Research Topic

My research topic consists of creating a machine learning model that can accurately classify different exercises being performed in a video. The model works by analyzing 32 different points on the person performing the exercise. These points create different joint angles that the model can use to classify which exercise is being done. I wrote a script that utilizes the MediaPipe library by Google to extract these key points from the dataset that I was working with. Once I had these data points stored in my database, I was able to perform exploratory data analysis as well as test different models. The main models that I tested were random forest, CNN, and LSTM. The model that performed the best was the random forest based on a singular frame within the video. The CNN model didn't have enough data to accurately classify the exercises. The LSTM model worked well but had lower marks than the random forest model.

Dataset

The dataset was sourced from Kaggle's WorkoutFitness Video Dataset. It includes hundreds of labeled workout clips across a wide range of exercises, such as bicep curls, squats, and tricep pushdowns. Each table consists of similar exercise videos.

Data Management

The data management part of this research project was somewhat complex. I extracted the data from the Kaggle notebook using their API. I saved the data in my google drive using a similar folder structure as the Kaggle dataset used. I then wrote a script to extract key point data utilizing mediapipe. This created a new table that had 32 different fields for x, y, z, and visibility, as well as columns for time stamps and labeling.

SQL

The table I created in supabase using sql to store all my data is called video frames and is defined by the following:

```
create table public.video_frames (
  frame_id serial not null,
  video_id uuid null,
  frame_number integer not null,
  timestamp_seconds double precision not null,
  x0 double precision null,
  y0 double precision null,
  z0 double precision null,
  visibility0 double precision null,
  x1 double precision null,
```

```
y1 double precision null,
z1 double precision null,
visibility1 double precision null,
x2 double precision null,
y2 double precision null,
z2 double precision null,
visibility2 double precision null,
x3 double precision null,
y3 double precision null,
z3 double precision null,
visibility3 double precision null,
x4 double precision null,
y4 double precision null,
z4 double precision null,
visibility4 double precision null,
x5 double precision null,
y5 double precision null,
z5 double precision null,
visibility5 double precision null,
x6 double precision null,
y6 double precision null,
z6 double precision null,
visibility6 double precision null,
x7 double precision null,
y7 double precision null,
z7 double precision null,
visibility7 double precision null,
x8 double precision null,
y8 double precision null,
z8 double precision null,
visibility8 double precision null,
x9 double precision null,
y9 double precision null,
z9 double precision null,
visibility9 double precision null,
x10 double precision null,
y10 double precision null,
z10 double precision null,
visibility10 double precision null,
x11 double precision null,
y11 double precision null,
z11 double precision null,
visibility11 double precision null,
```

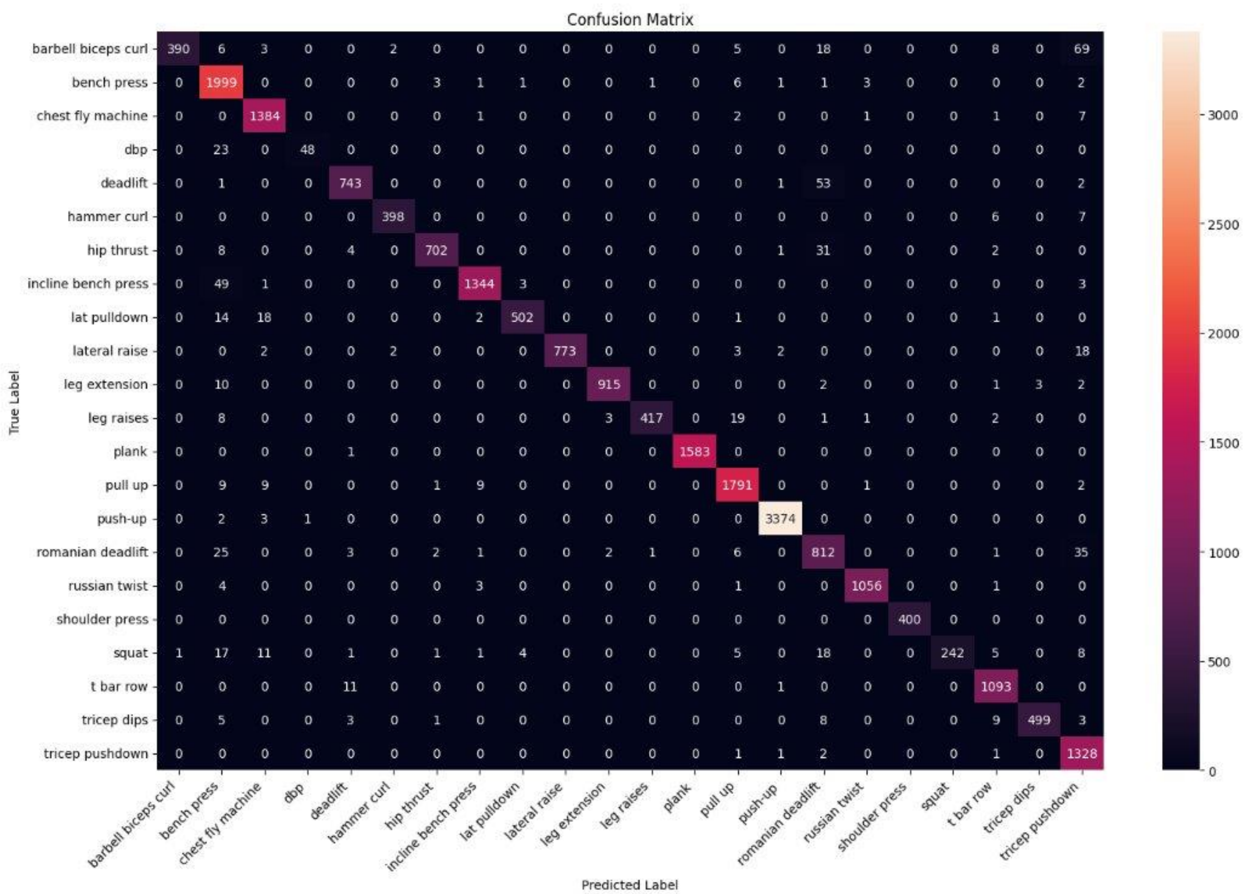
```
x12 double precision null,
y12 double precision null,
z12 double precision null,
visibility12 double precision null,
x13 double precision null,
y13 double precision null,
z13 double precision null,
visibility13 double precision null,
x14 double precision null,
y14 double precision null,
z14 double precision null,
visibility14 double precision null,
x15 double precision null,
y15 double precision null,
z15 double precision null,
visibility15 double precision null,
x16 double precision null,
y16 double precision null,
z16 double precision null,
visibility16 double precision null,
x17 double precision null,
y17 double precision null,
z17 double precision null,
visibility17 double precision null,
x18 double precision null,
y18 double precision null,
z18 double precision null,
visibility18 double precision null,
x19 double precision null,
y19 double precision null,
z19 double precision null,
visibility19 double precision null,
x20 double precision null,
y20 double precision null,
z20 double precision null,
visibility20 double precision null,
x21 double precision null,
y21 double precision null,
z21 double precision null,
visibility21 double precision null,
x22 double precision null,
y22 double precision null,
z22 double precision null,
```

```
visibility22 double precision null,
x23 double precision null,
y23 double precision null,
z23 double precision null,
visibility23 double precision null,
x24 double precision null,
y24 double precision null,
z24 double precision null,
visibility24 double precision null,
x25 double precision null,
y25 double precision null,
z25 double precision null,
visibility25 double precision null,
x26 double precision null,
y26 double precision null,
z26 double precision null,
visibility26 double precision null,
x27 double precision null,
y27 double precision null,
z27 double precision null,
visibility27 double precision null,
x28 double precision null,
y28 double precision null,
z28 double precision null,
visibility28 double precision null,
x29 double precision null,
y29 double precision null,
z29 double precision null,
visibility29 double precision null,
x30 double precision null,
y30 double precision null,
z30 double precision null,
visibility30 double precision null,
x31 double precision null,
y31 double precision null,
z31 double precision null,
visibility31 double precision null,
x32 double precision null,
y32 double precision null,
z32 double precision null,
visibility32 double precision null,
source_video text null,
constraint video_frames_pkey primary key (frame_id),
```

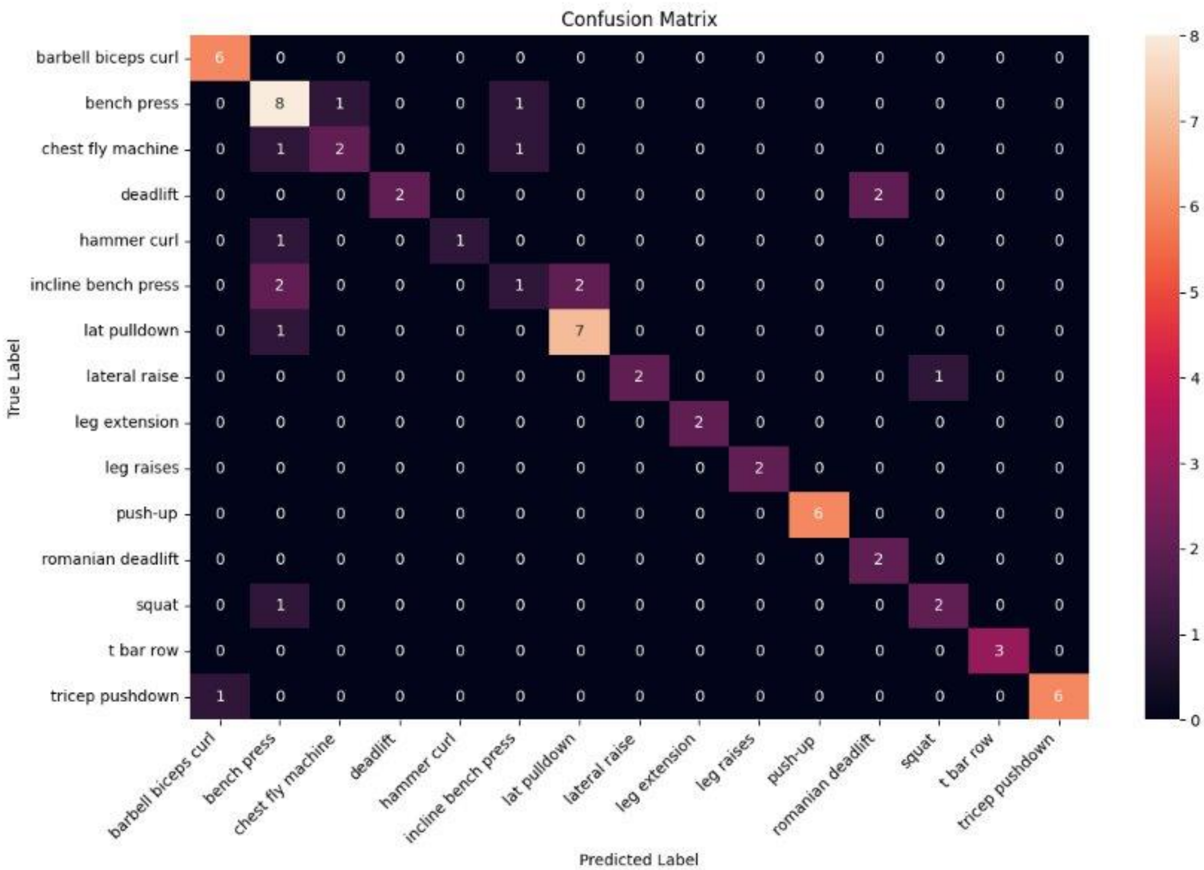
```
  constraint video_frames_video_id_fkey foreign KEY (video_id) references videos (video_id)
on delete CASCADE
) TABLESPACE pg_default;
```

Python

The Random Forest model proved to be the most effective for this classification task. My implementation began with careful data preprocessing, which included normalizing the coordinate data and calculating joint angles from the 32 key points provided by MediaPipe. Feature engineering played a crucial role, as we created meaningful features from the raw coordinate data, including relative positions and angles between key body points. Using scikit-learn's RandomForestClassifier with optimized hyperparameters, we trained the model on 80% of the dataset with cross-validation. The model achieved 90% accuracy on the test set, with particularly strong performance in classifying exercises like leg extensions, push-ups, and barbell bicep curls. The Random Forest model's success can be attributed to its ability to handle non-linear relationships and its robustness to noise in the input data. While the model showed strong overall performance in distinguishing between different exercises, it did have some difficulty differentiating between similar movements like regular bench press and incline bench press.



While Long Short-Term Memory (LSTM) networks are typically well-suited for sequential data like exercise movements, my implementation faced several challenges that limited its effectiveness. The temporal nature of the data required more complex preprocessing, and despite theoretically being able to capture movement patterns over time, the model achieved lower accuracy than the Random Forest approach. Additionally, the LSTM model required significantly more computational resources and training time. The sequential nature of the data processing made real-time predictions more challenging, which was an important consideration for practical application.



The Convolutional Neural Network (CNN) approach was primarily limited by insufficient data volume, as CNNs typically require large datasets for effective training. The spatial relationship between keypoints didn't translate well into the CNN architecture, and the model showed signs of overfitting despite various regularization attempts. We observed training stability issues and inconsistent results across different training runs, which ultimately led us to favor the Random Forest approach. The CNN's performance limitations highlighted the importance of having a sufficiently large and diverse dataset when working with deep learning models.

Data Visualization (Python)

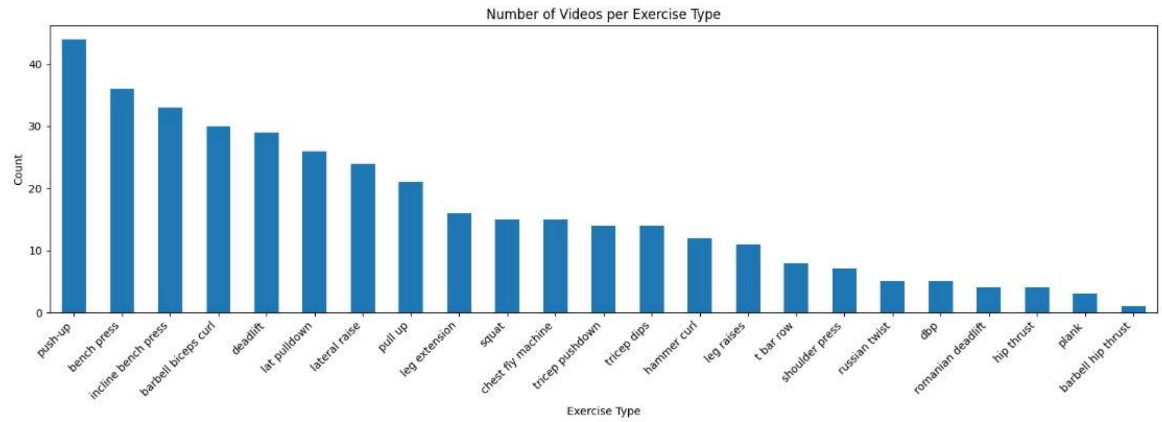
I will utilize power bi to create confusion matrices, accuracy trends, and detailed breakdowns of exercise classifications. I will also use the drill down features to dive deeper into the specific exercise analytics.

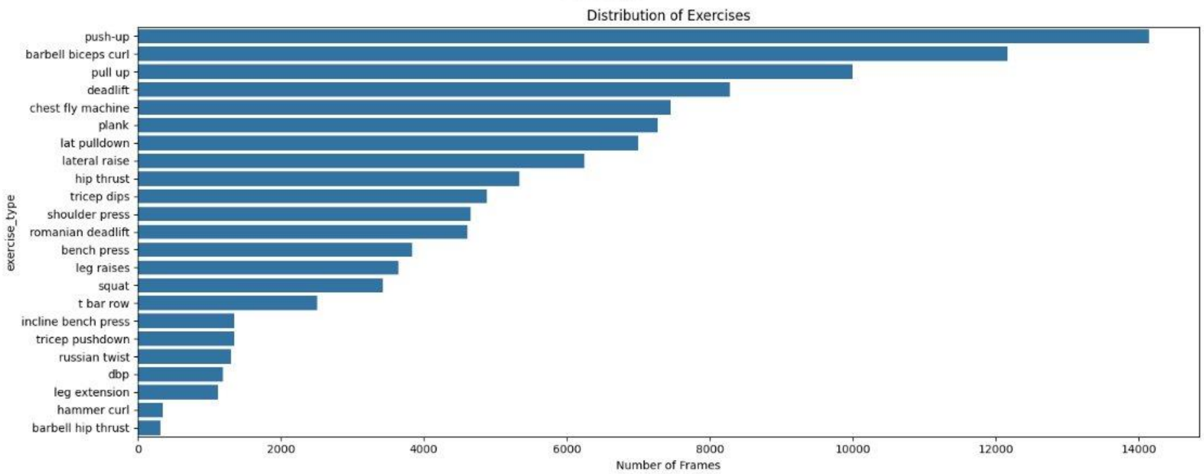
```
Number of unique videos: 377
Number of unique exercises: 23

Exercise Distribution (number of videos per exercise):
exercise_type
push-up                 44
bench press             36
incline bench press     33
barbell biceps curl     30
deadlift                29
lat pulldown            26
lateral raise           24
pull up                 21
leg extension           16
squat                   15
chest fly machine       15
tricep pushdown         14
tricep dips             14
hammer curl             12
leg raises              11
t bar row                8
shoulder press           7
russian twist            5
dbp                      5
romanian deadlift        4
hip thrust               4
plank                    3
barbell hip thrust       1
Name: count, dtype: int64

Frames per Video Statistics:
count      377.000000
mean       298.214854
std        613.627944
min          3.000000
25%         93.000000
50%        162.000000
75%        292.000000
max      10182.000000
dtype: float64

Number of coordinate columns: 133
Number of visibility columns: 33
```


Number of Videos per Exercise Type
Count
0
10
20
30
40
push-up
bench press
incline bench press
barbell biceps curl
deadlift
lat pulldown
lateral raise
pull up
leg extension
squat
chest fly machine
tricep pushdown
tricep dips
hammer curl
leg raises
t bar row
shoulder press
russian twist
dbp
romanian deadlift
hip thrust
plank
barbell hip thrust
Exercise Type


Distribution of Exercises
exercise_type
push-up
barbell biceps curl
pull up
deadlift
chest fly machine
plank
lat pulldown
lateral raise
hip thrust
tricep dips
shoulder press
romanian deadlift
bench press
leg raises
squat
t bar row
incline bench press
tricep pushdown
russian twist
dbp
leg extension
hammer curl
barbell hip thrust
0
2000
4000
6000
8000
10000
12000
14000
Number of Frames

```
Number of unique videos: 377
Number of unique exercises: 23

Exercise Distribution (number of videos per exercise):
exercise_type
push-up                44
bench press            36
incline bench press    33
barbell biceps curl    30
deadlift               29
lat pulldown           26
lateral raise          24
pull up                21
leg extension          16
squat                  15
chest fly machine      15
tricep pushdown        14
tricep dips            14
hammer curl            12
leg raises             11
t bar row               8
shoulder press          7
russian twist           5
dbp                     5
romanian deadlift       4
hip thrust              4
plank                   3
barbell hip thrust      1
Name: count, dtype: int64

Frames per Video Statistics:
count      377.000000
mean       298.214854
std        613.627944
min          3.000000
25%         93.000000
50%        162.000000
75%        292.000000
max      10182.000000
dtype: float64

Number of coordinate columns: 133
Number of visibility columns: 33
```

```
Exercise types found:
exercise_type
push-up                 14141
barbell biceps curl     12165
pull up                 10000
deadlift                 8276
chest fly machine        7447
plank                    7267
lat pulldown             7000
lateral raise            6245
hip thrust               5337
tricep dips              4885
shoulder press           4654
romanian deadlift        4609
bench press              3835
leg raises               3643
squat                    3428
t bar row                2506
incline bench press      1350
tricep pushdown          1346
russian twist            1305
dbp                      1189
leg extension            1129
hammer curl               357
barbell hip thrust        313
Name: count, dtype: int64

Total frames: 112427
Number of exercises: 23
Calculating joint angles...
```

Interpretation of Results

The random forest model that looked at the data frame by frame performed the best when classifying exercise type. The CNN model had very low accuracy when looking at entire videos. This is likely due to the insufficient amount of data being used to train the model. The LSTM model that looked at each video performed better than the CNN model, but significantly worse than the random forest model. For these reasons, the random forest is the model that I would use to classify exercise type.

Actionable Recommendations

I now have a model that can classify an exercise from our dataset with 90% accuracy.

References

Kaggle Dataset: Workout/Exercises Video