Ira A. Fulton Schools of Engineering
Arizona State University

CSE 572: Data Mining

Project 2: Machine Model Training

Purpose

In this project, you will use a training dataset to train and test a machine model. The purpose is to distinguish between meal and no meal time series data.

Objectives

Learners will be able to:

- Develop code to train a machine model.
- Assess the accuracy of a machine model.

Technology Requirements

- Python 3.10.9
- scikit-learn==1.1.1
- pandas==1.4.3
- numpy==1.23.1
- scipy==1.8.1

Project Description

In this project, you will train a machine model to assess whether a person has eaten a meal or not eaten a meal. A training data set is provided.

Please watch the **three introductory videos on Project 2** before beginning. These are located in your course *Welcome and Start Here* section.

- Project 2: Machine Model Training Introductory Video 1
- Project 2: Machine Model Training Introductory Video 2
- Project 2: Machine Model Training Introductory Video 3

Note: Project details in the Overview Document were recently updated since the recording of the videos, so some directions or items may not match. Please follow the Overview Document directions to complete your project correctly.

Directions

Meal data can be extracted as follows:

- From the InsulinData.csv file, search the column Y for a non NAN non zero value. This time indicates the start of meal consumption time tm. Meal data comprises a 2hr 30 min stretch of CGM data that starts from tm-30min and extends to tm+2hrs.
- No meal data comprises 2 hrs of raw data that does not have meal intake.

Extraction: Meal data

Start of a meal can be obtained from InsulinData.csv. Search column Y for a non NAN non zero value. This time indicates the start of a meal. There can be three conditions:

1. There is no meal from time tm to time tm+2hrs. Then use this stretch as meal data
2. There is a meal at some time tp in between tp>tm and tp< tm+2hrs. Ignore the meal data at time tm and consider the meal at time tp instead.
3. There is a meal at time tm+2hrs, then consider the stretch from tm+1hr 30min to tm+4hrs as meal data.

Extraction: No Meal data

Start of no meal is at time tm+2hrs where tm is the start of some meal. We need to obtain a 2 hr stretch of no meal time. So you need to find all 2 hr stretches in a day that have no meal and do not fall within 2 hrs of the start of a meal.

Handling missing data:

You have to carefully handle missing data. This is an important data mining step that is required for many applications. Here there are several approaches:

1. Ignore the meal or no meal data stretch if the number of missing data points in that stretch is greater than a certain threshold.
2. Use linear interpolation (not a good idea for meal data but maybe for no meal data)

3. Use polynomial regression to fill up missing data (untested in this domain).

Choose wisely.

Feature Extraction and Selection:

You have to carefully select features from the meal time series that are discriminatory between meal and no meal classes.

Test Data:

The test data will be a matrix of size N×24, where N is the total number of tests and 24 is the size of the CGM time series. N will have some distribution of meal and no meal data.

Note here that for meal data you are asked to obtain a 2 hr 30 min time series data, while for no meal you are taking 2 hr. However, a machine will not take data with different lengths. Hence, in the feature extraction step, you have to ensure that features extracted from both meal and no meal data have the same length.

Output format:

You have to output an N×1 vector of 1s and 0s, where if a row is determined to be meal data, then the corresponding entry will be 1, and if determined to be no meal, the corresponding entry will be 0.

- This vector should be saved in a "**Result.csv**" file.

Given:

- Meal Data and No Meal Data of subjects 1 and 2
- Ground truth labels of Meal and No Meal for subjects 1 and 2

Using Python, train a machine model to recognize whether a sample in the training data set represents a person who has eaten (Meal), or not eaten (No Meal). The training data set contains ground truth labels of Meal and No Meal for 5 subjects.

You will need to perform the following tasks:

1. Extract features from Meal and No Meal training data set.
2. Make sure that the features are discriminatory.
3. Train a machine to recognize Meal or No Meal data.

4. Use k fold cross validation on the training data to evaluate your recognition system.
5. Write a function that takes a single test sample as input, and outputs 1 if it predicts the test sample as meal or 0 if it predicts test sample as No meal.

Submission Directions for Project Deliverables

Deliverables:

- Two python files: 1) train.py and 2) test.py
- The train.py reads CGMData.csv, CGM_patient2.csv and InsulinData.csv, Insulin_patient2.csv, extracts meal and no-meal data, extracts features, trains your machine to recognize meal and no-meal classes, and stores the machine in a pickle file (Python API pickle).
- The test.py reads test.csv which has the N x 24 matrix and outputs a Result.csv file which has N x 1 vector of 1s and 0s, where 1 denotes meal, 0 denotes no meal.
- Assume that CGMData.csv, CGM_patient2.csv and InsulinData.csv, Insulin_patient2.csv files are all in your compilation and execution folder. Avoid using static paths.

Submission Guidelines:

- Please submit a zipped file containing train.py and test.py as "**yourfirstname_lastname_Project2.zip**".
    - Do not create an additional folder; just zip the files directly.
- The submission space is located at the bottom of Week 4 under "Week 4: Graded Coursework" as "**Project Assignment: Project 2: Machine Model Training Submission**".

Evaluation

The autograder in Coursera will evaluate your code as well as the accuracy of your results based on a set of Meal and No Meal data that is not included in the training set.

- 50 points for developing a code in Python that takes the given dataset, extracts Meal and No Meal data, and trains a machine model
- 20 points for developing a code in Python that implements a function to take a test input and run the trained machine to provide the class label as output

- 30 points will be evaluated on the accuracy, F1 score, Precision, and Recall results obtained by your machine.

Note: The autograder performs a few mathematical calculations on the mean and standard deviation, and your F1 and accuracy score should be higher than these calculations, respectively. Below are the mean and standard deviation for F1 and accuracy:

- Mean of F1 : 0.3831
- Standard deviation of F1 : 0.11
- Mean of Accuracy : 0.5098
- Standard deviation of Accuracy : 0.07

Common Errors:

- **ModuleNotFoundError**: You are trying to access or use modules that are not found in the grader. Use the modules mentioned in the Technology Requirements section.