Machine Learning Engineer Nanodegree 2019

Capstone Project

Early identification of Prediabetic using machine learning

Asis Pattnaik
July 2019

Table of Contents

1. Definition

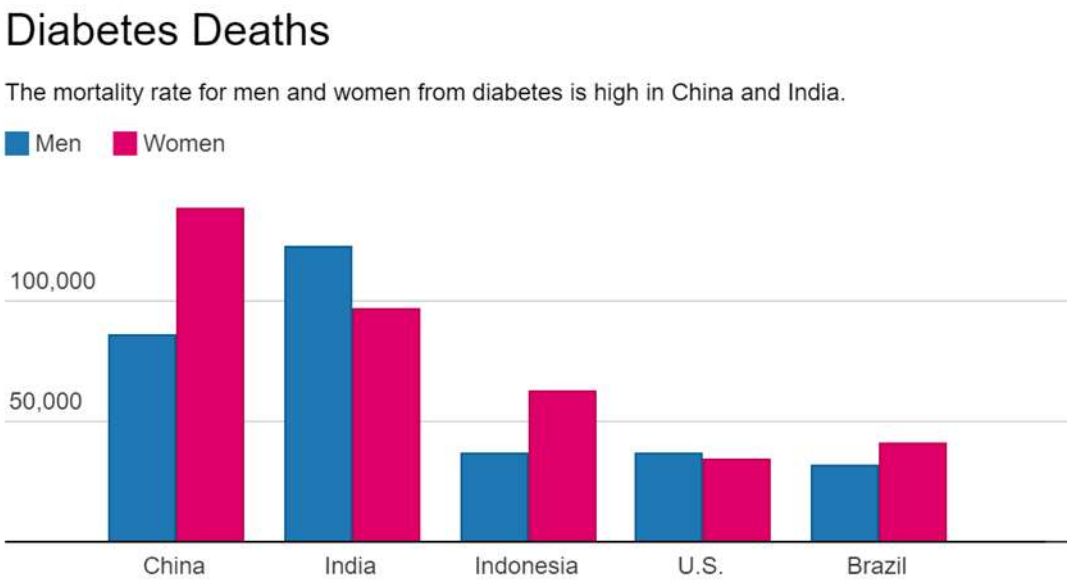
Diabetes is the cause of major health crisis across the world. An early diagnosis and appropriate lifestyle change is the best solution for the Type 2 diabetes which is the major cause behind this outbreak.

"Sugar is now more dangerous than gunpowder." - Yuval Noah Harari: Author Homo Deus

Udacity capstone project proposal Review link:

https://review.udacity.com/?utm_campaign=ret_000_auto_ndxxx_submission-reviewed&utm_source=blueshift&utm_medium=email&utm_content=reviewsapp-submission-reviewed&bsft_clkid=615120ee-6fa1-469d-b7d9-899d30a46c7f&bsft_uid=98400d5d-5a35-4074-98b6-9f8046188f26&bsft_mid=b8b4d28f-d1ea-4f0f-8e8d-5b9af008c06f&bsft_eid=6f154690-7543-4582-9be7-e397af208dbd&bsft_txnid=90d2741a-170a-4af7-b3b0-e4d7765aec09#!/reviews/1903521

1.1. Project Overview



Source: United Nations World Health Organization Get the data (2016)

1. An estimated 122,700 men aged 30 and above die from diabetes in India every year. Now India has 61.3 million diabetes patients. The numbers are estimated to rise to 101.2 million by 2030.
2. Out of all the cases of diabetes, Type 2 diabetes mellitus, constitutes more than 95%. The type 2 diabetes can go unnoticed for a long time, till it manifests in the form of metabolic syndromes like Retinopathy, Neuropathy, Nephropathy and Cardio vascular diseases. The prediabetic stages also carry high risk for cardiovascular diseases (CVDs).
3. Many studies have found that that this large-scale epidemic is faced not only in India, but across the world.
4. The type 2 Diabetes is a lifestyle disease. Due to ignorance, socio economic reasons, unhealthy food habits and sedentary lifestyle…Diabetes has become such a big problem. If it can be diagnosed at a prediabetes level, many young and old folks alike can get the treatment in time and will be benefited from it.

1.2. Problem Statement

The diabetes It constitutes impaired fasting glucose (IFG) and impaired glucose tolerance (IGT).

Due to the asymptomatic nature of disease and the low disease awareness among the population, diagnosis of the disease is delayed by several years. As a result, many subjects already have vascular complications at the time of diagnosis of diabetes.

1.3. The current challenges

India has nearly 3.0% of adults with prediabetes. The challenge is very high conversion rate from prediabetes to diabetes (18% per year).

Many studies show that Asian Indians have more body fat for any given body mass index (BMI) compared with other countries in Europe, America and Africa. Indians also have higher levels of central obesity (measured as waist circumference [WC], WHR, visceral fat, and posterior subcutaneous abdominal fat). The prevalence of overweight in adults in India increased from 9·0% in 1990 to 20·4% in 2016 in every state across the country.

For every 100 overweight adults aged 20 years or older in India, there were 38 adults with diabetes, compared with the global average of 19 adults in 2016.

1.4. Metrics

The correlation between the diabetes, BMI (Body Mass Index) and Age has been found to be high. As the age advances, a high BMI person has a higher chance of developing diabetes. It is also highly dependent on the food habits. As has been found out for the same BMI there is a significant difference in the percentage of people who are affected from different geographical locations. There are genetic, socio cultural, lifestyle differences and the pollution level has been found to be impacting the diabetes occurrences.

The aim is to develop a model that can predict the chances of a person being a diabetes / pre-diabetes.

The Fasting Blood Sugar level is one of the key indicators of the diabetes. The model predicts the chances of the person falling into a specific group “Diabetes”, ”Pre-diabetes”, ”Non-diabetes” .

2. Analysis

2.1. Data Exploration

The dataset that will be used is available from the “data.gov.in” on the Annual health survey 2014: Clinical, Anthropometric & Bio-chemical (CAB) Survey.

https://data.gov.in/resources/clinical-anthropometric-bio-chemical-cab-2014-survey-data-district- sitamarhi-bihar

The survey was carried out in the following nine states in India:

5. RAJASTHAN
6. UTTAR PRADESH
7. ODISHA
8. ASSAM
9. MADHYA PRADESH
10. CHHATTISGARH
11. UTTARAKHAND
12. BIHAR
13. JHARKHAND

The survey was carried out in 288 districts in these states. Out of the indices captured in this survey the following indices will be analysed for prediction of prediabetes.

1. Age
2. Weight in Kgs
3. Height in Cms
4. Fasting Blood sugar level
5. Blood Pressure
6. Haemoglobin level
7. Iodine content
8. Pulse Rate
9. Rural-Urban

10. Sex
11. State

Classification of the populace based on the risk that they will develop diabetes will help in providing the right Medicare in right time.

Since the data collected are categorical i.e the Age, BMI Index, Haemoglobin level, Blood pressure, iodine level, Marital status, these information need to be One hot encoded before applying the

classification techniques. We also have to identify the specificity and sensitivity factors to measure the accuracy.

After performing the data cleansing of the data that are not available and are important for this analysis the final dataset size is = 871582

- The number of participants that are found to be normal (not diabetic) = 603545
- The number of participants who were found to be pre-diabetic or diabetic = 268037

The participants will be classified into three classes i.e. who are diabetic, Prediabetic and non- diabetic.

The present dataset is not balanced as the number of non-diabetic instances are three time that of the other class.

To ensure that the three classes of participants have equal representation, we prepared the stratified data sample taking 25000 samples from each category. Thus, there are total 75000 entries in the dataset. Each class is equally represented.

The Training and Validation data will be distributed as below.

- The training set is 60% of the dataset
- The Validation set is 20% of the dataset
- The Testing set is 20% of the dataset

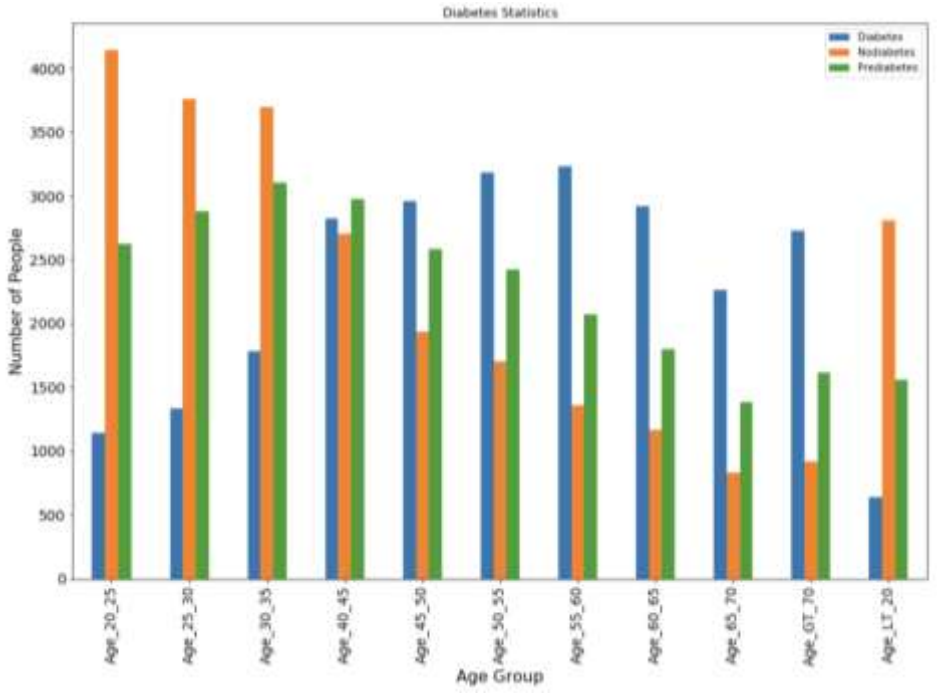
stratified k-fold cross-validation will be used for creation of the train, test and validation set.

We have created a Dataset after data cleansing and enrichment. The one-Hot-Encoding is implemented to take care of the categorical data into numerical representation so that the regression could be performed

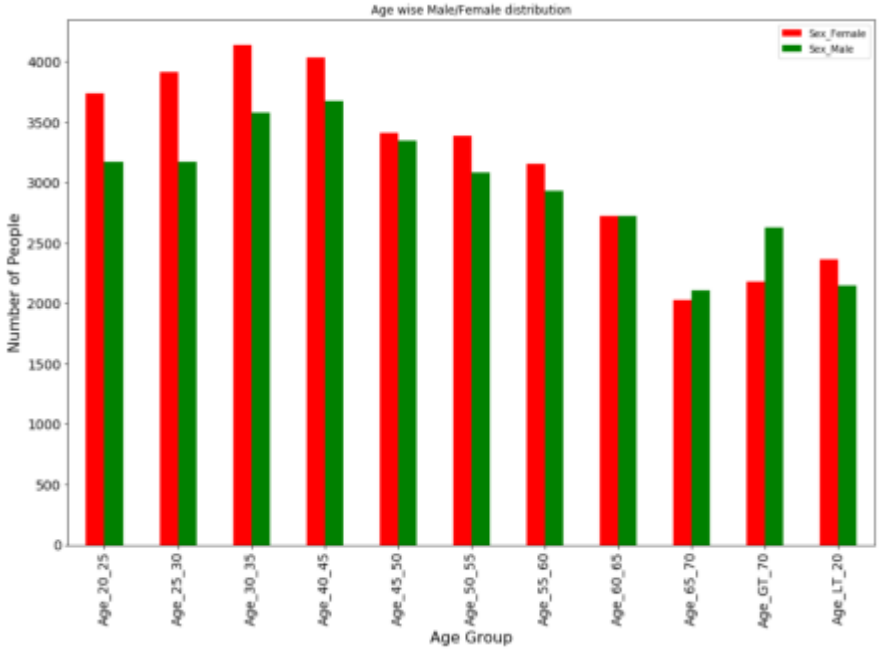
Description	Number of Entries
Sugar_Level	75000
Hb_Low	75000
Hb_High	75000
Age_LT_20	75000
Age_20_25	75000
Age_25_30	75000
Age_30_35	75000
Age_35_40	75000
Age_40_45	75000
Age_45_50	75000
Age_50_55	75000
Age_55_60	75000
Age_60_65	75000
Age_65_70	75000
Age_GT_70	75000
BMI_LT_18	75000
BMI_18_20	75000

BMI_20_22	75000
BMI_22_24	75000
BMI_24_26	75000
BMI_26_28	75000
BMI_28_30	75000
BMI_30_32	75000
BMI_GT_32	75000
Diabetes	75000
Prediabetes	75000
Nodiabetes	75000
BP_HIGH	75000
BP_LOW	75000
BP_Normal	75000
Pulse_Rate	75000
Iodine_level	75000
Sex_Female	75000
Sex_Male	75000
rural_urban_Rural	75000
rural_urban_Urban	75000
state_code_ASSAM	75000
state_code_BIHAR	75000
state_code_CHHATTISGARH	75000
state_code_JHARKHAND	75000
state_code_MADHYA PRADESH	75000
state_code_ODISHA	75000
state_code_RAJASTHAN	75000
state_code_UTTAR PRADESH	75000
state_code_UTTARAKHAND	75000

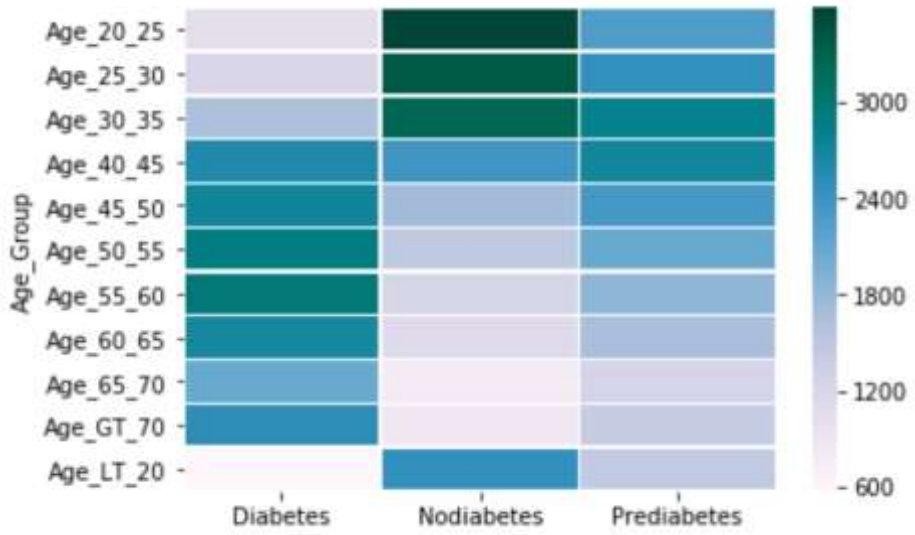
2.2. Exploratory Visualization



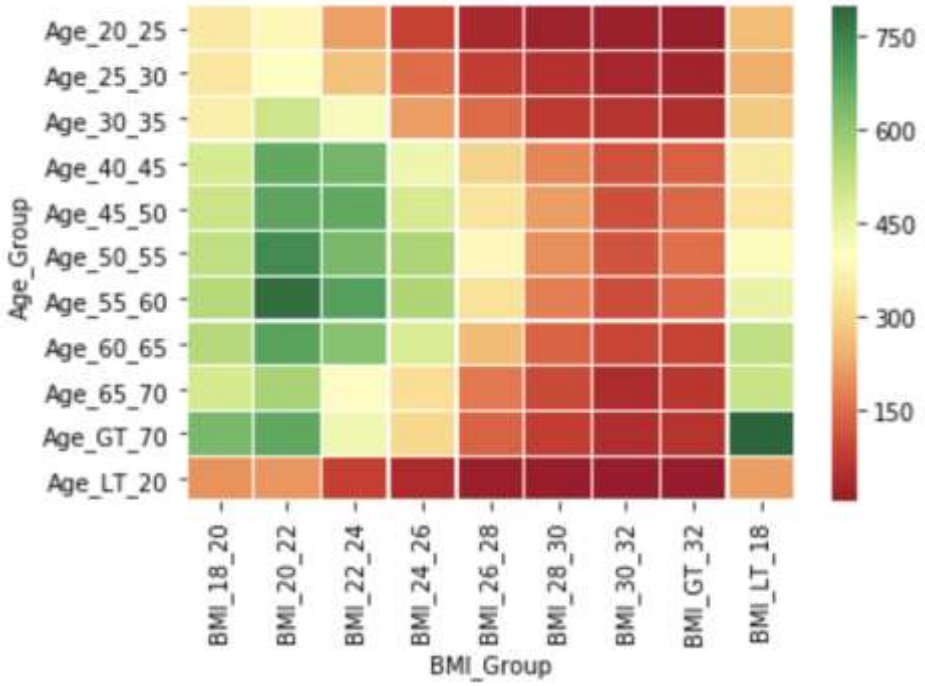
(Fig: 2.2.1: Population spread across age groups)



(Fig: 2.2.2: Male/ Female distribution)



(Fig: 2.2.3: Heatmap - Diabetes status across age groups)



(Fig: 2.2.4: Heatmap – BMI / Age distribution)

2.3. Algorithms and Techniques

At present the following are some of the criteria considered for the detection of early diabetes / prediabetes cases.

Fasting blood sugar test: A blood sample was taken after an overnight fast.

- **Normal**: A fasting blood sugar level less than 100 mg/dL (5.6 mmol/L)
- **Prediabetes**: A fasting blood sugar level from 100 to 125 mg/dL (5.6 to 6.9 mmol/L)
- **Diabetes**: A fasting blood sugar level > 126 mg/dL (7 mmol/L) or higher on two separate tests

Body mass index: A high BMI has strong correlation with diabetes. Worldwide the BMI higher than 25 is considered to be high risk. This figure is 23 for Asians.

Blood pressure : Systolic (Top Number) / Diastolic (Bottom number)

- (90/60) or less: May have low blood pressure.
- (90/60 - 120/80): The blood pressure reading is ideal and healthy
- (120/80 -140/90): A normal blood pressure reading but it is a little higher than it should be
- (>=140/90 over several weeks): May have high blood pressure (hypertension).
- Systolic >= 140 - High blood pressure, regardless of your bottom number.
- Diastolic >= 90 - High blood pressure, regardless your top number.
- Systolic <= 90 - Low blood pressure, regardless of your bottom number.
- Diastolic <= 60 - Low blood pressure, regardless of your top number.

Age : If the age is more than 45 years is advised to receive an initial blood sugar screening.

Hamoglobin level :

- Men : >= 13.5 g/dL
- Wemen: >= 12 g/dL

Algorithm:

The Softwares used:

- Language : Python
- Application: Jupiter Notebook
- Libraries: SK Learn, Keras, Matplotlib, Pandas, Numpy, Graphviz

The analysis is done using the regression analysis using the Decision tree method….Since the target variable can take a discrete set of values, in this case classification trees method is implemented.

- The Training data has 41 features.
- The header set : ["Diabetic", "Non-Diabetic", "Pre_Diabetic"]

Also the Sequential modelling using Keras is used for the analysis.

- The features matrix shape: 41
- The outcomes matrix shape: 3

2.4. Benchmark

Bench mark: Considering the health industries recommendation that a person having BMI > 25 and above and Age > 45 are prone to prediabetic or diabatic. In the current dataset the number of instances of people who are more than 45 years old and have BMI > 25 are 34746. There are 16176 people who are nondiabetic and 18570 are diabetic / prediabetic. The prediction accuracy of the criteria that people with BMI>25 and age more than equals to 45: 53.44 %

3. Methodology

3.1. Data Pre-processing

The data collected are categorical: Age, Sex, Haemoglobin level, Pulse rate, Blood Sugar level, Sex… For processing these were classified into groups. Once the classification is done, the data base is converted based on one hot encoding for each group.

AGE = ['Age_45_50', 'Age_25_30', 'Age_GT_70', 'Age_55_60', 'Age_LT_20','Age_60_65', 'Age_30_35', 'Age_20_25', nan, 'Age_50_55','Age_40_45', 'Age_65_70']

Following Python code was used for the Age categorization.

```
df['Age_LT_20'] = [1 if X <= 20 else 0 for X in df['Age'] ]
df['Age_20_25'] = [1 if X > 20 and X <= 25 else 0 for X in df['Age'] ]
df['Age_25_30'] = [1 if X > 25 and X <= 30 else 0 for X in df['Age'] ]
df['Age_30_35'] = [1 if X > 30 and X <= 35 else 0 for X in df['Age'] ]
df['Age_35_40'] = [1 if X > 35 and X <= 40 else 0 for X in df['Age'] ]
df['Age_40_45'] = [1 if X > 40 and X <= 45 else 0 for X in df['Age'] ]
df['Age_45_50'] = [1 if X > 45 and X <= 50 else 0 for X in df['Age'] ]
df['Age_50_55'] = [1 if X > 50 and X <= 55 else 0 for X in df['Age'] ]
df['Age_55_60'] = [1 if X > 55 and X <= 60 else 0 for X in df['Age'] ]
df['Age_60_65'] = [1 if X > 60 and X <= 65 else 0 for X in df['Age'] ]
df['Age_65_70'] = [1 if X > 65 and X <= 70 else 0 for X in df['Age'] ]
df['Age_GT_70'] = [1 if X > 70  else 0 for X in df['Age'] ]
```

BMI : It is calculated based on the weight and height of a person. After which the people were distributed in the following segments.

```
[ 'BMI_LT_18', 'BMI_18_20', 'BMI_20_22', 'BMI_22_24',
  'BMI_24_26', 'BMI_26_28', 'BMI_28_30', 'BMI_30_32',
          'BMI_GT_32']
```

```
df['BMI']= df['Weight_in_kg']/((df['Length_height_cm']/100)**2)
```

```
df['BMI_LT_18'] = [1 if X < 18 else 0 for X in df['BMI']]
df['BMI_18_20'] = [1 if X >= 18 and X < 20 else 0 for X in df['BMI']]
df['BMI_20_22'] = [1 if X >= 20 and X < 22 else 0 for X in df['BMI']]
df['BMI_22_24'] = [1 if X >= 22 and X < 24 else 0 for X in df['BMI']]
df['BMI_24_26'] = [1 if X >= 24 and X < 26 else 0 for X in df['BMI']]
df['BMI_26_28'] = [1 if X >= 26 and X < 28 else 0 for X in df['BMI']]
df['BMI_28_30'] = [1 if X >= 28 and X < 30 else 0 for X in df['BMI']]
df['BMI_30_32'] = [1 if X >= 30 and X < 32 else 0 for X in df['BMI']]
df['BMI_GT_32'] = [1 if X >= 32 else 0 for X in df['BMI']]
```

Hemoglobin Level:

Hb_Level = [‘Hb_High’,’Hb_Low’]

```
if (df['Sex'] == 'Female').all():
    df['Hb_Low'] =  [1 if x < 12.0   else 0 for x in df['Haemoglobin_level'] ]
else:
    df['Hb_Low'] =  [1 if y < 13.5  else 0 for y in df['Haemoglobin_level'] ]
```

```
if (df['Sex'] == 'Female').all():
    df['Hb_High'] =  [1 if x >= 15.5   else 0 for x in df['Haemoglobin_level'] ]
else:
    df['Hb_High'] =  [1 if y >= 17.5  else 0 for y in df['Haemoglobin_level'] ]
```

Pulse rate: The Pulse rate can vary from less than 55 to more than 82. Low pulse rate is healthy sign; however, it also depends on the sex and age of a person. Based on the criteria it is distributed from 0 to 6. 0 being bad and 6 being very good.

```
if (df['Sex'] == 'Female').all():
    df.loc[(df['Pulse_rate'] <= 59), 'Pulse_Rate'] =  6
    df.loc[(df['Pulse_rate'] <= 64) & (df['Pulse_rate'] > 59), 'Pulse_Rate'] =  5
    df.loc[(df['Pulse_rate'] <= 68) & (df['Pulse_rate'] > 64), 'Pulse_Rate'] =  4
    df.loc[(df['Pulse_rate'] <= 72) & (df['Pulse_rate'] > 68), 'Pulse_Rate'] =  3
    df.loc[(df['Pulse_rate'] <= 76) & (df['Pulse_rate'] > 72), 'Pulse_Rate'] =  2
    df.loc[(df['Pulse_rate'] <= 82) & (df['Pulse_rate'] > 76), 'Pulse_Rate'] =  1
    df.loc[(df['Pulse_rate'] > 82), 'Pulse_Rate'] =  0
else:
    df.loc[(df['Pulse_rate'] <= 55), 'Pulse_Rate'] =  6
    df.loc[(df['Pulse_rate'] <= 61) & (df['Pulse_rate'] > 55), 'Pulse_Rate'] =  5
    df.loc[(df['Pulse_rate'] <= 65) & (df['Pulse_rate'] > 61), 'Pulse_Rate'] =  4
    df.loc[(df['Pulse_rate'] <= 69) & (df['Pulse_rate'] > 65), 'Pulse_Rate'] =  3
    df.loc[(df['Pulse_rate'] <= 74) & (df['Pulse_rate'] > 69), 'Pulse_Rate'] =  2
    df.loc[(df['Pulse_rate'] <= 81) & (df['Pulse_rate'] > 74), 'Pulse_Rate'] =  1
    df.loc[(df['Pulse_rate'] > 81), 'Pulse_Rate'] =  0
```

3.2. Implementation

The supervised learning methods is used for this prediction modelling.

The following Classification techniques are implemented :

DecisionTreeClassifier from “sk learn”:

The supervised learning Dicision tree algorithm is used in this case from sk learn. It is non-parametric supervised learning method used for regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. It takes as input two arrays: an array X, sparse or dense, of size [75000_samples, 41_features] holding the training samples, and an

array Y of integer values, size [75000_samples], holding the class labels for the training samples:

We have also used the deep learning

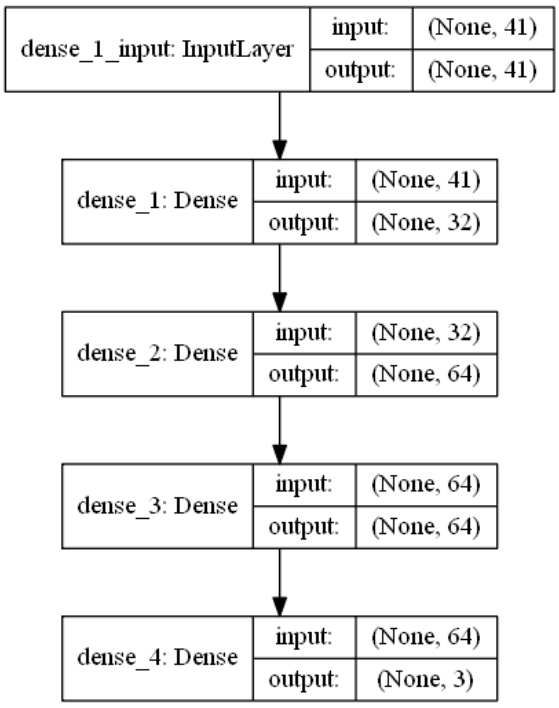
Sequential modelling from Keras:

- The features matrix shape: 41
- The outcomes matrix shape: 3

The following learning process parameters are configured:

- Optimizers: rmsprop, Adam
- Activation Functions: Relu, Sigmoid, Softmax,
- Loss Functions: binary_crossentropy
- List Matrics: accuracy

Implementation - 01: 4 dense layers were used in the model. Input layer of 32 nodes. two 64 node hidden layer and one 3 node output layer. Activation function used is Relu for the hidden layer and the output layer the 'sigmoid' activation function was used. The 'rmsprop' Optimizer function used for this model.



```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(features, outcomes, test_size=0.2, random_state=42)
```

```
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
```

```
model = Sequential()
model.add(Dense(32, activation='relu', input_dim=41))
model.add(Dense(64, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(3, activation='sigmoid'))
model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

```
# x_train and y_train are Numpy arrays --just like in the Scikit-Learn API.
model.fit(X_train, y_train, epochs=5,  batch_size=32)
```

```
Epoch 1/5
60000/60000 [==============================] - 6s 94us/step - loss: 0.6009 - acc: 0.6822
Epoch 2/5
60000/60000 [==============================] - 5s 83us/step - loss: 0.5924 - acc: 0.6879
Epoch 3/5
60000/60000 [==============================] - 5s 89us/step - loss: 0.5909 - acc: 0.6892
Epoch 4/5
60000/60000 [==============================] - 5s 86us/step - loss: 0.5900 - acc: 0.6898
Epoch 5/5
60000/60000 [==============================] - 5s 87us/step - loss: 0.5894 - acc: 0.6902
<keras.callbacks.History at 0x11b52a588>
```

3.3. Refinement

DecisionTreeClassifier: To start with I set the maximum leaf nodes to None

```
DecisionTreeClassifier(class_weight=None, criterion
='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity
_split=None,
            min_samples_leaf=1, min_samples_split=2
,
            min_weight_fraction_leaf=0.0, presort=F
alse, random_state=None,
            splitter='best')
```

The result: The training accuracy was 70.98% wile there is a significant loss in testing accuracy 41.83. Once I changed the Max_Depth to 3, the training and testing accuracy almost matched.

The Classification models gave modest results.

	Training Accuracy	Testing Accuracy
DecisionTreeClassifier 01:	70.98	41.83
DecisionTreeClassifier 01:	45.47	44.89

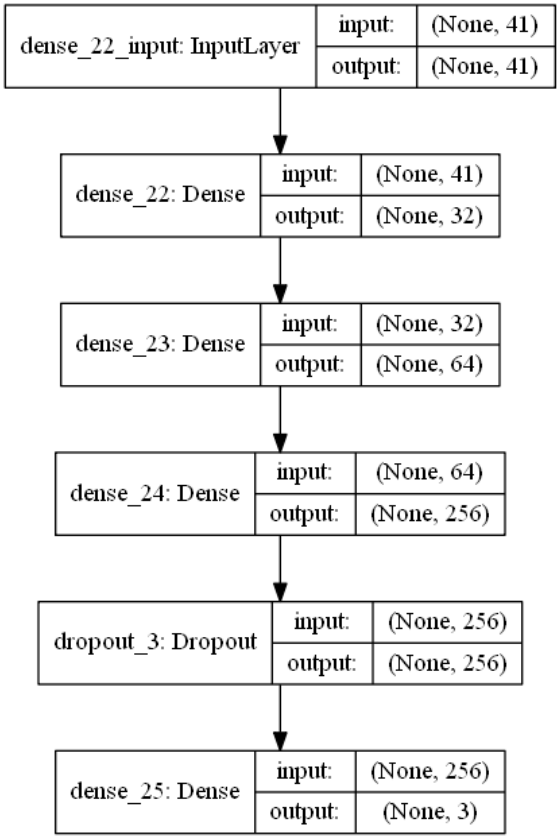
AdaBoostClassifier: We have kept the Max_Depth value to 3 to see that the model does not overfits.

```
AdaBoostClassifier(algorithm='SAMME.R',
          base_estimator=DecisionTreeClassifier(cla
ss_weight=None, criterion='gini', max_depth=3,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity
_split=None,
            min_samples_leaf=1, min_samples_split=2
,
            min_weight_fraction_leaf=0.0, presort=F
alse, random_state=None,
            splitter='best'),
          learning_rate=1.0, n_estimators=4, random
_state= 10)
```

XGBClassifier:

```
XGBClassifier(base_score=0.5, booster='gbtree', col
sample_bylevel=1,
       colsample_bynode=1, colsample_bytree=1, gamm
a=0, learning_rate=0.1,
       max_delta_step=0, max_depth=3, min_child_wei
ght=1, missing=None,
       n_estimators=100, n_jobs=1, nthread=None,
       objective='multi:softprob', random_state=0,
reg_alpha=0,
       reg_lambda=1, scale_pos_weight=1, seed=None,
 silent=None,
       subsample=1, verbosity=1)
```

Implementation 02: 4 dense layers were used in the model. 32 node input layer, one 64 node hidden layer, and one 256 node hidden layer. The output layer is of 3 node. Activation function used is Relu for the hidden layer and the output layer the ‘softmax’ activation function was used. The ‘Adam’ Optimizer function used for this model.



Model implementation:

```
model = Sequential()
model.add(Dense(32, activation='relu', input_dim=41))
model.add(Dense(64, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(3, activation='softmax'))
model.compile(optimizer='Adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

4. Results

4.1. Model Evaluation and Validation

The Classification models gave modest results.

	Training Accuracy	Testing Accuracy
• DecisionTreeClassifier :	45.77	44.89
• AdaBoostClassifier :	45.77	45.79
• XGBoost Classifier:	**49.79**	**48.67**

The Keras Sequential model gave the following results:

	Training Accuracy	Testing Accuracy
• Keras Sequential Model-01	68.80	68.52
• Keras Sequential Model-02	**69.85**	**69.00**

4.2. Justification

There were no difference noted in the prediction rate of Decision Tree and the Adaboost models.

The Xboost classifier provided a 4% better performance over the DecisionTree and Adaboost Models.

The Kears sequential model can predict at 68.5% accuracy which is 20% more than the other decision tree / adaboost / X-boost classifier.

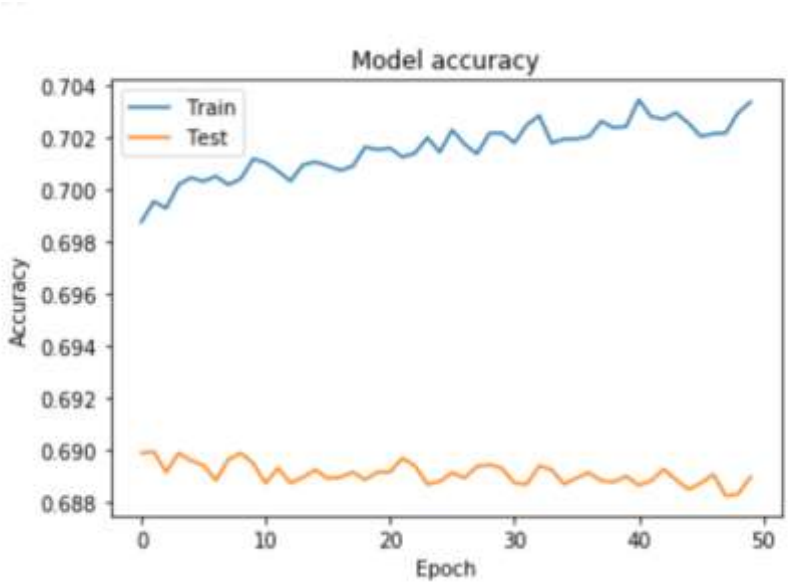
The further enhancement of the sequential model resulted in better training performance with the number of epochs (Training with 45000 entries, Validation: 15000 entries, Testing 15000 entries). The Trainin accuracy improved to 69.85% and the testing accuracy improved to 69.00% however the testing result remained at the

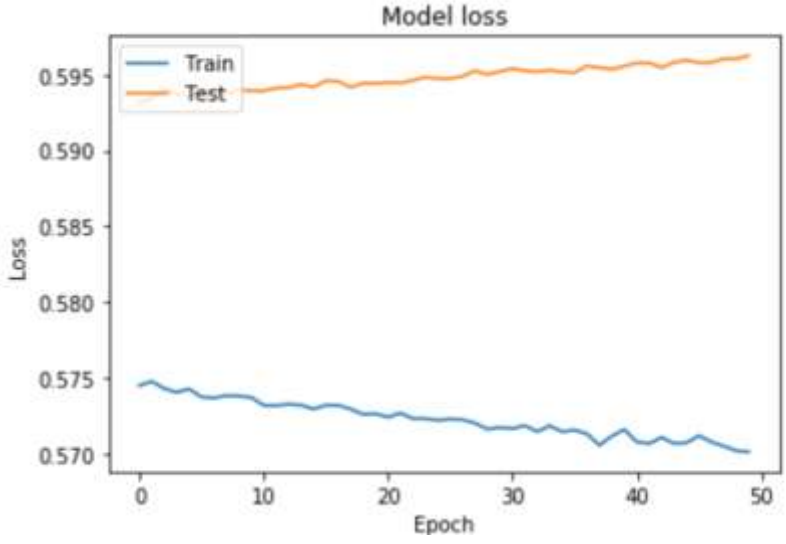
initial level. Hence it is not acceptable as the difference between the train and test performances increased.

5. Conclusion

5.1. Free-Form Visualization

The results from the Keras Sequential modeling are depicted below.





5.2. Reflection

It's impressive to see how in this case the sequential model is giving 20% better result. We started with the benchmark figure of 53% accuracy for age group > 45 and BMI 25.

The conventional DecisionTree model predicted with an accuracy of 48%. However, it is noteworthy that this figure is for all age groups.

5.3. Improvement

As we have seen the prediction performance with the Sequential model has been 22% better than that of the DecisionTreeClassification, we can further improve the sequential model. However, there could be a limit to this model if we can gather other factors that influences diabetes occurrences e.g. eating habits, Exercise, and occurrences if diabetes to any close family members.