Gym Posture Correction

Team: Anit Upadhyaya, Tejasvi Birdh, Vishal Paudel

Problem Statement

- To **classify the gym posture as correct or incorrect** while performing heavy-weight injury prone exercises.
- On top of classification, we are also providing **feedback** on the posture correction.

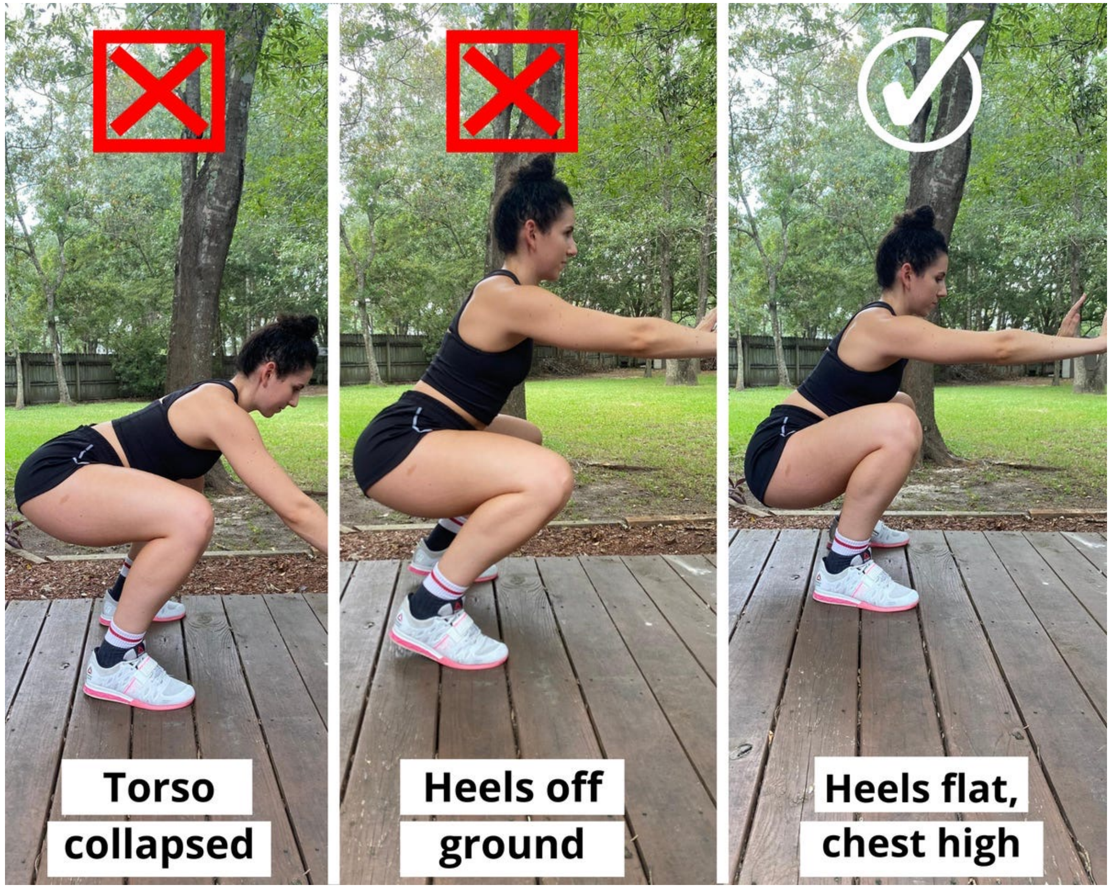


Figure 1

Fig. 1: Squat mistakes: Fix your form for squats and other exercises

Impact of Wrong Posture on Workout

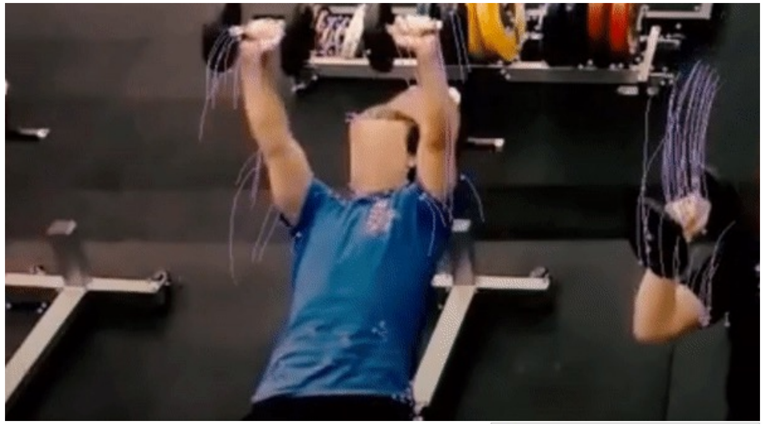
- Injury Risk
- Muscle Imbalance
- Joint Strain and Pain
- Reduced Effectiveness
- Long-term health issues
- Loss of motivation
- Financial Costs

Literature Survey

Gym Cam

Hypothesis:

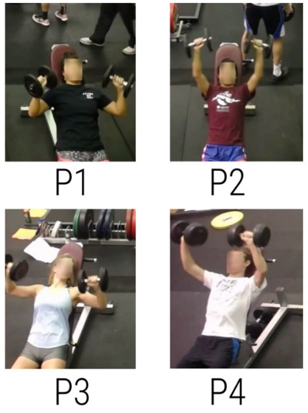
- Repetitive motion in gym (multi-user setting) represents exercises
- extraordinarily rare for two people to exercise at same rate
- to solve noise is to collect data in the user’s workout environment

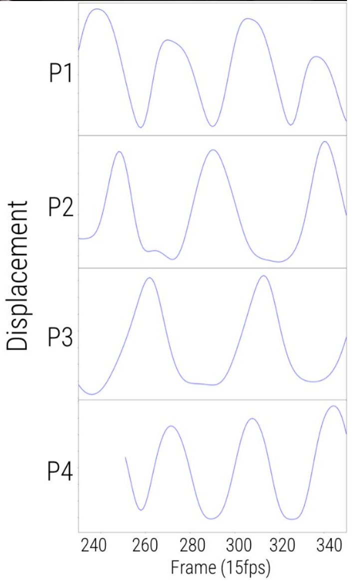


Data:

42 hours of university gym feed, spanning 5 days

Hand annotated 50% of the exercise key points



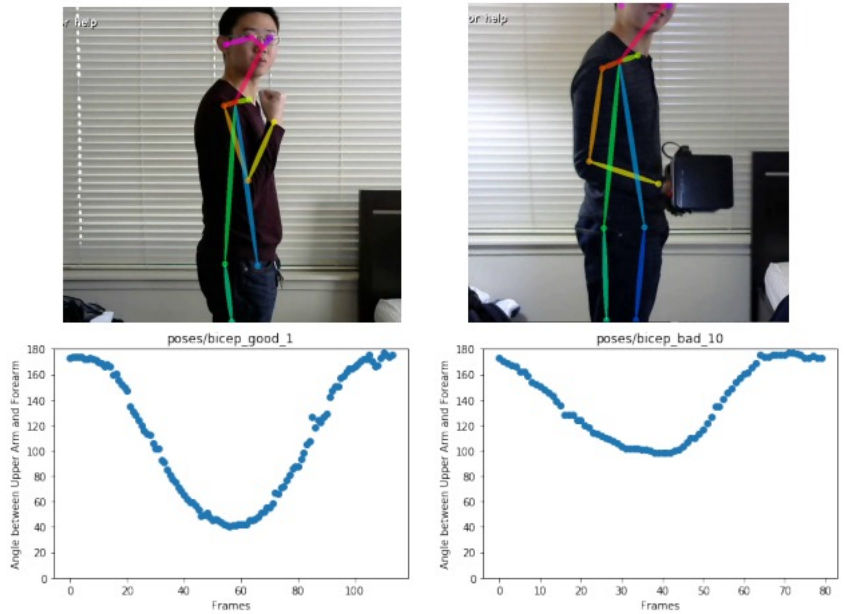


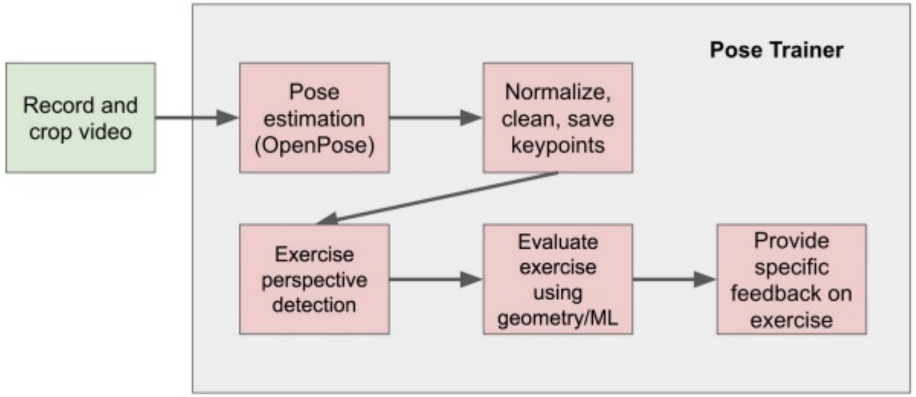
Results:
(1) Detect all exercise activities in the scene (acc. = 99.6%), then
(2) Disambiguate between simultaneous exercises (acc. = 84.6%), then
(3) Estimate repetition counts (± 1.7 counts)
(4) Recognize common exercise types (acc. = 93.6% for 5 most common exercise types).

Literature Survey

Pose Trainer

- Extracts insights from pose changes temporally.
- **Personalized feedback** on fitness exercise form.
- Uses **OpenPose** (which uses a multi-stage CNN).
- Uses pose estimation, **visual geometry**.
- **Hard-codes feedback** based on heuristics





for front raise:

'Your back shows significant movement. Try keeping your back straight and still when you lift the weight. Consider using lighter weight.'

'You are not lifting the weight all the way up. Finish with wrists at or slightly above shoulder level.'

Literature Survey

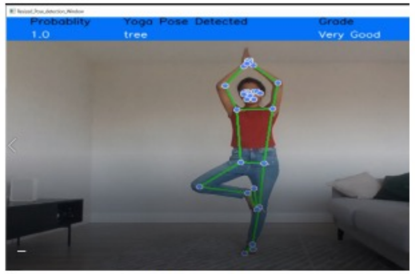
Real-Time Yoga Pose Detection using Machine Learning Algorithm

Figure 7: Confusion matrix - XgBoost Classifier

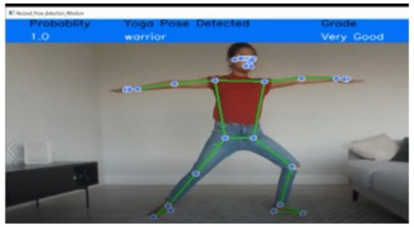
Figure 3: 3D Landmark data generation on warrior pose using Blazepose model

Classifier	Accuracy	precision	Recall	F1 Score
XgBoost	95.14%	95.36%	95.02%	95.17%
Random Forest	94.7%	95.22%	94.41%	94.75%
SVM	92.05%	91.89%	92.27%	91.95%
Decision Tree	86.75%	86.42%	87.15%	86.58%

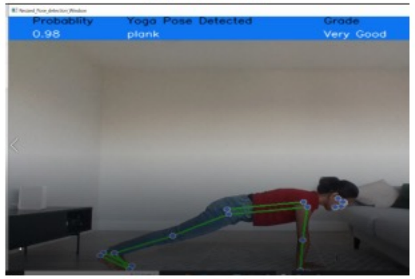
Table 1: Results From Experiment 3



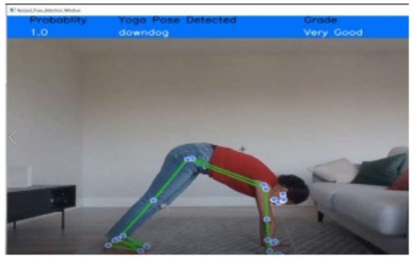
(a) Tree Pose



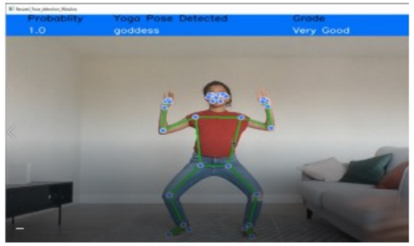
(b) Warrior Pose



(c) Plank Pose



(d) Downdog Pose



(e) Goddess Pose

Figure 8: Real-Time Yoga Pose Detection

Literature Survey

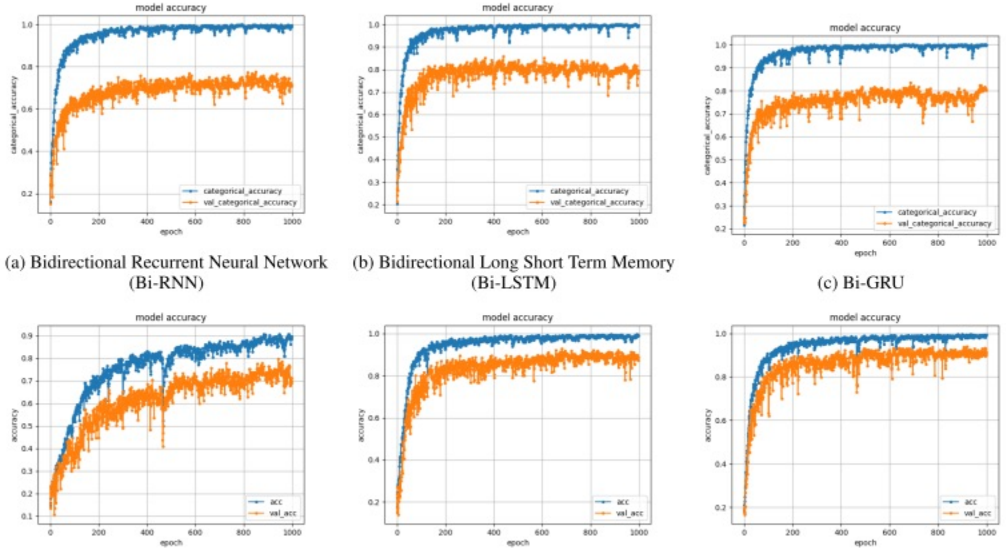
AI Trainer

(a) Camera 1 (b) Camera Holder Top view (c) Camera setup side view

FIGURE 4. Proposed experimental setup.

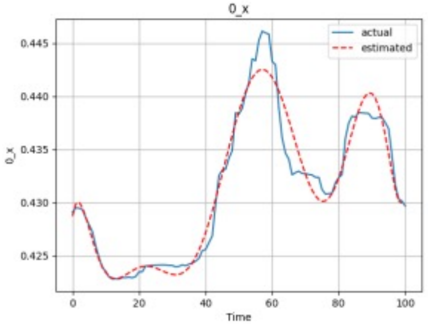
(a) Stereo Calibration Frame for Camera 0 (b) Stereo Calibration Frame for Camera 1 (c) Mono Calibration Frame for Camera 0

FIGURE 5. Images used for calibration process.



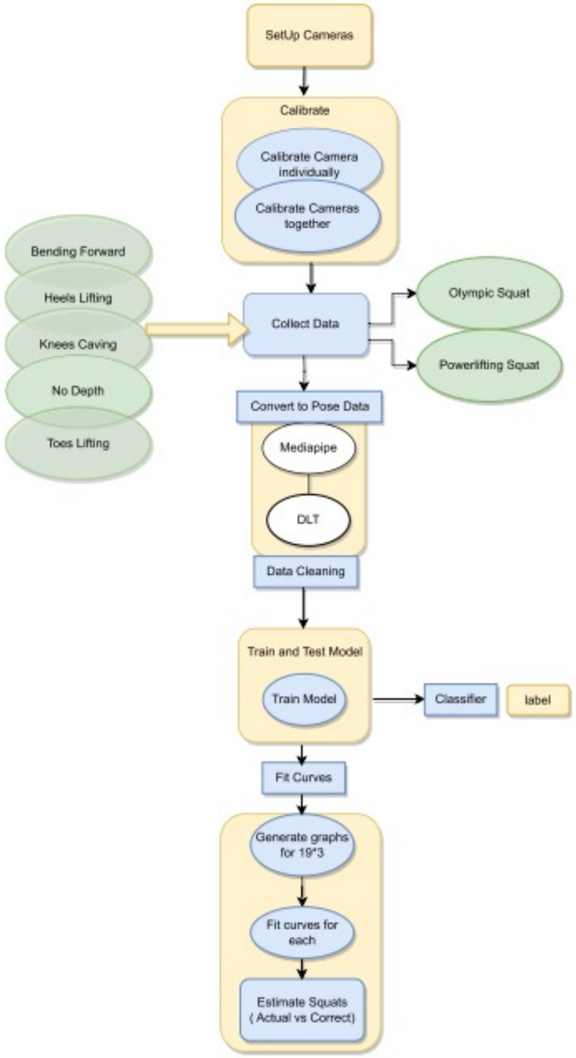
(a) Bidirectional Recurrent Neural Network (Bi-RNN) (b) Bidirectional Long Short Term Memory (Bi-LSTM) (c) Bi-GRU

(d) Bi-RNN with Attention (e) Bi-LSTM with Attention (f) Bi-GRU with Attention



(a) Olympic Squat, Camera 0, 0-x

Model Comparison	
Model Name	Accuracy
Bi-RNN	77.52%
Bi-LSTM	85.76%
Bi-GRU	83.52%
Bi-RNN with Attention	79.77%
Bi-LSTM with Attention	92.88%
Bi-GRU with Attention	**94.00%**



. **Process flow of the proposed approach for squat detection ...ction.**

Dataset & Data Collection

A. Self Collected video data

Plaksha Gymnasium

1. Taken **three sets of ten reps** each
2. from about **60 Plaksha students**,
3. Each video about a **minute long**
4. Video Feed about **200 minutes** in total
5. Female to Male ratio is about **1:4**

Dataset & Data Collection

A. Self Collected video data

Important points to note about our data:

1. **Fixed Front view**
2. **Fixed location in the gym**
3. **Homogenous FPS**
4. **Lighting condition same**

These are our **assumptions** about the sufficient information required for correction/feedback

Dataset & Data Collection

B. YouTube scrapped video of people doing squats.

We have managed to scrape ~~twenty~~ squat videos from YouTube

Dataset Augmentation

Approach-1

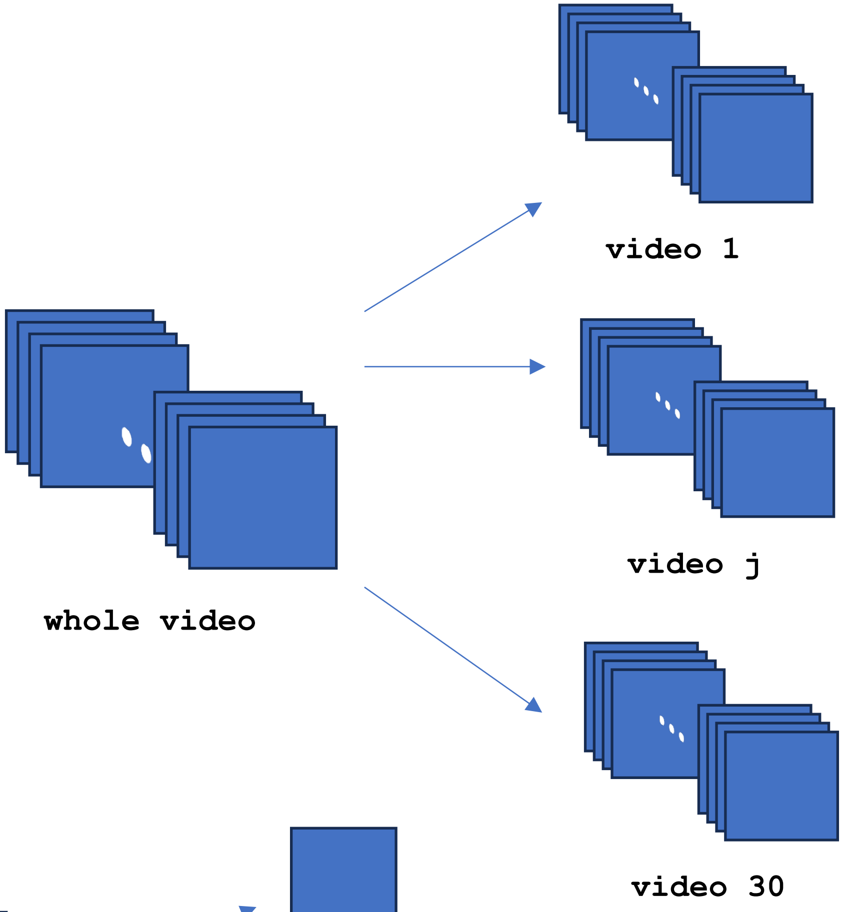
Assumptions:

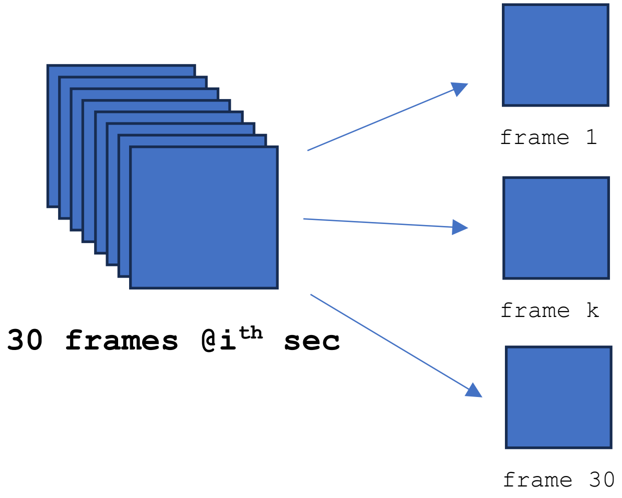
- 30 neighbouring frames have similar data
- Separating out 1st frame from all seconds still preserves the movement

Method:

- In this approach, we segregated similar frames collected at a second at equally timed interval into groups to generate multiple data points.

- This approach gave us datapoints in the shape of **(1800,30,33,3)** with a binary label(0 or 1) same as that of video.



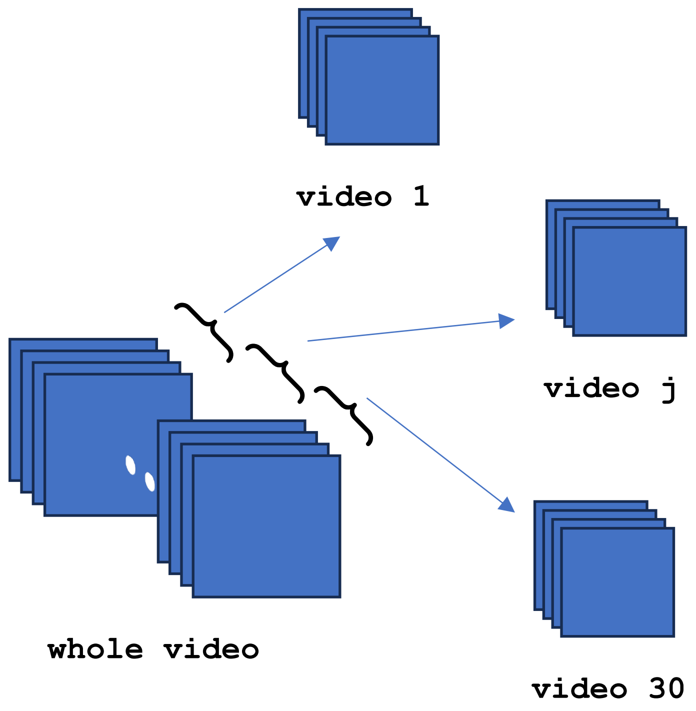


Dataset Augmentation

Approach-2

Assumptions:

- 30 non-local sequential frames have important information data
- Sliding groups of frames with internal gaps is like chopping the whole video into videos of lowered frame rates, but information(movement) remains same
- If whole video is correctly labelled, then such subparts can also inherit the same label



ML Methodology

- Which models were used? Random Forest, Decision Trees, XGBoost Classifier and LSTM.
- Many-to-One RNN based LSTM model used for binary classification.
- For feedback, VAE along with LSTM use to get good results.

ML Methodology

Decision Tree

- A decision tree is a predictive model in machine learning that resembles a tree-like structure.
- Internal nodes represent decisions based on specific features, branching into subsequent nodes and leaves.
- The model recursively partitions data, making decisions at each node based on the most significant features.

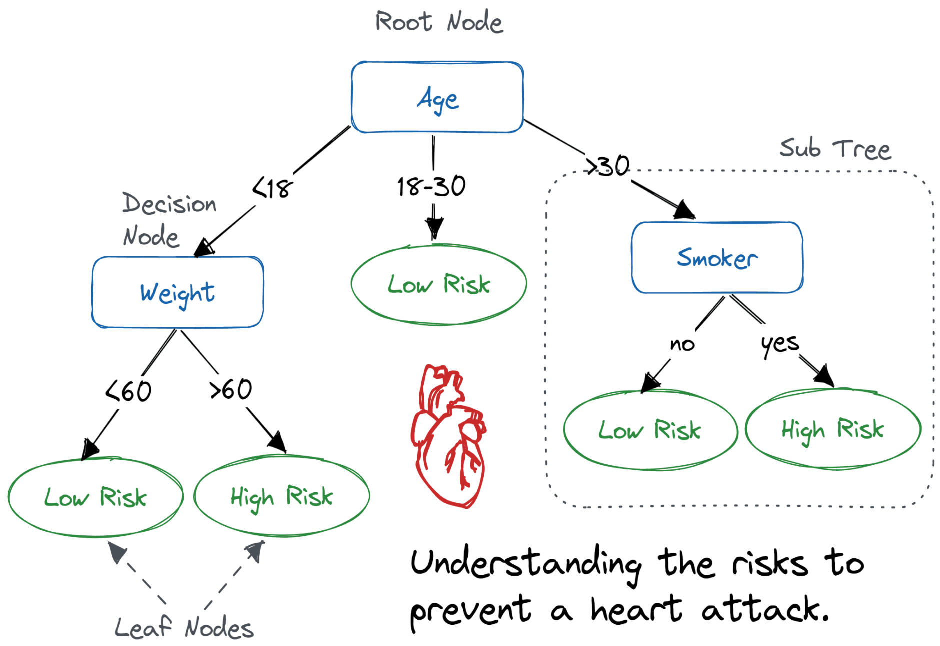


Fig. 2: Decision Tree for Heart Attack Prediction

Source: [Decision Tree Classification in Python Tutorial]

ML Methodology

Decision Tree

- **Decision Tree on Gym Data:** Shape transformation of our data from (1339,30,33,3) to (1339,2970) where 2970 comes from 30*33*3.

- Training decision tree ended up with a result of 0.61 accuracy.

```
Accuracy: 0.61
Confusion Matrix:
 [[123  59]
 [ 73  80]]
Classification Report:
              precision    recall  f1-score   support

           0       0.63      0.68      0.65       182
           1       0.58      0.52      0.55       153

    accuracy                           0.61       335
   macro avg       0.60      0.60      0.60       335
weighted avg       0.60      0.61      0.60       335
```

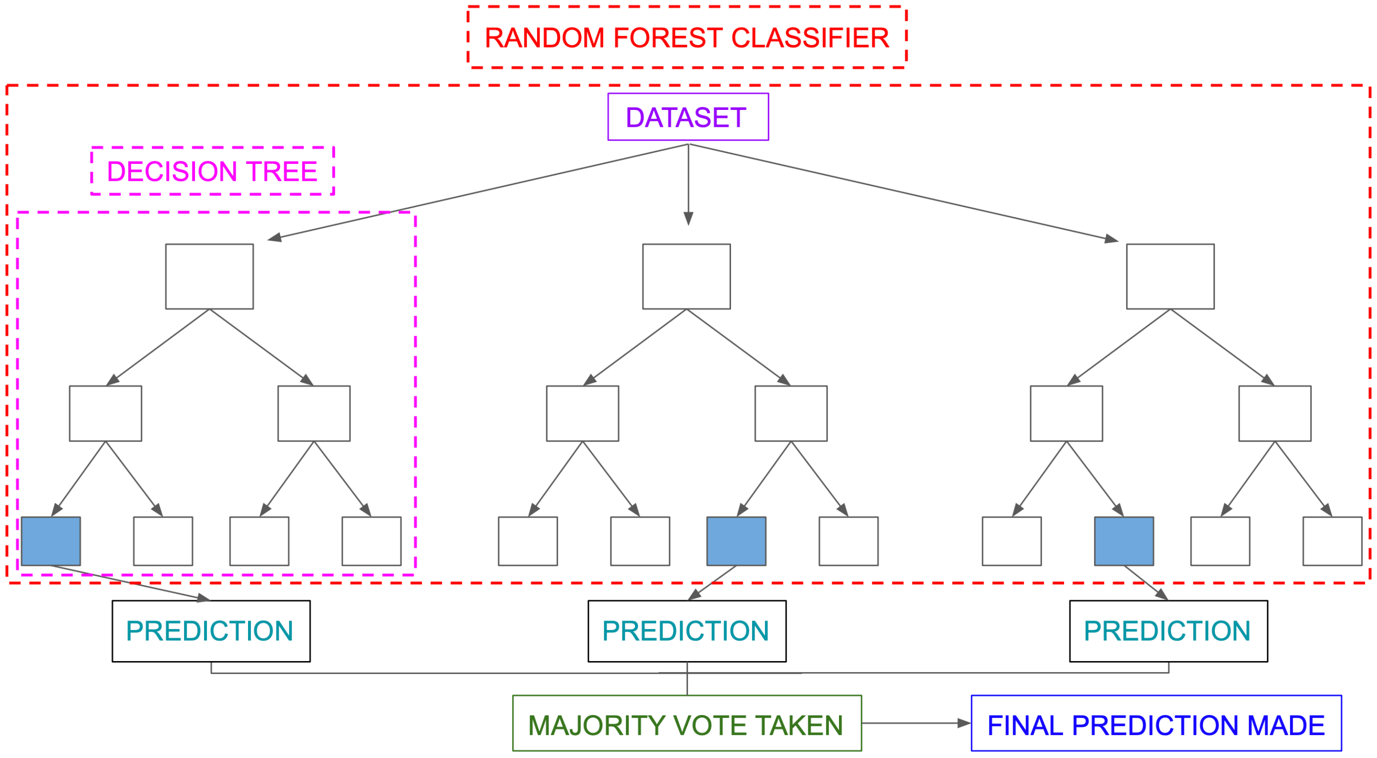
```
Decision Tree Summary:
|--- feature_294 <= 0.44
|   |--- feature_1545 <= 0.44
|   |   |--- class: 1
|   |--- feature_1545 >  0.44
|   |   |--- class: 0
|--- feature_294 >  0.44
|   |--- feature_1164 <= 0.54
|   |   |--- feature_137 <= 0.00
|   |   |   |--- feature_67 <= 0.44
|   |   |   |   |--- feature_559 <= 0.14
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- feature_559 >  0.14
|   |   |   |   |   |--- class: 0
|   |   |   |--- feature_67 >  0.44
|   |   |   |   |--- class: 1
|   |   |--- feature_137 >  0.00
|   |   |   |--- feature_1632 <= 0.44
|   |   |   |   |--- feature_531 <= 0.48
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- feature_531 >  0.48
|   |   |   |   |   |--- class: 0
|   |   |   |--- feature_1632 >  0.44
|   |   |   |   |--- class: 0
|   |--- feature_1164 >  0.54
|   |   |--- feature_175 <= 0.53
|   |   |   |--- feature_436 <= 0.27
|   |   |   |   |--- class: 1
|   |   |   |--- feature_436 >  0.27
|   |   |   |   |--- class: 0
|   |   |--- feature_175 >  0.53
|   |   |   |--- feature_76 <= 0.64
|   |   |   |   |--- feature_2102 <= 0.00
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- feature_2102 >  0.00
|   |   |   |   |   |--- class: 0
|   |   |   |--- feature_76 >  0.64
|   |   |   |   |--- class: 0
```

ML Methodology

Random Forest

- Random Forest is an ensemble model that aggregates predictions from multiple decision trees followed by majority voting to do the final prediction.

- By introducing randomness by **bootstrapped sampling** & **feature randomness,** the random forest model mitigates overfitting by enhancing the model's generalization capabilities.



Random Forest Classification

Source: Introduction to Random Forest in Machine Learning

ML Methodology

Random Forest

- In Random Forest, we also used the same rescaling of the dataset from (1339,30,33,3) to (1339,2970) where 2970 comes from 30*33*3.
- The accuracy of the random forest model turned out to be 0.40.
- This suggested us that in the decision tree there might be some sort of overfitting.

```
Random Forest Accuracy: 0.40
Random Forest Confusion Matrix:
 [[ 49 133]
 [ 68  85]]
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.42      0.27      0.33       182
           1       0.39      0.56      0.46       153

    accuracy                           0.40       335
   macro avg       0.40      0.41      0.39       335
weighted avg       0.41      0.40      0.39       335
```

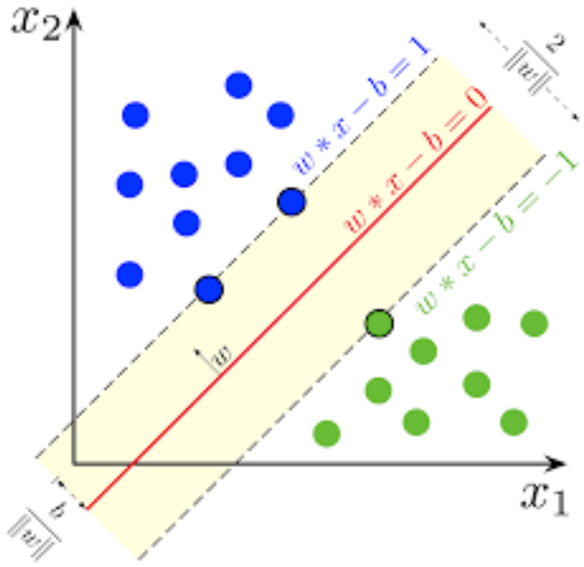
Random Forest Classification Report

ML Methodology

SVM Classifier

- SVM identifies a hyperplane that maximally separates two classes in a high-dimensional space.
- The accuracy of the SVM decreases as we go higher in dimensions i.e., the curse of dimensionality.
- As our dataset contains **33 features** it gave a low accuracy of **0.47**.

source: [Support vector machine - Wikipedia]



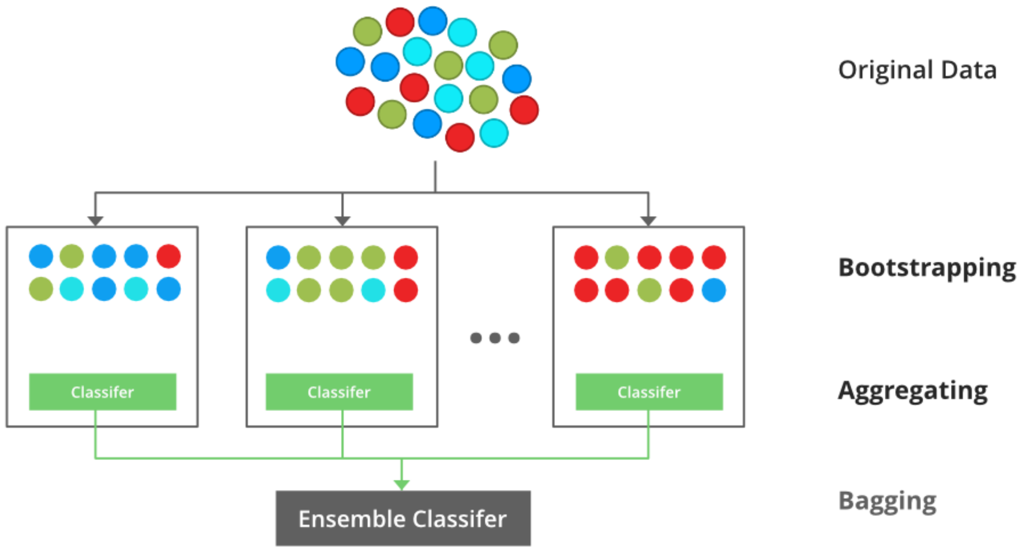
Support Vector Machines (SVM) Classifier

```
SVM Accuracy: 0.47
SVM Confusion Matrix:
 [[  3 179]
 [  0 153]]
SVM Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.02      0.03       182
           1       0.46      1.00      0.63       153

    accuracy                           0.47       335
   macro avg       0.73      0.51      0.33       335
weighted avg       0.75      0.47      0.31       335
```

ML Methodology

XGBoost Classifier

- XGBoost is a powerful machine learning algorithm for its efficiency and predictive accuracy.
- It operates by building an ensemble of decision trees sequentially, optimizing a global objective function.
- It incorporates regularization techniques and employs a gradient boosting approach.
- With XGBoost, we were able to get an accuracy of over **0.75**.



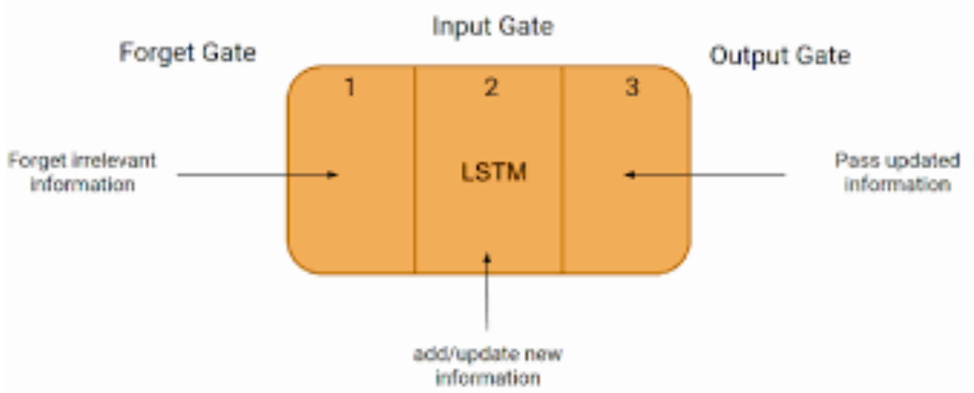

```
XGBoost Accuracy: 0.75
XGBoost Confusion Matrix:
 [[ 97  85]
 [  0 153]]
XGBoost Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.53      0.70       182
           1       0.64      1.00      0.78       153

    accuracy                           0.75       335
   macro avg       0.82      0.77      0.74       335
weighted avg       0.84      0.75      0.74       335
```

source: [Bootstrap aggregating - Wikipedia]

ML Methodology

Long-Short Term Memory

- Using LSTM for detecting correct or incorrect posture involves training a sequence model on time-series data capturing body positions over time.
- The model can predict and classify unseen sequences, helping to identify instances of improper body alignment or posture deviations.
- With LSTM, we were able to achieve an accuracy of 0.87 on our dataset.

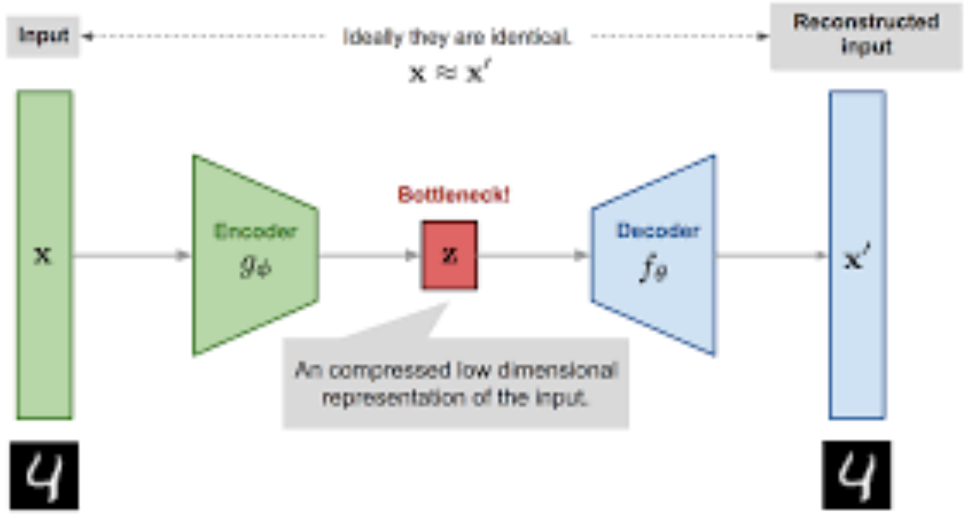


LSTM Model

Source: [What is LSTM?]()

Feedback Mechanism: Using Anomaly Detection in Time-Series

- The time-series anomaly will help us to figure out which features lead us towards wrong posture.
- Isolating these features will help us in providing the feedback for a specific wrong posture.
- This will be done by calculating the re-construction loss from the autoencoder & if this loss is more than a certain threshold those features will be the part of anomaly.



LSTM Autoencoder: Time-Series Anomaly Detection

Source: [LSTM Autoencoder for Anomaly Detection for ECG data]

Performance Metrics

- As the problem involves binary classification of posture (correct or incorrect).
- Hence, the most performance metric involved accuracy, precision, recall and f1-score.
- Our LSTM based model gave us an f1-score of 92.3. This score seems to be a good start to deploy ML model in the gymnasium along with keeping an instructor (human) in the loop.

Results

Model	Data-1	Data-2
Decision Tree	0.61	0.54
Random Forest	0.40	0.42
SVM Classifier	0.47	0.35
XGBoost Classifier	0.75	0.62
LSTM	0.87	0.79

Deployment Challenges

- Model compression over the hardware like TinyML.
- Difference in real-time accuracies of the LSTM model.
- Lack of data in getting proper classification & anomaly detection (due to less variability in data).

Demo Time!!

Thank You