Heart Health Report Automation

Umang Pandya

(Data Scientist, Business Analyst, Statistician)

Introduction:

The increase in the number of patients who wants to show the report to doctors of their blood and cardio in order to get the answer whether he/she has heart disease or not is a traditional way that is how it has happened till now to visit the cardiologist. My focus is mainly to reduce the emergency patients' waiting time by creating a machine learning algorithm that predicts accurately and give the patients answer whether they have heart disease or not who have reports. So that I can increase the efficiency of the process by this and emergency or other critical patients would have a priority visit to Doctor with reduced waiting time.

Data:

The source of the data is here and below are the credits.

Creators:

1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.

2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.

3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.

4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.

Donor: David W. Aha (aha '@' ics.uci.edu) (714) 856-8779

This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date.

Methodology:

I first wanted the data to be sorted by age and I did it accordingly.
The good thing is there are no NaN values or Null values in this dataframe which saved my time to be applied for imputation.Below are the attributes information:

Attributes Information:

- age
- sex
- chest pain type (4 values)
- resting blood pressure
- serum cholesterol in mg/dl
- fasting blood sugar > 120 mg/dl
- resting electrocardiographic results (values 0,1,2)
- maximum heart rate achieved
- exercise induced angina
- oldpeak = ST depression induced by exercise relative to rest
- the slope of the peak exercise ST segment
- number of major vessels (0-3) colored by fluoroscopy
- thal: 3 = normal; 6 = fixed defect; 7 = reversible defect

Actual Data (small preview)

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	29	1	1	130	204	0	0	202	0	0.0	2	0	2	1
1	34	1	3	118	182	0	0	174	0	0.0	2	0	2	1
2	34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
3	35	1	0	126	282	0	0	156	1	0.0	2	0	3	0
4	35	0	0	138	183	0	1	182	0	1.4	2	0	2	1

By looking at the table I was sure about one thing that there are some attributes as above that has high minkowski distance or Euclidean Distances with other variables which has values like 0,1,2. Those variables to be treated are ‘age’,‘trestbps’,’chol’,’thalach’. So I did the ***Normalization*** technique of feature scaling but I also kept the original data since I really wanted to compare the output of both treated and non treated data.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	0.092	1	1	0.411	0.645	0	0	0.638	0	0.0	2	0	2	1
1	0.121	1	3	0.421	0.650	0	0	0.621	0	0.0	2	0	2	1
2	0.110	0	1	0.381	0.678	0	1	0.620	0	0.7	2	0	2	1
3	0.101	1	0	0.362	0.811	0	0	0.449	1	0.0	2	0	3	0
4	0.119	0	0	0.468	0.621	0	1	0.617	0	1.4	2	0	2	1

Profuse Dispersion Reduction
P Sigma:

I used my own Feature scaling method that I call **P sigma or Profuse Dispersion reduction.** I call P sigma as multiplication of Z score with Coefficient of Variation
Why I do P sigma reduction which is my own method of scaling is because,

- It reduces the standard deviation.
- It give values between -1 to 1 normally where 0 is the mean.
- We can multiply these values with 100 and the result shows how far the observation is from the mean in percentage.
- Those values which are above 1 or below -1 can be easily considered as outliers and treated accordingly.
- Decreases the Minkowski or Euclidean distance in great amount.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	-0.466582	1	1	-0.012336	-0.171621	0	0	0.349845	0	0.0	2	0	2	1
1	-0.374613	1	3	-0.103505	-0.260956	0	0	0.162737	0	0.0	2	0	2	1
2	-0.374613	0	1	-0.103505	-0.147257	0	1	0.283021	0	0.7	2	0	2	1
3	-0.356219	1	0	-0.042726	0.145112	0	0	0.042454	1	0.0	2	0	3	0
4	-0.356219	0	0	0.048443	-0.256895	0	1	0.216197	0	1.4	2	0	2	1

Understand Psigma Theory:

The Standardization formula is:

$$Z = \frac{x - \bar{x}}{\sigma}$$

The formula of coefficient of Variation is:

$$Cv = \frac{\sigma}{\bar{x}}$$

$$P\sigma = Z \times Cv$$

$$P\sigma = \frac{x - \bar{x}}{\sigma} \times \frac{\sigma}{\bar{x}}$$

We can directly say that the formula for $Psigma$ or Profuse Dispersion Reduction is:

$$P\sigma = \frac{x - \bar{x}}{\bar{x}}$$

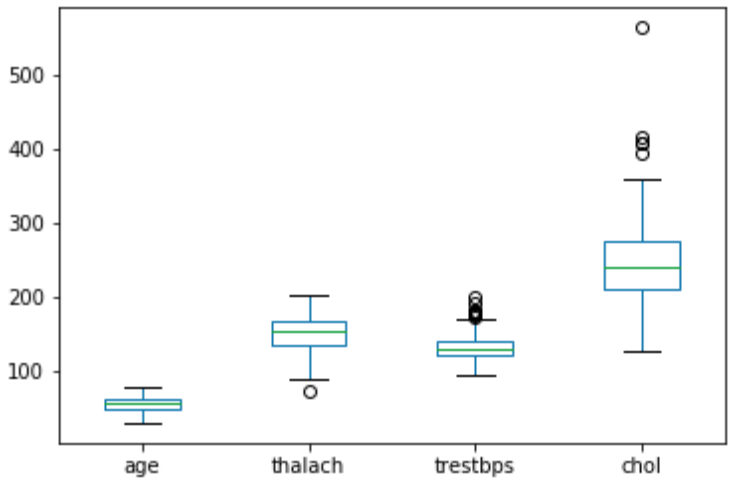
So we have now 3 types Data:

-Actual data
-Normalized Data
-Profuse Dispersion Reduction Data

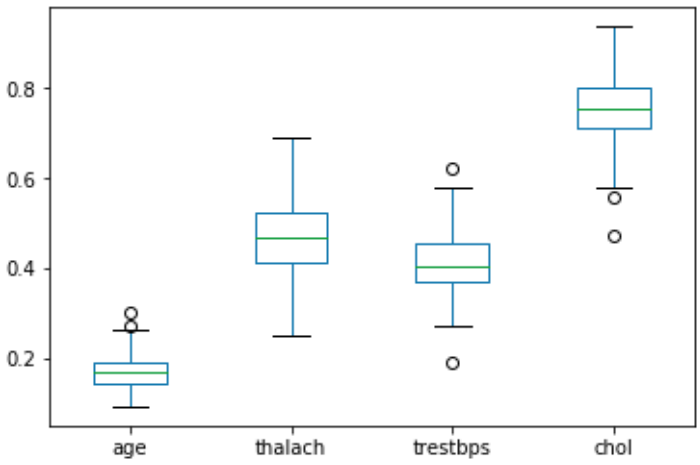
Below are some visualizations of how the feature scaling is affecting the data positively.

Boxplots:

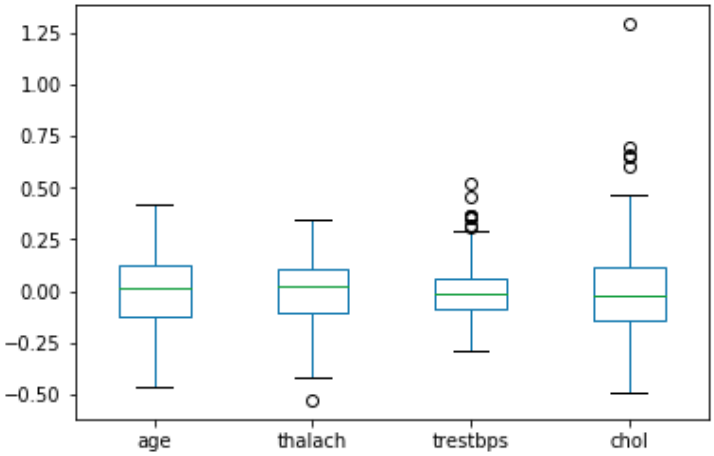
Actual Data:



Normalized:



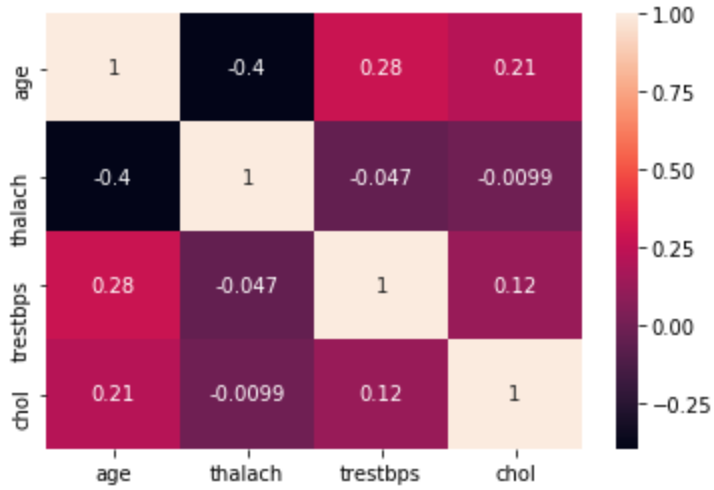
Profuse Dispersion Reduction:



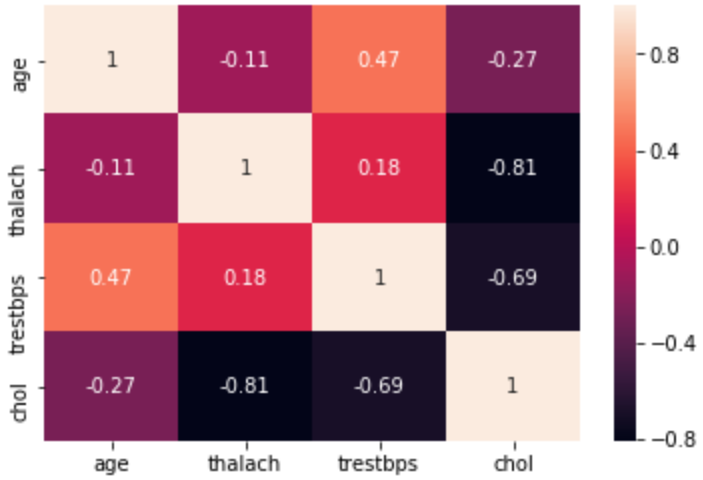
As we can see that feature scaling like Normalization has normalized the values and also did some outlier treatment. P sigma has not done much visually but has shifted all the values near 0 and decreased the standard deviation.

Correlation matrix:

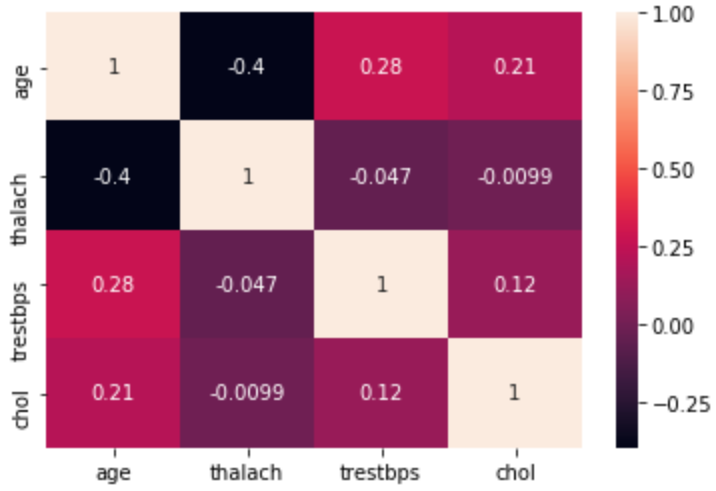
Actual Data:



Normalized:



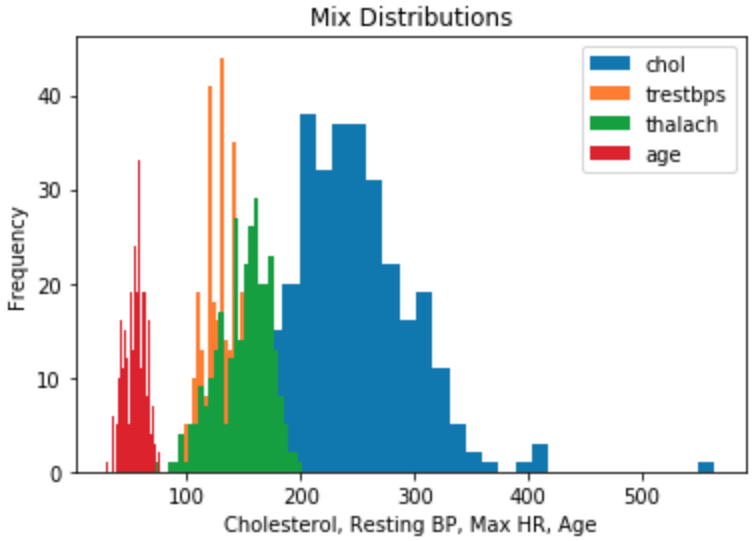
Profuse Dispersion Reduction:



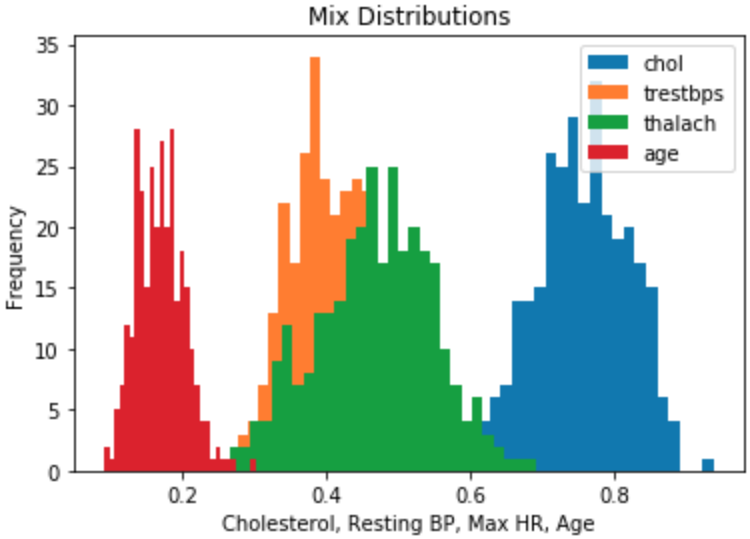
The Correlation matrix suggests that there is no correlation between these important variables which means there is no Multicollinearity however this project is based on classification supervised learning and ‘No multicollinearity’ may not affect or improve the model. Notice that the P sigma scaling didn't change the correlation however Normalized values did.

Distributions:

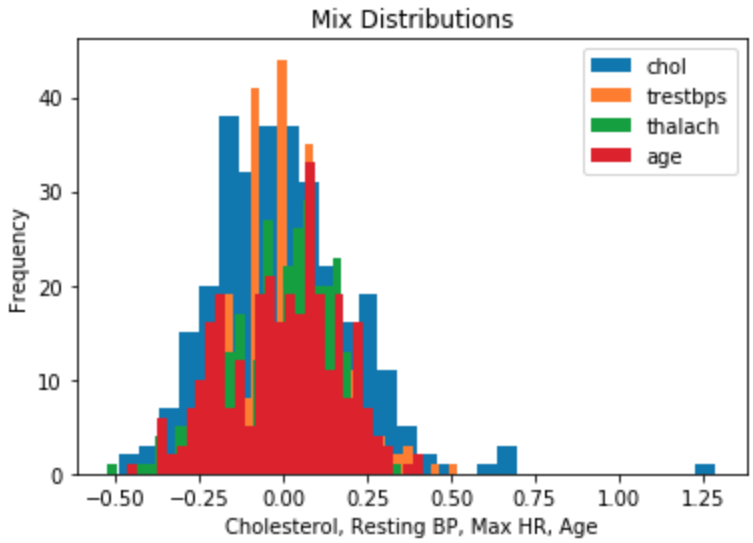
Actual Data:



Normalized:



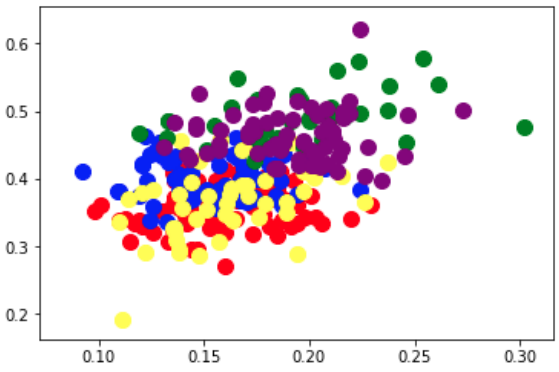
Profuse Dispersion Reduction:



So given above are the distributions for same 3 datas. We can clearly see how P sigma came in center as mean is 0 just like Normal distribution and all the attributes are in the same place. However the Actual Data and Normalized Data isn't. P sigma clearly decreases the Minkowski or Euclidean distances between Independent variables and it is visible.

Clustering Experiment:

There is another thing I wanted to examine which is if we do K means clustering and assign the cluster variable in the Normalized data how it will affect or improve the model. So I created one more variable of clusters by applying K means to make it act as independent variable for the algorithms below is the visualization of clusters.



So the fourth data is Normalized with Cluster variable. I decided to apply algorithms to all the 4 Datas to see if these affects or improves the models.

Since our target is whether a patient has Heart Disease of not which is (1,0), clearly it's a classification problem due to discrete, categorical outcome to be predicted.

Prior to create models it was important to do ***Principal Component analysis, Dimensionality reduction***. So I did PCA for all models except the actual data since I wanted to know what is the outcome without PCA as well.
After creating all the models I decided to check the accuracies though k Fold analysis since we do know that it is a better option that it counts the average accuracies based on various permutations of train data through k = n. I chose k = 10 in this scenario. Know that I also considered standard deviation of k fold analysis as an important aspect since it will show me the variability of the accuracies. The minimum is Standard Deviation the less is the variability and the Maximum is the accuracy the better model. So below is the outcome for the same.

Result:

	Algorithms	Normalized_with_Clusters	std1	Normalized_Data	std2	Actual_Data	std3	Psigma_scaling	std4
0	Logistic Regression	0.8282	0.0729	0.8395	0.0800	0.8458	0.0807	0.8019	0.0334
1	K Nearest Neighbor	0.8051	0.0610	0.8440	0.0838	0.6646	0.0627	0.8226	0.0719
2	Naive Bayes	0.7766	0.0602	0.8429	0.0666	0.8110	0.0820	0.8058	0.0556
3	Decision Tree	0.7479	0.0598	0.7349	0.0545	0.7934	0.0799	0.7429	0.0724
4	Random Forest	0.7049	0.0882	0.7694	0.0876	0.7770	0.0931	0.8172	0.0670
5	SVM Linear	0.8472	0.0445	0.8470	0.0748	0.8013	0.0648	0.8183	0.0370
6	SVM Sigmoid	0.8480	0.0522	0.8303	0.0578	0.5413	0.0010	0.7974	0.0632
7	SVM RBF	0.8350	0.0571	0.8258	0.0675	0.5413	0.0187	0.8306	0.0463

Looking at the Accuracy table I confirm that :

Support Vector Machine with kernel 'linear' shows Good accuracy of *0.8472* and Standard deviation is also less which is *0.0445* of **Normalized with cluster variable** data compared to other algorithms.

Also the **Support Vector Machine with kernel 'Radial basis function' on P sigma Data** also shows good accuracy of 0.8306 with less standard deviation of 0.0463.

I would take this into consideration that these two models are good enough for this automation of Heart Disease prediction.

Discussion:

The P sigma Profuse Dispersion Reduction is a Scaling method that I created is pretty much comfortable to me when it comes to exploring the data in a way that it shows in percentage if you multiply them with 100 and then that percentage is the distance from the mean of the data. Not only for this benefit but the Distribution is very sharp after applying the P Sigma with less Minkowski distance. Also, this is the first time that I tried a thought of using Unsupervised learning to create Cluster Variable and then do supervised learning on the model that includes this Cluster variable as an Independent Variable which in this model worked pretty great.

Conclusion:

This model is successful since the experiment of creating clusters and adding in Normalized data as well as P sigma Dispersion reduction data are showing maximum accuracy with minimum k fold standard deviation.