A

MAJOR PROJECT REPORT

on

Flat-Foot prediction using Machine Learning

Submitted in fulfillment for the award of the Degree of

Bachelor of Technology

In

Mechanical Engineering

By

R170581 B Lokesh

R170459 G Naveen Kumar

R161775 K Kalyani

R170857 K Vinod Kumar Reddy

Under the Guidance of

Mr. B. Imran Shareef

Assistant Professor,

Department of Mechanical Engineering,

IIIT RK Valley, RGUKT-AP.

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(A.P. Government Act 18 of 2008) RGUKT-RK VALLEY,

Idupulapaya, Y.S.R Kadapa(Dist)- 516330,

Andhra Pradesh.

A

MAJOR PROJECT REPORT

On

Flat-Foot prediction using Machine Learning

Submitted in fulfillment for the award of the Degree of

Bachelor of Technology

In

Mechanical Engineering

By

R170581 B Lokesh

R170459 G Naveen Kumar

R161775 K Kalyani

R170857 K Vinod Kumar Reddy

Under the Guidance of

Mr. B. Imran Shareef,

Assistant Professor,

Department of Mechanical Engineering,

IIIT RK Valley, RGUKT-AP.

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(A.P. Government Act 18 of 2008) RGUKT-RK VALLEY,
Idupulapaya, Y.S.R Kadapa(Dist)- 516330,

Andhra Pradesh.

Declaration

Project Title: **Flat-Foot prediction Using Machine Learning**

Degree for which the Project is submitted: **Bachelor of Technology in Mechanical Engineering**

We declare that the presented thesis represents largely our own ideas and work in our own words. Where others ideas or words have been included, we have adequately cited and listed in the reference materials. The thesis has been prepared without resorting to plagiarism. We have adhered to all principles of academic honesty and integrity. No falsified or fabricated data have been presented in the thesis. We understand that any violation of the above will cause for disciplinary action by the Institute, including revoking the conferred degree, if conferred, and can also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been taken.

Name of the Student: B. Lokesh
Regd. ID no.: R170581

Name of the Student: G. Naveen Kumar
Regd. ID no.: R170459

Name of the Student: K. Kalyani
Regd. ID no.: R161775

Name of the Student: K. Vinod Kumar Reddy
Regd. ID no.: R170857

Date:

CERTIFICATE

It is certified that the work contained in this thesis entitled **‘Flat-Foot prediction Using Machine Learning’** submitted **by G Naveen Kumar-R170459, K Vinod Kumar Reddy-R170857, B Lokesh-R170581, K Kalyani -R161775,** for the award of B.Tech. is absolutely based on his/her own work carried out under my/our supervision and that this work/thesis has not been submitted elsewhere for any degree.

Head of the Department
Mr. G Naveen Kumar
Assistant Professor
Dept. of Mechanical Engg.

Internal Guide
Mr. B. Imran Shareef
Assistant Professor
Dept. of Mechanical Engg.

External Examiner

1) ____________________________

2)____________________________

Abstract

Flatfoot is a condition in which the entire sole of the foot touches the floor while standing. It is a postural deformity in which the arches of the foot are collapsed and the entire sole of the foot lays against the ground.[1]

This report gives a detailed abstract on the biomechanics of the foot and ankle, the causes and effects of flatfoot. The report also explains the shoe types and their manufacturing and the correct kind of shoe that can solve the problem of flatfoot. This report centers around our machine learning project which aims to predic the foot type of students by distinguishing them as either flatfoot or normal. The article clearly illustrates the methodologies followed to collect data of footprints, the algorithm model used for prediction. We have also tried to recommend the scientifically correct shoe to wear for flatfooted people.

The report gives detailed information on the algorithms implemented to predict flat-foot and the scores each model has given. Also discussed about the algorithm which has given the best results.

Artificial Intelligence and Machine learning can be deployed for solving the issues those are complex and time consuming for humans. Machine learning can be trustfully used for solving problems like flatfoot which suffers around 20- 30 % population worldwide.

Lastly, the report summarizes the observations made in doing this project, the potential of machine learning in dealing problems related to biomechanics. This project is the first step towards tackling the flatfoot problem and it may enable all people to get consciousness about it and its causes and to wear the correct kind of shoes to prevent flatfoot.

Keywords: Flatfoot, Shoes, Machine Learning, Biomechanics of foot, Algorithms, Performance.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to **Mr. B**. **Imran Shareef,** our project guide, for valuable suggestions and keen interest throughout the progress of our course of research.

We are grateful to **Mr G. Naveen Kumar**, HOD of Mechanical engineering for providing excellent computing facilities and a congenial atmosphere for progressing with our project.

At the outset, we would like to thank RGUKT for providing all the necessary resources for the successful completion of our course work.

Your Sincerely,
B.Lokesh
G Naveen Kumar
K Kalyani
K Vinod Kumar Reddy

Contents

Chapter-1

Chapter-2

Literature Survey

Chapter-3

Prediction Models

Chapter-4

Chapter 1

1.1 Introduction

The project we intend to do for the final year project is flatfoot prediction in humans using machine learning models. Flatfoot, also called pes planus or fallen arches, is a postural deformity in which the arches of the foot are collapsed and the entire sole of the foot lays against the ground, either completely or nearly completely. The structure of flatfoot is related to the biomechanics of the lower leg and affects its functionality. Flatfoot can cause conditions related to the alignment of the foot, ankle, leg, pelvis and spine. Since these foot conditions are unstable, related joints experience excessive and unusual movements, and may become easily tired or damaged. In the long term, flatfoot can also cause numerous health problems, such as joint inflammation, heel pain syndrome, diverse foot deformities, pain and swelling.[1]

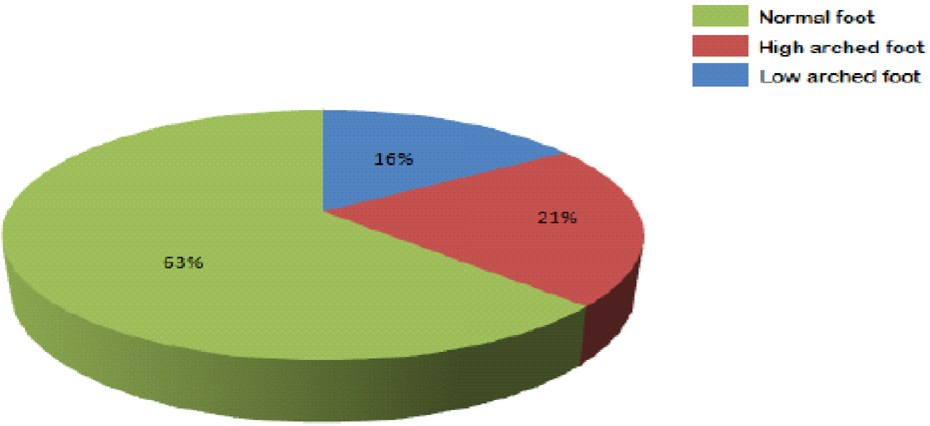


Fig. 1.1 Pie-chart-showing-distribution-of-arch-of-foot-in- adult[10].

Wearing shoes with correct fitment can overthrow the problem of flatfoot in children such that good arches are formed. Also getting doctor recommended Insoles or inserting a wedge into their footwear along the inside edge of an orthotic can do well to relieve the pain. [2]

Machine learning is a branch of artificial intelligence and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. The reason for using footprint as the index is that it is among the most popular and widely used methods of assessing Medial Longitudinal Arch (MLA) of foot [4]. Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data. Here in this project also, data we provided is the key for the model to decide itself what kind of foot it is.

Chapter 2

Literature Survey

2.1 History of Machine Learning [7]

Machine learning (ML) is an important tool for the goal of leveraging technologies around artificial intelligence. Because of its learning and decision-making abilities, machine learning is often referred to as AI, though, in reality, it is a subdivision of AI. Until the late 1970s, it was a part of AI's evolution. Then, it branched off to evolve on its own. Machine learning has become a very important response tool for cloud computing and e-Commerce, and is being used in a variety of cutting edge.

The Perceptron was one of the first algorithms to use artificial neural networks, widely used in machine learning. The Turing Test is a test of artificial intelligence proposed by mathematician Alan Turing. It involves determining whether a machine can act like a human, or if humans can't tell the difference between human and machine given answers.The Nearest Neighbor Algorithm was developed as a way to automatically identify patterns within large datasets. The goal of this algorithm is to find similarities between two items and determine which one is closer to the pattern found in the other item. This can be used for things like finding relationships between different pieces of data or predicting future events based on past events. The rise of machine learning in the 21th century is a result of Moore's Law and its exponential growth.

Current applications of Machine learning

- **Analyzing Sales Data:** Streamlining the data
- **Real-Time Mobile Personalization:** Promoting the experience
- **Fraud Detection:** Detecting pattern changes
- **Product Recommendations:** Customer personalization
- **Learning Management Systems:** Decision-making programs
- **Dynamic Pricing:** Flexible pricing based on a need or demand
- **Natural Language Processing:** Speaking with humans.

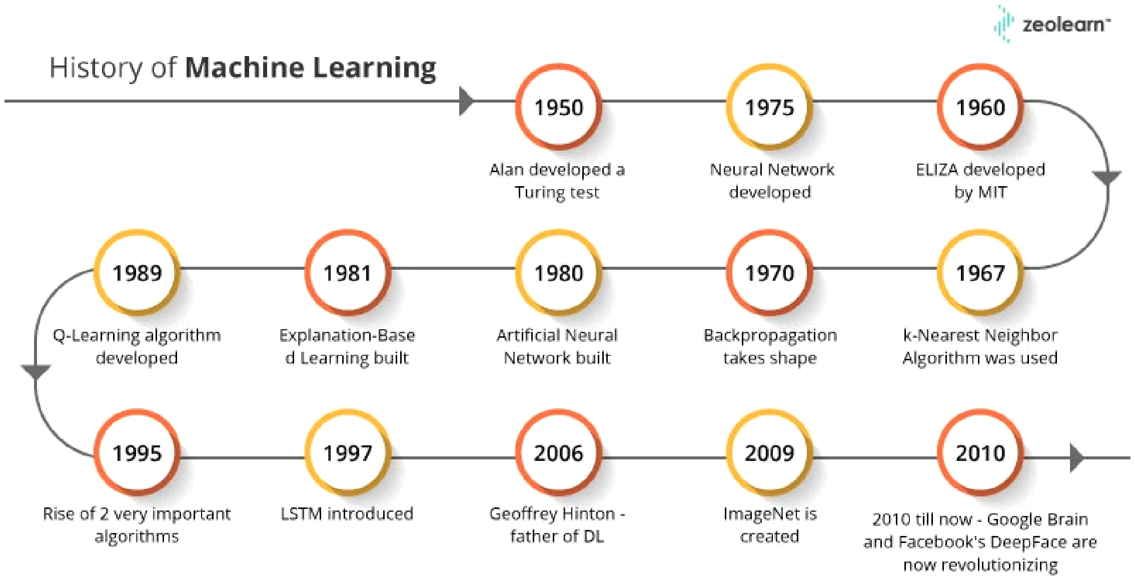

zeolearn
History of Machine Learning
1950
Alan developed a Turing test
1975
Neural Network developed
1960
ELIZA developed by MIT
1989
Q-Learning algorithm developed
1981
Explanation-Base d Learning built
1980
Artificial Neural Network built
1970
Backpropagation takes shape
1967
k-Nearest Neighbor Algorithm was used
1995
Rise of 2 very important algorithms
1997
LSTM introduced
2006
Geoffrey Hinton - father of DL
2009
ImageNet is created
2010
2010 till now - Google Brain and Facebook's DeepFace are now revolutionizing

Fig-2.1 Brief history of Machine Learning[5]

2.2 Biomechanics of foot [3]

Biomechanics of foot refers to the way in which the bones, muscles, tendons, and ligaments work together to support the body and allow for movement.

The human foot is a complex structure that is designed to support the weight of the body and provide stability while walking, running, and jumping. The foot is made up of 26 bones, 33 joints, and more than 100 muscles, tendons, and ligaments. The bones of the foot include the heel bone, the ankle bone, and the metatarsals (the bones in the arch of the foot).

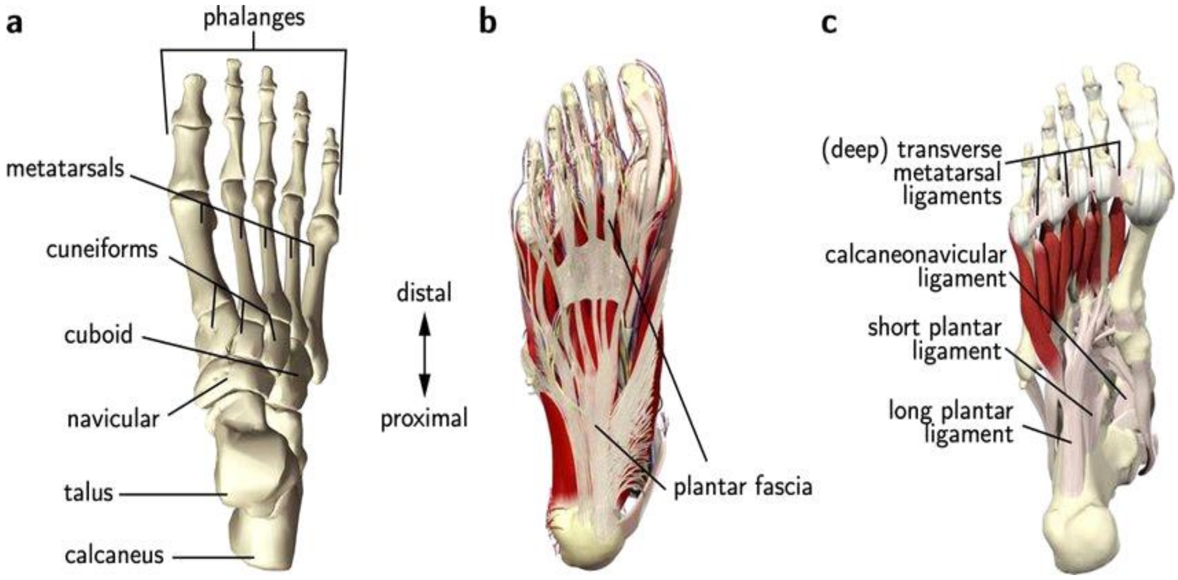


Fig-2. 2 : Biomechanics of foot[5]

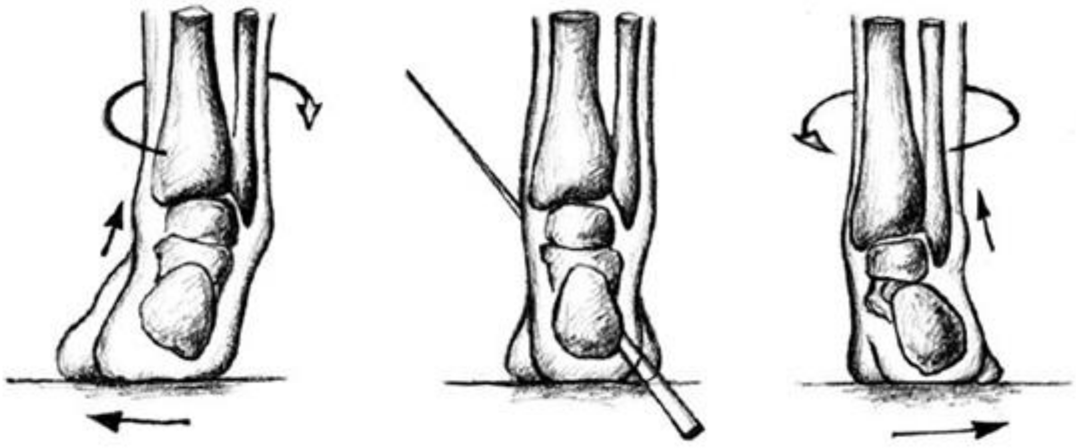


Fig 2.3 Neutral position shown in center, supinated position displayed on left and pronated position displayed on right[9].

These bones are connected by joints that allow for movement and support. The muscles and tendons in the foot help to control the movement of the bones and joints, and provide support and stability.

The arch of the foot is an important part of the biomechanics of the foot. The arch provides support and stability, and helps to distribute the weight of the body evenly across the foot. The muscles and tendons in the foot work together to control the movement of the arch, and to maintain the proper alignment of the bones and joints. Overall, the biomechanics of the human foot are complex and intricate, but they work together to support the body and allow for movement. Understanding the biomechanics of the foot can help to prevent injuries and improve performance in sports and other activities.

2.3 Flatfoot[1]

Flatfoot, also known as pes planus, is a condition in which the arch of the foot collapses and the entire sole of the foot comes into contact with the ground. This can cause pain, discomfort, and difficulty with activities such as walking, running, and standing for long periods of time.

There are several types of flatfoot, each with their own specific causes and symptoms. Congenital flatfoot, for example, is a condition present at birth, where the arch never develops. Acquired flatfoot is a condition that develops later in life, and can be caused by injury, disease, or the wear and tear of aging. The causes of flatfoot can vary, but it is often the result of a combination of factors, including genetics, obesity, and certain medical conditions. Flatfoot can also be caused by injury or overuse of the foot, such as from excessive running or standing for long periods of time.

Symptoms of flatfoot include pain and discomfort in the foot and ankle, difficulty standing for long periods of time, and difficulty walking or running. In severe cases, flatfoot can cause problems with the knees, hips, and lower back.

Treatment for flatfoot varies depending on the cause and severity of the condition. In some cases, treatment may include rest, ice, and over-the-counter pain medication. Orthotic devices, such as arch supports or custom-made shoe inserts, can also be helpful in relieving pain and discomfort. In more severe cases, surgery may be necessary to correct the condition. It is important to seek medical attention if you are experiencing pain or difficulty with activities such as walking or standing, as flatfoot can lead to further complications if left untreated.

Causes: [2]

- Deterioration or non-development of arches in childhood.
- Obesity.
- Injury to the foot or ankle.
- Rheumatoid arthritis
- More workload on feet in growing ages.

2.4 Shoe Design

Shoe design is a complex process that involves the collaboration of variety of individuals, including runners, craftsmen, technicians, scientists, and doctors. This article will focus on the complicated efforts that stand behind a relatively simple piece of athletic equipment: the running shoe.

The primary goal of running shoe engineers is to achieve an optimal shoe design for the “average” athlete. Average, in terms of human biomechanics, is a very tricky concept, since all people are anatomically and functionally different. Each individual is unique; differences in structure, movement, and gait pattern require footwear to vary from person to person. Efforts to meet this concern are further multiplied by the critical factors to be considered in the design of each shoe: shock absorption, flexibility, fit, traction, sole wear, breathability, weight, etc. Due to the diversity of the human form, it is impossible to provide for the needs of every runner on the planet. Shoe designers manage this overwhelming demand by supplying some standard, user-defined, foot-ground interface.

Biomechanics also plays an important role in the shoe testing process. Before the advent of dynamic mechanical methods for evaluating shoe performance, the primary concern of the shoe industry was to test materials for adhesion, attachment, seams, and fatigue due to the structural breakdown of the shoe with use. Biomechanical testing concentrates on the shoe in action and the resulting stresses imposed on the runner.

There are several methods for shoe testing. In the general process, advanced technology is used to determine the shoe requirements of runners in motion. Results of the test are analyzed and the opinions and feelings of the runner are considered. These results are combined to produce a shoe that both tests and fits well Factors tested both in and out of the laboratory include shock absorption, flexibility, heel counter stiffness, rear foot stability, overall rear foot control, sole wear test, traction and permeability to water. The testing procedures associated with these factors will not be described at length in this article. However, the following image provides a summary of selected running shoe features and corresponding design functions that have resulted from research.

2.5 Sole [6]

Soles are the fundamental part of shoes. They protect the bottom of the wearers' feet and ensure they do not develop severe pain afterward. Wearing shoes without soles is similar to walking barefoot. There are many different shoe materials available out there. You must know essential details about them to make informed decisions on materials to use for soles for your shoes. Selecting the best materials plays a significant role in ensuring comfort, functionality, and durability.

Soles of shoes are made from different materials because different types of shoes need different soles. For instance, the sole of hiking boots or running shoes cannot be the same as ballet shoes because they perform different functions.

2.6 Problem Statement

After acquiring knowledge in machine learning, the team recognized an opportunity to contribute to society by developing a model that addresses a problem that is often overlooked due to lack of awareness. Flatfoot is a condition that affects 20-30% of the population and can impact walking and running, as well as cause leg-related problems and associated health issues such as obesity and diabetes. Therefore, the team aimed to develop a prediction model using machine learning algorithms to identify flatfoot at a tender age with the hope of correcting the issue at an early stage.

Chapter-3

Prediction Models

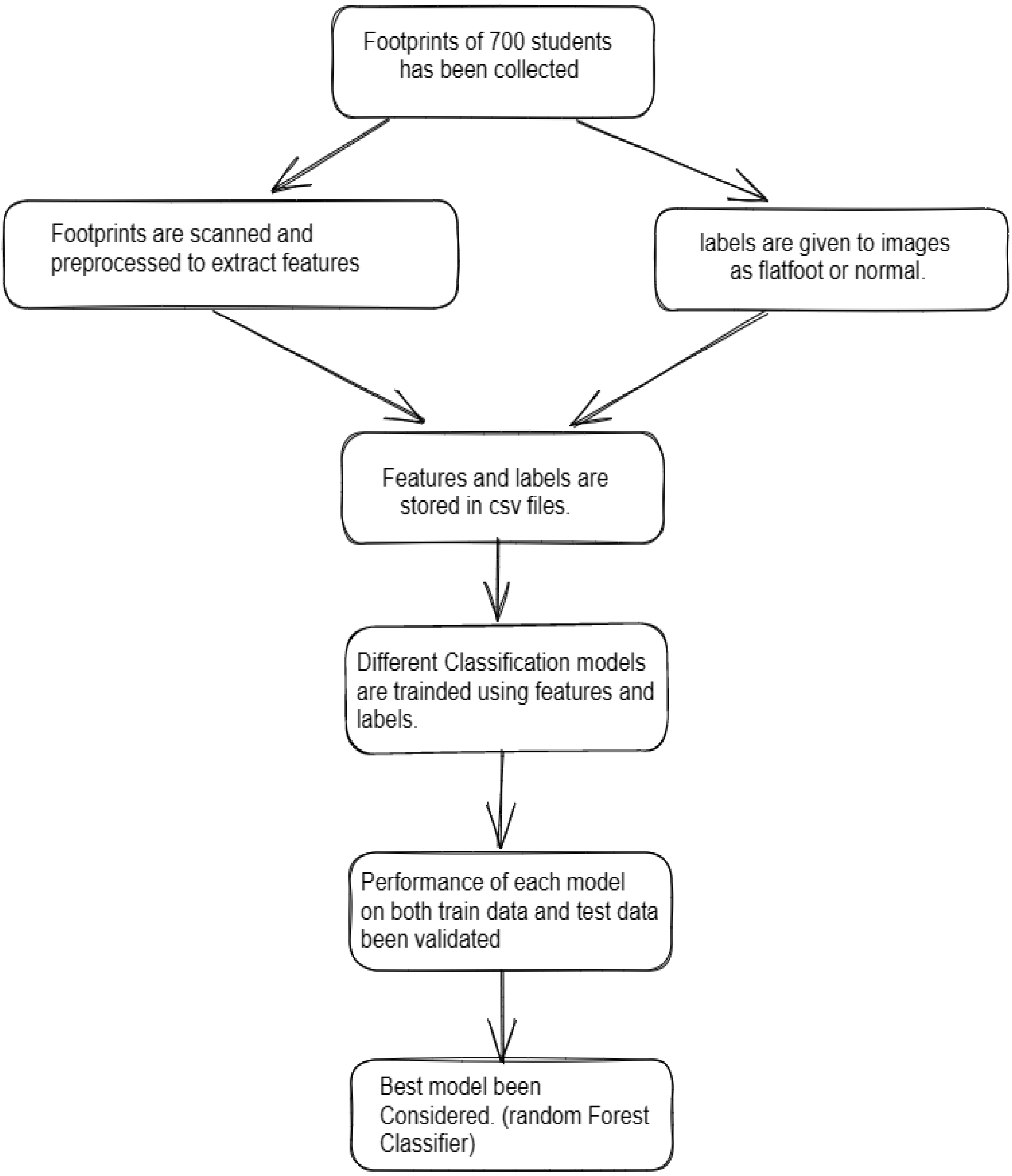

Footprints of 700 students has been collected
Footprints are scanned and preprocessed to extract features
labels are given to images as flatfoot or normal.
Features and labels are stored in csv files.
Different Classification models are trainded using features and labels.
Performance of each model on both train data and test data been validated
Best model been Considered. (random Forest Classifier)

Fig 3.1: Flowchart of the work carried out

3.1 Data Collection:

Collecting real time, authentic data of footprints is a tedious task as it involves the process of making setup for footprint collection, asking individuals and persuading them to give their footprints. We made a setup for this using Ink Pads with blue ink which are generally used for giving fingerprints. We used A-4 sheet papers to collect the prints. Then we scanned the sheets using a scanner. Along with footprint, student general details like age, foot size, height, weight are collected for later inference.

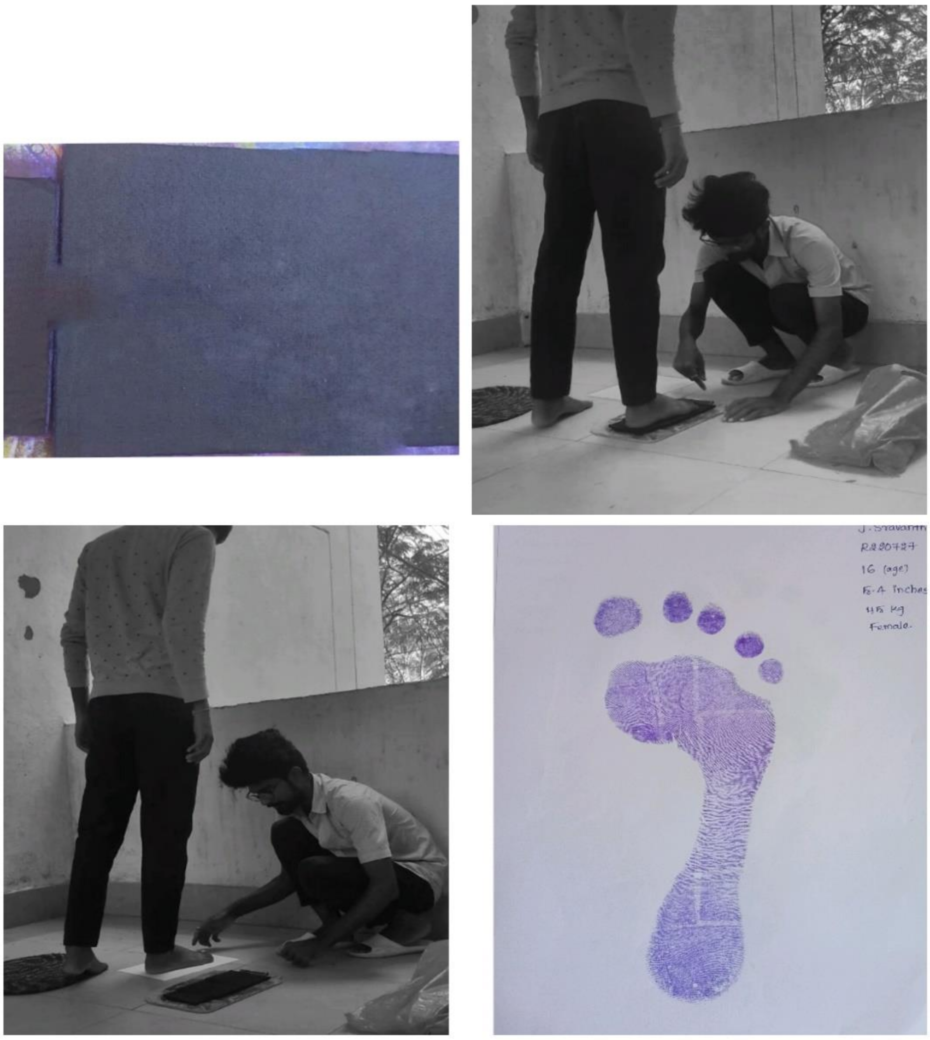


Fig 3.2 Data collection procedure. Ink pad setup, Imprinting, Resulted foot imprint

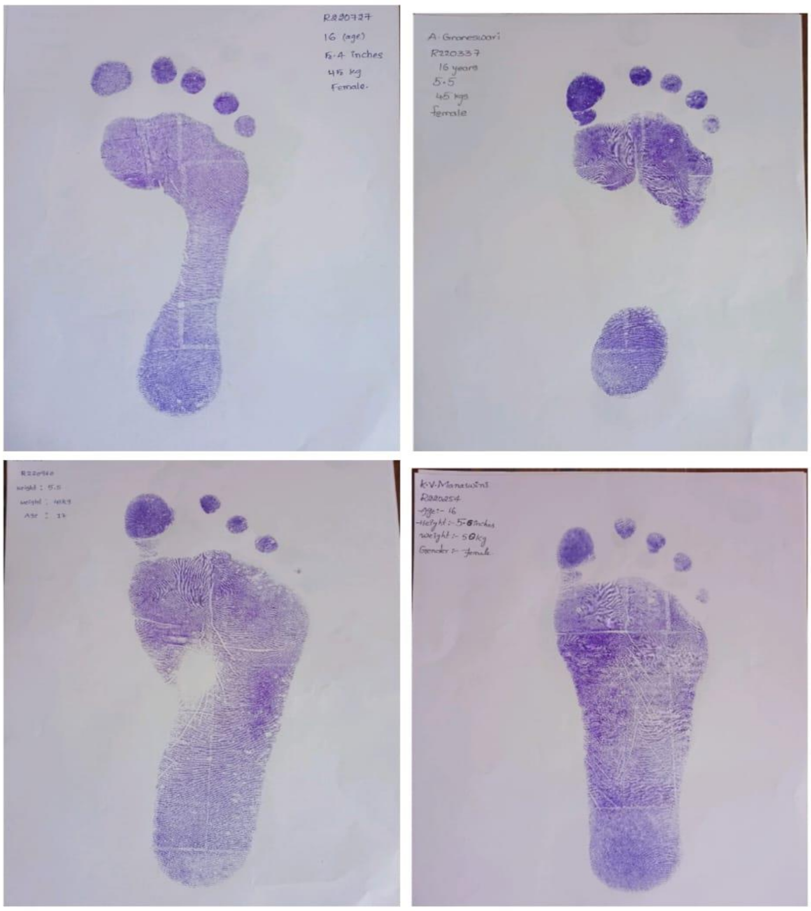


Fig 3.3 Different types of foot i.e. normal, high arch, moderate flat and flatfoot

3.2 Parameter Extraction

3.2.1 Sztriter-Godunov index (KY):

KY index is comprised of two parameters and represents the ratio of the length of AB line to AC. AC is a line plumbed from the center of medial longitudinal arch, (MLA) to the medial border of foot . AB line is defined as the distance from the center of MLA to the medial border of foot on the AC line. Based on this index, FF is present if the calculated number is < 0.47.

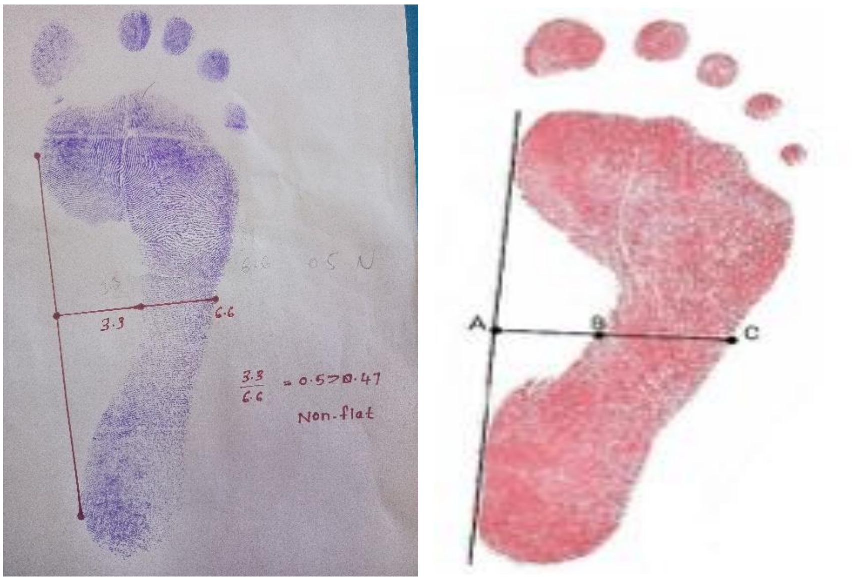


Fig-3.4 : (left) Image labelling (right) Sztriter-Godunov index[3]

Taken the parameters AC and AB and calculated the ratios for each image. If the ratio is < 0.47, it is considered as flatfoot, otherwise non-flat. The ratio, label for each image is taken and added to the csv file. The labels csv file contains the index and label of each image.

3.2.2 Image Preprocessing

Images are loaded and converted to grayscale using computer vision opencv tool. Then, Gaussian blur is applied to remove noise and thresholding to binarize the image. Next, the contours are found in the thresholded image and the largest contour is assume as the foot. The bounding rectangle of the largest contour is identified the image is cropped. We can adjust the parameters of the Gaussian blur and thresholding functions to achieve better results depending on the input image.

For each image features like arch length, arch-width, foot width, arch index and medial support are found. A feature list for storing the five features is created. The features of each image is appended into the list. Using the list a comma separated value extension, csv file is created with all the samples and their features. Labels dataset is a csv file containing index and label. Both the features.csv and labels.csv files are loaded into Google colab notebook IDE.

3.2.3 Extracting features from the Images:

To extract common features from footprints, image processing techniques are emplyed to identify the regions of interest (ROIs) in the footprints and extract the relevant features from those regions.

1. Arch Height: Arch height is the distance between the highest point of the arch and the ground. To extract the arch height, you can first identify the arch region in the footprint image using edge detection algorithms. Once identified the arch region is identified, the highest point of the arch is found and calculate the distance between that point and the ground.

2. Arch Length: Arch length is the distance between the two endpoints of the arch. To extract the arch length, the same edge detection techniques to identify the arch region are used and then calculate the distance between the two endpoints of the arch.

3. Foot Width: Foot width is the distance between the two widest points of the foot. To extract the foot width, the widest points of the foot are identified by using edge detection techniques and then calculate the distance between those points.

4. Arch Index: Arch index is the ratio of arch height to the foot length. To calculate the arch index, the foot length is extracted from the footprint image. This is done by identifying the heel and toe regions of the foot using edge detection techniques and then calculating the distance between those points. Once the foot length and arch height, are found the arch index as arch height divided by foot length are calculated.

5. Medial Support: To extract medial support from a footprint image, the arch of the foot can be analyzed to determine its height and level of support. This can be done by identifying the inner part of the foot, particularly the arch, and analyzing its shape and structure.

3.2.4 Over-Sampling the Flatfoot samples:

As a part of data exploration, label value counts are printed and found that label ‘N’ which is Non-Flatfoot is having 528 samples and label ‘F’ is having 103 samples.

It is found that the data has more non-flatfoot samples than flatfoot. It causes data imbalance leading to poor performance of the models. To overcome this issue SMOTE technique is used.

SMOTE, which stands for Synthetic Minority Over-sampling Technique, is a data augmentation technique used in machine learning to address the issue of class imbalance in the dataset. Class imbalance occurs when the number of instances in one class is significantly smaller than the number of instances in another class.

In such cases, the machine learning algorithm may not be able to learn the patterns in the minority class effectively, leading to poor performance. SMOTE helps to address this issue by generating synthetic examples of the minority class, which can be used to balance the number of instances in each class.

By using SMOTE, we can increase the number of instances in the minority class, which can help to improve the accuracy of the machine learning model. This is particularly important in applications where the minority class is of interest, such as fraud detection or medical

g Smote technique we have up-sample the flatfoot samples from 103 to 528 which matches the non-flatfoot samples count.

We have diagnosis, where correctly identifying instances of the minority class is crucial.

By up-sample the ‘F’ label using SMOTE technique from imblearn.over_sampling. This helps the ML models in learning the ‘F’ though it has very less samples compared to ‘N’.

Counter({'F': 528, 'N': 528})

3.3 Scikit -Learn:

Scikit-learn, also known as sklearn, is a popular Python library for machine learning. It provides a wide range of tools for data preprocessing, feature extraction, model selection, and evaluation, as well as various supervised and unsupervised learning algorithms.

Scikit-learn is built on top of other scientific libraries in Python, such as NumPy, SciPy, and matplotlib, and is designed to work seamlessly with these libraries. It provides a consistent and easy-to-use API for all of its algorithms, making it easy to experiment with different models and compare their performance. Scikit-learn includes a wide range of supervised learning algorithms, such as linear and logistic regression, decision trees, random forests, support vector machines, and neural networks. It also includes various unsupervised learning algorithms, such as clustering, dimensionality reduction, and anomaly detection.

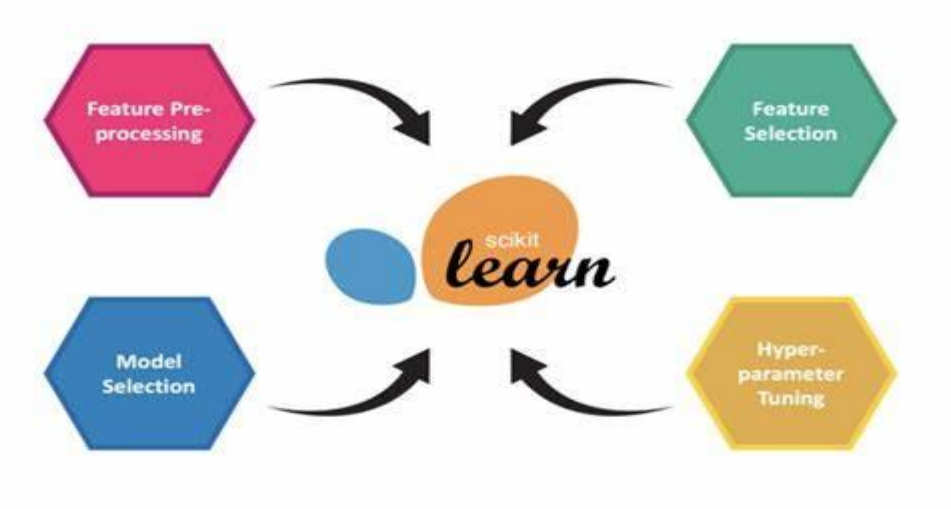


Fig-3.5: scikit-learn library [5].

In addition to its algorithms, scikit-learn also provides various tools for model selection and evaluation, such as cross-validation, hyper parameter tuning, and model persistence. It also includes many datasets for experimenting and benchmarking. Scikit-learn is widely used in academia and industry for various machine learning applications, such as natural language processing, image classification, and predictive modeling. Its popularity is due to its ease of use, flexibility, and scalability, as well as its active development community and extensive documentation.

3.4 Train-Test Split:

Now we will split the data into train test data. The train-test split in machine learning and data analysis is used to evaluate the performance of a model on an independent dataset. The main reason for using this technique is to estimate the generalization performance of a model, i.e how well it can perform on unseen data.

The process involves splitting a given dataset into two subsets: one for training the model and the other for testing the model’s performance. The training set is used to train the model, while the test set is used to evaluate how well the model performs on the data it has never seen before.

By using a train-test split, we can prevent the model from simply memorizing the data it has seen during training, which could result in overfitting. Overfitting occurs when the model fits the training data so closely that it does not generalize well to the new data .A train-test split helps us to evaluate the model’s ability to generalize to new data and identify any overfitting that may have occurred during training.

3.5 Sklearn Metrics

Precision, recall, accuracy, F1-score, and macro average are evaluation metrics used to measure the performance of a classification model. Each metric provides different information about the model's performance, and should be used in combination to obtain a comprehensive understanding of the model's strengths and weaknesses

1. **Precision:** Precision is the proportion of true positives (TP) among the total number of predicted positives (TP + FP). It measures how many of the predicted positive instances are actually positive. A high precision indicates that the model has a low false positive rate.

2. **Recall**: Recall is the proportion of true positives (TP) among the total number of actual positives (TP + FN). It measures how many of the actual positive instances are correctly predicted as positive. A high recall indicates that the model has a low false negative rate.
3. **Accuracy**: Accuracy is the proportion of correct predictions (TP + TN) among the total number of predictions. It measures how well the model performs in predicting both positive and negative instances. A high accuracy indicates that the model has a low overall error rate.

4. **F1-score:** F1-score is the harmonic mean of precision and recall, calculated as 2 * (precision * recall) / (precision + recall). It measures the balance between precision and recall, and is useful in cases where both high precision and high recall are desired.

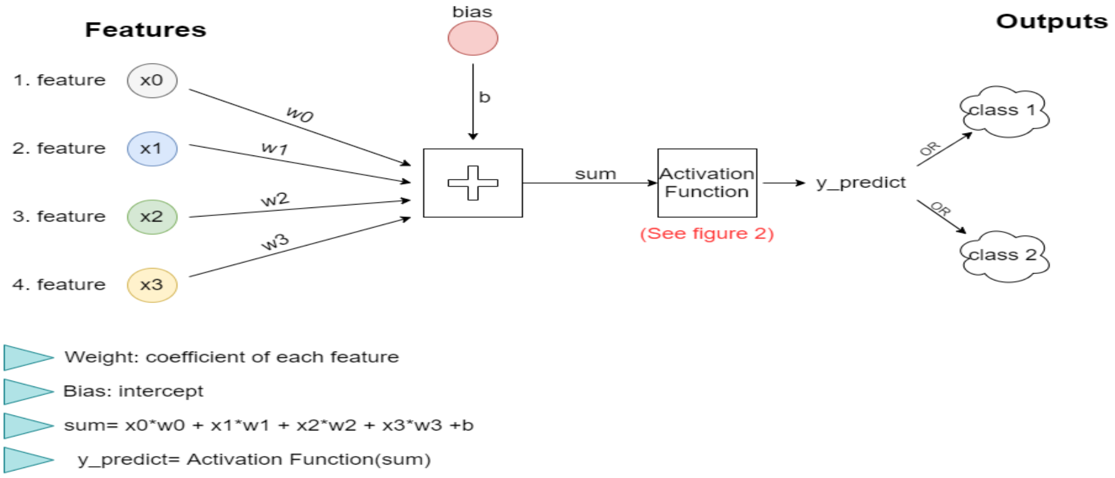
5. **Macro average:** Macro average is the average of the evaluation metrics calculated for each class. It is used in multi-class classification problems to measure the overall performance of the model across all classes, regardless of the class imbalance. It is computed by taking the arithmetic mean of the evaluation metric for each class.

6. **Confusion Matrix:** A confusion matrix is a table that is often used to evaluate the performance of a machine learning model for a binary classification problem. The table compares the actual labels of a dataset with the predicted labels generated by the model. The table consists of four components, namely True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN).
7. **True Positives (TP)**: The number of instances that are truly positive and are predicted to be positive by the model.
8. **False Positives (FP):** The number of instances that are actually negative but are predicted to be positive by the model.
9. **True Negatives (TN**): The number of instances that are truly negative and are predicted to be negative by the model.
10. **False Negatives (FN):** The number of instances that are actually positive but are predicted to be negative by the model.

3.6 Logistic Regression:

Logistic regression is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability that an instance of belonging to a given class or not. It is a kind of statistical algorithm, which analyze the relationship between a set of independent variables and the dependent binary variables. It is a powerful tool for decision-making. For example email spam or not.

It is referred to as regression because it takes the output of the linear regression function as input and uses a sigmoid function to estimate the probability for the given class. The difference between linear regression and logistic regression is that linear regression output is the continuous value that can be anything while logistic regression predicts the probability that an instance belongs to a given class or not.



Weight: coefficient of each feature

Bias: intercept

sum= x0*w0 + x1*w1 + x2*w2 + x3*w3 +b

y_predict= Activation Function(sum)

Fig -3.6: Logistic Regression Algorithmic Flowchart [5]

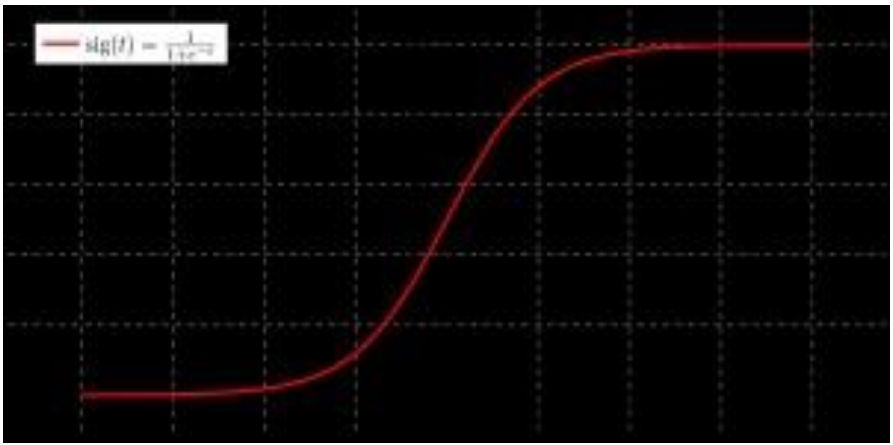


Fig – 3.6 : Logistic Regression sigmoid curve[5]

The results of the model performance are:

Train Report:

Train Score: 0.6465997770345596

Metrics	precision	recall	f1-score	Support
F	0.68	0.55	0.61	446
N	0.62	0.75	0.68	451
Accuracy			0.65	897
Macro avg	0.65	0.65	0.64	897
Weighted avg	0.65	0.65	0.64	897

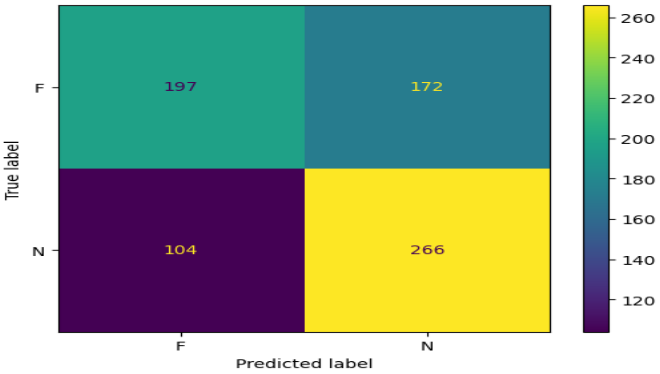


Fig 3.7 Confusion Matrix of Logistic regression

Test Report:

Test Score : 0.6037735849056604

	precision	recall	f1-score	Support
F	0.63	0.56	0.59	82
N	0.58	0.65	0.61	77
Accuracy			0.60	159
Macro avg	0.61	0.61	0.60	159
Weighted avg	0.61	0.60	0.60	159

3.6.1 Weights of the Model:

Weights are the parameters that a model learns from the training data to make predictions or classifications.

During the training process, a model iteratively adjusts its weights to minimize the difference between the predicted output and the actual output of the training data. These weights capture the relationships and patterns in the data that are relevant for making accurate predictions.

For example the weights learnt by the Logistic Regression model in this case is:

```
array([[-1.49681177e-03, -3.59575486e-02, 3.77860968e-02,
7.30994440e-02, 9.12955014e-05]])
```

3.7 SGD Classifier:

The word ‘stochastic‘ means a system or process linked with a random probability. Hence, in Stochastic Gradient Descent, a few samples are selected randomly instead of the whole data set for each iteration. In Gradient Descent, there is a term called "batch" which denotes the total number of samples from a dataset that is used for calculating the gradient for each iteration. In typical Gradient Descent optimization, like Batch Gradient Descent, the batch is taken to be the whole dataset. Although using the whole dataset is really useful for getting to the minima in a less noisy and less random manner, the problem arises when our dataset gets big.

Suppose, there are million samples in the dataset, so a typical Gradient Descent optimization technique is used, we will have to use all of the one million samples for completing one iteration while performing the Gradient Descent, and it has to be done for every iteration until the minima are reached. Hence, it becomes computationally very expensive to perform.

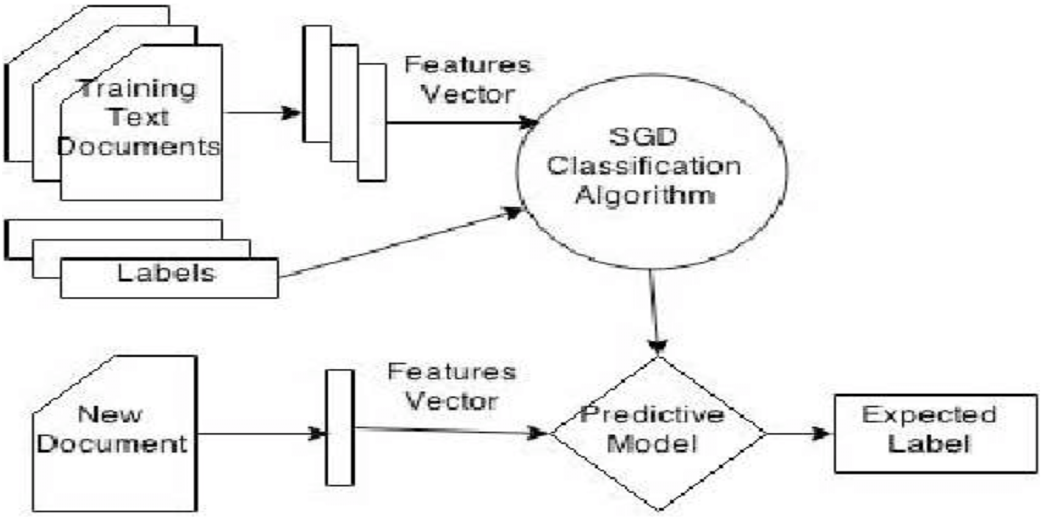


Fig – 3.8: SGD Classifier Flowchart[5]

This problem is solved by Stochastic Gradient Descent. In SGD, it uses only a single sample, i.e., a batch size of one, to perform each iteration. The sample is randomly shuffled and selected for performing the iteration. Stochastic Gradient Descent (SGD) is a variant of the Gradient Descent algorithm used for optimizing machine learning models. In this variant, only one random training example is used to calculate the gradient and update the parameters at each iteration.

The results of the model performance are:

Train Report:

Training Score 0.6048714479025711

	precision	recall	f1-score	Support
F	0.64	0.48	0.54	366
N	0.59	0.73	0.63	372
Accuracy			0.60	739
Macro avg	0.61	0.60	0.60	739
Weighted avg	0.61	0.60	0.60	739

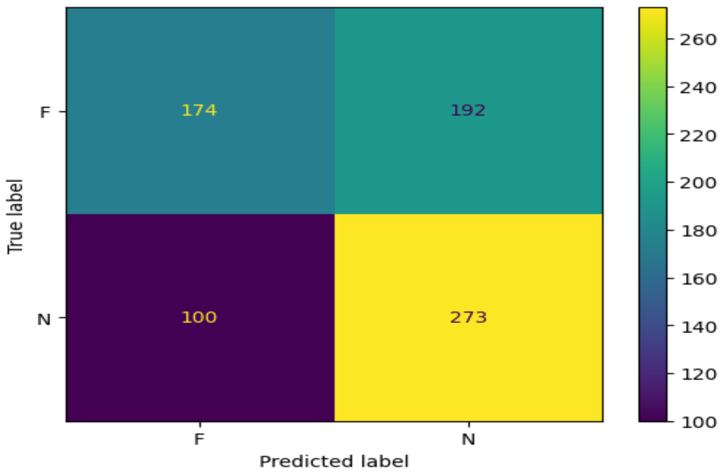


Fig -3.9: Confusion matrix of SGD Classifier model

Test Report:

Test Score: 0.5583596214511041

	precision	recall	f1-score	Support
F	0.59	0.45	0.51	162
N	0.54	0.67	0.60	155
Accuracy			0.56	317
Macro avg	0.56	0.56	0.55	317
Weighted avg	0.56	0.56	0.55	317

3.8 Support Vector Classifier (SVC):

Support Vector Machine Classifier is a type of machine learning algorithm used for classification tasks, particularly in binary classification problems, where the goal is to classify instances into one of two classes. In an SVC model, the algorithm learns to find the optimal boundary (or hyperplane) that separates the two classes of data points with the maximum margin. The data points that lie closest to the boundary and have the largest margin are known as support vectors.

SVCs are popular because they can handle both linearly separable and non-linearly separable datasets by using a technique called the kernel trick. The kernel trick maps the original feature space into a higher-dimensional space, where the data becomes more separable. Overall, SVCs are a powerful and versatile tool for classification tasks, and have been successfully applied to a wide range of problems in various domains such as text classification, image classification, and bioinformatics.

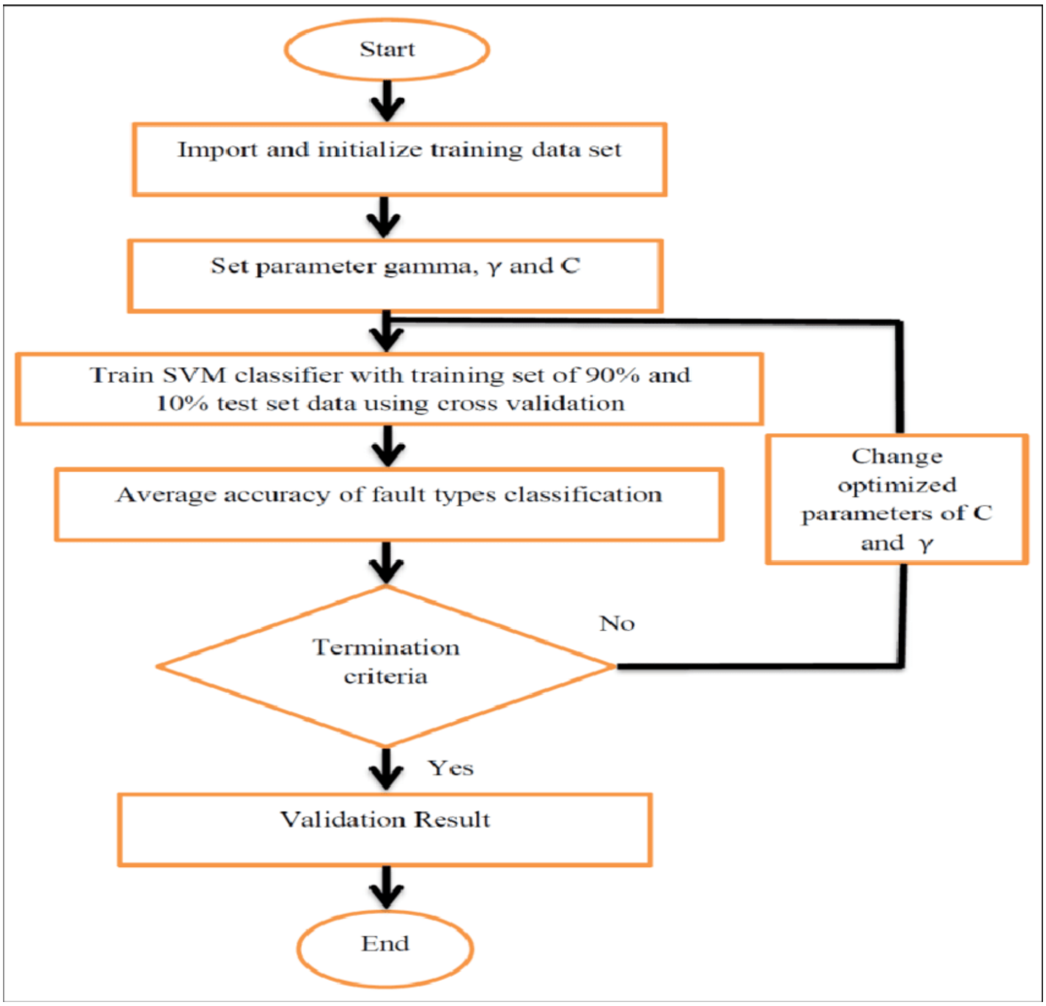

Start
Import and initialize training data set
Set parameter gamma, γ and C
Train SVM classifier with training set of 90% and 10% test set data using cross validation
Average accuracy of fault types classification
Change optimized parameters of C and γ
No
Termination criteria
Yes
Validation Result
End

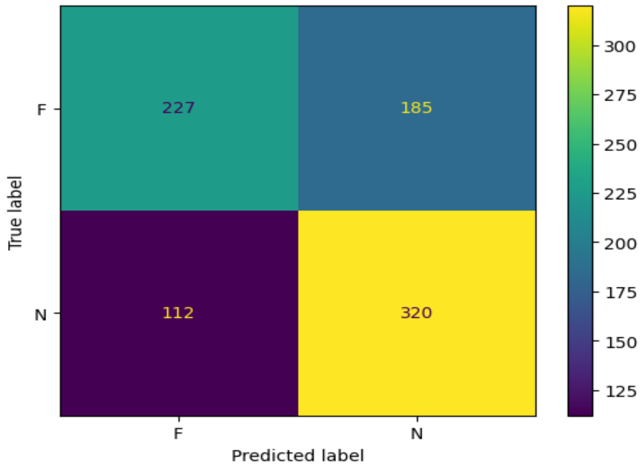
Fig -3.10: Support vector machine Classifier [5]

The results of the model performance are:

Training Report

Train Score: 0.6658767772511849

	precision	recall	f1-score	Support
F	0.67	0.55	0.60	412
N	0.63	0.74	0.68	432
Accuracy			0.65	844
Macro avg	0.65	0.65	0.64	844
Weighted avg	0.65	0.65	0.634	844



Test Report:
Test Score: 0.6403785488958991

	precision	recall	f1-score	Support
F	0.66	0.53	0.59	159
N	0.61	0.72	0.66	158
Accuracy			0.63	317
Macro avg	0.64	0.63	0.63	317
Weighted avg	0.64	0.63	0.63	317

3.9 Naive Bayes Algorithm:

Naive Bayes is a probabilistic machine learning algorithm that is commonly used for classification tasks. It is based on Bayes' theorem and assumes that the features in the data are conditionally independent given the class labels. Naive Bayes is called "naive" because it makes the simplifying assumption of feature independence, which may not hold in many real-world scenarios.

Multinomial Naive Bayes is a variant of Naive Bayes that is specifically designed for text classification tasks, where the features are typically word counts or frequencies. It assumes that the features are generated from a multinomial distribution and uses the maximum likelihood estimation to estimate the probabilities of each feature given the class label. Multinomial Naive Bayes is often used in natural language processing tasks such as sentiment analysis, spam filtering, and topic classification.

Compared to other machine learning algorithms, Naive Bayes is computationally efficient and requires a relatively small amount of training data. However, its performance may be suboptimal in situations where the feature independence assumption is violated or when the features are not well-suited to the multinomial distribution. In these cases, more sophisticated models such as logistic regression or support vector machines may be more appropriate.

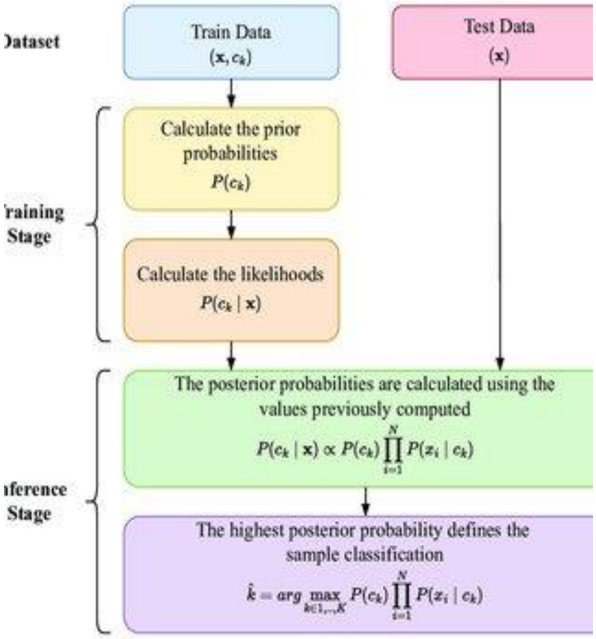
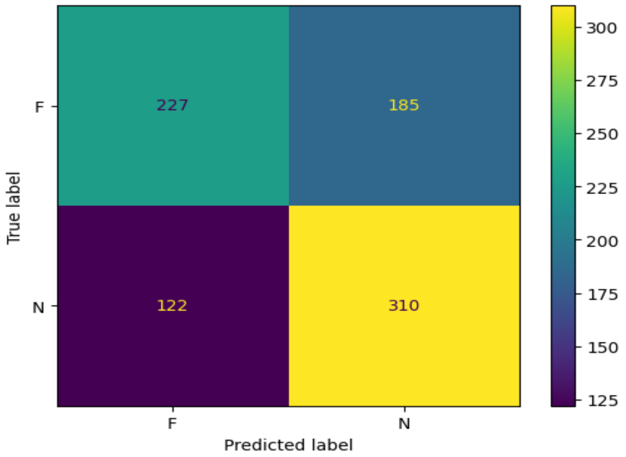

ataset
Train Data
$(\mathbf{x}, c_k)$
Test Data
$(\mathbf{x})$
raining Stage
Calculate the prior probabilities
$P(c_k)$
Calculate the likelihoods
$P(c_k \mid \mathbf{x})$
ference Stage
The posterior probabilities are calculated using the values previously computed
$P(c_k \mid \mathbf{x}) \propto P(c_k) \prod_{i=1}^{N} P(x_i \mid c_k)$
The highest posterior probability defines the sample classification
$\hat{k} = arg \max_{k \in 1,\dots,K} P(c_k) \prod_{i=1}^{N} P(x_i \mid c_k)$

Fig -3.11: Naïve Bayes classifier [5]

Training Report:

Training Score: 0.6362559241706162

	precision	recall	f1-score	Support
F	0.65	0.55	0.60	412
N	0.63	0.72	0.67	432
Accuracy			0.64	844
Macro avg	0.64	0.63	0.63	844
Weighted avg	0.64	0.64	0.63	844



Test Report:
Test Score : 0.6246056782334385

	precision	recall	f1-score	Support
F	0.65	0.51	0.57	156
N	0.61	0.74	0.67	161
Accuracy			0.62	317
Macro avg	0.63	0.62	0.62	317
Weighted avg	0.63	0.62	0.62	317

3.10 Decision Tree Classifier:

A decision tree classifier is a type of supervised learning algorithm that is commonly used for classification and regression tasks. It works by recursively partitioning the input space into smaller and smaller subsets based on the values of the input features, until a stopping criterion is met. The resulting tree structure is used to make predictions for new data points by following the path from the root node to the appropriate leaf node.

In a decision tree, each internal node represents a decision on a feature, and each branch represents the possible outcomes of that decision. The leaf nodes represent the predicted class labels or target values. The decision on which feature to split on at each node is made using a measure of impurity, such as entropy or Gini impurity. The goal is to find the split that maximally reduces the impurity of the resulting subsets.

Decision trees have several advantages, such as being easy to interpret and visualize, as well as being able to handle both numerical and categorical features. They can also handle missing values and outliers in the data. However, they may suffer from overfitting if the tree is too deep or if the training data is noisy or imbalanced. Techniques such as pruning, ensemble methods, and boosting can be used to address these issues and improve the performance of decision trees.

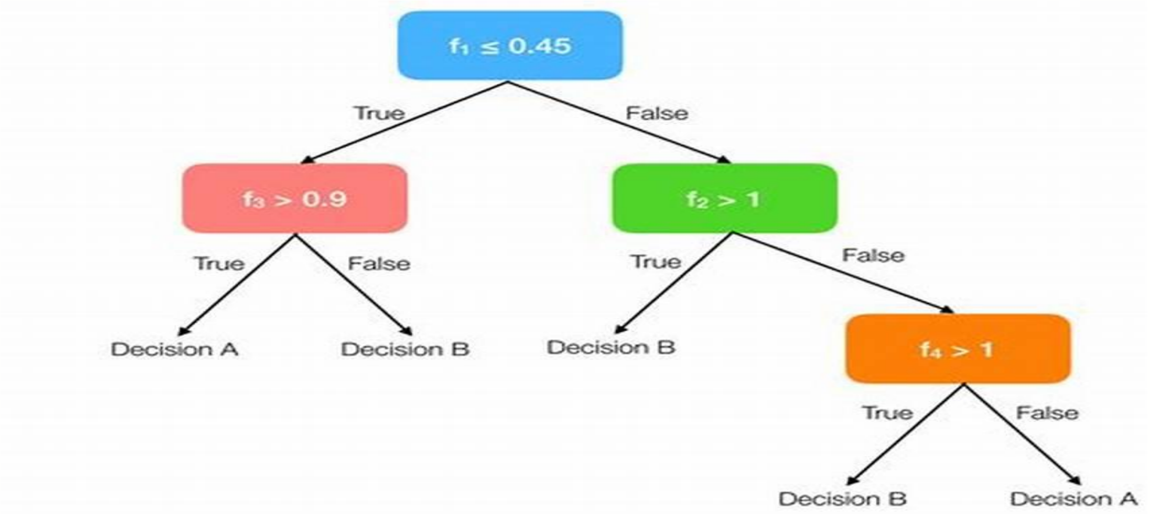


Fig -3.12: Decision tree classifier Flowchart [5]

The results of the model performace are:

Training Report:

Training Score : 0.9774881516587678

	precision	recall	f1-score	Support
F	0.97	0.99	0.98	412
N	0.99	0.97	0.98	432
Accuracy			0.98	844
Macro avg	0.98	0.98	0.98	844
Weighted avg	0.98	0.98	0.98	844

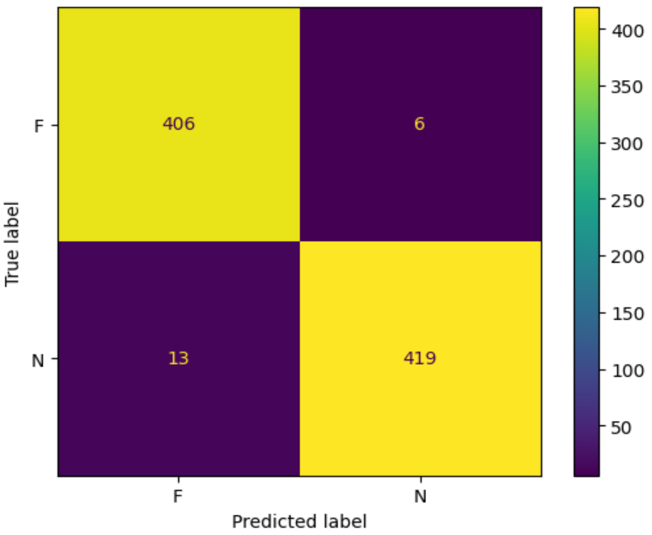


Fig -3.13 : Confusion matrix for Decision Tree Classifier

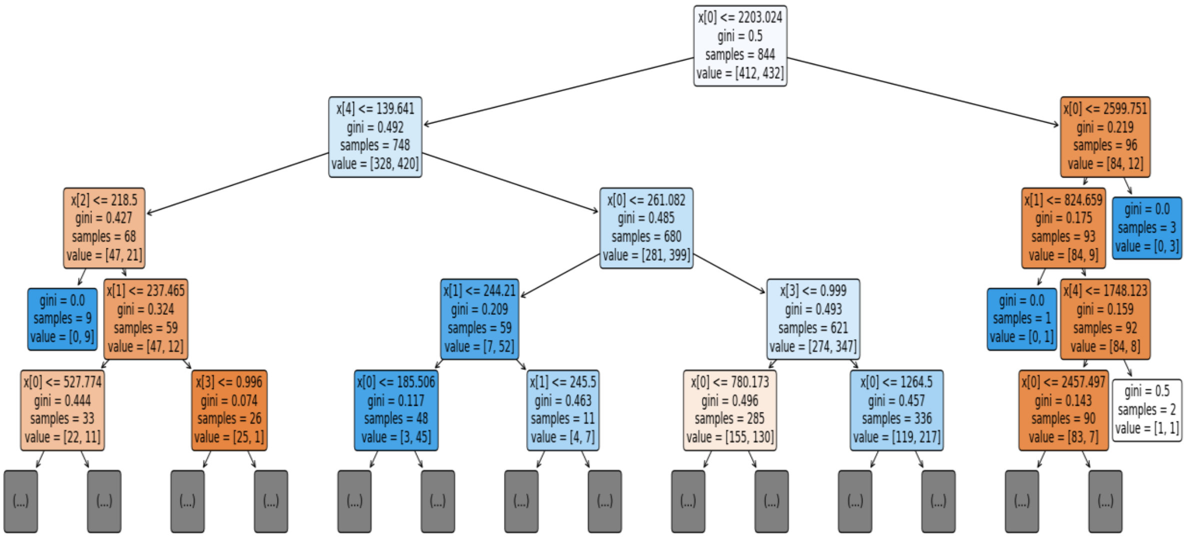


Fig -3.14 : Decision tree for the training data

Test Report:
Test Score : 0.7169811320754716

	precision	recall	f1-score	Support
F	0.74	0.75	0.74	116
N	0.69	0.68	0,68	96
Accuracy			0.72	212
Macro avg	0.71	0.71	0.71	212
Weighted avg	0.72	0.72	0.72	212

3.11 Random Forest Classifier:

Random Forest Classifier is an ensemble learning method that combines multiple decision trees to improve the performance and reduce overfitting. It is a popular machine learning algorithm for classification, regression, and feature selection tasks.

Random Forest Classifier builds multiple decision trees on randomly selected subsets of the training data and the features. The process of selecting a subset of the data and features is called bagging or bootstrap aggregation. Each decision tree is trained on a different subset of the data and features, resulting in a set of diverse models that can capture different patterns in the data.

During prediction, each tree in the forest independently predicts the class label or target value of the input data, and the final prediction is based on the majority vote or average of the individual predictions. This approach reduces the risk of overfitting and improves the accuracy and robustness of the model.

Random Forest Classifier has several advantages, including its ability to handle high-dimensional data, non-linear relationships, and missing values. It also provides estimates of feature importance, which can be useful for feature selection and interpretation. However, it may require more computational resources than single decision trees, and its performance may depend on the choice of hyperparameters such as the number of trees, the maximum depth of the trees, and the size of the feature subsets.Random Forest Classifier is a widely used algorithm in various applications such as bioinformatics, finance, marketing, and image processing.

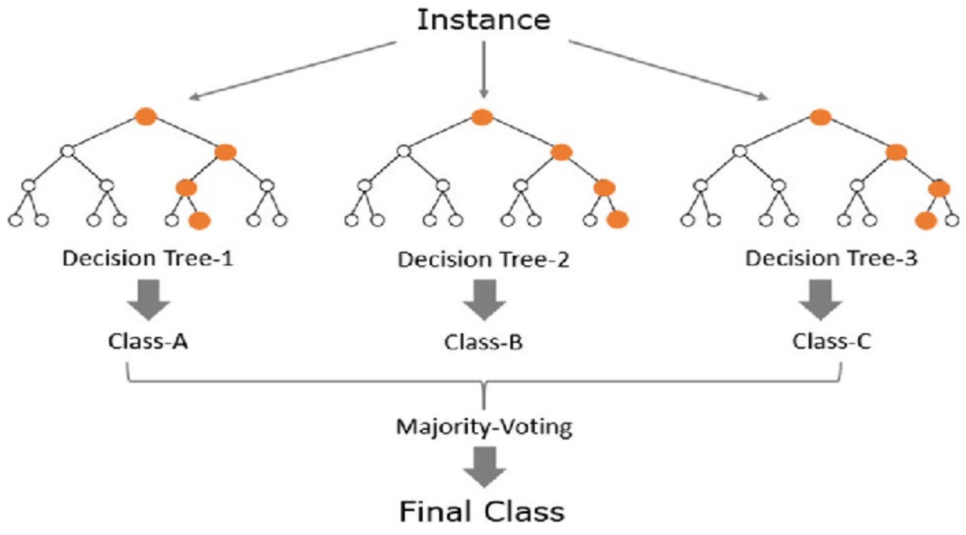


Fig -3.15: Random Forest Classifier Flowchart[5]

The results of the model performance are:

Training Report:

Training Score : 0.9976303317535545

	precision	recall	f1-score	Support
F	1.00	1.00	1.00	412
N	1.00	1.00	1.00	432
Accuracy			1.00	844
Macro avg	1.00	1.00	1.00	844
Weighted avg	1.00	1.00	1.00	844

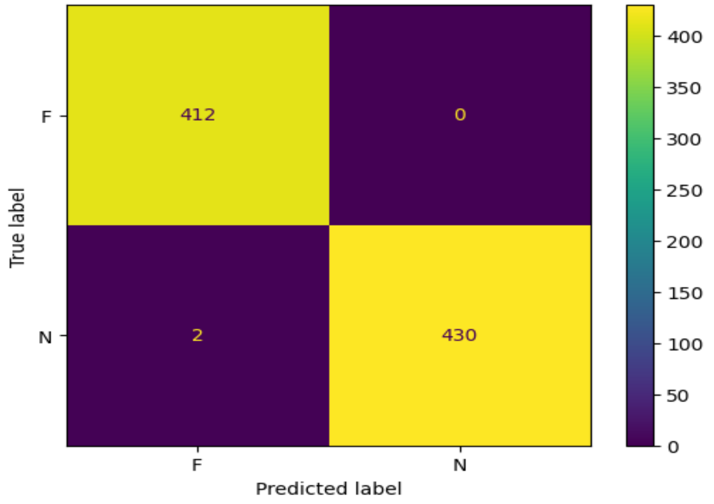


Fig-3.16 : Confusion Matrix of Random Forest Classifier

Test Report:
Test Score: 0.75934

	precision	recall	f1-score	Support
F	0.76	0.78	0.77	116
N	0.73	0.71	0.72	96
Accuracy			0.75	212
Macro avg	0.75	0.75	0.75	212
Weighted avg	0.75	0.75	0.75	212

3.12 Best shoe for flat foot[8]:

Sole Support

Most experts agree that wearing shoes with a supportive sole is best for flat feet. Look for a shoe with a firm but cushioned insole to support the surface of your foot. The sole of the shoe should be flexible but not floppy. It should move with your foot and provide support while you walk or stand.

Arch Support

Look for shoes that have extra support in the middle of the sole. A raised arch support can take pressure off the middle of your foot. This can help align your foot correctly so you don't twist your ankles, knees, or hips as you move.

Heel Support

Flip flops or sandles that don't wrap around your heel are a no-no for flat feet. These are too loose and don't provide the kind of supportive soles you need for flat feet. You want shoes that enclose your heel so that the sole of the shoe and your foot move together. Look for shoes that come up high enough in the back to keep your heel in contact with the sole of the shoe at all times.

Raised Heel

Shoes with a slightly raised heel take pressure off your mid-sole and relieve foot pain. The heel height doesn't need to be dramatic. A heel of an inch or two can be sufficient to help with flat foot pain. Look for raised heels in dress shoes as well as athletic shoes with a thicker heel sole.

Lace-ups

Shoes with laces or velcro straps allow you to customize the way the upper part of the shoe fits around your foot. You can adjust the lacing to accommodate the width of different parts of your foot so you get even support. Giving your foot the right amount of room inside the shoe allows you to evenly distribute your weight on your sole. By making sure your upper foot is snugly supported in the shoes, you can keep the sole firmly in place. Your foot will stay in contact with the supportive bottom of your shoe rather than sliding or lifting inside the shoe. This can help you correct gait issues you might develop when you're trying to hold a looser shoe in place.

3.13 Recommended models available in market for flat foot:

- **Best Everyday Sneakers:** Munro Gabbie Sneaker
- **Best Booties:** Munro Lexi Boot
- **Best Orthopedic:** Alegria Paloma Slip-On
- **Best Running Sneakers:** Asics Gel Kayano 28 Sneaker
- **Best Sandals:** Birkenstock Arizona Soft Footbed Sandal
- **Best Walking Shoes:** New Balance 1540v3 Sneaker
- **Best Slippers:** Vionic Indulge Sadie Mule Slipper
- **Best Heels:** Coach Cora Block Heel Penny Loafer
- **Best Flats:** Mephisto Emilie Flats
- **Best Flip-Flops:** Oofos Oolala Sandal
- **Best Slip-On Sneakers:** Adidas Puremotion Adapt Running Shoe

Chapter 4

4.1 Result and Discussions:

Parameters required to train a machine learning model were extracted from the footprint images. The training samples were analyzed manually to classify the type of foot as flatfoot or normal arch foot. SG index labeling was performed on the training samples manually. Different ML classification algorithms like Logistic Regression, SGD Classifier, Support vector Classifier, Multinomial NB, Decision tree classifier and Random Forest Classifier were trained using collected samples data.

Among the all classification models, Random Forest classifier trained predictive model has given the best results. It performed with a train score of 0.99 and a test score of 0.76 on the scale of 0 to 1. So Random Forest Classifier can be employed for solving this problem of Flat-foot prediction.

4.2 Conclusion

The team conducted extensive research on the biomechanics of the foot and ankle to identify the causes of flatfoot. Data on various types of shoes and insoles suitable for different situations was collected. The team gathered the necessary data for analysis and implementation of the problem, including researching machine learning models that could yield accurate results. Several algorithms, such as Logistic Regression, SGD Classifier, Support Vector Classifier, Multinomial NB, Decision Tree Classifier, and Random Forest Classifier were experimented. Among the various models tested, the Random Forest Classifier demonstrated the most promising results. By predicting the type of foot, the team aimed to recommend shoes that would best suit the specific foot type. The team provided detailed information about the shoes and their properties that would be suitable for individuals with flatfoot.

4.3 Future Scope

This project focused on the prediction of flatfoot, a condition that affects a significant percentage of the Indian population. Despite its prevalence, flatfoot is a neglected problem in India. Through this project, the team aimed to develop a machine learning algorithm that could accurately predict flatfoot using footprints and recommend suitable shoes based on the individual's foot shape. This project is an innovative experiment in India, as only a few foreign companies have conducted research on foot type prediction using sophisticated machine learning algorithms. In future, the team envisions the development of a machine that suggests suitable shoes for particular foot types using footprints. To improve the accuracy of the project, the team proposes the use of photogrammetry for creating 3D models of feet to measure and analyse the foot arch. This approach would yield more reliable and realistic data for the project. The team plans to collect samples of feet in 3D using photogrammetry and plot graphs of footprints in 3D to better understand their distributions.

Overall, this project provides valuable insights into the potential of machine learning to address neglected health problems such as flatfoot, and highlights the importance of early intervention and personalized recommendations for appropriate footwear.

Colab Link:
https://colab.research.google.com/drive/1S8XEKjHCSCU9mbxA-uknARQ9Lbk6BozI?usp=sharing

Features data:
https://drive.google.com/file/d/1U8tPRHCtgzY4giuvSVaAhqDDFQB8R039/view?usp=sharing

Labels data:
https://drive.google.com/file/d/1y2Q09BGlgESwfJDxWF3KxOK2uZ37DT/view?usp=sharing

References

1. Lung- Yoon Kim et al"Flat-Feet Prediction Based on a Designed Wearable Sensing Shoe, " IEEE Access..vol.8.
2. https://www.medicalnewstoday.com/articles/168608 Dated 26-01-2023
3. Robert Donatelli," Normal Biomechanics of the Foot and Ankle,"American Physical Theapy Association, November 1985, Vol. 7, No. 3.
4. Leila Ghazaleh et al"Comparing Three Footprint Grades to Evaluate Footprint Indexes for Flat Foot Diagnosis,"July 2019. Volume 9. No. 3.
5. Htttp://www.comunitymade.com/posts/different-materials-for-the-soles- Dated 06-01-2023.
6. https://www.dataversity.net/a-brief-history-of-machine-learning/ "A Brief History of Machine Learning" By Keith D. Foote on December 3, 2021.
7. The 12 Best Walking Shoes for Flat Feet of 2023 | Tested by Verywell Fit Dated on August 11, 2021.
8. https://www.health.com/style/walking-shoes-high-arches published on March 16, 2022