3.0 Data

Data is the basic and the most important thing in Machine Learning. more data leads more training accuracy. before training any Machine Learning or Deep Learning model we have to first collect the data and preprocess it. sometimes we have to extract the usefull data and enhance it in a reaspective way. also we have to make sure that our collected data contains all the parameters and all the possibilities of a perticular scenario. then we take the final data for the training.

3.1 Data Collection

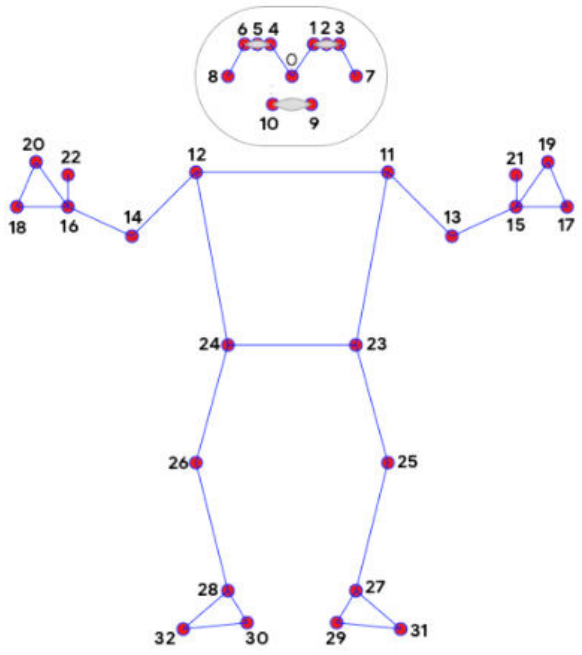
3.1.1 Data Format

In our usecase we have to collect the data of a person doing a perticular exercise. and this data should be in the continues form ie.data should contain the moving coordinate points. so we have to collect the data in the video format, and that video must have only one person doing a exercise in a correct manner and the coordinate points should be clearly visible.

3.1.2 Data Capture

For using this data we have to convert our video data to the categorical format. for that we are using Mediapipe Blazepose, an advanced pretrained Deep Learning Model that can detect the human pose landmarks in real time.

MediaPipe BlazePose is a ML solution for high-fidelity body pose tracking, inferring 33 3D landmarks and background segmentation mask on the whole body from RGB video frames utilizing the BlazePose research that also powers the ML Kit Pose Detection API. Current state-of-the-art approaches rely primarily on powerful desktop environments for inference, whereas the method achieves real-time performance on most modern mobile phones, desktops/laptops, in python and even on the web.



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

3.1.2 Categoriacal Data

After using the pretrained model we have to save this data in a categorical format, therefor we are saving this data in a comma sepereted value (csv) file. we write a python script for applying the pretrained model on the video data and gathering the coordinates of landmarks we capture and finally store the whole data in a csv file

some info about video_to_csv.py

3.2 Data Extraction

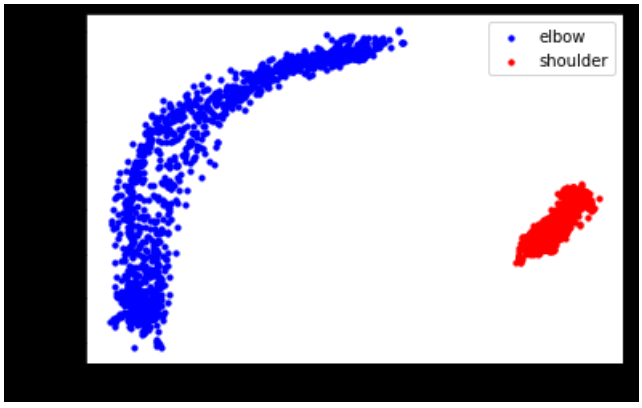
After getting the final raw data we have to extract the usefull data from it. this process depends on the type of exercise that the user doing. in a perticualr exercise not all the coordinate points are moving and not everything is useful. so we take only those coordinates's data that are only useful in a perticular exercise. extracted data points should have atleast one coordinate's data and it should not contain coordinates which are not in use. this may leads less training accuracy.

some more info about data extraction

3.3 Preprocessing Data

Feature scaling is one of the most important part in data preprocessing, as it includes Normalizing data, statndardizing data, etc. but in our scenario we have the data already in the range between 0 to 1. so its not necessary to normalize the data but the data we got is in the raw format, we have to convert this data in a usefull and universal form.

we have the data in a raw format and not in a stable form ie.data-points of the constant coordinates should be constant or same through the entire exercise. but as the data is extracted from a video so the coordinates are not stable. we have to normalize this data in a same manner. and for this process we are using different mathematical ways.



some more round-off values info

3.3.1 Slope

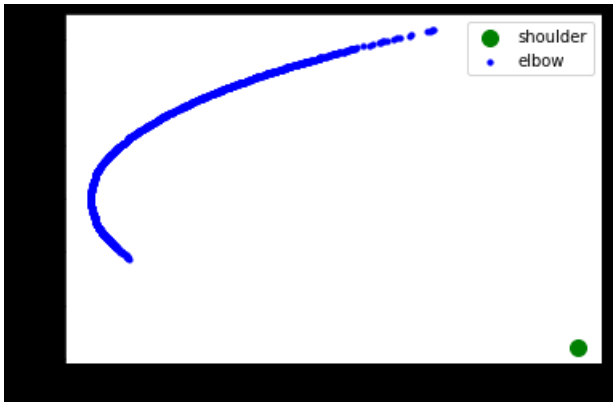
we have to find slope of the two coordinate points. for this process we are using the basic slope formula which needs both the X and Y coordinates of the two coordinate points.

some more info about slope use-case

3.3.2 Distance

we have to find the distance of two coordinate points. for this process we are using the basic distance formula which needs both the X and Y coordinates of the two coordinate points.

some more info about distance use-case



3.4 Enhancing Data

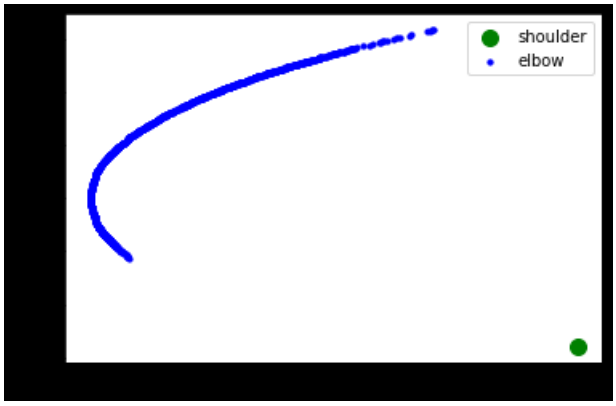
3.4.1 Need for Enhancing data

In our scenario we have to do classification as well as regression, and for classification we need at least two types of data. we are using binary classification so we need our data to be in the format of 2 labels, true and false or 1 or 0. but, our data consists of only one type of data ie. true data, which represents the coordinate of the data points that are in correct position. but we have to apply a classification algorithm on these data points. so we need both the data points of right and wrong exercisees. but its really complecated to record the wrong exercise data. because in order to collect the wrong data, we don't know the every possible worng exercise moves. but for classification we need the second type of the data.

3.4.1 Way to Enhance the data

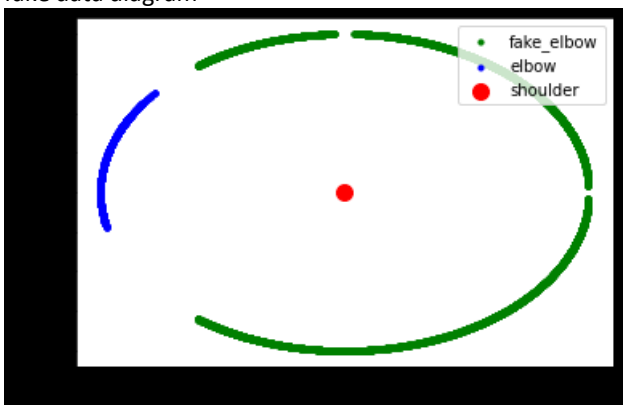
In order to generate the other type of data we have to observe the data we have. in our data we have the X and Y coordinates of 33 points. and if we consider a single point coordinate we have to generate every next possible point besides of the right or true data.

We Observe that, if we consider a point as constant and the other point is moving then they will form a curve.



and from this curve we can find the distance between the constant point and the moving point as well as the slope of the line formed by the points. our main goal is to draw a circle from this curve. by using distance between points as the radius and the Circle formula we can generate the possibe X and Y coordinates. after generating the fake coordinates we have to remove the coordinate points of the curve. while removing the curve points just remove some more extra points for better classification.

fake data diagram



3.5 Final Data

Final data will be in the form of X and Y coordinates of the perticular data points. now we can train the classifier of a regressor on this data but, we after applying either classifier or regressor or both we have to decode the results to the values suitable for the coordinates of the data points. for this we are using the slope, distance and some previous encoding method but in a reverse way to obtain the data points according to the original data points. in case of a classification the data will contain the X and Y coordinates of the point and a label describing the data, either right or wrong. and in a the case of regeression atlest two coordinates of the points ie. X1, Y1, X2, Y2 for better accuracy and the output coordinates for the predicted point ie. X3, Y3.