SMT. PARVATIBAI CHOWGULE
OF ARTS AND SCIENCE
GOA-403601, GOA(INDIA)
Computer Science and Engineering

Clustered Machine Learning for Predicting Diabetes

Submitted By:
Alexander Roque Rodrigues
SU170331
TYBSC

Submitted To:
Mrs. Ashweta Fondekar
Asst. Professor
Dept. of CSE

June-February
2019-2020
Graduation Dissertation

1 Acknowledgements

Contents

List of Figures

List of Tables

Abstract

Your abstract.

2 Introduction

In recent days, there has been a sharp increase in the cases of diabetes mellitus. Diabetes mellitus is on the rise amongst many people and the rate of contracting this lifestyle disease could be reduced significantly if proper measures and precautions were to be instilled amongst people the number of people can be reduced.

Machine learning is a growing field in computer science. With the development and introduction of many algorithms the prediction and accuracy of the predictions itself has improved substantially. Machine learning and healthcare systems are also becoming increasingly popular in the healthcare sector.

The project encompasses the qualities of Remote Patient Monitoring (RPM) and Clinical Decision Support (CDS). RPM provides medical facilities that have the ability to transmit patient data to healthcare professionals who might very well be halfway around the world. RPM can monitor blood glucose levels and blood pressure. It is particularly helpful for patients with chronic conditions such as type 2 diabetes, hypertension, or cardiac disease. Data collected and transmitted via PRM can be used by a healthcare professional or a healthcare team to detect medical events such as stroke or heart attack that require immediate and aggressive medical intervention. Data collected may be used as part of a research project or health study. RPM is a life-saving system for patients in remote areas who cannot access face-to-face health care. CDS analyzes data from clinical and administrative systems. The aim is to assist healthcare providers in making informed clinical decisions. Data available can provide information to medical professions who are preparing diagnoses or predicting medical conditions like drug interactions and reactions. CDS tools filter information to assist healthcare professionals in caring for individual clients.

The objective of this project is to create a system that is able to use the machine learning algorithms and predict the outcome of the parameters entered into the algo-

rithm and help the patient draw a conclusion whether or not he/she has the same traits exibhited by simillar patients that have diabetes. Also the system should have a UI that is capable of displaying the data of the patients to the doctor and to the patients themselves for further interpretation.

3 Software Requirements

The main function of this project is to enable the doctors to advise their patients with the help of the prediction software. The system should be accessible to the patient as wel

l as the doctor, therefore it should have a web interface for the two parties to interact with.

3.1 Functionalities for Doctors

As an owner of a doctors account a doctor should be able to:

- Add new patients.
- Add new observations for the machine learning algorithm to predict.
- Analyse the patients previous records.
- Leave notes for patients to act on.

3.2 Functionalities for Patients

As the patient, one should be able to:

- Should be able to see the predicted risk of developing diabetes.
- Should be able to view historic data.
- Should be able to view notes or suggestions left by doctor.

3.3 Functionalities for Master Nodes

As the patient, one should be able to:

- Should be able to see the predicted risk of developing diabetes.
- Should be able to view historic data.
- Should be able to view notes or suggestions left by doctor.

3.4 Functionalities for Slave Nodes

As the patient, one should be able to:

- Should be able to see the predicted risk of developing diabetes.
- Should be able to view historic data.
- Should be able to view notes or suggestions left by doctor.

4 Hardware Requirements

4.1 Raspberry Pi

According to raspberrypi.org, the Raspberry Pi 3 Model B is the earliest model of the third-generation Raspberry Pi. It replaced the Raspberry Pi 2 Model B in February 2016. Some of the key features of this single board computer or SBC are:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU.
- 1GB RAM.
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board.
- 100 Base Ethernet.
- 40-pin extended GPIO.
- 4 USB 2 ports.
- Full size HDMI.
- Micro SD port for loading the operating system and storing data.

In this project I will be using 3 Raspberry Pi's to implement a cluster and deploy the machine learning algorithm on.

4.2 Switch

A network switch was used in the project to make up for the lack of ethernet ports available on the router. The dumb network switch was able to connect upto 3 Raspberry Pi and one cable back to the network router itself to connect the switch to the main network.

4.3 Router

A router is required to assign internet protocol addresses to the nodes using dynamic host control protocol (DHCP). The router also is responsible for displaying the nodes

connected to the network thereby displaying hostnames and making it more easier to capture the addresses of each node.

4.4 Master Node

The master node is assigned the task of managing the the slave nodes connected to the network. The master node and the slave nodes should be connected to the same database to run and execute queries.

5 Technology Stack

5.1 Python

5.1.1 Pandas

Pandas, is a library that is required for loading the comma separated value file into python. Pandas is a package for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

5.1.2 Sklearn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

5.1.3 Numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

5.1.4 Itertools

The module standardizes a core set of fast, memory efficient tools that are useful by themselves or in combination. Together, they form an “iterator algebra” making it possible to construct specialized tools succinctly and efficiently in pure Python.

5.2 MYSQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter and YouTube.

5.3 Apache Web Server

The Apache HTTP Server, colloquially called Apache, is free and open-source cross-platform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

The vast majority of Apache HTTP Server instances run on a Linux distribution, but current versions also run on Microsoft Windows and a wide variety of Unix-like systems. Past versions also ran on OpenVMS, NetWare, OS/2 and other operating systems, including ports to mainframes.

5.4 PHP

PHP is a general-purpose programming language originally designed for web development. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

PHP code may be executed with a command line interface (CLI), embedded into HTML code, or used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. The web server outputs the results of the interpreted and executed PHP code, which may be any type of data, such as generated HTML code or binary image data. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications[8] and robotic drone control.

5.5 AJAX

Ajax is a set of web development techniques using many web technologies on the client side to create asynchronous web applications. With Ajax, web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page. By decoupling the data interchange layer from the presentation layer, Ajax allows web pages and, by extension, web applications, to change content dynamically without the need to reload the entire page.[3] In practice, modern implementations commonly utilize JSON instead of XML.

Ajax is not a single technology, but rather a group of technologies. HTML and CSS can be used in combination to mark up and style information. The webpage can then be modified by JavaScript to dynamically display—and allow the user to interact with—the new information. The built-in XMLHttpRequest object, or since 2017 the new "fetch()" function within JavaScript, is commonly used to execute Ajax on webpages allowing websites to load content onto the screen without refreshing the page. Ajax is not a new technology, or different language, just existing technologies used in new ways.

5.6 GitHub

GitHub is a global company that provides hosting for software development version control using Git. It offers all of the distributed version control and source code man-

agement (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

6 The Multi Layered Perceptron

6.1 Introduction

The multilayer perceptron (MLP) is the one of the most commonly used artificial neural networks. The name is a slight misnomer; a multilayer perceptron is not a single perceptron with multiple layers, but rather multiple layers of artificial neurons that can be perceptrons. The layers of the MLP form a directed, acyclic graph. Generally, each layer is fully connected to the subsequent layer; the output of each artificial neuron in a layer is an input to every artificial neuron in the next layer towards the output. MLPs have three or more layers of artificial neurons.

7 Writing Code

```
import itertools
import mysql.connector
from colorama import init
from colorama import Fore, Back, Style
from mysql.connector import Error
import masterenvironment as en
from os import system
from os import chdir
init()

def divide_chunks(l, n):
    for i in range(0, len(l), n):
        yield l[i:i + n]


def readNodes():
    global lineList
    with open(en.fileName) as f:
        lineList = f.readlines()
    lineList = [line.rstrip('\n') for line in open(en.fileName)]


def db_connect():
    curr_node = 0
    curr_chunk = 0
    print("Connecting to DB ", end="")
    try:
        mydb = mysql.connector.connect(
            host=en.host, user=en.user, passwd=en.passwd, database=en.
    database)
        mycursor = mydb.cursor()
        print(Fore.GREEN+"[SUCCESS]"+Style.RESET_ALL)
        pending = []
```

```
33          query = "select * from {} where PredictedOutcome {}".format("
    diagnosis", "IS NULL")
34          mycursor.execute(query)
35          myresult = mycursor.fetchall()
36          for x in myresult:
37              pending.append(x[0])
38
39          x = list(divide_chunks(pending, en.chunkSize))
40          executeStatus = 0
41          nodeCount = len(lineList)
42          nodeCount = nodeCount - 1
43          chunkCount = len(x)
44          while(executeStatus != 1):
45              temp = x[curr_chunk]
46              temp = str(temp)
47              temp = temp[1:-1]
48              temp = temp.replace(" ", "")
49              query = "ssh pi@{} python3 hive-ml/slave.py {}".format(
    lineList[curr_node], temp)
50              query = str(query)
51              system(query)
52              curr_chunk = curr_chunk + 1
53              if(curr_node == nodeCount):
54                  curr_node = 0
55              else:
56                  curr_node = curr_node+1
57              if(curr_chunk == chunkCount):
58                  executeStatus = 1
59
60      except Error as e:
61          print(Fore.RED+"[FAILED]"+Style.RESET_ALL)
62          print(e)
63          exit(0)
64
65  if __name__ == "__main__":
```

```
66      readNodes()
67      db_connect()
```

8 Tables

The table 1 is an example of referenced LaTeXelements.

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

Table 1: Table to test captions and labels

9 Conclusion