Future of Information and Communication Conference (FICC) 2018
5-6 April 2018 | Singapore

New Modification Version of Principal Component Analysis with Kinetic Correlation Matrix using Kinetic Energy

Sara K Al-Ruzaiqi
Computer Science Department
Loughborough University
Muscat, Oman
s.k.s.al-ruseiqi@lboro.ac.uk

Dr Christian W Dawson
Computer Science Department
Loughborough University
Loughborough, United Kingdom
c.w.dawson1@lboro.ac.uk

***Abstract*—Principle Component Analysis (PCA) is a direct, non-parametric method for extracting pertinent information from confusing data sets. It presents a roadmap for how to reduce a complex data set to a lower dimension to disclose the hidden, simplified structures that often underlie it. However, most PCA methods are not able to realize the desired benefits when they handle real world, and nonlinear data. In this work, a modified version of PCA with kinetic correlation matrix using kinetic energy is proposed. The features of this modified PCA have been assessed on different data sets of air passenger numbers. The results show that the modified version of PCA is more effective in data compression, classes reparability and classification accuracy than using traditional PCA.**

Keywords— Principle Component Analysis (PCA); kinetic correlation matrix; kinetic energy; algorithm; prediction

I. INTRODUCTION

Principal Component Analysis (PCA) is a classical multivariate data analysis technique, which is popular within linear feature extraction as well as the data compression of numerous uses [1]. PCA has been applied in numerous areas of information processing to prepare data due to its distinctive result of error reducing and correlating properties. PCA compresses most of the information in the first data space into a fewer features. It attempts to look for a subspace in which the variance is maximized [2]. The PCA subspace is spanned through the eigenvectors corresponding to the top eigenvalues of the sample covariance matrix. PCA also can be applied in data preparation for both supervised and un-supervised learning and recognition processes [3].

However, most PCA strategies might not result in desirable classification benefits when they cope with real world, nonlinear data. As nonlinear PCA and its variants can effectively capture the nonlinear relations, they might provide more effective power to cope with the real world, nonlinear data [4]. It is recognized that PCA is designed to find the most indicative vectors, i.e., the eigenvectors corresponding to the best eigenvalues of the sample covariance matrix.

As data with good spectral resolution results in unwanted data for classification, a proven way to conquer this issue is reducing the dimensionality of data space. Different feature extractions, as well as selection strategies, recommend using PCA, as it is highly effective and involves a mathematical process which transforms a selection of (possibly) correlated variables into a (smaller) selection of uncorrelated variables known as principal components [5].

The sheer size of data in the modern age is not only a challenge for computer hardware but also a bottleneck for the performance of many machine learning algorithms. Identifying patterns in data is one of the main goals of a PCA analysis, and it only works by reducing the data dimensionality only when there is strong correlation between the variables. In brief, PCA is a data analysis technique which finds directions of maximum variance in high-dimensional data and projects them onto a smaller dimensional subspace while retaining most of the information.

In this work, a modified version of PCA with kinetic correlation matrix using kinetic energy is proposed, where the transformed matrix is computed from samples of selected features only. The efficiency of the modified and traditional versions of PCA is compared by applying them to an air passenger dataset. The results show that the modified version of PCA is more effective in data compression, class reparability and classification accuracy than using traditional PCA.

II. MODIFICATION OF PCA

Since the original definition of PCA via approximating multivariate distributions by planes and lines [2], scientists have defined PCA from various elements [2], [3]. Among the definitions, utilizing the covariance matrix of the training sets to explain PCA is extremely well known in pattern recognition as well as the machine learning community.

Current implementations of PCA use a correlation matrix, the matrix obtained by pairwise correlation using Pearson correlation coefficient. However, in some cases the Pearson correlation coefficient could be limited in the sense that it fails to capture other properties of the data outside of the linear relation. For example, the correlation of two random vectors: $x=\{-4,-3,-2,-1,0\}$, $y=x^2$ => $Cor(x,y)=0$, using Pearson coefficient. However, this result is not capturing the non-linear relation between the two random vectors given by the functional transformation $(x)^2 \rightarrow (y)$ which means the

correlation is not zero (just non-linear). In order to improve this, the following two features have been introduced into traditional PCA in this work.

- *Information energy*: first introduced in 1966, is an analogy of the kinetic energy from physics to probability, which can be defined as follows:

  $x_1, x_2, \ldots, x_n$ and corresponding probabilities:

  $P=(P_1, P_2, \ldots, Pn)$

  $IE(p_1, p_2, \ldots, p_n)= \sum$

If the experiment has *n* outcomes, and every outcome has the same probability 1/*n*, then the information energy IE=1/*n*. If the experiment results in same outcome, then the probability for every outcome is 1 and the information energy has maximum value of IE=1.

The information energy increases when the randomness decreases. It is like reverse of Shannon entropy, for measuring bits of information to determine uncertainty. It is also an entropy, but the correct way to think about it is as 1/2∗m∗v2 of a random vector. Simple, but very powerful, the kinetic energy method works very well to improve the accuracy or improve some machine learning methods on row data especially if there are groups of categorical data, even if they are continuous they could be discretized.

- *Informational Correlation Coefficient*, also known as Onicescu's correlation coefficient, is a function of the joint probability density distribution of the two vectors x and y. Assume we have two random vectors x and y, the information correlation coefficient can be described as:

$$O(x,y) \quad \frac{\sum_k p^{(Pk)*p(Qk)}}{\sqrt{(P)*IE(Q)}} \qquad (1)$$

This is only applicable for the discrete data that we have dealt with in this research.

The Pearson correlation captures only linear properties of the manifold on which our raw data lives. For instance: if we take a random vector in R x=c(-4,-3,-2,-1,0,1,2,3,4) and y=x^2, Pearson or Spearman, will yield 0 correlation when in fact it is 0.5 because of the functional transformation x->^2. In this work, a new correlation coefficient, as a performance metric, instead of cross entropy as in the case of neural networks, or, in the case of genetic algorithms, as fitness functions, has been applied in the modified PCA.

Previously, PCA was utilized to decrease large data sets, correlated by a number of correlation metrics, or used in addition to deriving new features. Consequently, Pearson correlation or the covariance matrix is used to determine eigenvalues and eigenvectors. Having a completely different correlation metric that captures kinetic properties of two random vectors against one another has also been used in creating a modified version of original algorithm with this new correlation matrix.

Hence, we implemented new correlation metrics, and the new idea was to modify the original PCA for obtaining eigenvectors and eigenvalues for dimensionality reduction using a correlation matrix with our kinetic correlation coefficient.

III. Implementing Modified PCA

In this work, a new correlation coefficient method called Octave has been introduced. The correlation is used as a method for feature selection (calculated between two features) using Kinetic Energy. The new Octave correlation makes a useful contribution as it provides a new measure of dependence between random vectors that capture non-linear relationships as well.

The modified version of PCA was assessed using a data set of air passenger numbers, from where the features of the modified PCA were derived, using kinetic correlation metrics instead of Pearson correlation coefficient based on kinetic correlation theory.

A. Implementation Setup

This function returns information coefficient IC for two random variables defined as the dot product of probabilities corresponding to each class:

```
def  ic(vector1,vector2):
   a=vector1
  b=vector2
prob1=np.unique(a,return_counts=True)[1]/a.shape[0]
   prob2=np.unique(b,return_counts=True)[1]/b.shape[0]
   p1=list(prob1)
   p2=list(prob2)
   diff=len(p1)-len(p2)
   if diff>0:
      for elem in range(diff):
         p2.append(0)
   if diff<0:
      for  elem in range((diff*-1)):
         p1.append(0)
   ic=np.dot(np.array(p1),np.array(p2))
   return ic
```

And, having functions for kinetic energy of a vector and for information correlation, we can define a new function that computes kinetic correlation. This function will return correlation based on kinetic energy as illustrated below:

```
def  o(vector1,vector2):
     i_c=ic(vector1,vector2)
   o=i_c/np.sqrt(kin_energy(vector1)*kin_energy(vector2))
   return o
```

The formula is updated such that the denominator contains sqrt in order to have probabilities bounded between 0 and 1.

SHAPE will return the number of items in the numpy array in the form of a tuple, then creates a matrix with the number of rows initialized with zero values.

```
rows=data.shape[1]
rows
matrix= np.zeros((rows,rows))
```

Then the correlation matrix is created with the function o() that was defined previously, as shown in Table 1. The correlation matrix obtained by the Pearson method is also listed in Table 2 for comparison.

TABLE I. CORRELATION MATRIX WITH THE FUNCTION O()

	0	**1**	**2**	**3**	**4**	**5**
0	1.000	1.000	0.974	0.326	0.184	0.229
1	1.000	1.000	0.974	0.326	0.184	0.229
2	0.974	0.974	1.000	0.320	0.180	0.223
3	0.326	0.326	0.320	1.000	0.071	0.131
4	0.184	0.184	0.180	0.070	1.000	0.490
5	0.229	0.229	0.223	0.131	0.490	1.000

TABLE II. CORRELATION MATRIX ON BASIS OF 'PEARSON R' MODEL

	0	**1**	**2**	**3**	**4**	**5**
0	1.000	0.751	0.770	-0.041	-0.027	0.000
1	0.751	1.000	0.959	-0.013	-0.031	0.000
2	0.770	0.959	1.000	-0.020	-0.023	0.000
3	-0.041	-0.013	-0.020	1.000	0.216	0.000
4	-0.027	-0.031	-0.023	0.216	1.000	0.000
5	-0.000	-0.000	-0.000	0.000	0.000	1.000

B. Comparison of modified PCA with Kinetic Correlation Matrix from Kinetic Energy and PCA with Pearson R correlation

Our contribution is based on changing the correlation matrix that uses Pearson R correlation or, in some cases, the covariance, with a correlation matrix based on the Onicescu correlation coefficient. The results of testing the kinetic correlation of our data sets using the Pearson coefficient are shown in Fig. 1 and 2.

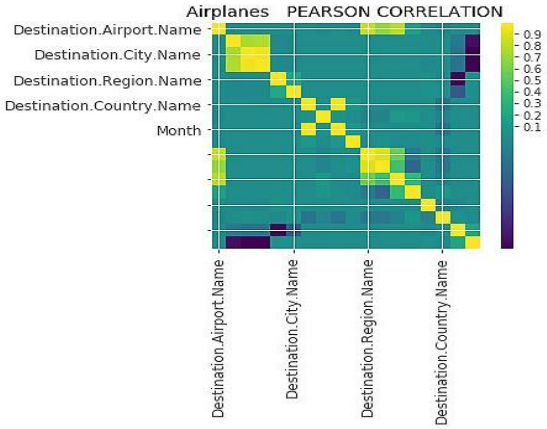

Fig. 1. Air passenger numbers data with Pearson Correlation.

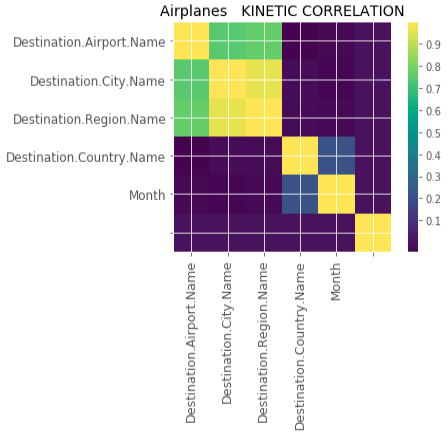

Fig. 2. A train passenger numbers data with Kinetic Correlation.

As expected the kinetic correlation has a much higher kinetic correlation matrix from kinetic energy than the Pearson one. Pearson's R is able to detect only linear relations in data. The graphs have the same list of seven columns on both x and y axis. The colouring of each particular square shows the actual correlation between the columns on the scale of 0 to 1.0. So, if the color is dark, there is low correlation and vice-versa.

C. Features Obtained from Kinetic Energy PCA Components

In this section, we implemented XGBoost [6]. This is an algorithm that has recently been dominating applied machine learning for structured or tabular data and it is designed for speed and performance. It has been applied here to a training data set of passenger numbers with a dataset of 51,983 observations with 9 variables. In order to get a better estimate of model performance, we used a variant of the famous 1-fold cross validation. We split dataset into a training set (75% of the data) and a test set (25% of the data) randomly for 1 different time and measure accuracy, false positive rate and false negative rate.

The XGBoost model was run within Python machine learning modules and the calculated mean values (Fig. 3) are very much nearer to the actual values of one.xgb.train, which is an advanced interface for training an XGBoost model.

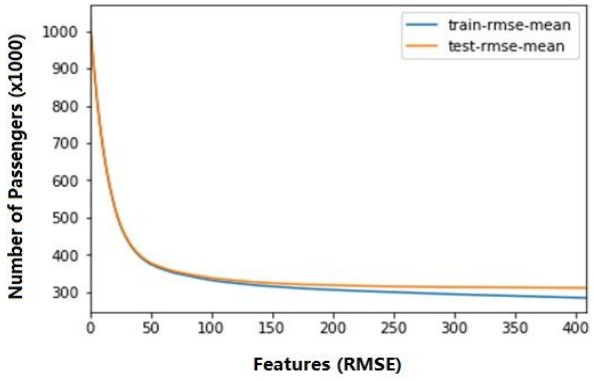

Fig. 3. The mean values of features obtained from Kinetic Energy PCA Components.

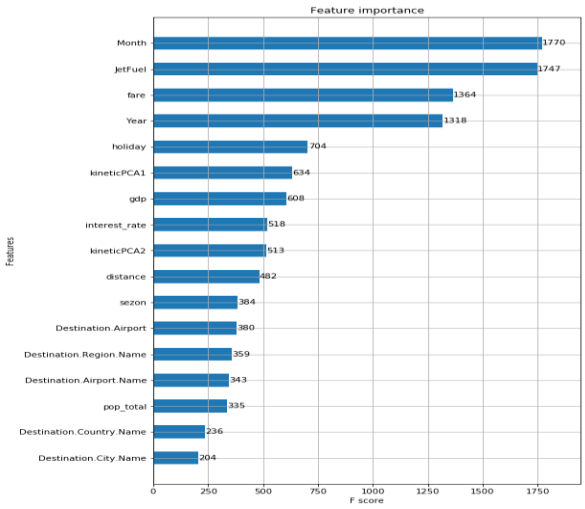

Fig. 4. Principal component analysis features (KineticPCA1 and KineticPCA2).

TABLE III. PREDICTION MODEL USING KINETICPCA1 AND KINETIC PCA2

	passangersPred5
0	515
1	636
2	621
3	624
4	607

Fig. 4 shows the features for predicting the number of passengers from most important to least important. Here it shows that JetFuel, Month, and fare are the most predictive values. As is noted in the plot below the features obtained from the modified PCA, called kineticPCA1 and kineticPCA2, are captured with reasonable influence after running the XGBOOST model and inspecting feature importance. The number of passenger predictions using the Prediction model is given in Table 3.

D. Features Obtained from Deep Learning Hidden Layers

In this step, we created a different engineered dataset in order to have diversity in multiple datasets. We have chosen at this step to add non-linear features that were extracted from an R implementation of a Deep Learning model.

We trained a deep learning neural network with 100 neurons in first hidden layer, 63 neurons in second hidden layer and 30 neurons in the third hidden layer and 15 neurons in last hidden layer. The number of features extracted from the deep learning model was the same number of neurons in each hidden layer. For a better selection of only important non-linear features, we computed correlations of each feature that was corresponding to each neuron in the hidden layer with our target variable. During the computing the correlations, we kept only one feature from each neuron, where is the maximum correlation compared with other features in the same hidden layer, and obtained final four non-linear features. An XGBOOST model was then run to see the behavior of that particular model on the newly created data set.

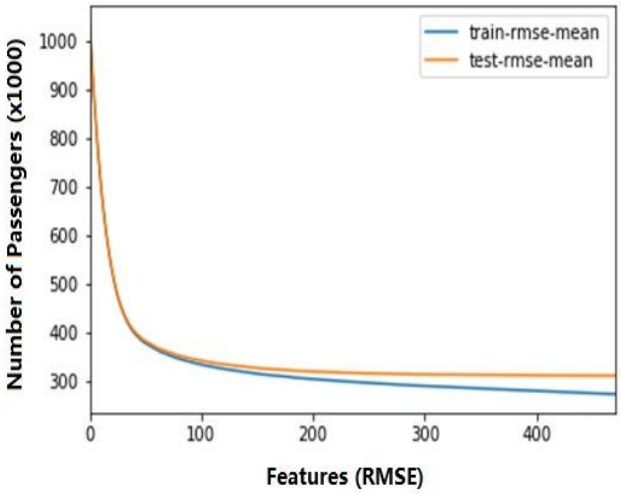

Fig. 5. The mean values of features obtained from deep learning hidden layers.

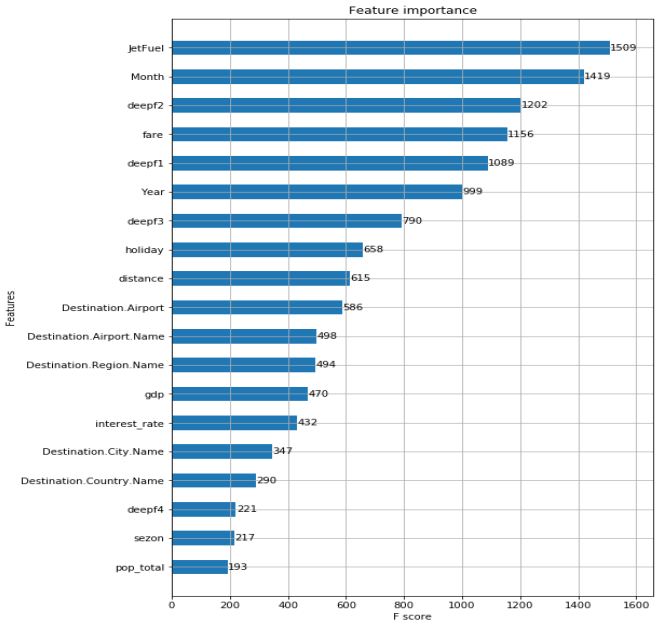

Fig. 6. Prediction model using deep learning hidden layers.

TABLE IV. PREDICTION MODEL USING DEEP LEARNING HIDDEN LAYERS

	passangersPred4
0	582
1	545
2	546
3	552
4	627

The plot in Fig. 5 shows that the mean values calculated are much nearer to one but differed more on the last set of inputs. Fig. 6 shows the features that are important for the number of passengers predicted from most important to least important. Here it shows JetFuel, Month, and fares have the highest predictive values. As observed from the plot the nonlinear features deepf1, deepf2, deep3, and deepf4 (obtained by the method described above) are very influential and are the ones with the highest influential impact captured by XGBOOST feature importance. The number of passengers predicted by using the Deep Learning Hidden Layers is given in Table 4.

E. Features Obtained from Genetic Algorithm

This feature was extracted from a genetic algorithm called *symbolic transformer*, which is an estimator that begins by building a population of naive random formulas to represent a relationship [7]. The formulas are represented as tree-like structures with mathematical functions being recursively applied to variables and constants. Each successive generation of programs is then evolved from the one that came before it by selecting the fittest individuals from the population to undergo genetic operations such as crossover, mutation or reproduction. The results are presented in Fig. 7.

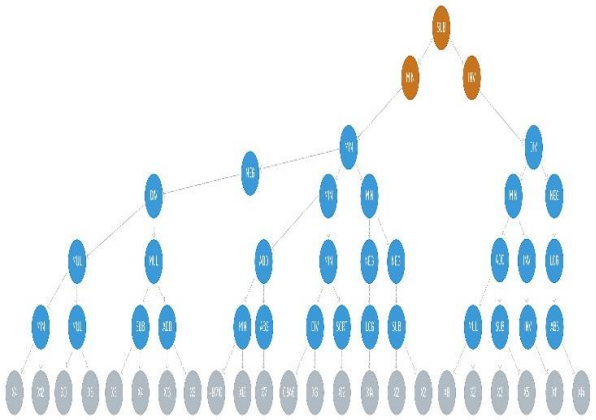

Fig. 7. Tree-like structures of the Genetic Algorithm.

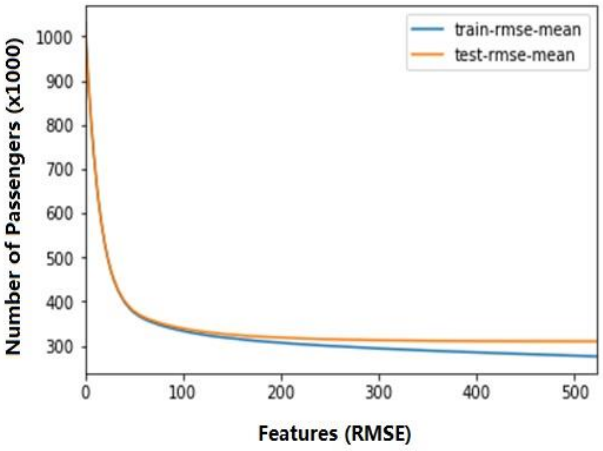

Fig. 8. The mean values of features obtained from Genetic Algorithm.

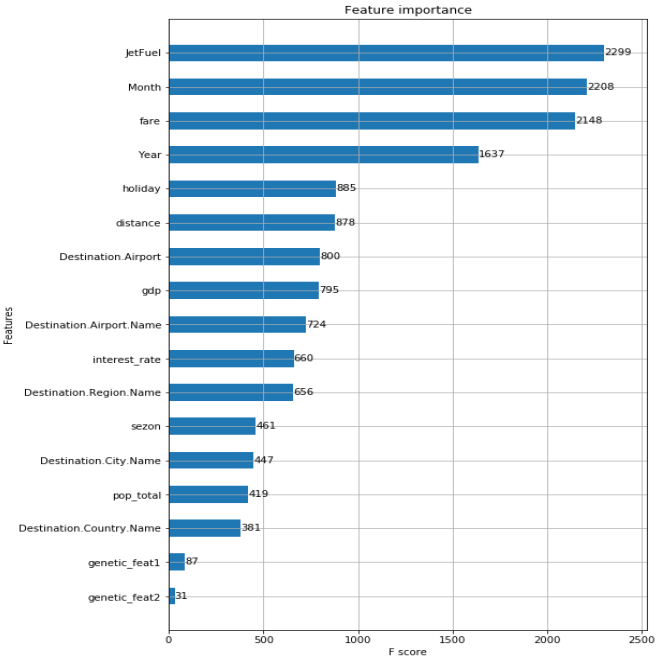

Fig. 9. Features importance obtained from Genetic Algorithm.

TABLE V. PREDICTION MODEL USING GENETIC ALGORITHM

	passangersPred2
0	529
1	611
2	600
3	609
4	607

In the genetic program, it is easy to observe different kinds of operations that the genetic algorithm produced. Two new features obtained from genetic transformer after running an XGBOOST model have been added into this algorithm. Fig. 8 shows that the mean values calculated are very near one, but differed more on last set of inputs. Fig. 9 shows that the features that are important for the number of passengers predicted from most important to least important. Here it shows that JetFuel, Month, and fare are the most predictive values. From the plot below the genetic features called genetic_feat1 and genetic_feat2, where captured with very small influence in contrast with our expectation when conducting the experiment. The number of passengers predicted by using the Deep Learning Hidden Layers is given in Table 5.

IV. CONCLUSION

In this work, a new modified version of PCA with kinetic correlation matrix using kinetic energy is presented. The features of this modified PCA have been assessed with different sets of air passenger data and compared to traditional PCA. The results of the modified version of PCA show that the kinetic correlation is much higher than that of the Pearson one, which makes lot of sense since Pearson's R is able to detect only linear relations in data. It turned out that the modified version of PCA is more effective in data compression, classes reparability and classification accuracy than those form traditional PCA.

Based on these results, the modified PCA can be applied to make clustering in hyper-dimensional space using kinetic correlation as a distance (increase performance) to make it run in real time in a future work. When coping with clustering, such as clustering algorithm, clustering K-means or in hierarchical clustering, it requires a for-loop at every point to get the nearest point from row vector. For *n* rows of data complexity will be of the order *n*^*n*, which is impossible to finish using this method. In two-dimensional space, there is a trick to fast implementation using divide and conquer, which has complexity *n* or log *n*. However, these problems can be solved by using modified PCA with properly added features.

In this work, only limited features of the modified PCA method were studied with one set of data. To fully understand and investigate the features of modified PCA, large subsets of data with more features should be considered.

ACKNOWLEDGMENT

I would like to address my special acknowledgements to all those people who provide me with data for my experiments. My warm appreciation is due to the Public Authority for Civil Aviation, Directorate General of Meteorology, and Ministry of Tourism in Oman.

REFERENCES

[1] Bengio, Y. (2013). Representation learning: a review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798-1828

[2] Timmerman, M. E. (2003). Principal component analysis (2nd Ed.). I. T. Jolliffe. Journal of the American Statistical Association, 98, 1082-108

[3] F. M. Palechor et al. (2017), "Cardiovascular Disease Analysis Using Supervised and Unsupervised Data Mining Techniques", Journal of Software, vol. 12, no.2, pp. 81-90

[4] Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9 (2579-2605), 85.

[5] Coates, A., & Ng, A. Y. (2012). Learning feature representations with k-means *Neural Networks:*

[6] Chen T. Q. and Guestrin C. (2016). XGBoost: A Scalable Tree Boosting System, KDD'16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages 785-794

[7] Lowe, D. G. (1999). *Object recognition from local scale-invariant features.* Paper presented at the The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.