Predict Onset of Diabetes
Big Data System Design - Final Report

John Carneiro - johnmcarneiro@gmail.com
Janakiram Sundaraneedi - janakiram_sundaraneedi@student.uml.edu
Jan Zaloudek - honzicekz@gmail.com

Outline

- Introduction
- Motivations
- Related Work
- Proposed Approach
- Evaluation
- Timeline
- Experimental Results and Discussions
- Conclusions and Future Work
- References

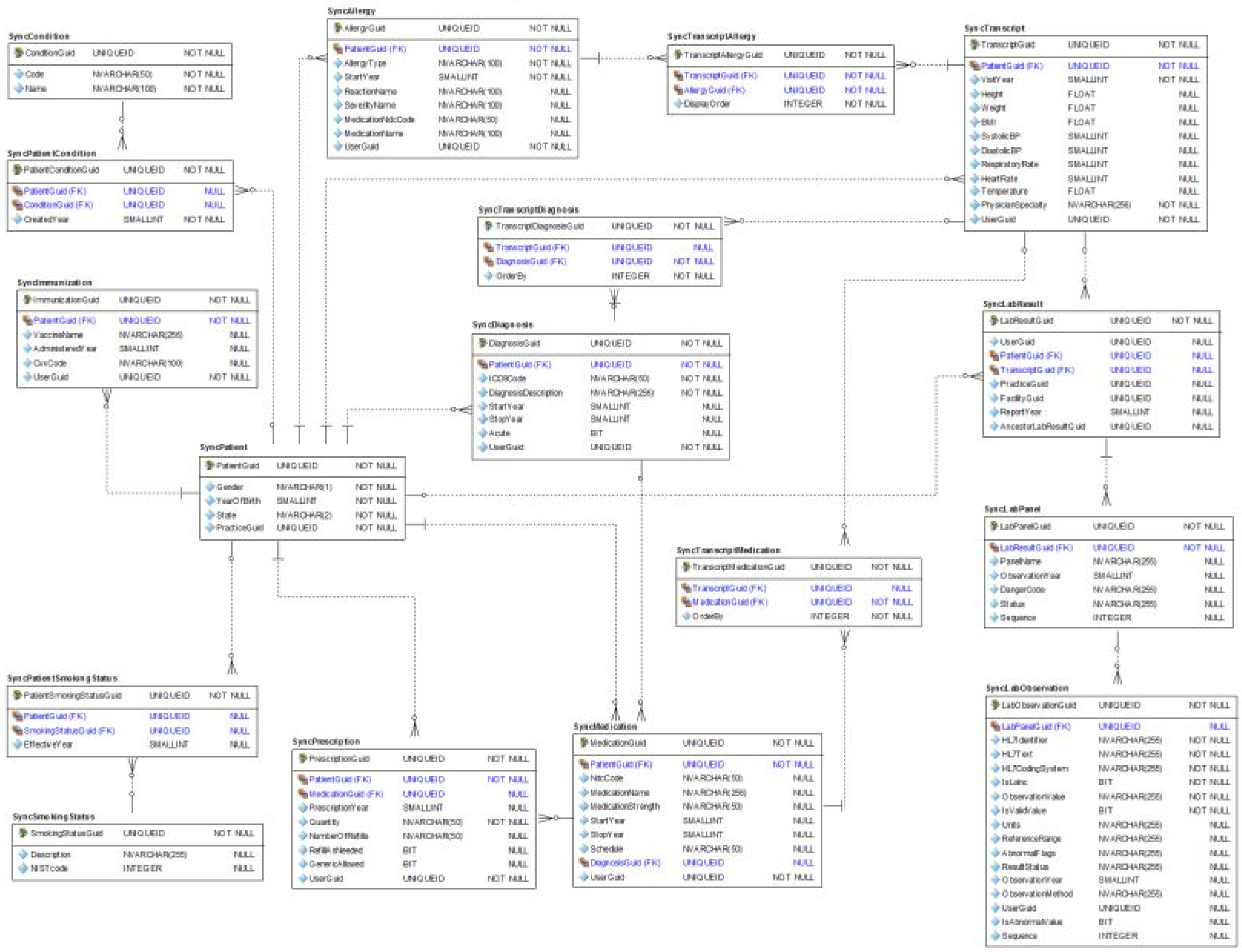
Introduction

The research/development question is to predict based on diagnostic measurements whether a patient has type 2 diabetes using the Kaggle based Practice Fusion Diabetes Classification dataset.

The CSV text dataset contains records of 10,000 de-identified medical records. Data includes: Gender, BMI, Height, Weight, SystolicBP, DiastolicBP, RespiratoryRate, HeartRate, Temperature, Meds, Diagnoses, Immunizations, Allergies

The ultimate goal of your proposed system is to perform this prediction in an automated way using big data system design (MongoDB, Machine Learning, Processing techniques such as Keras, TensorFlow, and Random Forest).

Practice Fusion De-Identified Data Set



Motivations

- This is interesting question for us due to a lot of our family members have diabetes
- It is important to develop a system to be able to predict the occurrence of diabetes if the patient’s lifestyle and eating habits are kept the same and possibly prevent the occurrence of diabetes if preventative steps are taken by patient
- 415 million people have diabetes worldwide
- 8.3% of the world adult population (equal parts men/women) have diabetes and is rising
- Diabetes doubles a person's risk of early death.
- 5 million deaths occur worldwide each year because of diabetes.
- The global economic cost of diabetes is estimated to be US $612 billion.

Related Work

The following are sources of existing research/development work has tried to answer the same or a similar question:

- https://www.kaggle.com/c/pf2012-diabetes (data set)
- Using the ADAP learning algorithm to forecast the onset of diabetes mellitus
- Developing risk prediction models for type 2 diabetes: a systematic review of methodology and reporting

A highly reliable and accurate automated prediction algorithm is still not available.
Also, it would be self-evident that an accurate automated prediction algorithm should work regardless of gender and ethnic background.

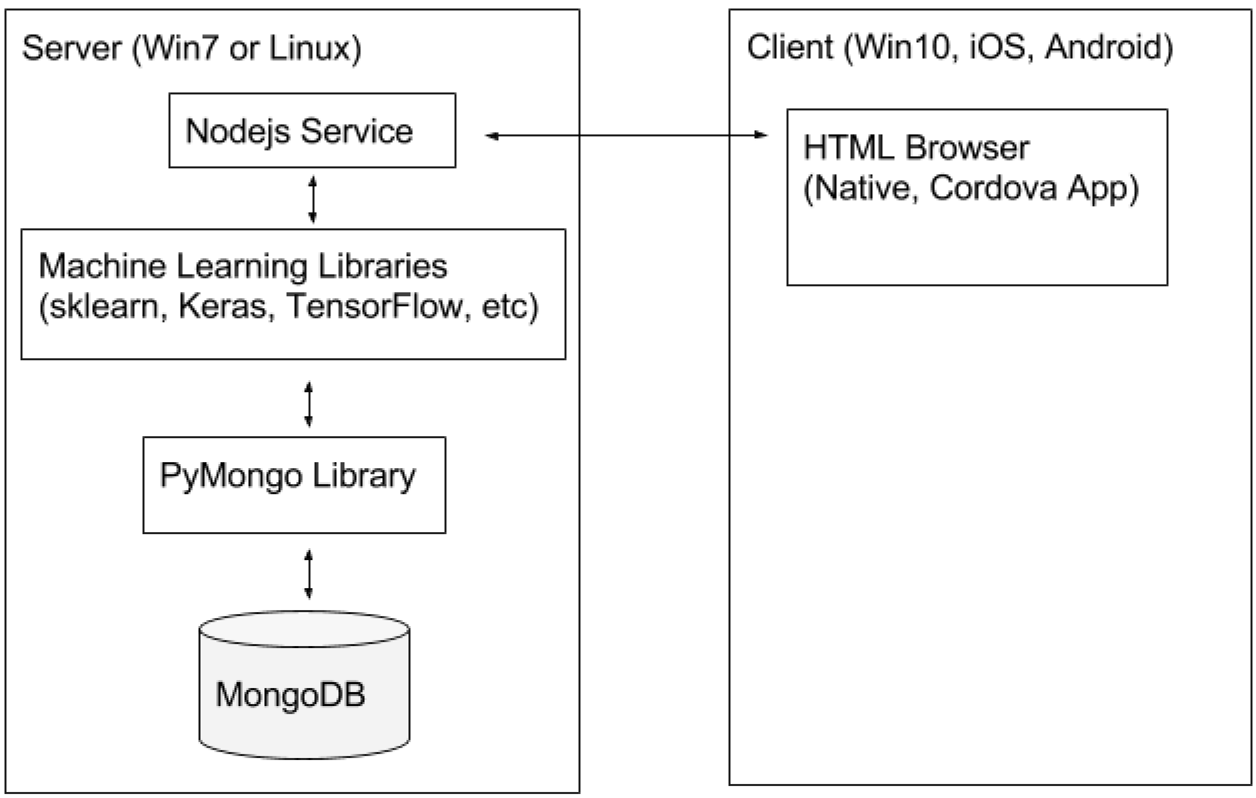
Proposed Approach

Plan for working out the solutions to the question:

- Research dataset
    - Survey knowledge experts (UML doctor, nursing staff, and people with the diabetes)
    - Improve via algorithms accuracy and precision level on missing values
    - Research ML algorithms, current solutions on kaggle.com
    - Use rapidminer.com, and Python ML libraries
    - Performance test ML algorithms
- Main features in your proposed system:
    - MongoDB (i.e. NoSQL DB), Python, Node.js)
    - Client Application HTML5, JS, Cordova mobile development front end (desktop, iOS, Android)
    - Data entry and/or upload data to add to global learning dataset
    - Matplotlib, plotly, Google Chart, D3, etc.
    - Based on input data, one or more users can be told whether or not he/she has or is susceptible to get type 2 diabetes (data can be optionally added to learning dataset to improve algorithm accuracy)
- Implementation approach:
    - MongoDB
    - Python, RapidMiner, RStudio
    - HTML5, JS, Node.js
    - Evaluated and used: Keras, TensorFlow, Scikit Learn
    - Evaluated Machine Learning Algorithm(s) (Classification, Regression, Clustering, Dimensionality reduction, Model selection, Preprocessing, NN) - those found exhibiting best performance
        - Keras based neural network
        - Random Forest Classifier
        - Gaussian Naive Bayes

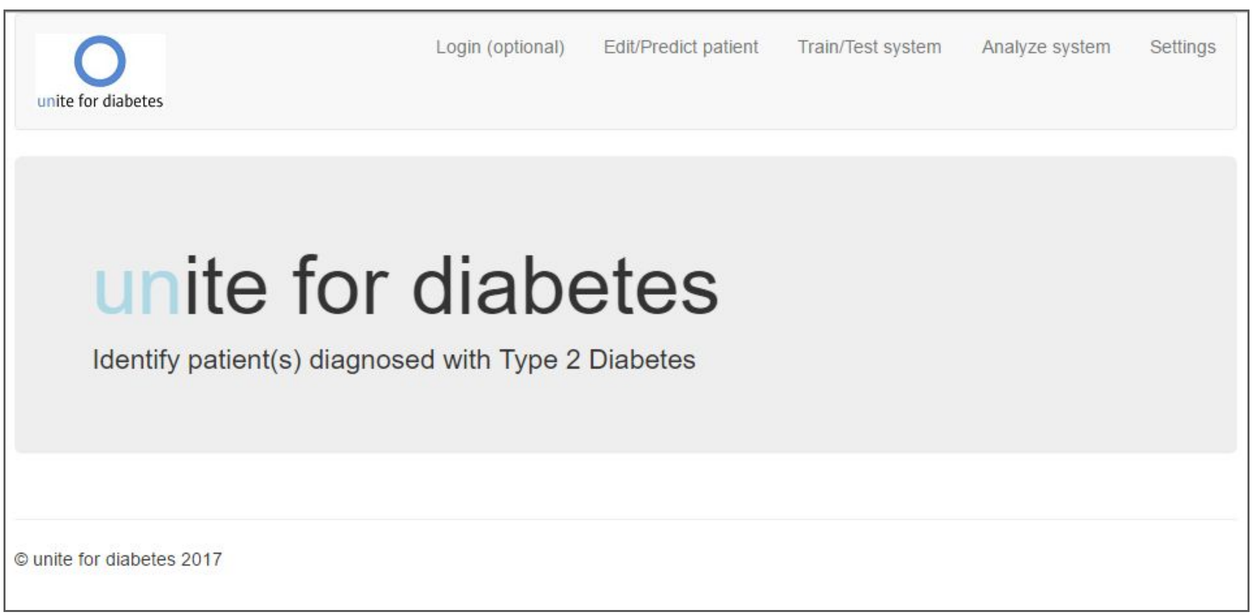
        - K-Nearest Neighbor Classifier
        - Decision Tree Classifier
    - After evaluating many ML algorithms, the following was settled on:
        - Random Forest Model - using the “scikit learn” python library
        - Keras for use as a high-level neural networks API, with TensorFlow - low-level neural networks API
        - Features used to : Age, Gender, BMI, Height, Weight, Systolic Blood Pressure, Diastolic Blood Pressure, Respiratory Rate, Heart Rate, Temperature, Medications, Diagnoses, and Lab Tests

System Block Diagram

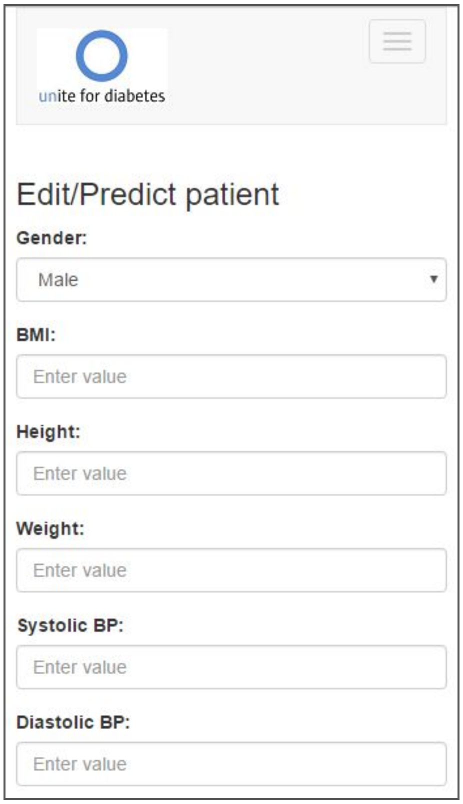


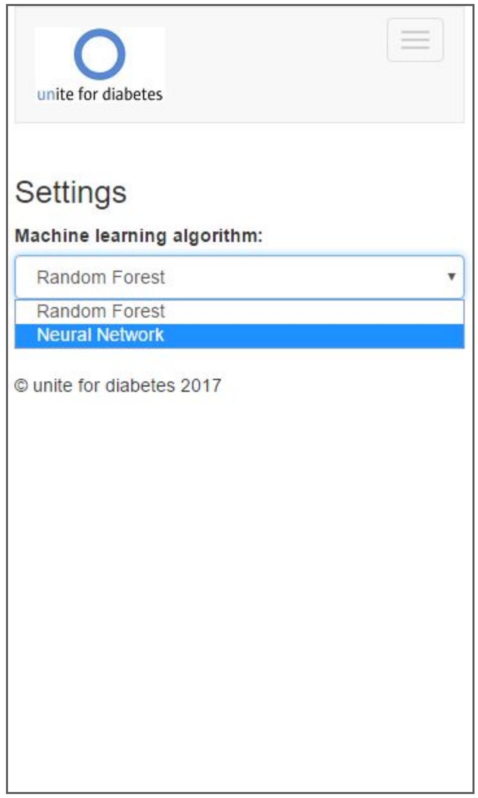
Mobile Application

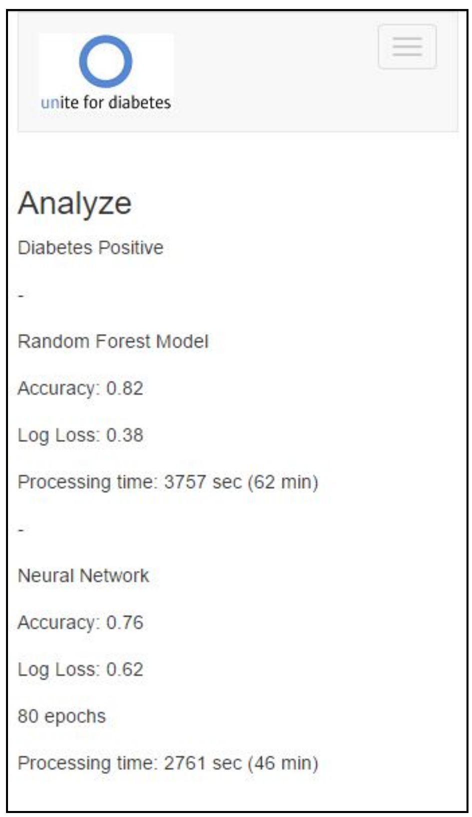
Tablet/Desktop Application



Application Screens







Evaluation

We evaluated our solution as follows (to demonstrate that our solution/answer is good/reasonable):

- Test ML algorithms locally with the stock dataset
- Preprocess dataset
- Find the algorithm that performs the best in a reasonable amount of processing time at a small scale first. Used Anaconda and Jupyter notebook extensively.
- Setup a MongoDB environment to handle the increase in size
- Integrate MongoDB component
- Test the scaled up components at the backend server side
- Test the scaled up components at the client side
- For Random Forest and NN, train on the training dataset, then test the test dataset
- Compare all results with those available online in previous studies
- Publish results on sites like github and kaggle.com to verify and respond to feedback

Timeline

- Week 1 (ending 2017-02-02) - Proposal presentation
- Week 2 (ending 2017-02-09) - Pre data set analysis and Github setup
- Week 3 (ending 2017-02-16) - Subject matter expert surveys, topic research
- Week 4 (ending 2017-02-23) - ML algorithm analysis
- Week 5 (ending 2017-03-02) - ML algorithm testing
- Week 6 (ending 2017-03-09) - ML algorithm testing
- Week 7 (ending 2017-03-16) - MongoDB integration
- Week 8 (ending 2017-03-23) - MongoDB integration
- Week 9 (ending 2017-03-30) - Front end development
- Week 10 (ending 2017-04-06) - Front end development
- Week 11 (ending 2017-04-13) - System performance testing (train/test)
- Week 12 (ending 2017-04-20) - System performance testing (train/test)
- Week 13 (ending 2017-04-27) - Final presentation/report

Experimental Results and Discussions

Random Forest Model

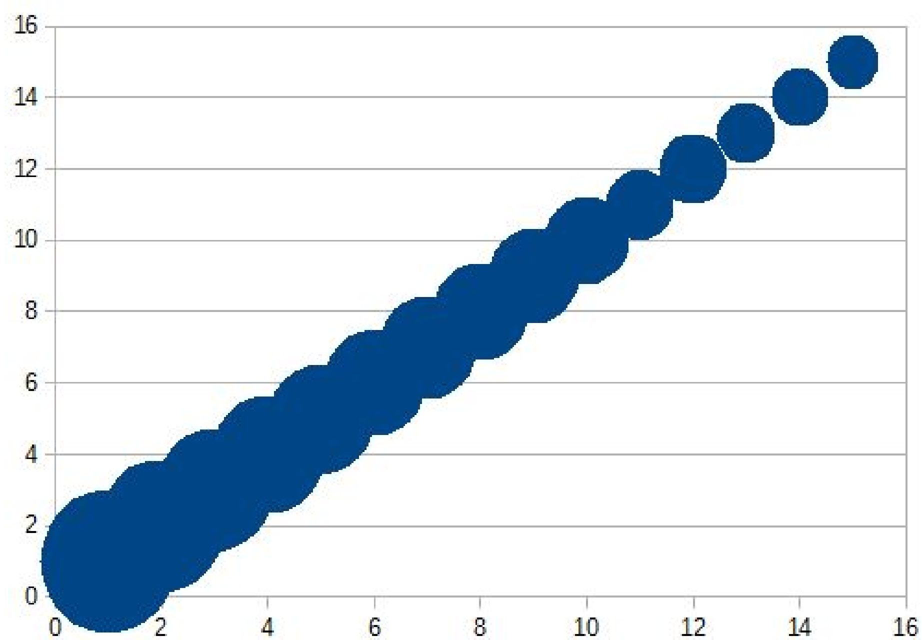
- Accuracy: 0.82
- Log Loss: 0.38
- Processing time: 3757 sec (62 min)

$$\log \text{loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i),$$

where N is the number of patients, $\log$ is the natural logarithm, $\hat{y}_i$ is the posterior probability that the i^{th} patient has diabetes, and y_i is the ground truth ($y_i = 1$ means the patient has diabetes, $y_i = 0$ means that he does not).

Dataset feature importance factors to determine diabetes positive (sorted high to low) after reading online medical documentation and preprocessing

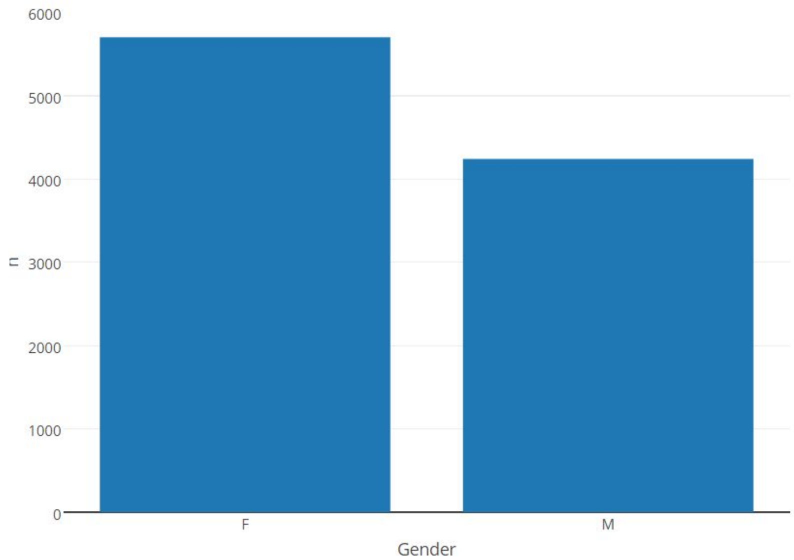
Patient-YOB, Patient-BMI, Patient-Weight, Patient-SystolicBP, Patient-DiastolicBP, Diag-Melenoma, Patient-Height, Patient-Temperature, Diag-Int-Pain, Diag-Resp, Diag-HeartValve, Med-Lisinopril, Diag-Osteoarthrosis, Diag-Dysphonia, Patient-Gender, Diag-HeartCongenital, Med-Simvastatin, Med-Lipitor, Med-Zocor, Diag-HeadInjury, Diag-Carcinoma, Diag-Carcinoma, Diag-Hyperhidrosis, Diag-Hyperhidrosis, Diag-Dysphagia, Med-Cozaar, Diag-Mastodynia, LabTestAuthorized, Diag-Colitis



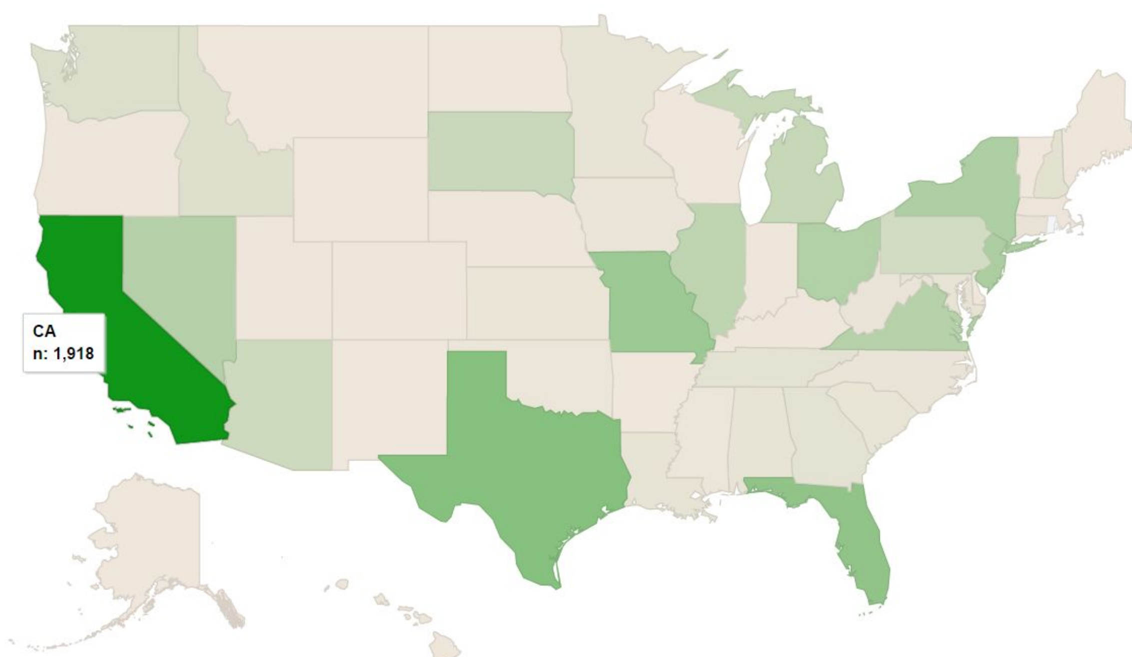
Decision Tree Classifier Investigation (using R)

http://www.rpubs.com/janakiram/add

Diabetes Positive by Gender



Diabetes Positive by State

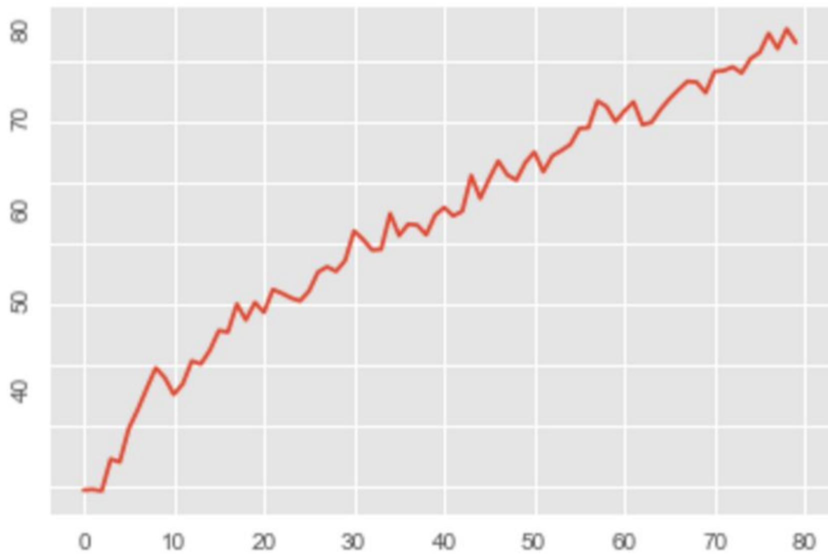


Random Forest Classifier Background (using Python)
https://en.wikipedia.org/wiki/Random_forest

Keras/TensorFlow Model

- Accuracy: 0.76
- Log Loss: 0.42
- 80 epochs

Accuracy improvement over epochs



Conclusions and Future Work

- Conclusion:
  - Our solutions where competitive against other teams who entered the Kaggle based Practice Fusion Diabetes Classification dataset contest (our best log loss 0.38, contest range 0.31 - 0.60)
  - Feature importance values corresponded with literature
- Future Work:
  - Overall, the model was a success, but there is room to improve the accuracy of prediction.
  - Use of GPUs to speed performance (local or remote third party (AWS))
  - Publish Diabetes predictor app to iOS, Android, Windows stores and public website to allow users determine likelihood of diabetes and/or to add their medical records to help train and improve accuracy of model
  - Partner with a government agency to help distribute app to promote healthier lifestyles

References

[1] Practice Fusion Diabetes Classification Challenge Background and Dataset
https://www.kaggle.com/c/pf2012-diabetes

[2] Deep Learning with Python by Jason Brownlee
https://machinelearningmastery.com/deep-learning-with-python

[3] Keras: Deep Learning library for Theano and TensorFlow
https://keras.io

[4] scikit-learn - Machine Learning in Python library
http://scikit-learn.org

[5] MongoDB - NoSQL database
https://www.mongodb.com

[6] PyMongo - MongoDB API
https://api.mongodb.com/python/current

[7] Cordova - HTML5 based Cross Platform Mobile App platform
https://cordova.apache.org

[8] American Diabetes Association. Economic costs of diabetes in the U.S. in 2012. Diabetes Care April 2013. 36(4):1033-1046
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3609540/

[9] Centers for Disease Control and Prevention: Diabetes successes and opportunities for population-based prevention and control. U.S. Department of Health and Human Services; 2011.
http://www.cdc.gov/chronicdisease/resources/publications/aag/ddt.htm
https://www.cdc.gov/diabetes/pdfs/newsroom/diabetes-materials-for-study-or-distribution-by-health-care-professionals.pdf

[10] Mokdad AH, Ford ES, Bowman BA, Dietz WH, Vinicor F, Bales VS, Marks JS. Prevalence of obesity, diabetes, and obersity-related health risk factors. JAMA. 2003; 289(1):76-79.
https://www.ncbi.nlm.nih.gov/pubmed/12503980

[11] National Diabetes Information Clearinghouse. Am I at risk for type 2 diabetes? Taking Steps to Lower Your Risk of Getting Diabetes. National Institute of Diabetes and Digestive and Kidney Diseases. NIH Publication No 12-4805. June 2012.
https://www.niddk.nih.gov/health-information/diabetes/overview/preventing-type-2-diabetes

[12] Haffner S, Lehto S, Ronnemaa T, Pyorala K. Mortality from Coronary Heart Disease in Subjects with Type 2 Diabetes and in Nondiabetes subjects with and without prior Myocardial Infarction. N Engl J Med. 1998; 339:229-234
http://www.nejm.org/doi/full/10.1056/NEJM199807233390404#t=article