# Tennis Swing Classification

**Contributions:**

- **Abhinit Giri**: Report Writing, Data Collection, Analysis, Model Training
- **Sid Mishra**: Report Writing, Data Cleaning, Analysis, Validation, Visualization

# Problem Statement

Large amounts of data collected during Tennis matches in Big Events

- Ball Tracking
- % of First Serve Rallies won
- Rallies Won

Primary collection is done through motion cameras and spectators real-time

Introduce new data

- # of classified swings (Forehand Backhand)
- Efficiency of Swing

Difficult to keep track of whilst tracking other activities, thus make it autonomous through a wearable device classifying swings

*Resembles Walk Detection Greatly (+ Gyroscope inclusion) :*

1. Data Collection
   1.1. Swing 50 Times - Recorded through Apple watch with Sensor Logger (2 min. approx.)
   1.2. Repeat for Forehand, Backhand, Serve
   1.3. Created helper functions to combine and categorize gyroscope and accelerometer data
2. Accelerometer Magnitude Calculation
   2.1. Not applicable to Gyroscope Data, as angular velocity sign is significant
3. Noise Removal
   3.1. Apply Low-Pass Butterworth Filter on calculated Accel. Magnitude
4. Feature Extraction on Time-Series Data and Angular Velocities
5. Create Feature Frames using functions defined above
6. Train Random Forest Classifier on 70-30% Split
7. Generate .pkl model for Real-Time Sensing

# Pipeline

# Modified Code

```
def starter(filepath):
    df = pd.read_csv(filepath, dtype={'time': 'float64'})
    df.dropna(inplace=True)
    df['time'] = df['time'].apply(unix_to_date)
    df = df.drop(['seconds_elapsed'], axis=1)
    df = df.reindex(columns=['time', 'x', 'y', 'z'])

    return df
```

```
def starter_g(filepath):
    df = pd.read_csv(filepath, dtype={'time': 'float64'})
    df.dropna(inplace=True)
    df['time'] = df['time'].apply(unix_to_date)
    df.rename(columns={'x': 'x_gyro'},inplace=True)
    df.rename(columns={'y': 'y_gyro'},inplace=True)
    df.rename(columns={'z': 'z_gyro'},inplace=True)

    df = df.drop(['seconds_elapsed'], axis=1)
    df = df.reindex(columns=['time', 'x_gyro', 'y_gyro', 'z_gyro'])

    return df
```

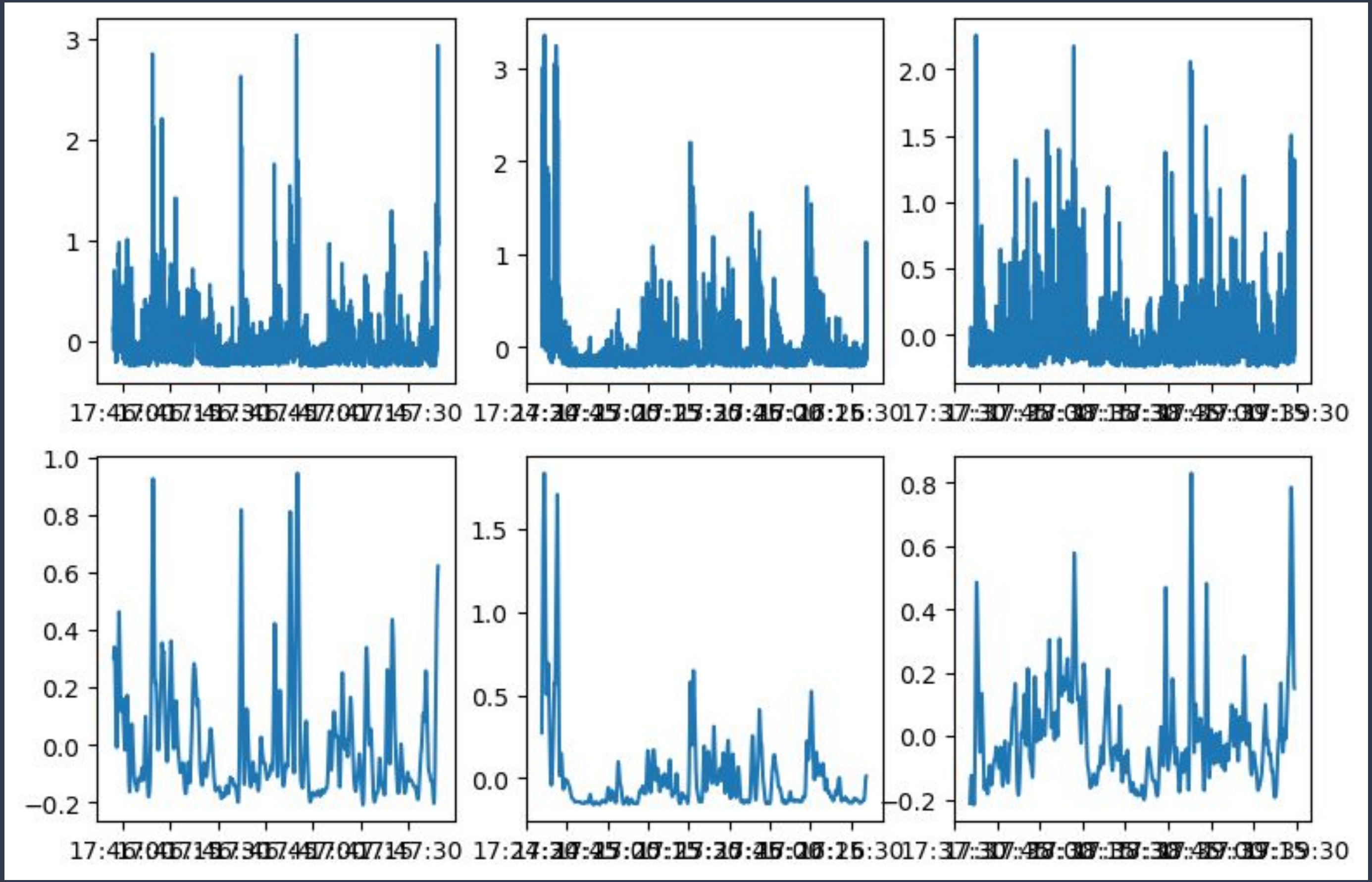
```
def add_features(window):
    features = {}
    features['avg'] = window['accel_mag'].mean()
    features['max'] = window['accel_mag'].quantile(1)
    features['med'] = window['accel_mag'].quantile(0.5)
    features['min'] = window['accel_mag'].quantile(0)
    features['q25'] = window['accel_mag'].quantile(0.25)
    features['q75'] = window['accel_mag'].quantile(0.75)
    features['std'] = window['accel_mag'].std()
    features['xg_avg'] = window['x_gyro'].mean()
    features['yg_avg'] = window['y_gyro'].mean()
    features['zg_avg'] = window['z_gyro'].mean()

    df = pd.DataFrame([features])
    return df
```

```
def combine(acg,gyro):
    combined_df = pd.merge(acg,gyro,on="time",how='left')
    combined_df.drop_duplicates(subset=acg, inplace=True)
    combined_df = combined_df.dropna()
    return combined_df
```

# Key Parameters and Features

- Window Size - 1 Second Windows for capturing the entirety of each swing, while ignoring the motion after the swing
- Cutoff Frequency - 4hz Found to be the best at capturing information
- Sampling Rate - 500hz set from Apple Watch Sensor Logger
- Feature Importance(By Information Gain ):
    - Minimum (Time-Series): Separating initial Serve assumption into Backhand
    - Q75: 4th Quartile Acceleration Magnitude representing end of swing
    - Angular X Velocity: Classifying Forehand/Backhand

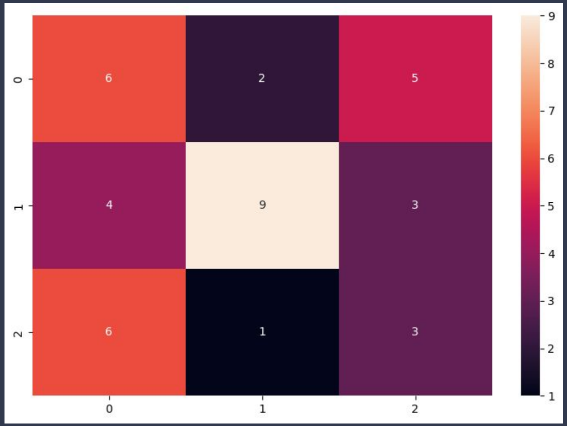
# Noise Reduced Acceleration Data



# Model Training - Confusion Matrices

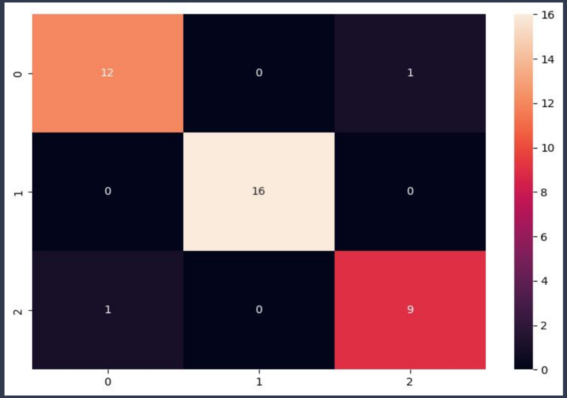
## No Noise Reduction:

- Many False Predictions, not appropriate
- 46 % Accuracy on Test Set

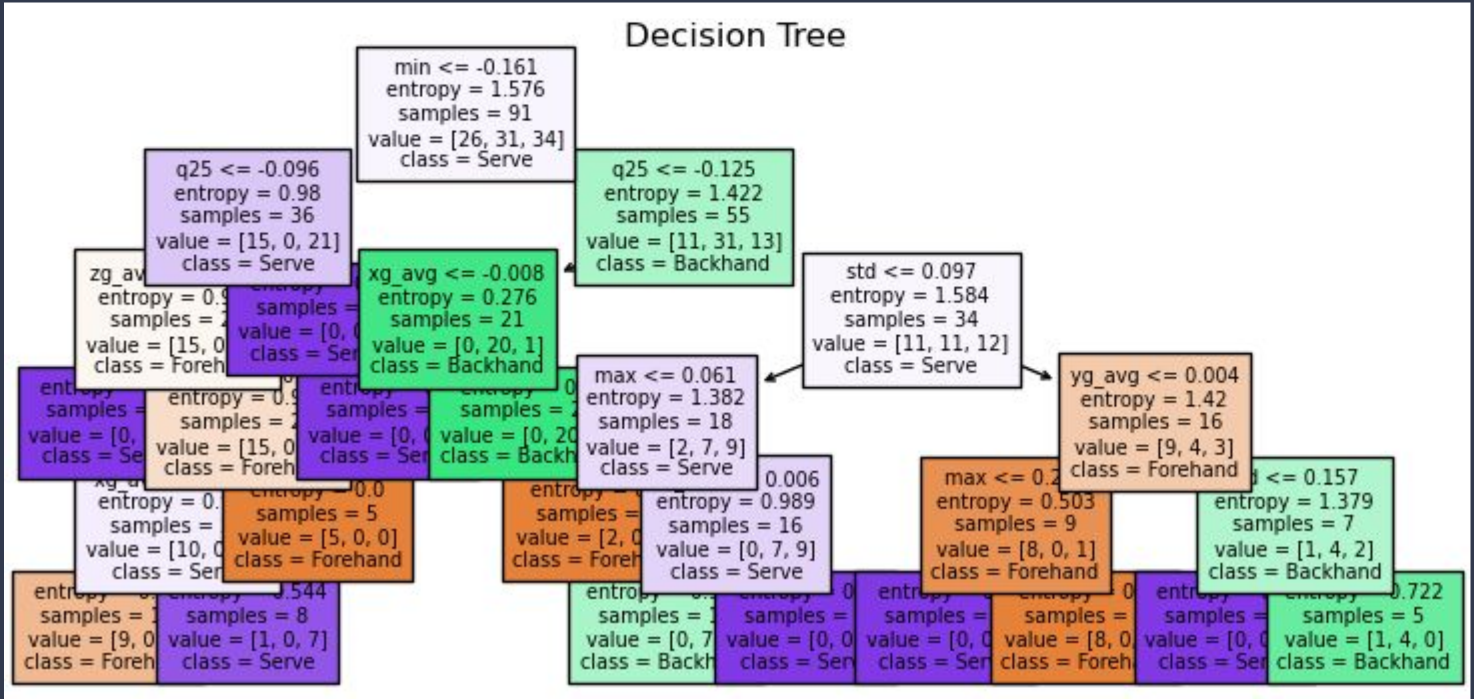



## With Noise Reduction:

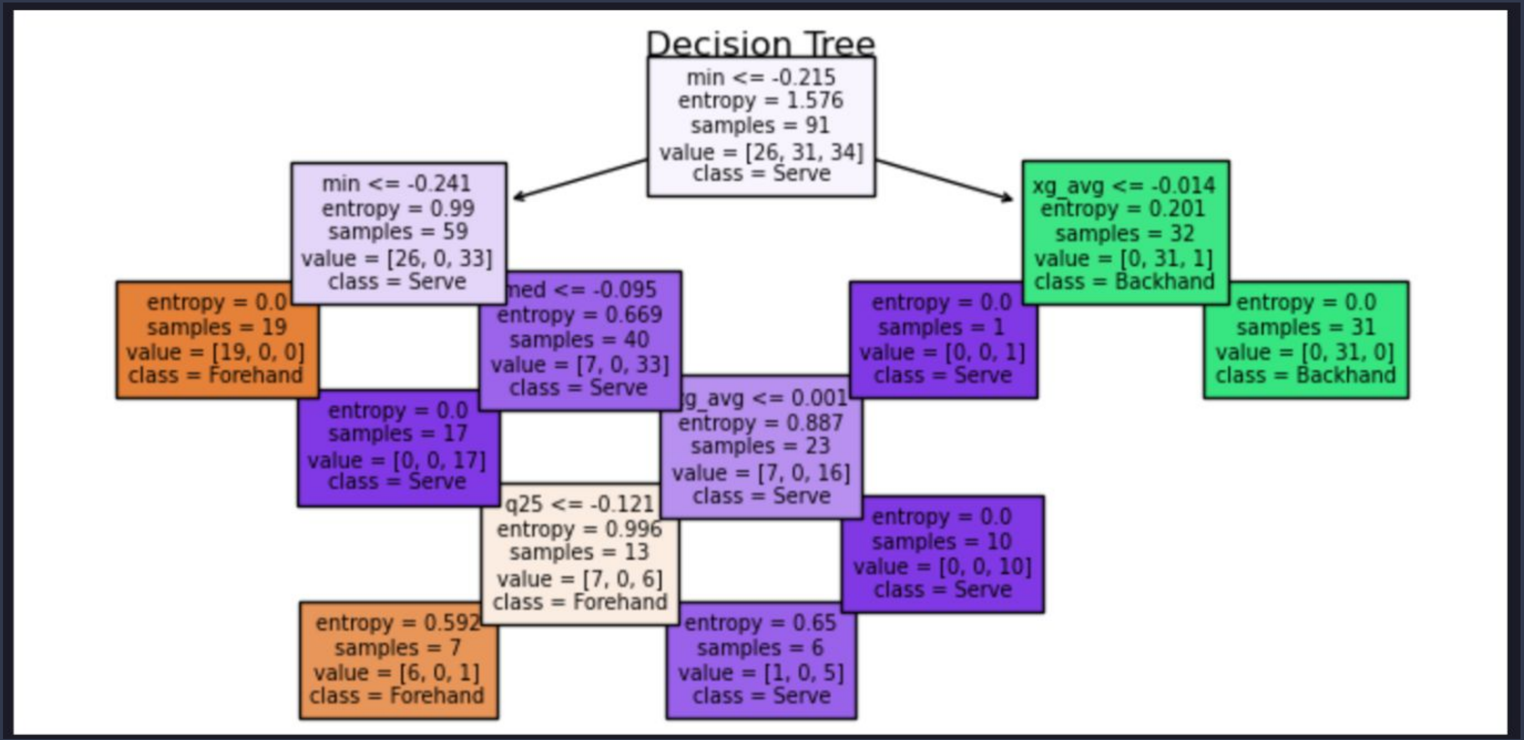
- Very Little False Positive Prediction
- 94% Accuracy on Test Set



# Unfiltered Decision Tree



# Filtered Decision Tree



# Potential Application

**Current Usage:**

- Analysis tool for live analytics during matches

**Future Additions:**

- Applicable to tennis players interested in “science-based” swinging
  - To make informed decisions on how to improve their game.
  -
- **Better for monitoring practice sessions through tallying progress**
  - Ex. Swings till muscle exhaustion

# Challenges and Realizations

- Data Collection: Not enough data could be collected due to not having readily available resources to collect data, as only the apple watch that we borrowed from our friend for a limited time was able to properly record the data needed

- Data Collection #2: The Collection application was a very challenging step as SensorBox proved to be faking a lot of data, thus Sensor Logger was effective. Additionally, the sensor logger would stop recording data from the watch when the phone screen automatically turned off which was discovered later on

- Data Method: Most effective means of collection is through wearable device like Apple Watch, however, variability in data is extremely high due to the watch shifting on wrist

- Noise Reduction: Finding an appropriate Cutoff Frequency to best train the model to be more accurate

# Learnings from the Project

- **Data Collection**: New experience to use Apple Watch Sensor Logger to poll into a CSV File (Compared to Phone previously)

- **Data Cleaning**: Understanding what filters to utilize when, and implementing various dataframe functionalities to clean raw data such as:
    - dropna()
    - dropduplicates()
    - reindex()

- **Data Validation**: Visualizing Raw Data Real Time to ascertain watch doesn't stop polling as data is collected
    - Never thought this would be a necessary step, valuable to know

- **Sampling**: Didn't realize how much the sampling rate needed to be tweaked along with the cutoff for the low pass filter to ascertain the best values.

- **Feature Addition**: How to work with large DataFrames and combining data

- **Data Testing**: Referenced the testing done from assignment three to tweak the all_data_to_combined_csv, create a visualization tree, confusion matrix, and check the accuracy.

# How to improve the Project Further

- **Data Collection**: Add instances of other individuals doing forehands, backhands, serving, so data isn't skewed by a single individual's performance.
  - Biggest Flaw with the current implementation is the lack of diverse data
  - Other calculations for swing efficiency and placement for the data can also be considered to further improve the data
- **Classifications**: Add more classifications for other types of swings such as forehand slice, backhand slice, volleys, etc.
- **Learning Model**: Utilize different models that might be better such as Support Vector Machine or Convoluted Neural Networks.

# References

- ResearchGate
    - Provided reference instance of classifying tennis swings using accelerometer and gyroscope data
- Assignments 1,2,3 - CS328
    - Creating the baseline that made this possible