prediction** - Predicted class (0 = No Heart Disease, 1 = Heart Disease)

Usage

Running the Script in Databricks

1. Upload **heartTraining.csv** and **heartTesting.csv** to **FileStore**:
   - Training File: `dbfs:/FileStore/shared_uploads/michael.ghattas@du.edu/heartTraining.csv`
   - Testing File: `dbfs:/FileStore/shared_uploads/michael.ghattas@du.edu/heartTesting.csv`
2. Run the **Lab7.py** script in a Databricks notebook to train and evaluate the model.

Sample Output

```
+---+----------------------------------------+----------+
|id |probability                             |prediction|
+---+----------------------------------------+----------+
|0  |[0.539360261019413,0.46063973898058697] |0.0       |
|1  |[0.6821448414294567,0.31785515857054325]|0.0       |
|2  |[0.7281946430459562,0.27180535695404384]|0.0       |
|3  |[0.9110378181272513,0.08896218187274874]|0.0       |
|4  |[0.6087956558943428,0.3912043441056572] |0.0       |
|5  |[0.34723544326991723,0.6527645567300828]|1.0       |
|...|...                                     |...       |
|99 |[0.38013603417836567,0.6198639658216343]|1.0       |
+---+----------------------------------------+----------+
```

Notes

- The script automatically handles whitespace issues in column names to prevent indexing errors.
- **“chol”** is cast to **DoubleType** to avoid **IllegalArgumentException** during feature assembly.
- The age binning function is optimized for efficient category conversion.

Next Steps

- Evaluate model accuracy using precision, recall, and F1-score.
- Experiment with hyperparameter tuning for improved performance.
- Integrate cross-validation for more robust model evaluation.
