**USER IDENTIFICATION FROM WALKING DATA**

A Course Project report submitted

in partial fulfillment of requirement for the award of degree

**BACHELOR OF TECHNOLOGY**

in

**ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**

by

M.Nikhilesh (2103A51172)

D.Vamshi kumar (2103A51125)

Ch.Sumanth (2103A51123)

Under the guidance of

**Mr. D. Ramesh**

Assistant Professor, Department of CSE.



**Department of Computer Science and Artificial Intelligence**



**Department of Computer Science and Artificial Intelligence**

**CERTIFICATE**

This is to certify that project entitled **“USER IDENTIFICATION FROM WALKING DATA** " is the bonafide work carried out by **MAJHI NIKHILESH, D VAMSHI KUMAR, CH SUMANTH** as a Course Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING** during the academic year 2022-2023 under our guidance and Supervision.

**Mr. D. Ramesh Dr. M Sheshikala**

Asst. Professor, Assoc. Prof. & HOD (CSE),

S R University, S R University,

Ananthasagar, Warangal. Ananthasagar, Warangal.

**ACKNOWLEDGEMENT**

We express our thanks to Course co-coordinator **Mr. D. Ramesh, Asst. Prof.** for guiding us from the beginning through the end of the Course Project. We express our gratitude toHead of the department CS&AI, **Dr. M. Sheshikala, Associate Professor** for encouragement, support and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, School of Computer Science and Artificial Intelligence, **Dr C. V. Guru Rao**, for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

**ABSTRACT**

Mobile and wearable technology are becoming more prevalent and play an increasingly significant part in our daily lives. Mobile devices are used to store highly sensitive data, such as vital documents and files. Using data from accelerometers in mobile and wearable devices, we want to categorize people and identify their routines. Previous research on gait analysis revealed the best results in terms of minimizing mistake rates in spotting a person using their motion data.

**CONTENTS**

***ABSTRACT iv***

**Chapter No. Title Page No.**

**1 INTRODUCTION** 01

1.1 Problem Statement 01

1.2 Existing System 01

1.3 Proposed System 02

1.4 Objectives 02

1.5 Architecture 02

**2 Literature survey** 03

2.1 Analysis of the Survey 03

**3. Data pre-processing** 05

3.1 Dataset Description 05

3.2 Pre-processing through Standardscaler 07

3.2 Data Visualization 08

**4. METHODOLOGY** 12

4.1 Procedure to solve 12

4.1.1 Using KNN 12

4.1.3 Using Support Vector Machine 13

4.2 Software Description 14

4.2.1 Through Decision Tree 14

4.2.3 Through Support vector Machine

4.2.3 Through KNN

4.2.3 Through Naïve Bayes 20

**5.**  **RESULTS**  29

**6.** **CONCLUSION** 30

6.1 Conclusion 30

6.2 Future scope 30

**REFERENCES** 31

# CHAPTER 1

# INTRODUCTION

In recent years, with the rapid development of wearable technology, collecting walking data has become an increasingly popular way to monitor human activities. Walking data can provide valuable insights into an individual's gait and movement patterns, which can be used to identify and track users. Machine learning techniques have been widely employed to extract meaningful information from walking data and to develop user identification models.

The purpose of this report is to investigate the effectiveness of machine learning algorithms in identifying users from walking data. Specifically, we explore the performance of various classification models, such as support vector machines, decision trees, and random forests, on a dataset of walking data collected from multiple users. We also evaluate the impact of different feature selection techniques on the classification accuracy and investigate the generalization ability of the models to new users.

The findings of this report have important implications for developing effective user identification systems using walking data. The results can help inform the design of personalized healthcare and fitness applications, security systems, and smart home devices that rely on user identification.

## Problem Statement

The problem addressed in this report is the identification of users from walking data using machine learning techniques. The goal is to develop accurate and reliable classification models that can differentiate between different users based on their gait and movement patterns. This problem is important for a variety of applications, including personalized healthcare and fitness monitoring, security systems, and smart home devices that require user identification. The challenge is to extract meaningful features from walking data and to develop classification models that can generalize well to new users. The report aims to investigate the effectiveness of different machine learning algorithms and feature selection techniques in addressing this problem and to provide insights into the development of user identification systems using walking data.

## Existing System

Predicting health of fetal has been studied extensively. Works related include classification model that was built KNN and SVM which helps with better precision than some simple classification. SVM is better in dealing with datasets with more dimensions and it is less prone to over fitting and under fitting. To support advanced KNN; basic indicators such as mean, variance and standard deviation are required.

## Proposed System

With the assist of dataset obtained we create 4 different machine algorithms, specifically KNN and Logistic Regression and examine the outcomes of accuracy and find which models performs better and is reliable.

## Objectives

* Compare the accuracy in 2 specific classification-based system learning algorithms.
* To establish machine learning algorithms are reliable for automatic results.
* Smoother the troublesome method throughout the child’s fetal health and mothers’ maternity.

## Architecture

The Supervised-Learning-approach as a qualitative-data with KNN classification, logistic Regression and its target to predict health of fetal, which might be normal, suspect or pathological.

## CHAPTER 2

## LITERATURE SURVEY

**2.1 ANALYSIS OF the survey**

The dataset that has been used is the CTG data which is observed to be beneficial to identify the abnormalities. The visual analysis along which the decision support system focuses has been made on the machine learning models that have been used. As the machine learning doesn’t perform well on the basis of accuracy the ensemble model has been used which has bagged an accuracy of 99.02% after the 10-fold cross validation has been employed. Hence, this can be used to classify the normal and the pathological cases of the ctg data. We will have data and with the help of that data inputs that we have we predict the fetal health.

A standard procedure done during the third trimester is fetal monitoring. Fetal monitoring is checking the health of the unborn baby. Fetal growth entirely depends on the mother's health. To avoid such complications, a continuous measuring of fetus health and growth rate is done with cardiotocography. The cardiotocography aims to track the fetus' heartbeat and parallelly measure the mother's uterine contractions. This process would be performed during the final trimester, once the fetus' growth functions fully with heart rate. This method is considered cost-effective and straightforward, and hence, it is to be carried out by medical experts for early detection of fetal status and to reduce fetal mortality. The result of cardiotocographic (CTG) will trace uterine contraction of the mother, most importantly heart rate of fetus, occurrence of acceleration, series of deceleration, and much more complicated measure of fetus. There are many ML techniques available for classifying the fetus's normal, suspected, and pathological stages. The results show that the ML technique will form a framework widely used for the automated system in analyzing early fetal health.

Artificial Intelligence (AI) techniques have recently provided a significant decision for early diagnosis and multi-classifications. A significant comparison was made between 15 ML techniques defending healthy vs. affected fetuses. The features are extracted from CTG signal recording. These effectively evaluate large amounts of real-time data to provide better solutions and develop a framework for other models to perform classification. CTC signals to directly assess the heart rate of the patients and give accurate results and updates to the medical experts. The effectiveness of using cardiotocography is discussed for the wellbeing of the fetal during labor. CTG is responsible for measuring the fetal heart rate and the contraction of the womb; hence this plays a vital role in assessing fetal before birth and also during labor.

This would also measure the frequency of baby movement. A recent rush in the deployment of two ML techniques, namely K-nearest neighbors (K-NN) Classifier, Logistic Regression which are evaluated on high dimensional data, proves that the classifier KNN fairly dominates the other technique in giving accurate diagnostic indices. Both the Classifiers work well and use 30 ranked features to determine common risk factors in the prediction model. While using ML algorithms, feature extraction and selection are among the most used methods to select optimized features for prediction in the model.

## CHAPTER 3

## DATA PRE-PROCESSING

### 3.1 Dataset Description

The dataset collects data from an Android smartphone positioned in the chest pocket. Accelerometer Data are collected from 5 participants walking in the wild over a predefined path.

**TABLE 1: Attributes of the Cardiotocographic data**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute (s) | Range | Attribute | Range |
| Fetal heart rate baseline | (160–106) | Width of FHR histogram | (3–180) |
| Accelerations | (0–0.019) | Minimum of FHR histogram | (50–159) |
| Fetal movements | (0–0.481) | Maximum of FHR histogram | (122–238) |
| Uterine contractions | (0–0.015) | Count of histogram peaks | (0–18) |
| Light decelerations | (0–0.015) | Count of histogram zeros | (0–10) |
| Severe decelerations | (0–0.001) | Histogram mode | (60–187) |
| Prolonged decelerations | (0–0.005) | Histogram mean | (73–182) |
| Percentage of time with abnormal short-term variability (S-TV) | (12–87) | Histogram median | (77–186) |
| Mean value of S-TV | (7–0.2) | Histogram variance | (0–269) |
| Percentage of time with abnormal | (0–91) | Histogram tendency | 1 |
| long-term variability (L-TV) | | | |

Mean value of L-TV (0–50.7) Fetal status (1-Normal, 2-Needs Reassurance, 3-Pathological)

Medical experts classify recorded data manually and fit it into the range where the value gets satisfied, and an apprehensive state is given when the value does not provide the range.

ML algorithms can assist doctors in concluding the fetal health status by predicting the health state. The CTG data collected by the University of Porto from 1831 recordings are used as a

benchmark dataset to assess the performance of the ML models. These recordings are automated by SisPorto2.0 (CTG Analysis Program) and are made available in UCI-ML. The attribute details of the CTG data are presented in Table 2.

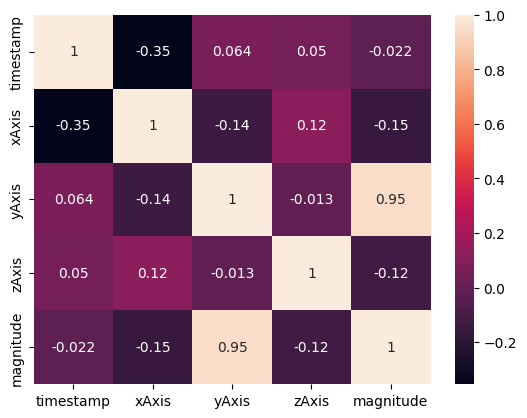
**3.2 PRE-PROCESSING THROUGH STANDARDSCALER**

Many machine learning algorithms work better when features are on a relatively similar scale and close to normally distributed. We use ‘StandardScaler’, a scikit-learn method to preprocess data.

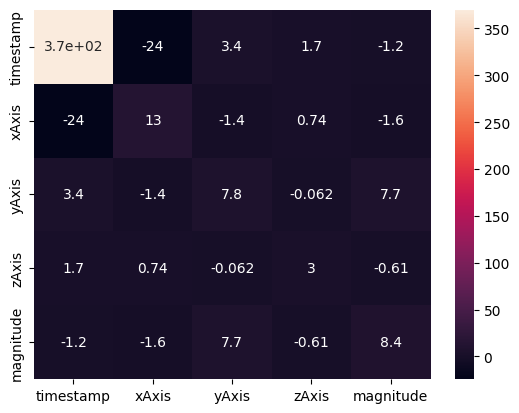
* Scale generally means to change the range of the values. The shape of the distribution doesn’t change. Think about how a scale model of a building has the same proportions as the original, just smaller. That’s why we say it is drawn to scale. The range is often set at 0 to 1.
* Standardize generally means changing the values so that the distribution’s standard deviation equals one. Scaling is often implied.
* StandardScaler standardizes a feature by subtracting the mean and then scaling to unit variance. Unit variance means dividing all the values by the standard deviation.
* StandardScaler results in a distribution with a standard deviation equal to 1. The variance is equal to 1 also, because variance = standard deviation squared and 1 squared = 1, StandardScaler makes the mean of the distribution approximately 0.

**3.3 DATA VISUALIZATION**

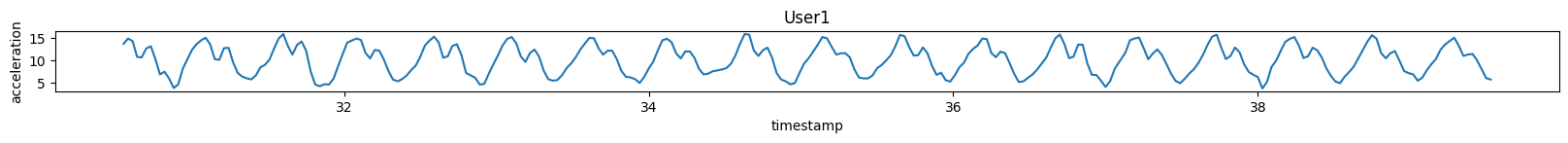
**Correlation matrix**

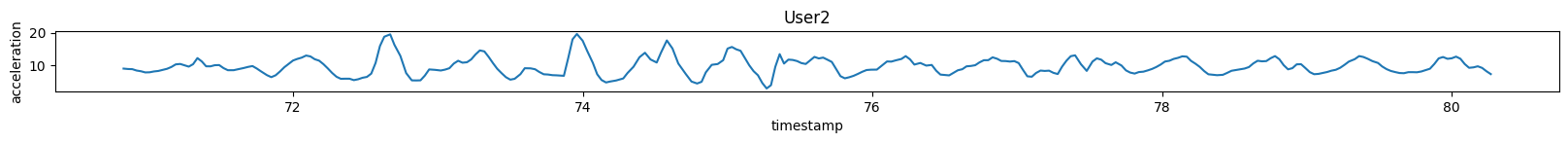


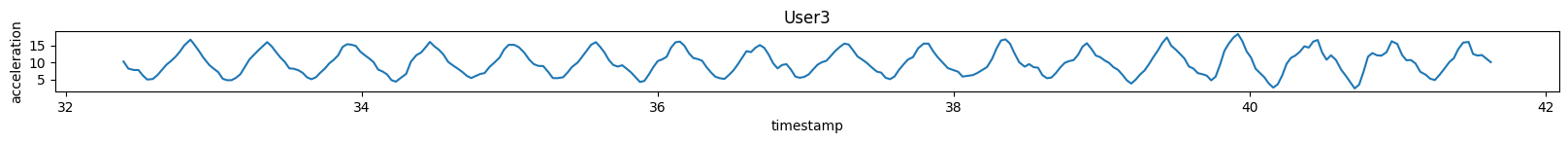
**Covariance Matrix**

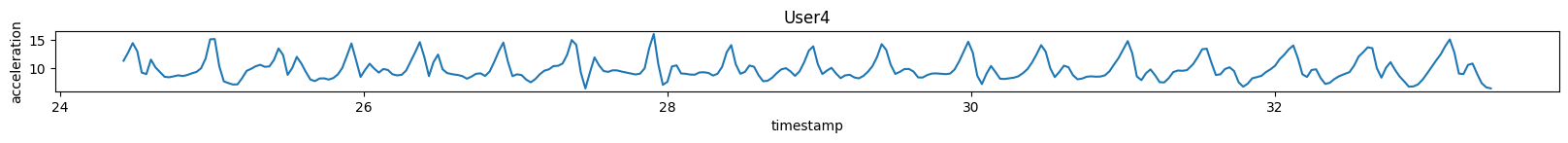


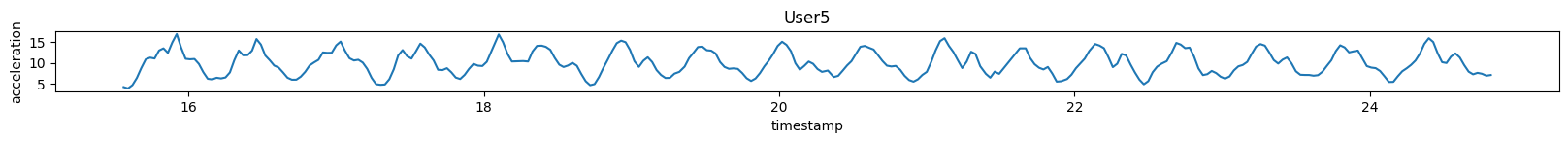
**Plotting graph for timestamp vs acceleration**





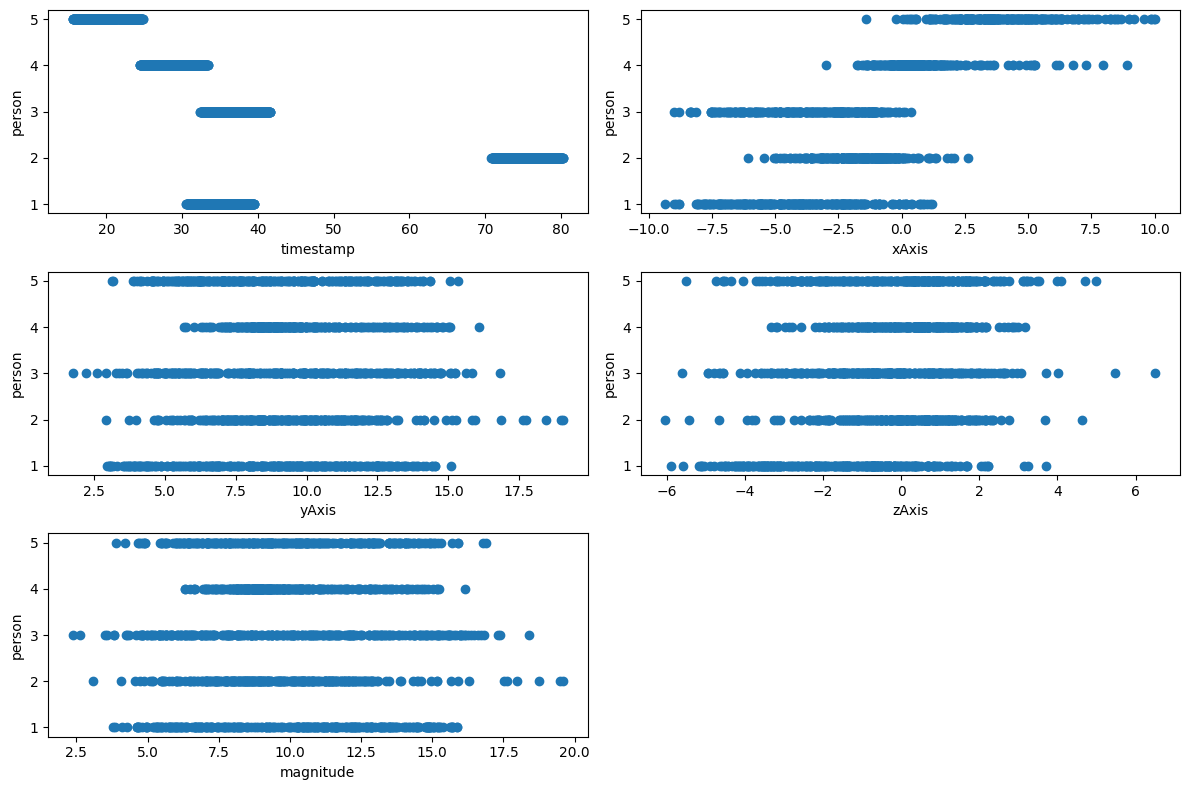




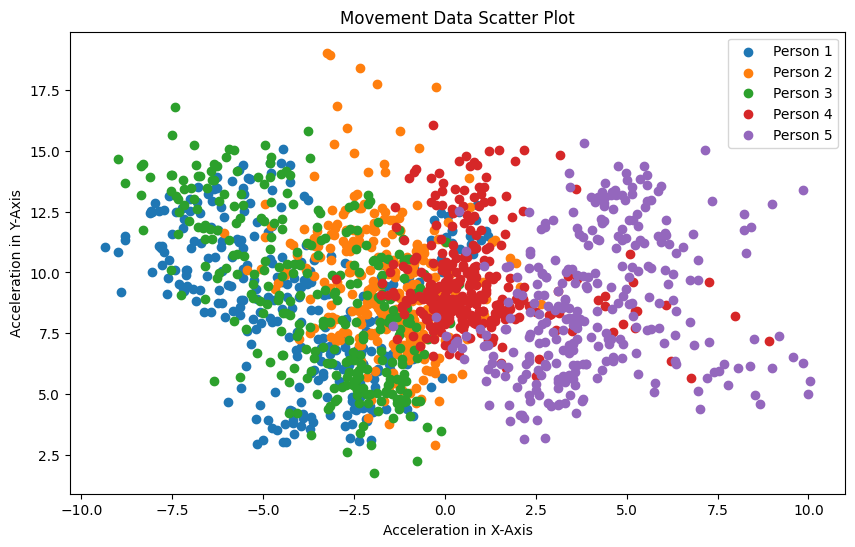


**Plotting each feature against target variable**

*Scatter Plot*



**Scatter plot for xAxis vs yAxis accleration**



# CHAPTER 4

# METHODOLOGY

**4.1. PROCEDURE TO SLOVE THE GIVEN PROBLEM**

**4.1.1 Using KNN**

The K-NN working can be explained on the basis of the below algorithm:

**Step-1:** Select the number K of the neighbors

**Step-2:** Calculate the Euclidean distance of **K number of neighbors**

**Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.

**Step-4:** Among these k neighbors, count the number of the data points in each category.

**Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.

**Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:

Firstly, we will choose the number of neighbors, so we will choose the k=5.

Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry.



By calculating the Euclidean distance, we get the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B.

As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

## 4.2 SOFTWARE DESCRIPTION

We used the Google colab service to test our machine learning algorithms written in Python. The jupyter notebooks with the results shown below.

**Importing Required Libraries**

import numpy as np  
import matplotlib.pyplot as plt  
import pandas as pd  
import seaborn as sns

**Importing classes for machine learnig models**

from sklearn.neighbors import KNeighborsClassifier  
from sklearn.tree import DecisionTreeClassifier  
from sklearn.svm import SVC  
from sklearn.naive\_bayes import GaussianNB  
models = {  
 'Decision Tree':DecisionTreeClassifier(),  
 'Support Vector Machine':SVC(),  
 'Naive Bayes':GaussianNB(),  
 'KNN':KNeighborsClassifier(n\_neighbors=5)  
}

**Loading data into a dataframe using pandas**

dataset=pd.read\_csv('Features.csv',names=['person','time','xAxis','yAxis','zAxis','magnitude'])

user\_list=['user1','user2','user3','user4','user5']

**Separating target and independent variables**

X=dataset.iloc[:,1:].values  
y=dataset.iloc[:,0].values

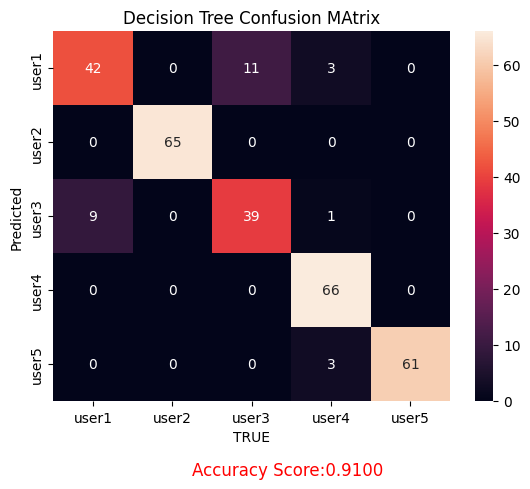
**Splitting data into test and train set**

from sklearn.model\_selection import train\_test\_split  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

from sklearn.metrics import confusion\_matrix,accuracy\_score  
accscores=[]

## 4.2.1 THROUGH Decision Tree Implementation

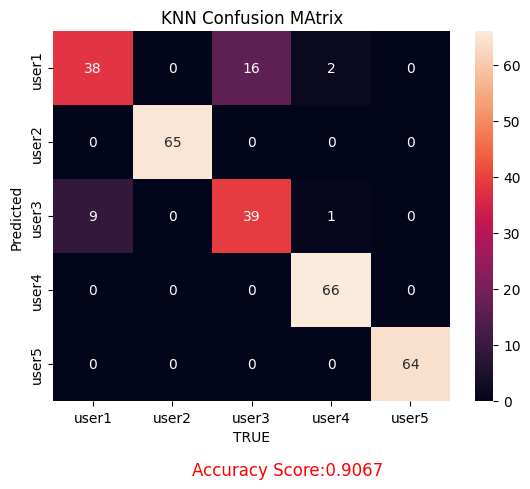
name='Decision Tree'  
model=models[name]  
model.fit(X\_train, y\_train)  
y\_pred = model.predict(X\_test)  
cm = confusion\_matrix(y\_test, y\_pred)  
accscore=accuracy\_score(y\_test,y\_pred)  
accscores.append(accscore)  
  
sns.heatmap(cm,annot=True,xticklabels=user\_list,yticklabels=user\_list)  
plt.xlabel('TRUE')  
plt.ylabel('Predicted')  
plt.title(name+' Confusion MAtrix')  
plt.text(1.75,6,f'{"Accuracy Score:"}{accscore.item():.4f}',fontsize='12',color='red')  
plt.show()



## 4.2.2 THROUGH KNN

**K-Nearest Neighbor Implementaion**

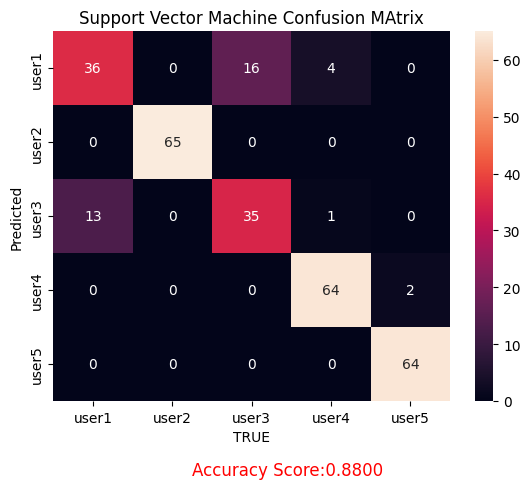
name='KNN'  
model=models[name]  
model.fit(X\_train, y\_train)  
y\_pred = model.predict(X\_test)  
cm = confusion\_matrix(y\_test, y\_pred)  
accscore=accuracy\_score(y\_test,y\_pred)  
accscores.append(accscore)  
  
sns.heatmap(cm,annot=True,xticklabels=user\_list,yticklabels=user\_list)  
plt.xlabel('TRUE')  
plt.ylabel('Predicted')  
plt.title(name+' Confusion MAtrix')  
plt.text(1.75,6,f'{"Accuracy Score:"}{accscore.item():.4f}',fontsize='12',color='red')  
plt.show()



## 4.2.3 THROUGH Support vector Machine

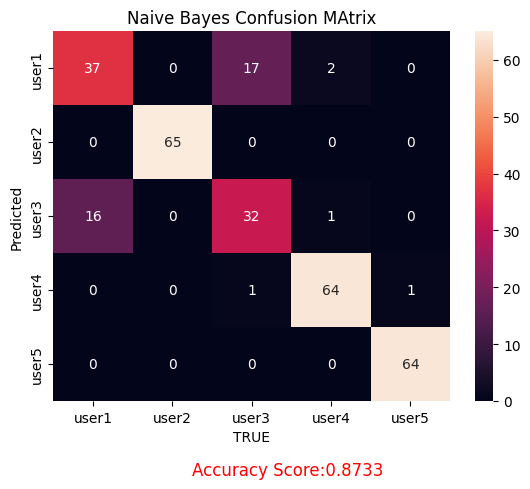
**Support Vector Machine Implementation**

name='Support Vector Machine'  
model=models[name]  
model.fit(X\_train, y\_train)  
y\_pred = model.predict(X\_test)  
cm = confusion\_matrix(y\_test, y\_pred)  
accscore=accuracy\_score(y\_test,y\_pred)  
accscores.append(accscore)  
  
sns.heatmap(cm,annot=True,xticklabels=user\_list,yticklabels=user\_list)  
plt.xlabel('TRUE')  
plt.ylabel('Predicted')  
plt.title(name+' Confusion MAtrix')  
plt.text(1.75,6,f'{"Accuracy Score:"}{accscore.item():.4f}',fontsize='12',color='red')  
plt.show()



## 4.2.4 THROUGH Naive Bayes

name='Naive Bayes'  
model=models[name]  
model.fit(X\_train, y\_train)  
y\_pred = model.predict(X\_test)  
cm = confusion\_matrix(y\_test, y\_pred)  
accscore=accuracy\_score(y\_test,y\_pred)  
accscores.append(accscore)  
  
sns.heatmap(cm,annot=True,xticklabels=user\_list,yticklabels=user\_list)  
plt.xlabel('TRUE')  
plt.ylabel('Predicted')  
plt.title(name+' Confusion MAtrix')  
plt.text(1.75,6,f'{"Accuracy Score:"}{accscore.item():.4f}',fontsize='12',color='red')  
plt.show()



*DECISION TREE HAS BEST PERFORMANCE*

### CONCLUSION AND FUTURE SCOPE

### CHAPTER 5

### RESULTS

**Accuracy tHROUGH KNN**

|  |  |
| --- | --- |
| ML model | Accuracy |
| KNN | 0.8733333 |

**Accuracy ThrouGh naïve BAYES**

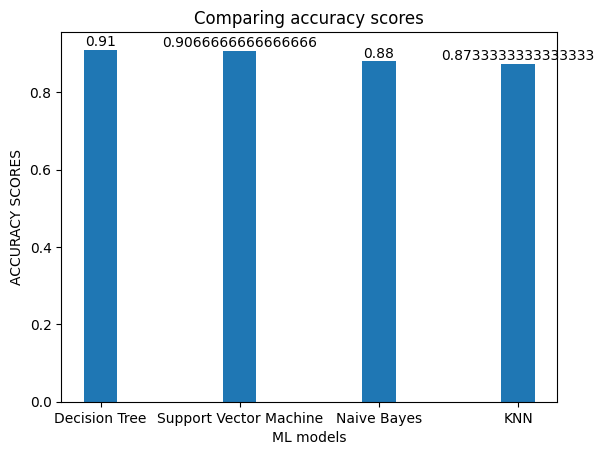
|  |  |
| --- | --- |
| ML model | Accuracy |
| Naïve bayes | 0.88 |

**Accuracy FRom SUPPORT VECTOR MACHINE**

|  |  |
| --- | --- |
| ML model | Accuracy |
| Support vector Machine | 0.908666 |

**Accuracy FROM DECISION TREE**

|  |  |
| --- | --- |
| ML model | Accuracy |
| Decision tree | 0.91 |



### CHAPTER 6

### 6.1 CONCLUSION

Finally, after performing all the steps needed to get the results from preparation to preprocessing to performing the models (Logistic regression and KNN) we conclude that the KNN model with 90.90909090909091 percent accuracy performs relatively better than Logistic Regression of 90.36885245901639 percent accuracy.

## 6.2 FUTURE SCOPE

As there is a lot of possibility of improvement in this based on the data as modern real time data can be collected which can be used to test all the different models that are present and to create a new accuracy based on this. Another thing that can be done is to test the model and also create a comparison on the new data. The data collection would take a long time hence till then multiple times the data should be collected from different sources.

### REFERENCES

1. <https://www.karger.com/Article/Fulltext/505021>
2. <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
3. <https://onlinelibrary.wiley.com/doi/full/10.1111/exsy.12899>
4. <https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification>
5. <https://www.researchgate.net/publication/356666279>
6. <https://www.ijert.org/research/fetal-health-prediction-using-classification-techniques-IJERTV10IS110182.pdf>