**ACCURACY ENHANCEMENT FOR MEDICAL DATA WITH THE IMPLEMENTATION OF ML ALGORITHMS**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology/Master of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

**Nagandla Avinash Pamulapati Jagadeesh Shaik Faiyajuddin**

**(AP20110010066) (AP20110010386) (AP20110010363)**

**A picture containing text

Description automatically generated**

Under the Guidance of

Prof. / Dr. pradyut kumar sanki sir

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**[December, 2022]**

# Certificate

Date: 13-Dec-22

This is to certify that the work present in this Project entitled “**Accuracy enhancement for the medical data with the implementation of ml algorithms**” has been carried out by **Nagandla Avinash , Pamulapati Jagadeesh, Shaik Faiyajuddin** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

**Supervisor**

****

Prof. / Dr. pradyut kumar sanki sir

Department of ECE, SRM University -AP

Andhra Pradesh.

# Acknowledgements

We thank pradyut kumar sanki sir, our professor in charge, for help and direction in finishing our project on this subject. This work has given us an opportunity to explore new topics which we are curious about. Your insightful counsel and recommendations were quite beneficial to us as we finished the project. we will be always grateful to you for this.

# Table of Contents

[Certificate i](#_Toc121866214)

[Acknowledgements iii](#_Toc121866215)

[Table of Contents v](#_Toc121866216)

[Abstract vii](#_Toc121866217)

[Abbreviations ix](#_Toc121866218)

[1. Introduction 1](#_Toc121866219)

[1.1 Terminologies related to the regression analysis 1](#_Toc121866220)

[1.1.1 Why do we use Regression analysis 3](#_Toc121866221)

[2. Methodology 5](#_Toc121866222)

[2.1 PCA 5](#_Toc121866224)

[2.1.1 PLSR 11](#_Toc121866225)

[PLSR Implementation 13](#_Toc121866226)

[3. Discussion 27](#_Toc121866227)

[4. Concluding Remarks 33](#_Toc121866228)

[5. Future Work 35](#_Toc121866229)

[References 37](#_Toc121866230)

# Abstract

The objective of a regression model in machine learning is to build a mathematical equation that defines the outcome variable (y) as a function of one or multiple predictor variables (x). later, this equation can be used to predict the y based on new values of predictor x. The main concepts considering in this ml regression analysis as a statistical process consisting of a set of ml methods. Here, regression analysis refers to the method of studying the relationship between independent variable and dependent variable. This paper discusses various works on various regressions and compares their performance using to optimize the best outcome and precision. In order to determine a regression model efficiency, it must be correlated with the actual values obtained for the illustrative variables.

# Abbreviations

PCA Principal Component Analysis

PLSR Partial Least Squares Regression

GPR Gaussian Process Regression

# Introduction

Regression is a technique for explore the relationship between independent variable or features and a dependent variable or outcome. Its used as a method for predictive modelling in machine learning in which algorithm is used to predict continuous outcomes.

Suppose there is a company A (marketing company), who does various advertisement every year and get sales on that. Now, the company wants to do the advertisement in the future years and wants to know the prediction about sales for this year. To solve this type of prediction problem in ml, we need regression analysis.

## Terminologies related to the regression analysis

* **Dependent Variable:** The major factor in the regressionanalysis that we want to anticipate or understand is called the dependent variable. It is also called destination variable.
* **Independent Variable:** Factors that affect dependent variables or that are used in predicting the values of dependent variables are called independent variables, a., also called as a predictor.
* **Outliers:** The outlier is a finding that contains either a very low value or a very high value in comparison to other observed values. An outlier may hamper the result, so it should be avoided.
* **Multicollinearity:** If the independent variables are closely related to each other, then this condition can arise. It causes problems when trying to rank which variable is most influential in a dataset. Collinearity should not be present in the data set, as it would cause undue confusion and instability during analysis.
* **Underfitting and Overfitting:** If our algorithm performs well on the training data set but not so well on the test data set, this is known as overfitting. If our algorithm does not perform at all in either case, this is called underfitting.

### Why do we use Regression analysis

Regression analysis helps in the prediction of a continuous variable. There are various scenarios in the real world where we need some future predictions such as medical conditions, weather condition, sales prediction, marketing trends, etc., for such cases we need some methods which can make predictions more perfectly or accurately. for such case we need Regression analysis which is a statistical method and used in machine learning and data science.

# Methodology

There are many types of regressions in machine learning. But In this we want to discuss about some types of regressions in machine learning.

* PCA (Principal Component Regression)
* PLSR (Partial Least Squares Regression)
* GPR (Gaussian Process Regression)



## PCA

Principal Component Analysis (PCA) is one of the most commonly used un-supervised machine learning algorithms across a variety of applications like : exploratory data analysis, dimensionality reduction, information compression and more. Reducing the number of variables of a data set naturally comes at the expense of accuracy. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without external variables to process. The main idea of PCA is to reduce the number of variables of a data set, while preserving more information as much as possible.

Step by step explanation of PCA:

Step-1:- First we have to do standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis. Mathematically, this can be done by subtracting the mean and dividing by the standard deviation for each value of each variable or by calculating the mean for the data sets.

Step-2:- After the standardization is done successfully. Now, all the values/variables will be transformed to the same scale. In this step we will do covariance matrix computation. Why we want to compute? What is the use of these computation? Let, discuss how it helps. Sometimes variables are highly correlated in such a way that they contain redundant information. So in order to identify these correlations, we compute the covariance matrix. The covariance matrix is a m \* m symmetric matrix( where m is the number of dimensions). The number of covariance values are 2 power n(2^n). since the covariance of a variable with itself is its variance

(cov(a,a)=var(a)) and so on. W.K.T the covariance is not more than a table that summarizes the correlations between all the possible pairs of variables.

Step-3:- In this step we compute the eigen vectors and eigen values of the covariance matrix to identify the principal components. Eigen vectors and eigen values are the linear algebra concepts based. Here, the principal components are new variables that are constructed as linear combination or mixtures of the initial variables. These combinations are done in such a way that the new variables are uncorrelated and mist of the information within the initial variables is squeezed or compressed into the first components. PCA tries to put maximum possible information in the first component. Let see about why we want know about eigen vectors and eigen values. In this concept we know that every eigen vector has an eigen value and their number is equal to the number of dimensions of the data. For example, we have a 2-dimensional data set and in this data set we can see 2 variables, 3 eigen vectors with corresponding 2 eigen values. By ranking your eigen vectors in order of their eigen values from highest to lowest we get the principal components in order of significance. For example if we got 2nd eigen value is greater than 1st eigen value then we take the big value for next step calculation.

Step-4:- In this step we will compute the values into normalized vectors or feature vectors. By computing the normalized vectors we get a n\*m matrix( here m=1) with the normalized vector values we form a equation with the help of variables. i.e., x(position 1) + y(position 2)+….. here, x, y,… values are normalized vector values.

The main idea we understand that PCA simplifies the complexity in high-dimensional data while retaining treads and patterns. It improves the performance of ML algorithm as it eliminates correlated variables that don’t contribute in any decision making.

PCA Table:-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL.No** | **Blood pressure** | **Cholesterol(x)** | **Age(y)** | **Prediction values** |
| 1 | 120 | 126 | 38 | 131.60 |
| 2 | 125 | 128 | 40 | 134.103 |
| 3 | 130 | 135 | 42 | 141.382 |
| 4 | 121 | 140 | 42 | 146.159 |
| 5 | 135 | 130 | 44 | 137.200 |
| 6 | 140 | 145 | 46 | 152.117 |

Number of features =2 (x,y)

Number of samples =6

By following above steps we can solve it in mathematical modelling ,

Step -1: mean of x = sum of samples/number of samples.= 134

mean of y= sum of samples/number of samples.= 42

Step -2: covariance of (x,x) = Cov(x,x) = Σ ((xi – x) ^2 / (N – 1) =54.8

covariance of (x,y) = Cov(x,y) = Σ ((xi – x) \* (yi –y ) / (N – 1)=16

covariance of (y,x) = covariance of (x,y)=16

covariance of (y,y) = Cov(y,y) = Σ ((yi – y) ^2 / (N – 1)=8

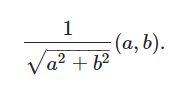
now, covariance matrix =

step –3: eigen value = det(s-λI) = 0

then λ1=59.7471339 and λ2=3.052860105

eigen vector of λ1 =

step -4: normalized eigenvector



Step -5: deriving the new data, Pci= 0.9553374346(xi) – 0.2953975(yi)

We get prediction values for pc1, pc2, pc3, pc4, pc5, pc6 are 131.60, 134.103, 141.382, 146.159, 137.200,152.117.

### PLSR

PLSR stands for partial least squares. It is a multivariate linear regression technique that is used to analyze the relationship between a response variable and a limited number (e.g., 1-2) of predictors. The technique finds predictors with maximum correlation to the response variable in directions that are orthogonal to each other (i.e. a "non response" variable). The technique is easily implemented, but its success depends on the selection of appropriate parameters.

Step by step explanation of PLSR:

* + 1. Preprocess the data by removing missing values, scaling the data so that all variables are on the same scale, and removing outliers.
    2. Construct the X and Y matrices, where X is the matrix of predictor variables and Y is the matrix of response variables.
    3. Compute the singular value decomposition (SVD) of the X matrix to identify the underlying factors that explain the relationships between the predictor and response variables.
    4. Choose the number of components to use in the PLSR analysis. This is typically done by selecting the top k components that explain the most variance in the data, where k is the number of dimensions you want to reduce the data to.
    5. Compute the weights and biases for the PLSR model using the SVD results and the X and Y matrices.
    6. Use the PLSR model to make predictions of the response variable based on the predictor variables. This is done by transforming the predictor variables into the new coordinate system defined by the components, and then using the equations for the new coordinate system to compute the predicted response variable.
    7. Making predictions: Finally, the PLSR model can be used to make predictions of the response variable based on the predictor variables.

### PLSR Implementation

#import required libraries

from sys import stdout

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from scipy.signal import savgol\_filter

from sklearn.cross\_decomposition import PLSRegression

from sklearn.model\_selection import cross\_val\_predict

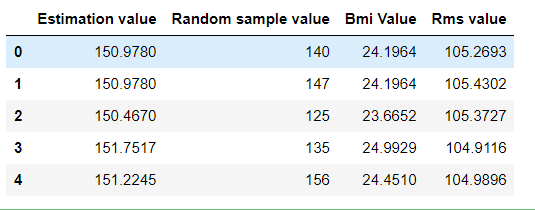
from sklearn.metrics import mean\_squared\_error, r2\_score

#import dataset

df = pd.read\_excel("Simplified All samples.xlsx")

#first 5 rows of the dataframe

df.head()



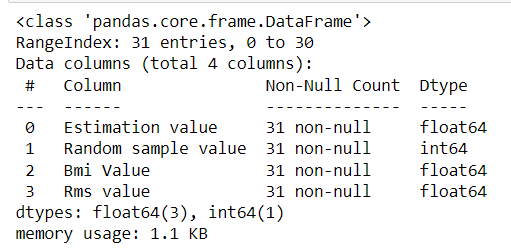
#Number of rows and Columns

df.shape



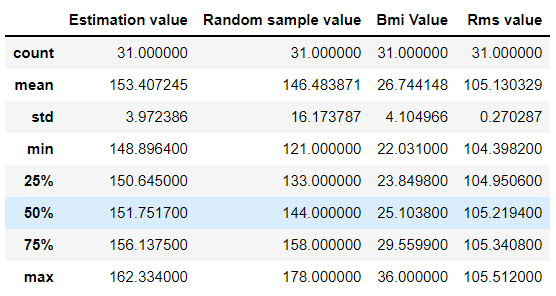
#To see if there are any null values

df.info()



#Description of the data

df.describe()



#splitting data into train & test

y= df.iloc[ : , :-3].values

X= df.iloc[ : , 2:4].values

#Number of rows and columns of testing data

X.shape



#Number of rows and columns of training data

y.shape



# split data

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.33, random\_state=42)

# create PLSRegression model

from sklearn.cross\_decomposition import PLSRegression

from sklearn.model\_selection import cross\_val\_predict

from sklearn.metrics import mean\_squared\_error, r2\_score

pls = PLSRegression(n\_components=2)

# fit the model

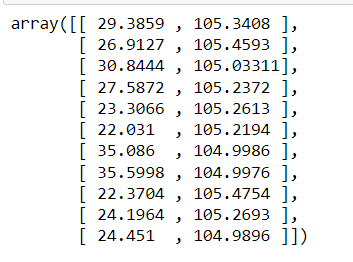
pls.fit(X\_train, y\_train)

# Make predictions using the test data

y\_pred = pls.predict(X\_test)

#Training set

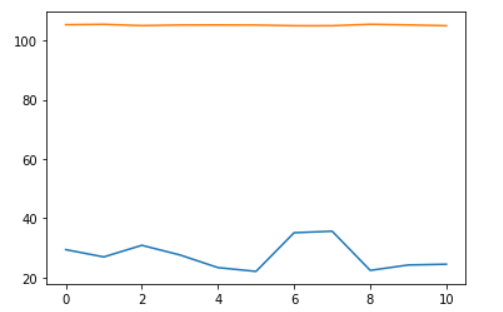
X\_test



#Ploting training set

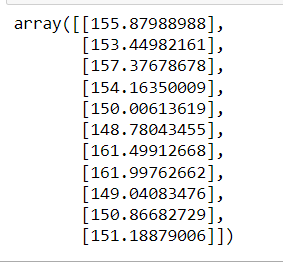
plt.plot(X\_test)

plt.show()



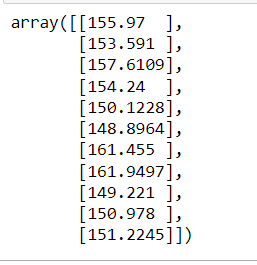
#Predicted values

y\_pred



#Estimated Values

y\_test



# Calculate the mean squared error

from sklearn.metrics import mean\_squared\_error

mse = mean\_squared\_error(y\_test, y\_pred)

# Print the mean squared error

print(mse)

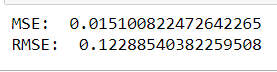


from sklearn.metrics import mean\_squared\_error

import math

print('MSE: ',mean\_squared\_error(y\_test, y\_pred))

print('RMSE: ',math.sqrt(mean\_squared\_error(y\_test, y\_pred)))



# R\_Square and Adjusted R Square

import statsmodels.api as sm

X\_addC = sm.add\_constant(X)

result = sm.OLS(y, X\_addC).fit()

print(result.rsquared, result.rsquared\_adj)



#Accuracy

score = pls.score(X\_test,y\_test)

print(score\*100,'%')

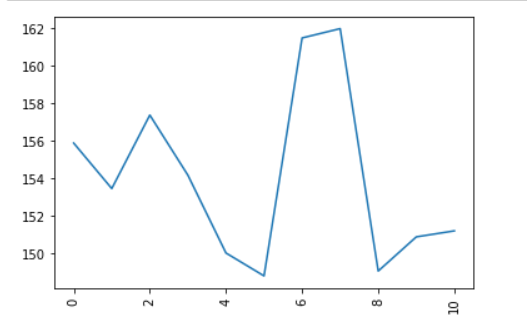


# plot the predicted Values

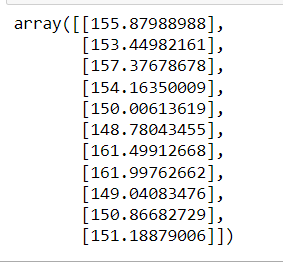
plt.plot(y\_pred)

plt.xticks(rotation=90)

plt.show()



**output:**



# Discussion

Data : (PLSR)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Estimation value | Random sample value | Bmi Value | Rms value | |
| 150.978 | 140 | 24.1964 | 105.2693 |  |
| 150.978 | 147 | 24.1964 | 105.4302 |  |
| 150.467 | 125 | 23.6652 | 105.3727 |  |
| 151.7517 | 135 | 24.9929 | 104.9116 |  |
| 151.2245 | 156 | 24.451 | 104.9896 |  |
| 149.1619 | 132 | 22.3127 | 104.6437 |  |
| 151.1212 | 166 | 24.3417 | 104.3982 |  |
| 149.7927 | 140 | 22.8928 | 104.8567 |  |
| 150.1228 | 134 | 23.3066 | 105.2613 |  |
| 148.8964 | 141 | 22.031 | 105.2194 |  |
| 149.071 | 159 | 22.2129 | 105.1068 |  |
| 149.8569 | 124 | 23.03 | 105.1945 |  |
| 149.221 | 121 | 22.3704 | 105.4754 |  |
| 150.823 | 178 | 24.0344 | 105.1527 |  |
| 150.824 | 122 | 24.0344 | 104.8665 |  |
| 153.591 | 157 | 26.9127 | 105.4593 |  |
| 156.347 | 166 | 29.7777 | 105.2576 |  |
| 154.24 | 157 | 27.5872 | 105.2372 |  |
| 150.85 | 146 | 25.1038 | 105.512 |  |
| 158.234 | 166 | 31.7397 | 105.3914 |  |
| 153.256 | 155 | 26.564 | 105.3667 |  |
| 156.305 | 132 | 29.7339 | 105.2643 |  |
| 162.334 | 154 | 36 | 104.7911 |  |
| 157.6109 | 144 | 30.8444 | 105.0331 |  |
| 161.9497 | 166 | 35.5998 | 104.9976 |  |
| 152.3823 | 167 | 25.6543 | 104.8975 |  |
| 157.7327 | 143 | 31.2174 | 105.2573 |  |
| 155.97 | 128 | 29.3859 | 105.3408 |  |
| 153.1069 | 170 | 26.4071 | 104.7463 |  |
| 161.455 | 142 | 35.086 | 104.9986 |  |
| 155.97 | 128 | 29.3859 | 105.3408 |  |

**Solving the data mathematically using PLSR algorithm:**

By using mathematical steps of PLSR, we calculate the above data table,

Number of features =2 (x,y)

Number of samples =31

By following above steps we can solve it in mathematical modelling ,

Step -1: mean of x = sum of samples/number of samples.= 105.13

mean of y= sum of samples/number of samples.= 26.74

Step -2: covariance of (x,x) = Cov(x,x) = Σ ((xi – x) ^2 / (N – 1) =0.07

covariance of (x,y) = Cov(x,y) = Σ ((xi – x) \* (yi –y ) / (N – 1)=-0.228

covariance of (y,x) = covariance of (x,y)= -0.228

covariance of (y,y) = Cov(y,y) = Σ ((yi – y) ^2 / (N – 1)= 16.85

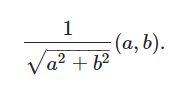
now, covariance matrix =

step –3: eigen value = det(s-λI) = 0

then λ1=16.78 and λ2=0.067

eigen vector of λ1 =

step -4: normalized eigenvector



Step -5: y= a+bx

a=y-bx = 132.2274

b=r. (sy/sx) = -1.0034

r=/ = -0.066

Sx= = 0.270

Sy= = 4.105

c = -0.0136

d = -0.99

Step- 6: Predicting the data

Prediction = a – b (( c \* xi ) – ( d – yi ))

By solving the above equation by imputing the xi and yi test data from the data frame we will get the predicted values like 150.978

150.978, 150.467, 151.7517, 151.2245, 149.1619, 151.1212, 149.7927, 150.1228, 148.8964, 149.071, 149.8569, 149.221, 150.823, 150.824, 153.591, 156.347, 154.24, 150.85, 158.234, 153.256, 156.305, 162.334, 157.6109, 161.9497, 152.3823, 157.7327, 155.97, 153.1069, 161.455, 155.97

# Concluding Remarks

PCA attempts to explain the variance-covariance structure of a data set. The goal is to increase the variance of the features themselves so that information loss is greatly reduced. PCA is a Dimensionality Reduction algorithm. PLSR and PCA are both used for dimension reduction.

In plsr To maximise inter-class variance, use partial least squares with the annotated label. The principal components are orthogonal in pairs. The primary components are designed to maximise correlation.

The main distinction is that PCA is an unsupervised method, whereas PLSR is a supervised method. PLSR predicts more accurately than PCA. Therefore PLSR Is better than PCA by seeing predictions values which were solved in mathematical way for the sample data and original data.

# Future Work

GAUSSIAN REGRESSION:

In future we will implement and solve gaussian regression process which is also another type of regression technique in machine learning. We have already researched and worked on it a little. So, we will implement the gaussian regression technique in the coming future.

# References

1. <https://www.researchgate.net/publication/316652806_Principal_Component_Analysis>
2. <https://pubmed.ncbi.nlm.nih.gov/12758135/>
3. <https://towardsdatascience.com/principal-component-regression-clearly-explained-and-implemented-608471530a2f>
4. <https://en.wikipedia.org/wiki/Principal_component_regression>
5. <https://www.sciencedirect.com/topics/nursing-and-health-professions/partial-least-squares-regression>
6. <https://en.wikipedia.org/wiki/Partial_least_squares_regression>
7. <https://towardsdatascience.com/intro-to-gaussian-process-regression-14f7c647d74d>
8. <https://www.educba.com/regression-in-machine-learning/>