# Pune Vidyarthi Griha’s College of Engineering and Technology & G.K. Pate (Wani) Institute of Management, Pune- 411009.

**(Affiliated to Savitribai Phule Pune University)**



**A Project Based Learning II Report**

**On**

**“Heart Disease Classification”**

**By**

**Student’s Name Seat Numbers**

**1. Kasturi Prasanna Bhandare F190070044**

**2. Mahak Sanjay Gautam F190070135**

**3. Pranali Ramesh More F190070283**

**4. Shivam Avinash Kandhare F190070193**

**Under The Guidance of-**

**Pranav Jaipurkar**

**(M.E. Computer Engineering)**

**Department of Computer Engineering**

**Academic Year: - 2020-2021**

**Pune Vidyarthi Griha’s College of Engineering and Technology & G.K. Pate (Wani) Institute of Management, Pune- 411009.**

**(Affiliated to Savitribai Phule Pune University)**



***Table of content***

|  |  |
| --- | --- |
| **Content** | **Page no.** |
| Table of figures | 1. | |
| Acknowledgement | 2. | |
| Abstract | 2. | |
| Chapter 1: Introduction  1.1 Problem Definition  1.2 Motivation  1.3 Objectives | 3.  3.  4.  4. | |
| Chapter 2: Datasets | 5. | |
| Chapter 3: Methods and Algorithms Used  3.1 Logistic Regression  3.2 Support Vector Machines (SVM)  3.3 K-nearest neighbor (K-NN)  3.4 SVM with PCA  3.5 K-NN with PCA | 5.  6.  6.  7.  8.  9. | |
| Chapter 4: Experiments   1. Logistic Regression 2. Support Vector Machines (SVM) 3. K-nearest neighbor (K-NN) 4. SVM with PCA 5. K-NN with PCA | 9.  10.  11.  12.  14.  15. | |
| Chapter 5: Discussion on result | 16. | | |
| Chapter 6: Conclusion | 17. | | |

***Table of figures***

|  |  |
| --- | --- |
| **Name of figure** | **Page no** |
| **Fig 1:** Original dataset | 5. |
| **Fig 2:**  Logistic Regression:  Training Confusion matrix | 10. |
| **Fig 3**: Logistic Regression:  Testing Confusion matrix | 10. |
| **Fig 4:** SVM: Training Confusion matrix | 11. |
| **Fig 5:** SVM: Testing Confusion matrix | 12. |
| **Fig 6**: K-NN: Training Confusion matrix | 12. |
| **Fig 7:**  K-NN: Testing Confusion matrix | 13. |
| **Fig 8**: SVM with PCA:  Training Confusion matrix | 14. |
| **Fig 9:** SVM with PCA:  Testing Confusion matrix | 14. |
| **Fig 10**: K-NN with PCA:  Training Confusion matrix | 15. |
| **Fig 11**: K-NN with PCA:  Testing Confusion matrix | 16. |

**1.**

***Acknowledgement***

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. I am very grateful to my project supervisor Pranav Jaipurkar Sir, for the guidance, inspiration and constructive suggestions that helpful me in the preparation of this project.

***Abstract***

Our project objective is to detect whether patients have heart disease or not by given a number of features from patients. The motivation of our project is to save human resources in medical centers and improve accuracy of diagnosis. In our project we use different methods to detect heart disease such as SVM (Support Vector Machines), K-NN (K-nearest neighbor), SVM with PCA (SVM with Principal component analysis) and K-NN with PCA.

And among all these models SVM gave us the best accuracy of **94.50%**.

**2**.

***CHAPTER 1: INTRODUCTION***

According to the World Health Organization, every year 12 million deaths occur worldwide due to heart disease. The load of cardiovascular disease is rapidly increasing all over the world from the past few years. Many researches have been conducted in attempt to pinpoint the most influential factors of heart disease as well as accurately predict the overall risk. Heart Disease is even highlighted as a silent killer which leads to the death of the person without obvious symptoms. The early diagnosis of heart disease plays a vital role in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications. This project aims to predict future heart disease by analyzing data of patients which classifies whether they have heart disease or not using machine-learning algorithms.

**1.1 Problem Definition**

The major challenge in heart disease is its detection. There are instruments available which can predict heart disease but either they are expensive or are not efficient to calculate chance of heart disease in human. Early detection of cardiac diseases can decrease the mortality rate and overall complications. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more sapience, time and expertise. Since we have a good amount of data in today’s world, we can use various machine learning algorithms to analyze the data for hidden patterns. The hidden patterns can be used for health diagnosis in medicinal data.

**3.**

**1.2 Motivation**

Machine learning techniques have been around us and has been compared and used for analysis for many kinds of data science applications. The major motivation behind this research-based project was to explore the feature selection methods, data preparation and processing behind the training models in the machine learning. With first hand models and libraries, the challenge we face today is data where beside their abundance, and our cooked models, the accuracy we see during training, testing and actual validation has a higher variance. Hence this project is carried out with the motivation to explore behind the models, and further implement Logistic Regression 2 model to train the obtained data. Furthermore, as the whole machine learning is motivated to develop an appropriate computer-based system and decision support that can aid to early detection of heart disease, in this project we have developed a model which classifies if patient will have heart disease in ten years or not based on various features (i.e., potential risk factors that can cause heart disease) using logistic regression. Hence, the early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine.

**1.3 Objectives**

The main objective of developing this project is:

1. To develop machine learning model to predict future possibility of heart disease by implementing Logistic Regression.

2. To determine significant risk factors based on medical dataset which may lead to heart disease.

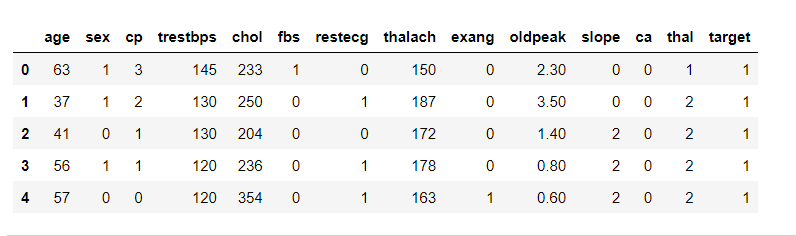
3. To analyze feature selection methods and understand their working principle.

**4.**

**CHAPTER 2: DATASETS**

The dataset was provided to us by our mentor. It provides patient information which includes 303 records and 14 attributes. The attributes include: age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, exercise induced angina, ST depression induced by exercise relative to rest, slope of the peak exercise ST regiment, number of major vessels Thal, and target ranging from 0 to 1, where 0 is absence of heart disease. The data set is in csv (Comma Separated Value) format which is further prepared to data frame as supported by panda's library in python.

***Fig 1: Original dataset***



The education data is irrelevant to the heart disease of an individual, so it is dropped. Further with this dataset pre-processing and experiments are then carried out.

**CHAPTER 3: METHODS AND ALGORITHMS USED**

The main purpose of designing this system is to predict the ten-year risk of future heart disease. We have used Logistic regression as a machine-learning algorithm to train our system and various feature selection algorithms like SVM, K-NN, SVM with PCA and K-NN with PCA. These algorithms are discussed below in detail.

**5.**

**3.1 Logistic Regression**

Logistic Regression is a supervised classification algorithm. It is a predictive analysis algorithm based on the concept of probability. It measures the relationship between the dependent variable and the one or more independent variables (risk factors) by estimating probabilities using underlying logistic function (sigmoid function). Sigmoid function is used as a cost function to limit the hypothesis of logistic regression between 0 and 1 (squashing) i.e.,

0 ≤ hθ (x) ≤ 1.

In logistic regression cost function is defined as:

𝐶𝑜𝑠𝑡(hθ(x), y) = {

− log(ℎ𝜃(𝑥)) 𝑖𝑓 𝑦 = 1

− log (1 − ℎ𝜃(𝑥)) 𝑖𝑓 𝑦 = 0

Logistic Regression relies highly on the proper presentation of data. So, to make the model more powerful, important features from the available data set are selected using.

**3.2 Support Vector Machine(SVM)**

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

**6.**

We are given a training dataset of 

where the yi are either 1 or −1, each indicating the class to which the point xi belongs. Each xi is a p-dimensional real vector. We want to find the "maximum-margin hyperplane" that divides the group of points xi for which yi=1 from the group of points for which yi=-1 which is defined so that the distance between the hyperplane and the nearest point xi from either group is maximized.

Any hyperplane can be written as the set of points x satisfying  where w is the (not necessarily normalized) normal vector to the hyperplane. This is much like Hesse normal form, except that w is not necessarily a unit vector. The parameter determines the offset of the hyperplane from the origin along the normal vector w.

**3.3 K-Nearest Neighbour (K-NN)**

In statistics, the ***k*-nearest neighbors' algorithm** (***k*-NN**) is a non-parametric classification method first developed by Evelyn Fix and Joseph Hodges in 1951, and later expanded by Thomas Cover. It is used for classification and regression. In both cases, the input consists of the *k* closest training examples in data set. The output depends on whether *k*-NN is used for classification or regression:

* In ***k-NN classification***, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive integer, typically, small). If *k* = 1, then the object is simply assigned to the class of that single nearest neighbor.
* In ***k-NN regression***, the output is the property value for the object. This value is the average of the values of *k* nearest neighbors.

**7**.

*k*-NN is a special case of a variable-bandwidth, kernel density "balloon" estimator with a uniform kernel.

The naive version of the algorithm is easy to implement by computing the distances from the test example to all stored examples, but it is computationally intensive for large training sets. Using an approximate nearest neighbor search algorithm makes *k-*NN computationally tractable even for large data sets. Many nearest neighbor search algorithms have been proposed over the years; these generally seek to reduce the number of distance evaluations actually performed.

*k-*NN has some strong consistency results. As the amount of data approaches infinity, the two-class *k-*NN algorithm is guaranteed to yield an error rate no worse than twice the Bayes error rate (the minimum achievable error rate given the distribution of the data). Various improvements to the *k*-NN speed are possible by using proximity graphs.

For multi-class *k-*NN classification, Cover and Hart (1967) prove an upper bound error rate of



where is the Bayes error rate (which is the minimal error rate possible), is the k-NN error rate, and M is the number of classes in the problem. For and as the Bayesian error rate approaches zero, this limit reduces to "not more than twice the Bayesian error rate".

**3.4 SVM with PCA**

In This Algorithm We Apply SVM and PCA together to find Accuracy and others Parameter.

**More About PCA (Principal component analysis):**

PCA is a most widely used tool in exploratory data analysis and in machine learning for predictive models. PCA is an unsupervised statistical technique used to examine the interrelations among a set of variables. It is also known as a general factor analysis where regression determines a line of best fit. High dimensionality means that the dataset has a large number of features. The primary problem associated with high-dimensionality in the machine learning field is model overfitting, which reduces the ability to generalize beyond the examples in the training set. The ability to generalize correctly becomes exponentially harder as the dimensionality of the training dataset grows, as the training set covers a dwindling fraction of the input space.

**8.**

**3.5 K-NN with PCA**

Based on a K-Nearest Neighbor (KNN) regression, a Principal Component Analysis (PCA) is applied to reduce redundancy information and data dimensionality. In a PCA-KNN model, the historical data set as input is generated by a sliding window, transformed by PCA to principal components with rich-information, and then input to KNN for prediction. Principal Component Analysis (PCA) is an unsupervised, non-parametric statistical technique primarily used for dimensionality reduction in machine learning.

Goals of PRINCIPAL COMPONENT ANALYSIS:

* Identify patterns in data.
* Detect the correlation between variables.
* Reduce the dimensions of a d-dimensional dataset by projecting it onto a
* k-dimensional subspace (where k<d)

**CHAPTER 4: EXPERIMENTS**

The dataset consists of 303 observations with 0 missing data and 165 observations to be risked for heart disease.

Since our project is a classification problem, we use accuracy, precision, recall, and F1 score to evaluate the models. We would like to introduce the meaning of TP, FP, TN and FN. A true positive (TP) is a positive outcome predicted by the model correctly while a false positive (FP) is a positive outcome predicted by the model incorrectly. A true negative (TN) is a negative outcome predicted by the model correctly while a false negative (FN) is a negative outcome predicted by the model incorrectly.

Some of the Machine Learning Models we are going to see are:

⦁ Logistic Regression

⦁ Support Vector Machine (SVM)

⦁ K-Nearest Neighbours Classifier(K-NN)

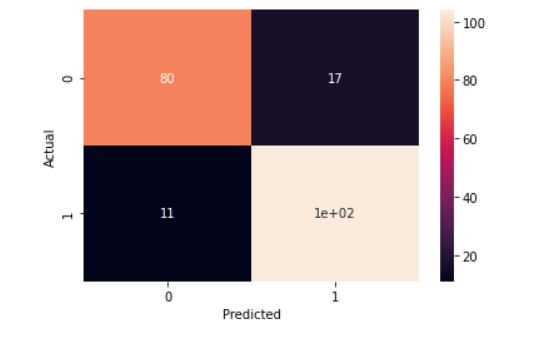
⦁ SVM with PCA

⦁ K-NN with PCA

**9.**

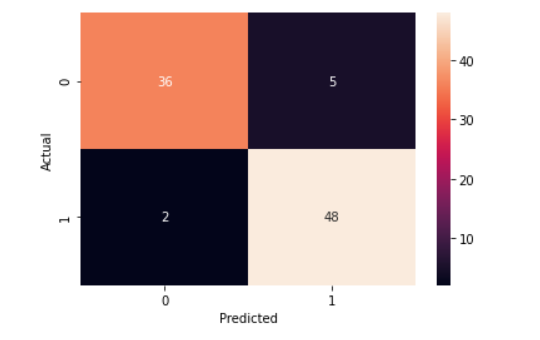
**A. Logistic Regression**

Here is the confusion matrix of the Logistic Regression:



***Fig 2. Training Confusion Matrix***

According to Sklenar, for Logistic regression it is better to use sklearn.linear model. Then import the Logistic Regression. The accuracy for training set comes as **78.77%.**



***Fig 3. Testing Data set***

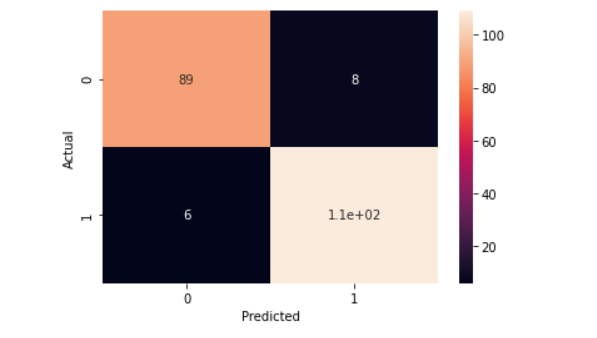
The testing accuracy comes as **92.30%**. Here testing accuracy comes more than training set because training set might be noise, depending on machine learning Algorithm, though it does not evaluate the correctness of the model.

**10.**

The advantage of the Logistic Regression is that it does not need too much computational resources and it is highly interpretable. So, it is easy and sufficient to apply Logistic Regression. However, the limitation of Logistic Regression is that it assumes linearity between the features of the dataset. In the real world, the data is rarely separable, neither as our dataset. That is why we cannot reach a very high accuracy of 95%.

**B. Support Vector Machine (SVM)**

Here is the confusion matrix for SVM are:

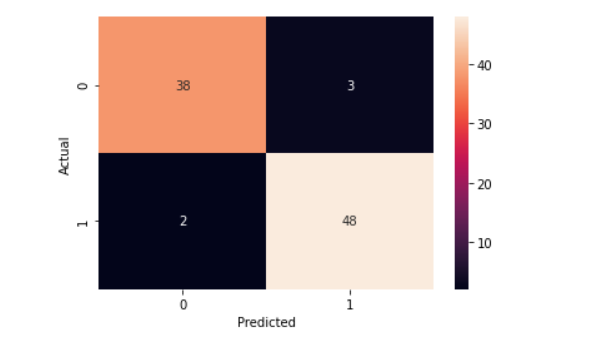


***Fig 4. Training Confusion Matrix for SVM***

For Support Vector Machine (SVM) we import it from SVC.

The train Accuracy for SVM has come as **78.77%.**

**11.**



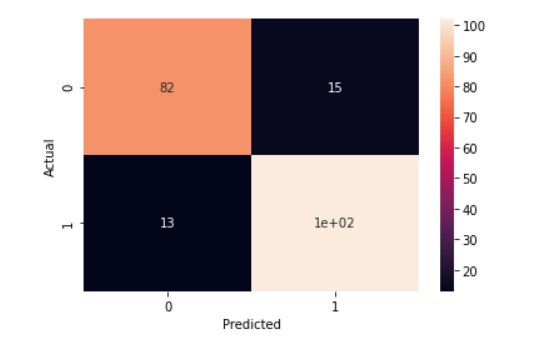
***Fig 5. Testing Confusion Matrix for SVM***

Here also, the test accuracy some more than train accuracy which is **94.50%**. This is good compared to logistic regression.

The advantage of SVM is that it is very efficient with high dimensional spaces. The main disadvantage is that the SVM has many parameters that needs to be correctly chosen to achieve the best performance. For safety we just use the default parameters of SVM.

**C. K-nearest neighbor (K-NN)**

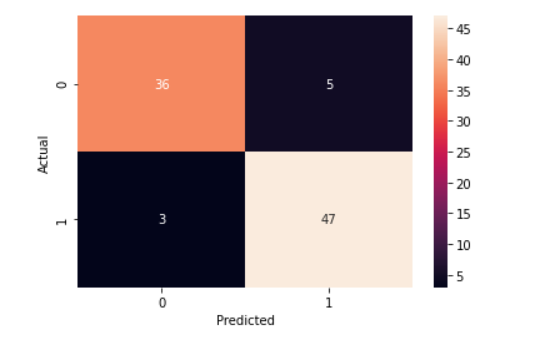
The Confusion matrix for K-NN is:



***Fig 6. Training Confusions Matrix for K-NN***

**12.**

For K-NN we import from Kneighbors Classifier from sklearn.neighbours . The train accuracy for K-NN comes as **75.94%.**



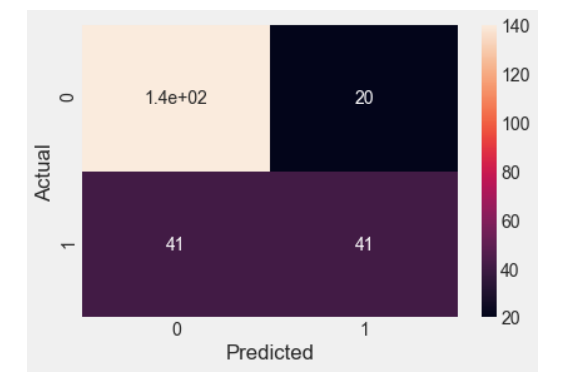
***Fig 7. Testing Confusion matrix for K-NN***

The above confusion plot is of K-NN. The testing accuracy for K-NN comes **91.20%.** It is quite good.

The advantage of K-NN classifier is that it requires no training data before making predictions, new data can be added seamlessly which will not impact the accuracy of the algorithm. It is simple algorithm versatile in nature and gives high accuracy. The disadvantage of K-NN classifier is that it does not work good with large data set and in high dimensions.

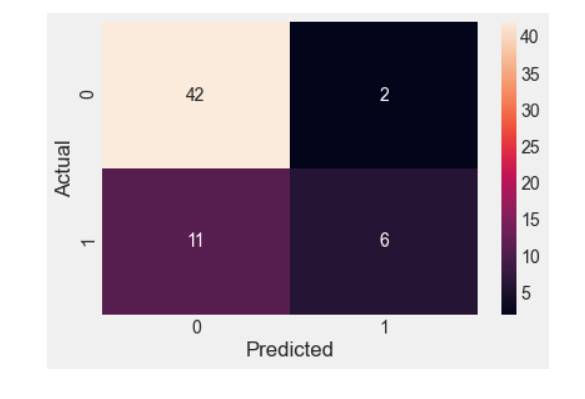
**13.**

**D. SVM with Principal Component Analysis(SVM with PCA)**



***Fig 8. Training Confusion Matrix for SVM with PCA***

We have first applied test train split then Standard scaling and then applied PCA in this SVM with PCA Module. The training accuracy comes as **77.47% for n=7.**



***Fig 9. Testing Confusion for SVM with PCA***

The testing accuracy for SVM with PCA comes as **87.85% for n=7**. It is found that when the principal component value increases the accuracy also increases.

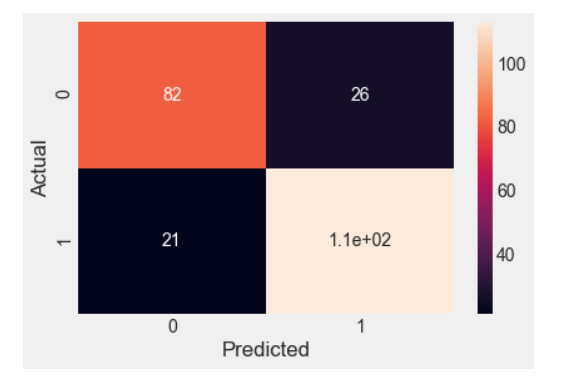
**14.**

As there are many principal components as there are variable in data, principal component is constructed in such a manner that the first principal components account for largest possible variance in data set.

So, as principal component increases to n=7 its accuracy increases.

**E. K-NN with Principal Component Analysis**

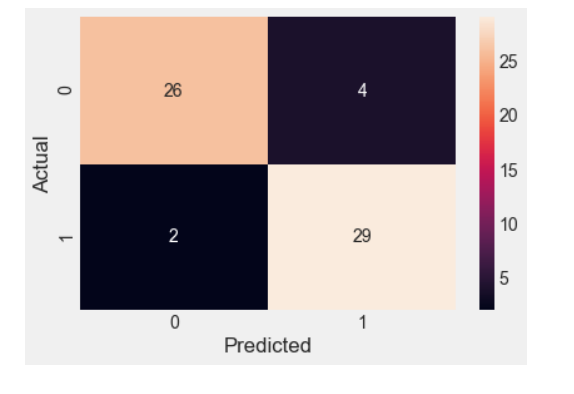
The confusion matrix for K-NN is:



***Fig 10. Training Confusion Matrix for K-NN with PCA***

Here we first apply PCA, then import knearest neighbor's classifier model and then create k-nn classifier then train the model using training set then predict the response for test data and then plot confusion matrix. The training accuracy comes as **80.57%. for n=11**

**15.**



***Fig 11. Testing Confusion Matrix for K-NN with PCA.***

The testing Accuracy of it is **90.16% for n=11**

Based on K-Nearest Neighbor (KNN) regression principal component analysis is applied to reduce redundancy information and Dimensionality.

**CHAPTER 5: DISCUSSION ON RESULTS**

When performing various methods of feature selection, testing it was found that SVM (Support Vector Machine) And K-NN (K-nearest neighbor classifier) gave us the best results among others. The various methods tried were SVM (Support Vector Machine) with linear kernel And K-NN (K-nearest neighbor classifier) and SVM with PCA gave very less accuracy as compared to others in range of 77% to 87% and K-NN with PCA module gave moderate accuracy as compared to SVM and KNN. The accuracy that was seen in SVM is 94% being maximum. Though both module SVM and KNN gave accuracy in the range of 85% to 95%.

**16.**

***RESULT OF DIFFERENT METHODS***

|  |  |  |
| --- | --- | --- |
| **Methods** | **Train accuracy** | **Test accuracy** |
| Logistic Regression | 78.77% | 92.30% |
| SVM | 78.77% | 94.50% |
| K-NN | 75.94% | 91.20% |
| SVM with PCA | 77.47% | 87.85% |
| K-NN with PCA | 80.57% | 90.16% |

**CHAPTER 6: CONCLUSION**

A cardiovascular disease detection model has been developed using three ML classification modelling techniques. This project predicts people with cardiovascular disease by extracting the patient medical history that leads to a fatal heart disease from a dataset that includes patients’ medical history such as chest pain, sugar level, blood pressure, etc. This Heart Disease detection system assists a patient based on his/her clinical information of them been diagnosed with a previous heart disease. The algorithms used in building the given model are Logistic regression, SVM (Support Vector Machine), KNN (K-nearest neighbor classifier), SVM with PCA and KNN with PCA. Use of more training data ensures the higher chances of the model to accurately predict whether the given person has a heart disease or not. Therefore, in conclusion this project helps us predict the patients who are diagnosed with heart diseases by cleaning the dataset and applying SVM gave better testing accuracy.

***REFERENCES:***

https://towardsdatascience.com/predicting-presence-of-heart-diseases-using-machine-learning-36f00f3edb2c

https://www.kaggle.com/faressayah/predicting-heart-disease-using-machine-learning

**Link of project file-** <final_group3_mlProject.ipynb>

**17.**

**Thank you**