README - Prediction Engine

Notebook:	4th Year Project		
Created:	2019-03-27 11:30 AM	**Updated:**	2019-03-28 9:23 AM
Author:	Mohamed Hozayen		
URL:	https://www.overleaf.com/3586987798rdzmfnnwkycz		

README - Prediction Engine

Diabetes Analytics and Recommendation Engine
IoT Lab, Department of Systems and Computer Engineering
Carleton University

This file documents all files within the Prediction Engine folder. Prediction Engine (PE) is a sub-project of the Diabetes Analytics and Recommendation Engine (DARE) project.

The PE predicts future glucose level based on previous glucose level signal.

- Short-term predictions are achieved by using a slide window and regression models; weighted linear regression, support vector regression (SVR), random forest regression (RFR).
    - Short term prediction refer to predictions in the next half an hour to the next two hours approximately.
- Long-term predictions are based on the time-series analysis concept along with neural network models; recurrent neural network (RNN), feed forward neural network (FFNN), long-shot term memory (LSTM)
    - Implementation is not fully done.
    - Neural Network learn the behavior of a signal for a profile to approximatrly predict glucose level over long periods (eg. next day to a week).
- Note: the current PE is only using the history of glucose level of a patient to predict future glucose level due to data constraint and limitations. Optimally, the following inputs shall be utilized over long periods: insulin intake, exercise, diet, sleep, stress, glucose level history, heartbeat, etc. Most of these inputs are collected by smart watches.

Authors:
Lama Yassin Kassab, LamaYassinKassab@cmail.carleton.ca
Leen Yassin Kassab, LeenYassinKassab@cmail.carleton.ca
Pooria Shafia, PooriaShafia@cmail.carleton.ca
Joe Samuel, me@joefs.com

Mohamed Hozayen, mohamedhozayen@cmail.carleton.ca

Supervisor: Dr. Zied Bouida - Dr. Mohamed Ibnkahla

Sections

1. Documents
2. Literature
3. Long-Term Predictions
4. Short-Term Predicitons
5. Development and Run-time

1- **Documents**
Contains related major documents to the project

- E-Health Group - Project Proposal "E-Health Group - Project Proposal.pdf"
  - Initial proposed ideas, however, adjustments are addressed in the progress report and more in the final report
- Progress Report "Progress Report - Dec 2018_Revised.pdf"
  - The description of the PE in the progress report is not clear, please use the final report and this PE folder
- Oral Presentation "Presentation_Jan.pptx"
  - on Predictions Engine only, full presentation is in PE folder
- Poster 2019 - "Poster2019.pptx"
  - slide 1 and slide 2 were presented in the poster fair 2019
  - slide 3 is a compressed version of slide 1 and slide 2 presented in Data Day 6.0 conference; 2nd place winners
- Machine learning Summaries - "ML_Summary_1.pdf" "ML_Summary_2.pdf"
  - This is a summary for an online course taken on udemy.com called "MachineLearnign A-Z: Hands-On Python & R In Data Science"
  - Source code, data, examples are provided in superdatascience.com/pages/machine-learning

2- **Literature**
This folder contains papers that were used to do the literature review on the subject.

3- **Long-Term Predictions**
This folders contain one uncompleted python script.
"tt_NN.py" is being used to learn the detailed implementation of time series analysis using neural network.
"sts" frames a time series as a supervised learning dataset

Tutorial link: https://www.tensorflow.org/tutorials/keras/basic_regression

4- **Short-Term Predictions**
Short-term predictions folder contains majority of source code for the current running PE. It also contains the dataset PE used and output results.

- All python files (file extension is .py) documentations are within the files themselves. Any text editor could open them for reading.
    - How to run the python files is at the end of this README

- Data

"data-points_cases.csv" contains the synthesize data; it has six healthy patient and six diabetic with their prospective personal information and glucose level over 24 hours in five minute interval (data is points are every five minutes). This file was delivered by the Data Generation team; Leen, Lama, and Pooria.

- Graphs Diabetes "graphs_d"

These are the output graphs after running SVR and LR on all diabetic patients with different window sizes and prediction intervals.

- Graphs Healthy "graphs_h"

These are the output graphs after running SVR and LR on all healthy patients with different window sizes and prediction intervals.

- Slide Window Simple Linear Regression "SW SLR"

These graphs were one of the first graphs that were generated in the PE. They are prototype models as a start point.
These are the output graphs after running SLR on a patient with different window sizes and a fixed prediction interval (predict the next five minutes which means the following data point).
EDS is error difference square, this takes the square of the difference between each predicted and actual value. The EDS plot is useful for watching a detailed performance rather than a single value such as accuracy.
EDST is error difference square total; the summation of EDS. The lower the EDST the better the model.

- Slide Window Simple Linear Regression Weights "SW SLR Weights"

Building on SW SLR, the SLR model with weights is applied. Basically, the closer the points to current time, the more weights the have in the regression model to predict.

- "presentation_grapgh_results.pdf"

This is a summary report on the results of "SW SLR" and "SW SLR Weights" presented in the weekly meetings.
However, the last table in that file was added later after running more regression models such as SVR and RFR. It contains predictions and EDS graphs.
ws -> Window Size
ws = 6, means predict based on the past half an hour (6 x 5 = 30 minutes)
L_Pred ->Predictive Interval
L_Pred = 4, means predict glucose level after 20 minutes from now (4 x 5 = 20 minutes)

- "presentation_result_summary_table.pdf"

This file summarizes the last table in "presentation_grapgh_results.pdf" in numbers instead of graphs. This was made to be on the presentation slides in a readable format.

5- Development and Run-time

Note: pip program eases the process of installation from the command line terminal

Language: Python 3.6.5 www.python.org (or any version above 3.6.0) (MUST HAVE)

IDE: Spyder www.spyder-ide.org (no need to download from this website if you are using anaconda)

- Any other IDE is fine. (Jupyter is also recommended)
- Spyder is good for simplicity and variable explorer

Platfrom: Anaconda www.anaconda.com

1. Open anaconda then Spyder from anaconda (or spyder directly)
2. Browse a working directory where the python files are. In the case how the files are provided, the directory would be ".....\Prediction Engine\Short-Term Predictions"

    - This could be done by file explorer or the folder icon on top right of the spyder IDE

3. Open the file to be run

    - Highlight the code section to be run
    - press "control + enter" to run the highlighted text
    - Or run all the file by pressing the green run button above (F5 shortcut)

Long-Term Predictions
Long-term predictions utilizes neural network thus further addition installation are required.

Don't install Tensorflow and Keras from these links if Anaconda is being used - see below instructions for Anaconda

- Tensorflow github.com/tensorflow/tensorflow
- Keras https://keras.io/

If Anaconda is being used:

- Deep Learning environment need to be installed in Anaconda to run neural network.
- Open Anaconda Terminal and type "conda install -c conda-forge keras"
- On Anaconda, Keras packages integrate tensorflow.
- Close Anaconda and reopen it
- Change Application on from "base (root)" to "deep_learning"