**Algorithms for Estimation of BP**

**Machine Learning**

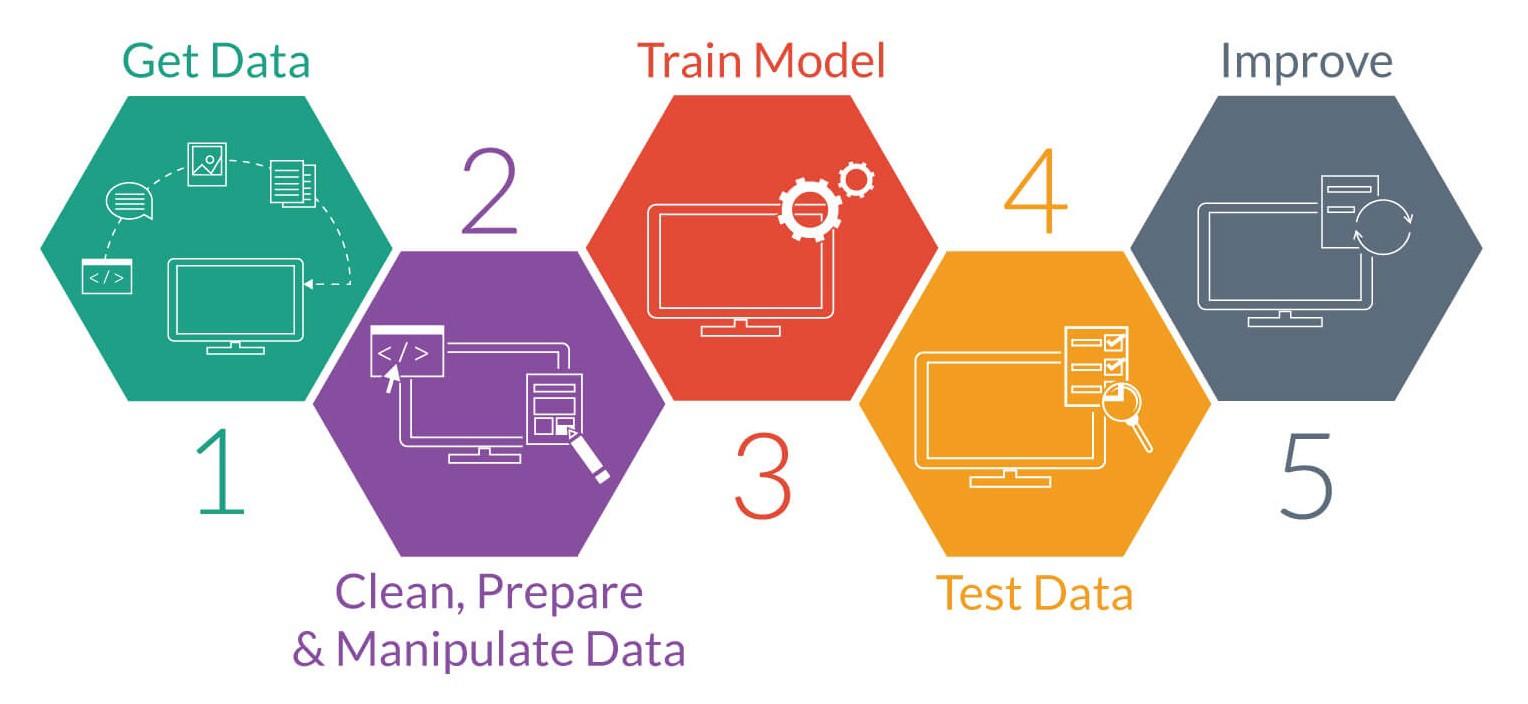
It is the field of study in which a computer program learns from previous experience/ data to perform a specific related task.

Major Algorithm Division:

* **Supervised**: In this type of algorithm, input-output pair is given with labeled training data. For each input ‘x’ algorithm maps the output ‘y’
* **Unsupervised**:It finds the relation and groups in the data by forming clusters in the set of unlabeled data.

**Steps to proceed in a machine learning problem**

1. **Define the problem** - At first, the problem should be clearly defined in terms of mathematics or statistics to be solved by machine learning
2. **Prepare Data** - Next is to be aware of the data, and how it should be manipulated
3. **Data Selection** - Eventually not all available data is useful, it is important to section the data to be applicable
4. **Data Preprocessing**- Before the data is used for modelling, it is normalized/ preprocessed for singular operations
5. **Feature Engineering** -It is done by using domain knowledge of available data to create features
6. **Model Selection** - The model is the algorithm which could fit the data into an estimation process
7. **Model Evaluation** - The selected model is run on the data to find the results
8. **Improve Results** - Once the algorithm puts the results, feedback can be done to improve the results
9. **Deploy**- The algorithm is deployed into the final model for usage in the form of apk files etc.



**Estimation of Blood Pressure for Machine Learning**

The blood pressure is measured using the hardware setup defined by ADC and microcontroller. Once the data is fetched and sent to receiver, it is used to extract features which would estimate the blood pressure for the corresponding signal. The signals collected are ECG and PPG which are measured using PTT (Pulse Transit Time). The post processing is done on Python GUI which plots the output and displays the result.

The process of understanding the GUI and machine learning setup is divided into two :

Python GUI - To plot the regenerated signal and display the BP estimation

Machine Learning Algorithm - To fit the model for estimation of BP from previously available data.

***Python GUI***

***How to use python for BP estimation GUI***

Step 1 : Download Anaconda2 for Python 2.7 (<https://conda.io/docs/user-guide/install/download.html> )

Step 2: Installation of packages: First open the Anaconda Prompt and run the following commands for installation:

1. Package: Scipy ; Command : conda install -c anaconda scipy
2. Package : PySerial ; Command : conda install -c anaconda pyserial
3. Package : Matplotlib ; Command : conda install -c conda-forge matplotlib

Step 3: Open Anaconda Prompt and type ‘idle’. From file open ‘get\_signal.py’, set the serial com port (as found in device manager(Ports and com ports)) . Then go to run module after setting up the hardware and run the code.

***Flow Control***

**Reconstruction of data:** The data which is received from serial port is in digital form. It is collected in different array to be finally plotted to find the parameters. But before that the signal is passed through various filters in order to get best results. ECG and PPG processing is done by:

**ECG:** For ECG, the R- peaks are to be found and the difference between the consecutive R- peaks. To find the R peaks,

1. DC Cancellation and Normalization

The DC component of the signal doesn’t affect the calculations. Thus we subtract the mean/ average of the values from the signal to cancel DC component.

*x = x − mean(x)*

And then signal is normalized to fit in the desired range of -1 to 1 by dividing the signal by maximum of the absolute values.

*x = x / max( abs( x ) )*

1. Low Pass Filtering

The low pass filter is used to filter out all the noise with frequency higher than the defined threshold value. The signal is passed through a designed digital filter.

1. Thresholding and Peak Location Determination

For finding the peaks, one threshold value is chosen and all the peaks above that value are counted. For example:

threshold = mean(x) + 0.2 \* max (x)

The peak is found between the two set of values and thus all points are found which cross the threshold value and peaks are detected in between of those values.

**PPG** : To estimate BP from PPG signal, methods like finding time difference between peak values can’t work because of lack of relation between the two quantities. Thus machine learning is used to train the model to predict the BP from PPG . The following features are extracted to train the model:

1. Systolic Peak Amplitude (Asys)

2. Valley Amplitude (Aval)

3. Dicrotic Notch Amplitude (Adn)

4. Systolic Area (SysArea)

5. Dicrotic Notch Area(DNArea)

6. Total Area(TotArea)

7. Peak Interval(PeakInt)

8. Pulse Height(PeakHt)

9. Pulse Interval(PulseInt)

10.Augmentation Index(AugIndex)

Thus the feature matrix is of the form:

For SBP (Systolic Blood Pressure) :

SBPfeatureVector = ​[ Asys Aval Adn SysArea DNArea TotArea PeakInt PeakHt PulseHt AugIndex SBP] ​

For DBP (Diastolic Blood Pressure)

DBPfeatureVector = ​[ Asys Aval Adn SysArea DNArea TotArea PeakInt PeakHt PulseHt AugIndex DBP]

To find these features, first all points of systolic peaks, valleys and dicrotic notches.

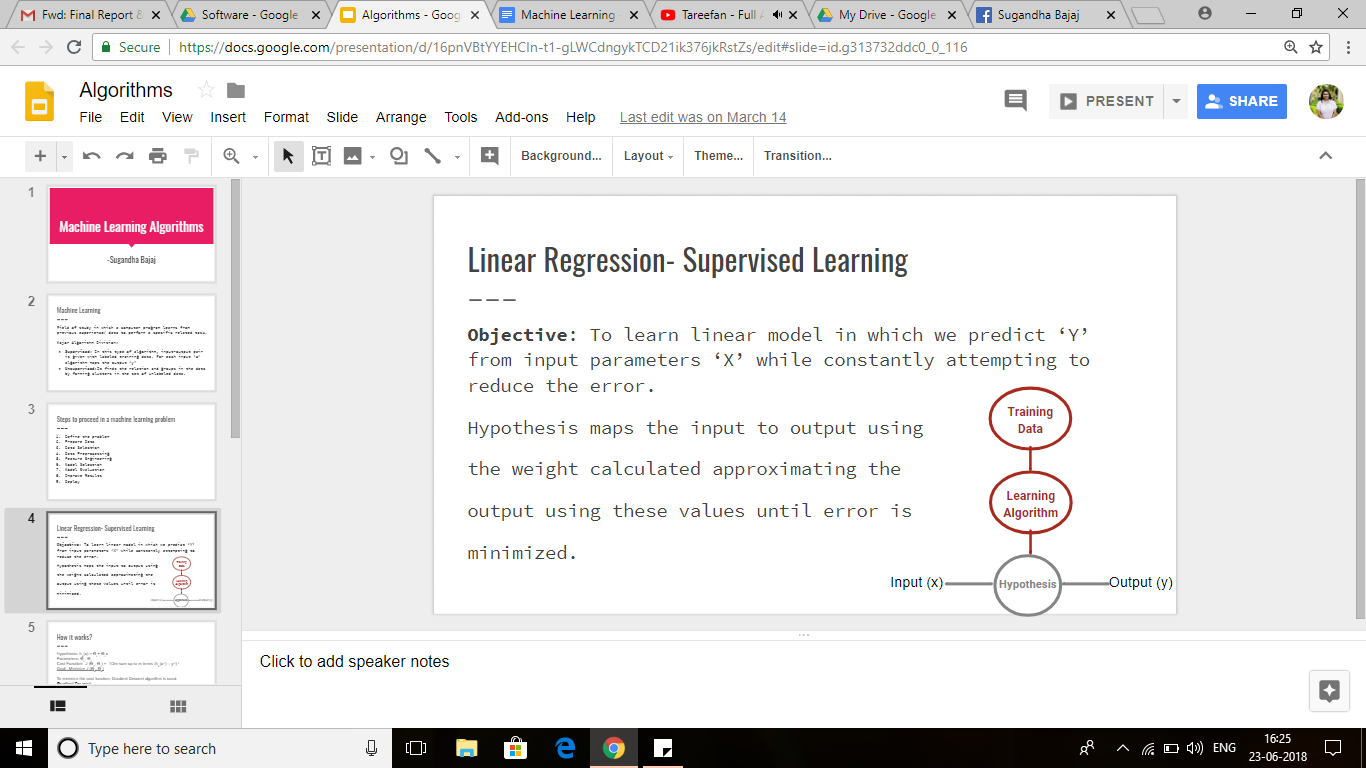
Systolic peaks are detected in same way as of ECG peaks by DC shift, normalization and LPF etc. To find the valley, the systolic peaks are taken into account and then minimum in between each pair is found. Once all these features are extracted, relation is established between PPG data and BP through machine learning.

***Machine Learning Process***

There are broadly two types of machine learning problems: Supervised and Unsupervised learning.

Linear Regression- Supervised Learning

**Objective**: To learn linear model in which we predict ‘Y’ from input parameters ‘X’ while constantly attempting to reduce the error.Hypothesis maps the input to output using the weight calculated approximating the output using these values until error is minimized.



**How it works?**

Hypothesis: hѲ(x) = Ѳ0+ Ѳ1x

Parameters: Ѳ0, Ѳ1

Cost Function: J (Ѳ0, Ѳ1) = 1/2m sum up to m terms (hѲ(x(i)) – y(i)) 2

Goal : Minimize J (Ѳ0, Ѳ1)

To minimize the cost function, Gradient Descent algorithm is used.

**Gradient Descent**:

The algorithm aims to find minimum point of model’s cost function by iteratively getting a better & better approximation. To begin, parameters are at initial guess which might minimize the function. Then partial derivative of loss function is taken in which we keep changing the parameter value until it converges to minimum after certain iterations.

The following process is repeated until convergence:

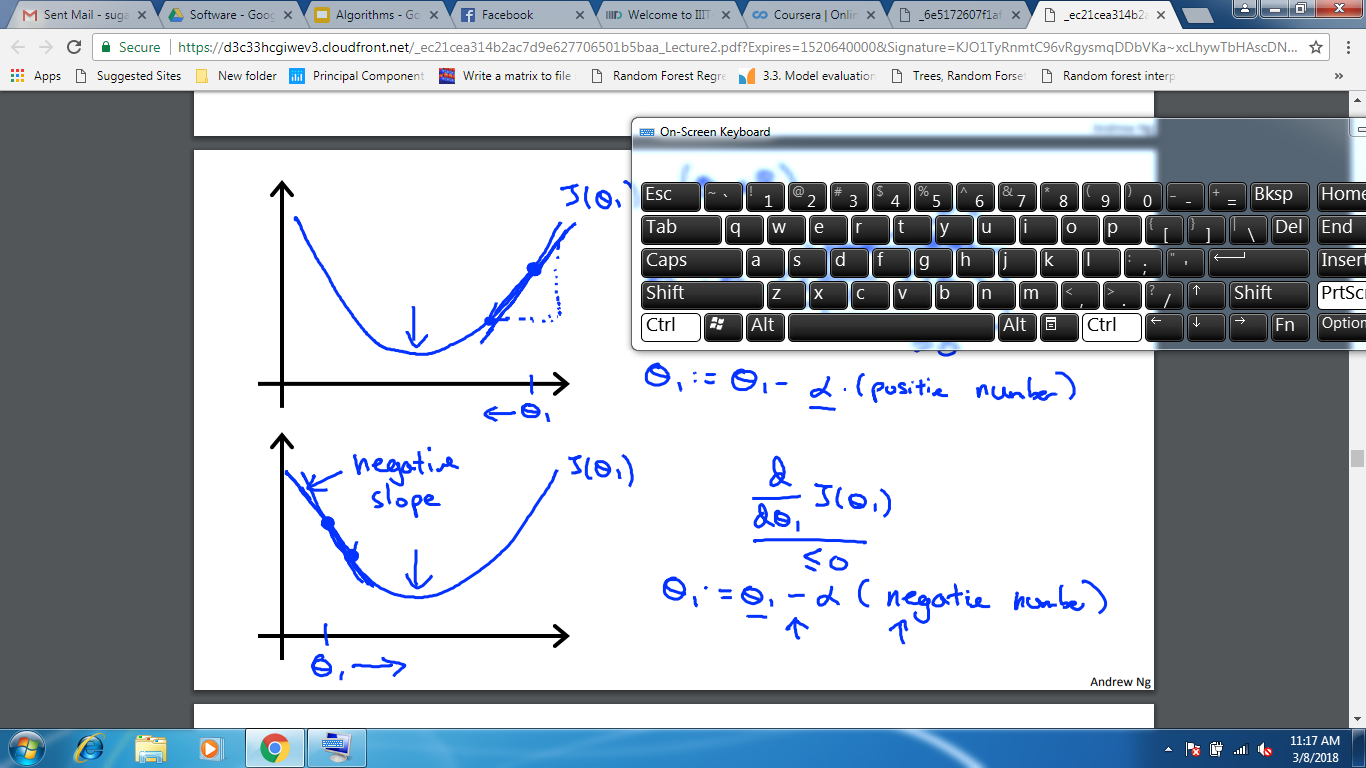
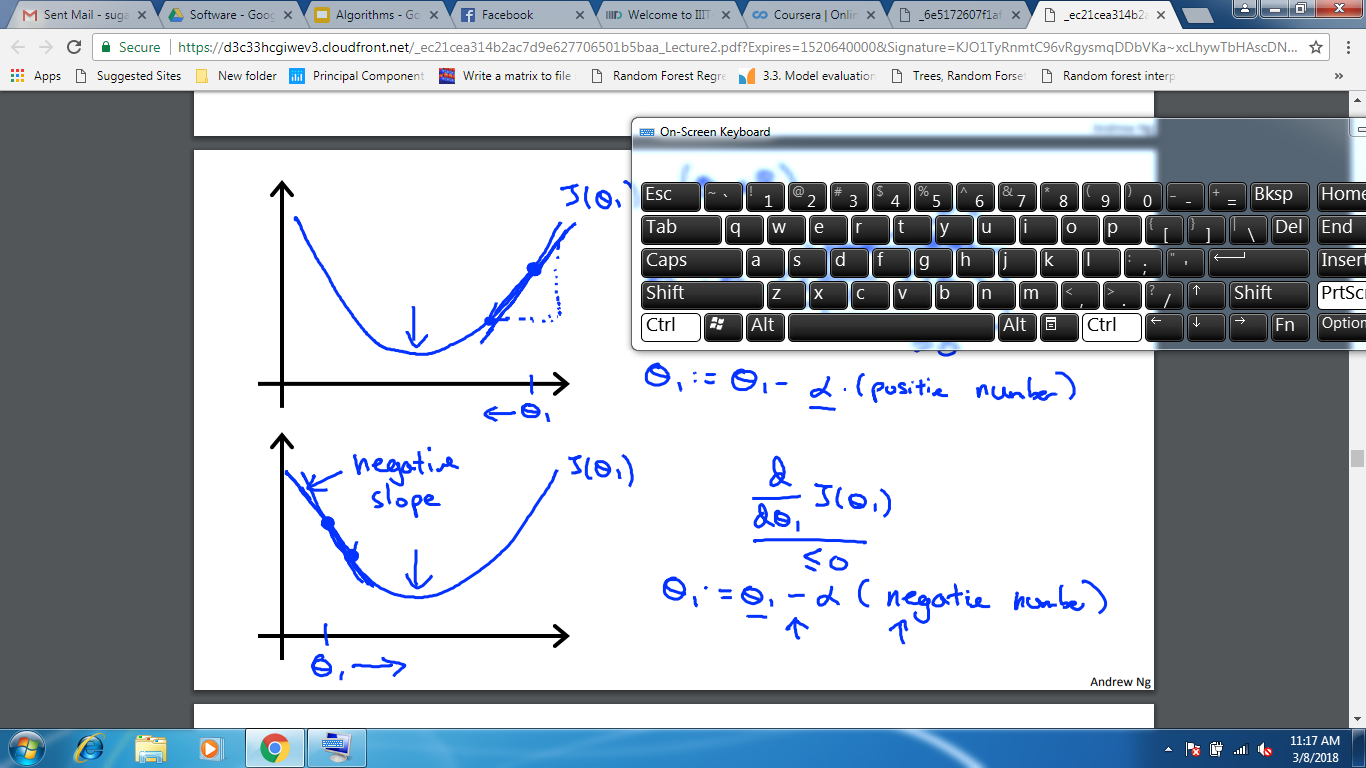
Ѳj = Ѳj - ᾀ d (J (Ѳ0, Ѳ1)) / /dѲj

Where ᾀ is the learning rate.

If alpha is too small, gradient descent can be slow

Else if alpha is too large, gradient descent can overshoot. Might also fail to converge, or even diverge.

As in the figures, at every point it decides which direction to go as per the slope to find the minima.



**Why this method?**

It is simple, easy to implement method which approximately predicts the output as a linear function of input features. Also, by plotting learning curve, we could understand how the algorithm is underfitting or overfitting the data and correct the algorithm accordingly.

**Code:**

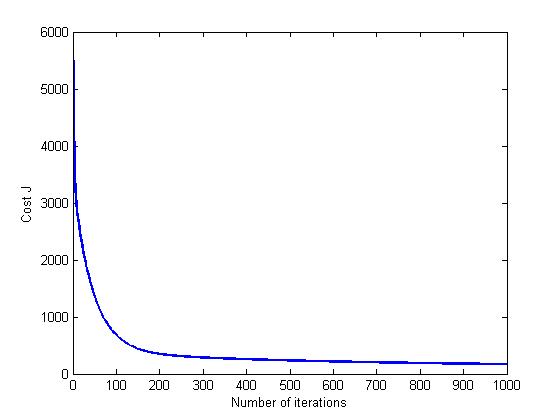
run.m : This is the main file which calls the other functions. Here one can choose to process the output equation for Diastolic BP or Systolic BP.

Extract features.m : It is used to extract the features from the given BP matrix which contains the data and information of 94 samples.

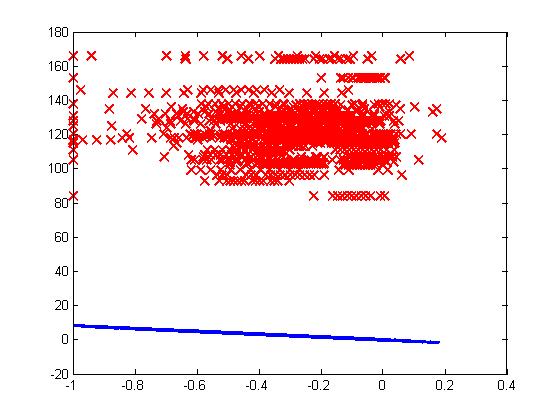
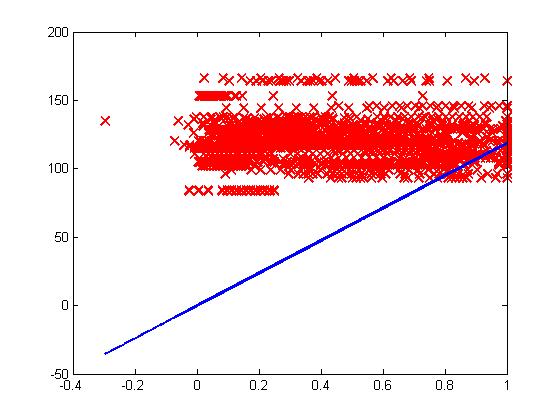
gradientDescentMulti.m: Calculates the cost function and iteratively tries to decrease it as much as possible.

**Observations:**

* The following graph depicts the performance error rate of the algorithm. As seen, the cost function decreases with increase in number of iterations.

****

* The next two images show the data fitting curve of the linear function and the data. The red ‘x’ marks here denote the data and the blue line is the prediction line. It is clearly seen from the graph that prediction line is either hardly fitting the data or not able to fit properly.

****

**Result:**  Although algorithm is able to predict the required BP, the prediction function fails to completely fit on the feature variable. Thus it was needed to find feature importance for the proper prediction of Blood Pressure.

As the results fairly fit the data, initially the algorithm was used to predict the BP. The equation used to determine BP is:

**DBP EQUATION**

DBP = ​73.9188 + Asys \*0.564005 + Adn \*1.77362 + Aval \* -2.19562 + SysArea \*-0.280499 + DNArea \*0.181496 + TotArea \* -0.0990031 + PeakInt \* 0.928643+ PulseHt \* 2.75962 + PulseInt \* -0.140988 + AugIndex \* -0.787559

**SBP EQUATION**

SBP = ​119.726 + Asys \*1.38196 + Adn \*9.53411 + Aval \* -5.07123 + SysArea \*-0.983105 + DNArea \* 0.585099 + TotArea \* -0.398006 + PeakInt \* 1.20928 +PulseHt \* 6.4532 + PulseInt \* -1.16069 + AugIndex \* -0.681099

The values of SBP and DBP are calculated and displayed in their respective position on the display panel.

* **As it was important to increase the accuracy of the system, few other techniques were tested with the available data.**

**Random Forest**

It creates the forest with no. of trees. Higher the number of trees,higher the accuracy.

When training data for current tree is drawn , ⅓ of data is left which is used to get a running unbiased estimate of classification error as the trees are added to the forest.

Once, after the tree is built, proximities are computed for each pair of cases. At the end of run, proximity is normalized and used in replacing missing data etc.

**Why this method?**

Easy to use and explain, follows same approach as humans, simplified by visualization, does not overfit the data. Specifically for our problem, it gives us the idea about the feature importance and better approximation results.

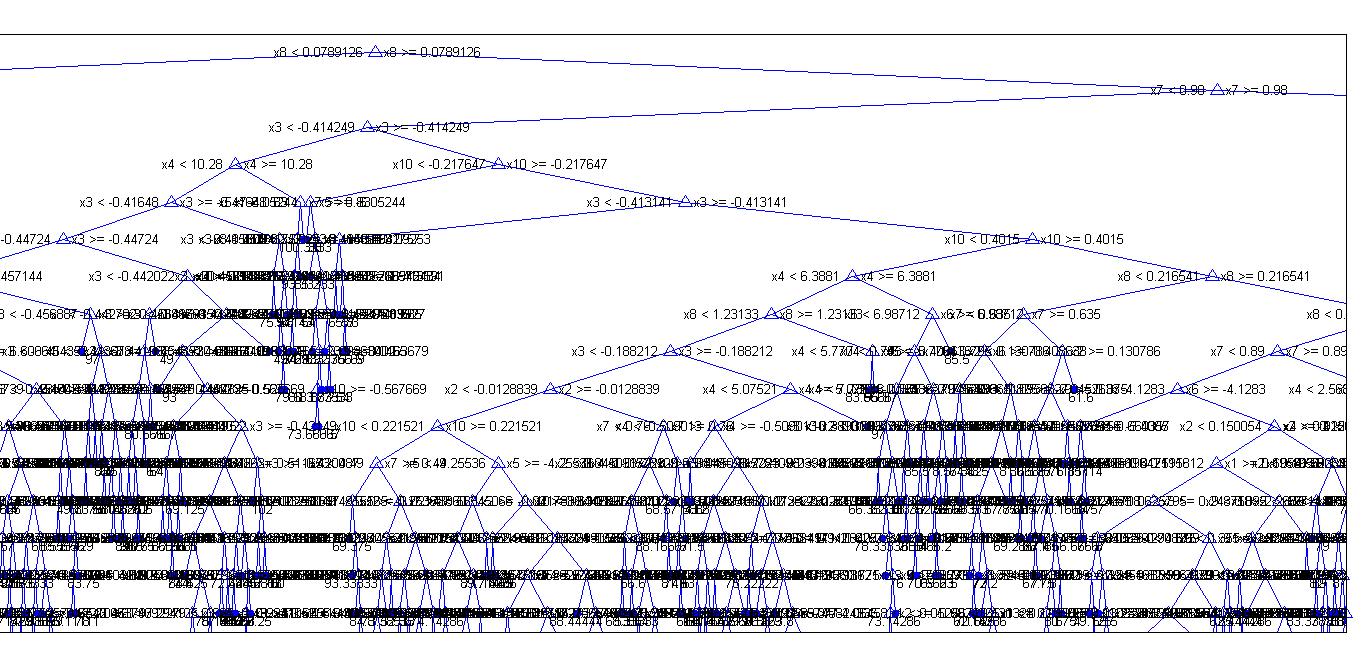
**Code:**

random.m : This is the main file which contains all the functions of the run. Here one can choose to process the output equation for Diastolic BP or Systolic BP.

Firstly training and test set is created by splitting the data into 80:20 ratio. Then TreeBagger function is called which created the tree in the regression mode and initiates the random forest algorithm. Once the tree is made, and the YFIT parameter is calculated by averaging all the nodes, the prediction function is used to find the output on the test set and validate it to calculate the accuracy of the system.

Extract features.m : It is used to extract the features from the given BP matrix which contains the data and information of 94 samples.

**Observations:**



**Result:**

**Accuracy**: 93%

**Tree Details:**

Ensemble with 165 bagged decision trees:

Training X: [6000x10]

Training Y: [6000x1]

Method: regression

Nvars: 10

NVarToSample: all

MinLeaf: 5

FBoot: 1

SampleWithReplacement: 1

ComputeOOBPrediction: 0

ComputeOOBVarImp: 0

Proximity: []