System Requirements Specification Index

For

Python pytorch sleep analysis

1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Sleep Quality Prediction Use Case

Sleep quality is one of the most important indicators of overall health and wellbeing. By analyzing lifestyle factors such as sleep duration, stress level, daily activity, and disorders, we can build a predictive model to classify a person's sleep quality. This system helps individuals understand their sleep health and take proactive steps toward improvement.
The use case leverages machine learning with PyTorch to process a dataset, train a model, and make predictions for new users. Below are function-wise instructions to implement the pipeline.

Function-Wise Instructions

- **Define a function load_dataset(path="Sleep_health_and_lifestyle_dataset.csv").**
  - The function should:
    o Use pandas to load the dataset from the given path.
    o Return the DataFrame so it can be used for preprocessing.
- **Define a function preprocess(df, fit=True).**
  - The function should:
    o If fit=True:
      - Drop rows with missing values in *Quality of Sleep*.
      - Bin *Quality of Sleep* into 3 categories (0=Poor, 1=Average, 2=Good).
      - Encode categorical columns (*Gender*, *Sleep Disorder*) using LabelEncoder, ensuring "None" is included.
      - Scale numeric features with StandardScaler.
      - Split into train/test sets using StratifiedShuffleSplit.
      - Return X_train, y_train, X_test, y_test, and input_dim.
        o If fit=False:
      - Apply stored encoders and scaler.
      - Return the transformed features as a torch tensor.
- **Define a class SleepDataset(Dataset).**
  - The class should:
    o Store features and labels as tensors.
    o Implement __len__ to return number of samples.
    o Implement __getitem__ to return (X[idx], y[idx]).
- **Define a function build_model(input_dim, num_classes=3).**
  - The function should:
    o Build a neural network using nn.Sequential:
      - Linear(input_dim → 16) + ReLU + Dropout(0.3)
      - Linear(16 → 8) + ReLU
      - Linear(8 → num_classes)
        o Return the model.
- **Define a function train_and_evaluate(model, train_loader, test_loader, device="cpu", epochs=15, lr=0.01).**
  - The function should:
    o Use CrossEntropyLoss and Adam optimizer.
    o Train over multiple epochs: forward pass, compute loss, backward pass, optimizer step.
    o Track and print training accuracy and validation accuracy per epoch.
- **Define a function evaluate(model, loader, device="cpu").**
  - The function should:
    o Set model to evaluation mode.

o Perform forward passes without gradients.
o Compute and return accuracy.

- **Define a function save_model(model, path="sleep_model.pth").**
    - The function should:
      o Save the model's state_dict to the given path using torch.save.
- **Define a function load_model(path, input_dim, num_classes=3).**
    - The function should:
      o Rebuild the model with build_model().
      o Load weights from the saved file using torch.load.
      o Return the model in eval mode.
- **Define a function predict_new_user(model, user_dict).**
    - The function should:
      o Convert user_dict into a DataFrame with 1 row.
      o Call preprocess(fit=False) to transform features.
      o Forward pass through the model.
      o Apply softmax to get probabilities and argmax for predicted class.
      o Map predicted index to label: {0: Poor (Abnormal), 1: Average, 2: Good}.
      o Return (predicted_label, probabilities_array).
- **Define a function main().**
    - The function should:
      o Load the dataset.
      o Preprocess into train/test splits.
      o Create Dataset/DataLoader objects.
      o Build, train, and evaluate the model.
      o Save and reload the model.
      o Read new user data from "new_user.txt"
      o Predict and print results.

<u>Execution Steps to Follow:</u>

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal
3. This editor Auto Saves the code
4. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page)
5. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
6. To setup environment:

You can run the application without importing any packages

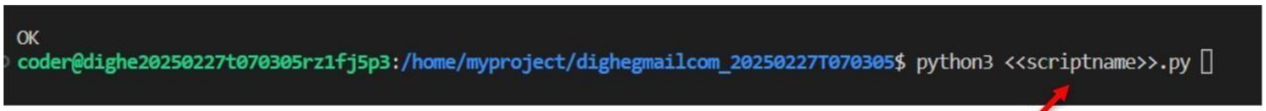
7. To launch application:

python3 main.py

To run Test cases:

python3 -m unittest

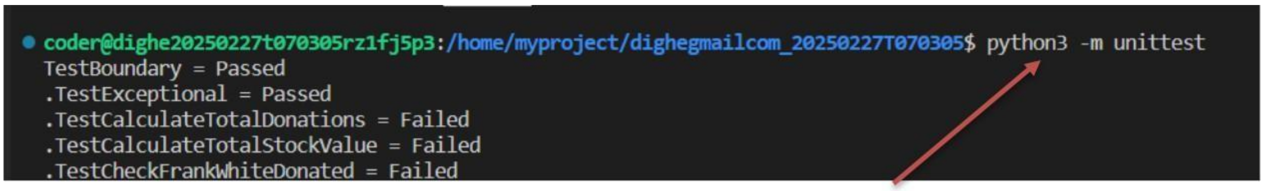
<u>**Screen shot to run the program**</u>



```
OK
coder@dighe20250227t070305rz1fj5p3:/home/myproject/dighegmailcom_20250227T070305$ python3 <<scriptname>>.py
```

To run the application

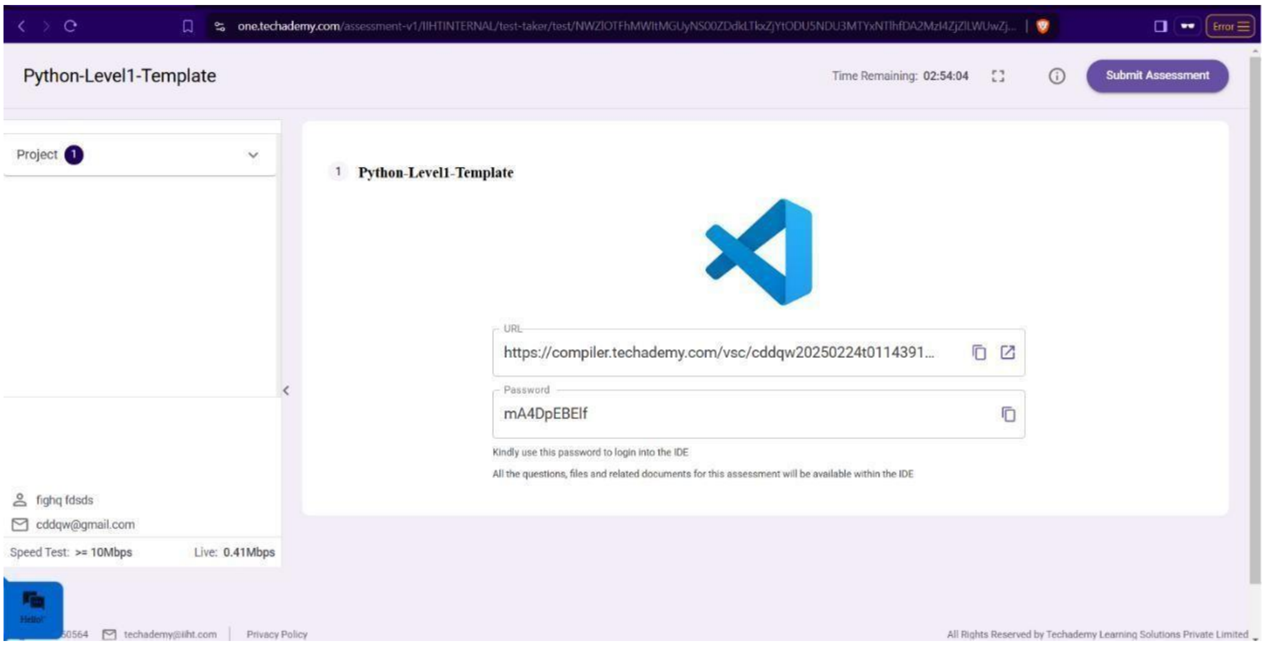
python3 main.py



```
● coder@dighe20250227t070305rz1fj5p3:/home/myproject/dighegmailcom_20250227T070305$ python3 -m unittest
  TestBoundary = Passed
  .TestExceptional = Passed
  .TestCalculateTotalDonations = Failed
  .TestCalculateTotalStockValue = Failed
  .TestCheckFrankWhiteDonated = Failed
```

To run the testcase

- **python3 -m unittest**



one.techademy.com/assessment-v1/IIHTINTERNAL/test-taker/test/NWZlOTFhMWItMGUyNS00ZDdkLTkzZjYtODU5NDU3MTYxNTlhfDA2Mzl4ZjZlLWUwZj...

Python-Level1-Template

Time Remaining: 02:54:04

1 Python-Level1-Template

URL
https://compiler.techademy.com/vsc/cddqw20250224t0114391...

Password
mA4DpEBElf

Kindly use this password to login into the IDE

All the questions, files and related documents for this assessment will be available within the IDE

fighq fdsds
cddqw@gmail.com
Speed Test: >= 10Mbps Live: 0.41Mbps

50564 techademy@iiht.com | Privacy Policy

All Rights Reserved by Techademy Learning Solutions Private Limited

8. Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click

on “Submit Assessment” after you are done with code.

-----x-----