Machine Learning: Risk Classification

EMGT 311: Data Science in Systems Management

June 5, 2025

Department of Engineering Management, Isabel Andaya & Evan Chang

Introduction

- Work related musculoskeletal disorders (WMSDs) are common in jobs requiring repetitive lifting, awkward postures, or high force levels.
- EMG sensors capture muscle strain and fatigue, IMU records motion and posture.
- This project will collect EMG/IMU data from repetitive lifting to develop and evaluate machine learning models for ergonomic assessment.

Problem Statement

The risk for WMSDs while performing certain tasks is currently unknown and must be predicted in order to prevent long term injury.

Objective/ Research Questions/Hypothesis

- Develop a machine learning model that uses EMG and IMU data to predict occupational risk during repetitive lifting tasks.

Methodology - Data collection

- Two test subjects perform high risk and low risk repetitive task for 20 reps
  - High risk and low risk classified by change of weight
- Wearable sensors collect EMG and IMU data in a timetable

EMG Methodology - Data Preprocessing / Processing

- Coerce all columns to numeric
- Rename EMG columns
- Extract rolling window features (extract_rolling_EMG_feature)
  - Slide a window of 1269 samples (1-second) with a 50% overlap
  - Compute mean, max, min, SD, RMS for from each sensor
- Append “Label” column for “High” or “Low” risk
- Replace NaNs with column mean

```
emg_columns = [
    "Sensor 1 (right bicep)",
    "Sensor 2 (right delt)",
    "Sensor 3 (left bicep)",
    "Sensor 4 (left delt)"
]
n_samples = df.shape[0]
feature_rows = []

for start in range(0, n_samples - window_size + 1, step_size): #loop over and select window size rows from start
    window = df.iloc[start : start + window_size]
    row_feats = []

    for col_name in emg_columns: #for eachcolum, comput the mean, max, min, std, rms
        emg_vals = window[col_name].to_numpy()
        mu_emg  = np.mean(emg_vals)
        max_emg = np.max(emg_vals)
        min_emg = np.min(emg_vals)
        std_emg = np.std(emg_vals)
        rms_emg = np.sqrt(np.mean(emg_vals**2))

        row_feats.extend([mu_emg, max_emg, min_emg, std_emg, rms_emg]) #append

    feature_rows.append(row_feats)

col_names = []
for s, label in enumerate(emg_columns, start=1): #add into column names
    short_name = f"S{s}"
    col_names += [
        f"μ_EMG_{short_name}", f"Max_EMG_{short_name}", f"Min_EMG_{short_name}",
        f"Std_EMG_{short_name}", f"RMS_EMG_{short_name}"
    ]

return pd.DataFrame(feature_rows, columns=col_names)
```

IMU Methodology - Data Preprocessing / Processing

- Coerce all columns to numeric
- Extract rolling window features (extract_rolling_IMU_feature)
  - Slide a window of 1269 samples (1-second) with a 50% overlap
  - Extract sensor data from accelerometer and gyroscope()
    - Peak Acceleration, Mean Acceleration, Total Acceleration Magnitude, Range of acceleration
    - Peak Angular Velocity, Mean Angular Velocity, Total Angular Velocity, Range of Angular Velocity
- Append “Label” column for “High” or “Low” risk
- Replace NaNs with column mean

```python
def extract_rolling_imu_features(df: pd.DataFrame, window_size: int = 1269, step_size: int = 634) -> pd.DataFrame:
    # Column names for IMU data
    acc_cols = [" ACC X (G)", " ACC Y (G)", " ACC Z (G)"]
    gyro_cols = [" GYRO X (deg/s)", " GYRO Y (deg/s)", " GYRO Z (deg/s)"]

    n_samples = df.shape[0]
    feature_rows = []

    for start in range(0, n_samples - window_size + 1, step_size):
        window = df.iloc[start : start + window_size]
        row_feats = []

        ax = window[" ACC X (G)"].to_numpy()
        ay =  (variable) window: DataFrame )
        az = window[" ACC Z (G)"].to_numpy()
        acc_mag = np.sqrt(ax**2 + ay**2 + az**2)

        peak_acc  = np.max(acc_mag)
        mean_acc  = np.mean(acc_mag)
        total_acc = np.sum(acc_mag)
        range_acc = peak_acc - np.min(acc_mag)

        row_feats.extend([peak_acc, mean_acc, total_acc, range_acc])

        gx = window[" GYRO X (deg/s)"].to_numpy()
        gy = window[" GYRO Y (deg/s)"].to_numpy()
        gz = window[" GYRO Z (deg/s)"].to_numpy()
        gyro_mag = np.sqrt(gx**2 + gy**2 + gz**2)

        peak_gyro  = np.max(gyro_mag)
        mean_gyro  = np.mean(gyro_mag)
        total_gyro = np.sum(gyro_mag)
        range_gyro = peak_gyro - np.min(gyro_mag)

        row_feats.extend([peak_gyro, mean_gyro, total_gyro, range_gyro])

        feature_rows.append(row_feats)

    col_names = [
        "PeakAcc", "MeanAcc", "TotalAcc", "RangeAcc",
        "PeakGyro", "MeanGyro", "TotalGyro", "RangeGyro"
    ]
```

Methodology - Coding

Import libraries:

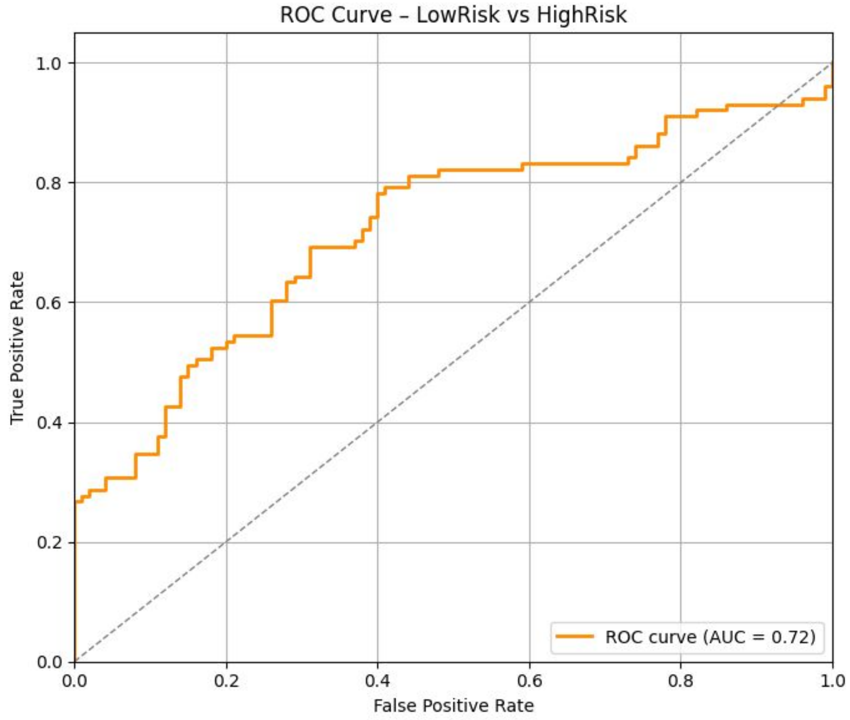
- pandas
- NumPy
- Matplotlib
- scikit-learn
  - model_selection, preprocessing, linear model, metrics, ensemble

Methodology - Analysis

- Train and evaluate Logistic regression
- Train and evaluate Random forest classifier

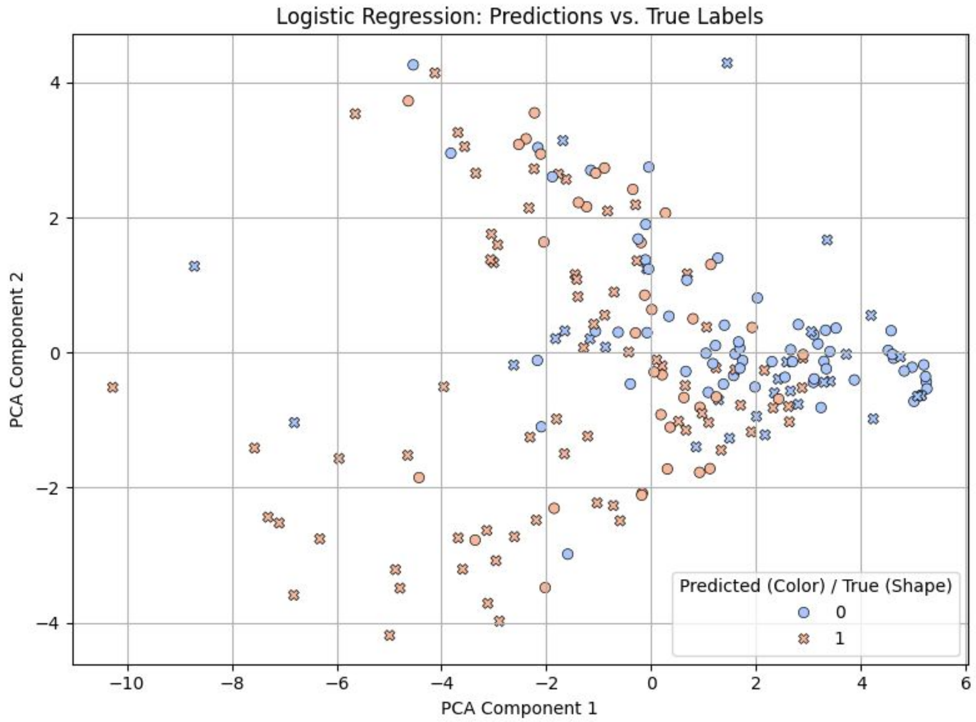
Methodology - Visualization

- Logistic regression: ROC curve
- Curve stayed well above the dashed diagonal
- curve gradually rises indicating moderate sensitivity at various false positive rate
- AUC .72 meaning 72% chance model assign a higher probability to a randomly chosen HighRisk



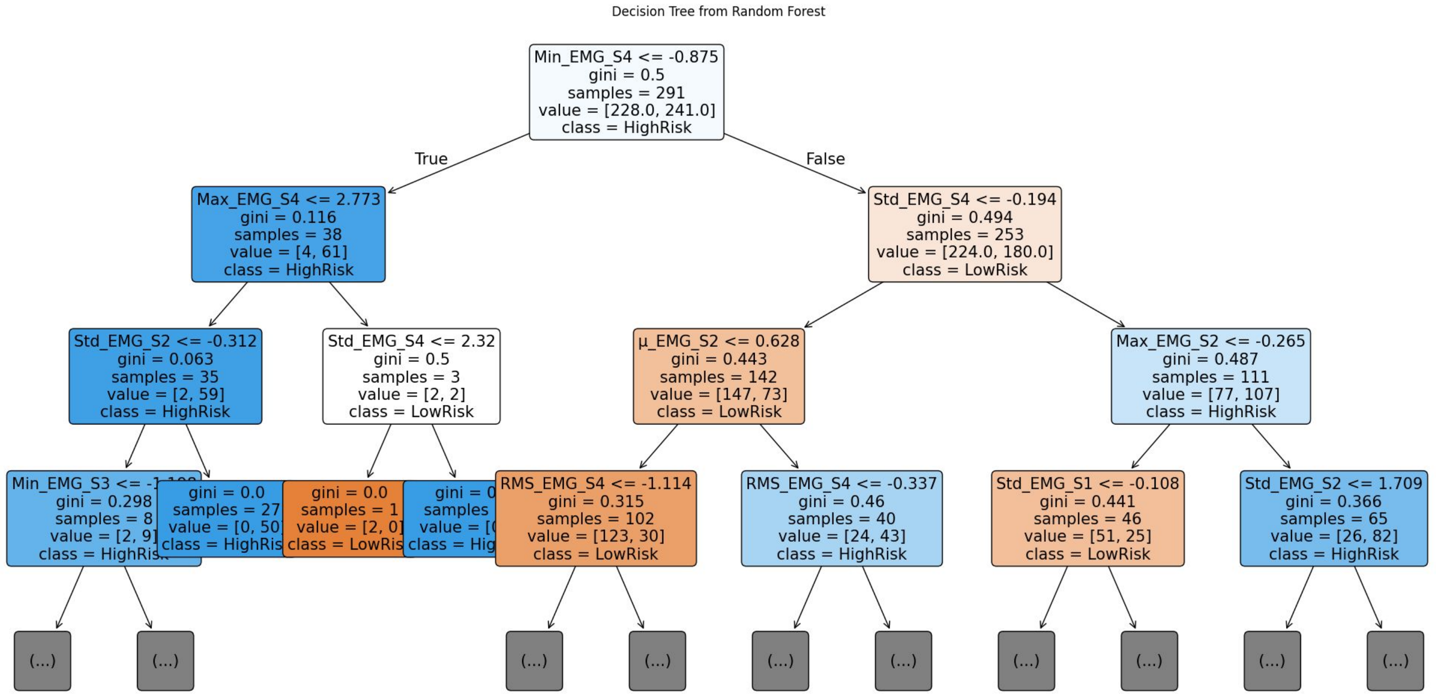
Methodology - Modeling

- Logistic regression: Scatter Plot
- Moderate class overlap
- Low Risk Correctly Predicted - blue circle/red x correctly predicted highrisk
  - mostly correctly predicted
- High Risk Correctly Predicted
  - struggle with isolated HighRisk Case
- X and Y axis - linear combination of variance in features



Methodology - Modeling

- Min_EMG_S4 <= -0.875
    - a relatively strong muscle relaxation or negative voltage deflection
- White = gini(pure class)
- Blue = highrisk
- Orange = lowrisk



Machine Learning Models Development

Logistic Regression

- Interpretable boundary
- Good baseline for linear separability

Machine Learning Models Development

Random Forest classifier

- Captures nonlinear patterns

Evaluation Metrics - EMG

```
Random Forest with EMG Rolling-Window Features
Accuracy (70/30 split): 0.88

Confusion Matrix:
[[82 18]
 [ 7 94]]

Classification Report:
              precision    recall  f1-score   support

     LowRisk       0.92      0.82      0.87       100
    HighRisk       0.84      0.93      0.88       101

    accuracy                           0.88       201
   macro avg       0.88      0.88      0.88       201
weighted avg       0.88      0.88      0.88       201


Mean Squared Error (MSE): 0.12
Root Mean Squared Error (RMSE): 0.35
R² Score (Goodness of Fit): 0.50
```

```
Accuracy (70/30 split): 0.67

Confusion Matrix:
[[63 37]
 [30 71]]

Classification Report:
              precision    recall  f1-score   support

     LowRisk       0.68      0.63      0.65       100
    HighRisk       0.66      0.70      0.68       101

    accuracy                           0.67       201
   macro avg       0.67      0.67      0.67       201
weighted avg       0.67      0.67      0.67       201


Mean Squared Error (MSE): 0.33
Root Mean Squared Error (RMSE): 0.58
R² Score (Goodness of Fit): -0.33
```

Results - EMG

88% accuracy

- Better at predicting low risk than high risk
- Could have improved by removing first and last segments of data
- Random Forest display better result than logistic regression
  - EMG data often has non linear pattern
  - less sensitive to outlier

Evaluation Metrics - IMU

```
Accuracy        : 0.52
Precision       : 0.78
Recall          : 0.07
F1 Score        : 0.13

Confusion Matrix:
[[98  2]
 [94  7]]

Classification Report:
              precision    recall  f1-score   support

     LowRisk       0.51      0.98      0.67       100
    HighRisk       0.78      0.07      0.13       101

    accuracy                           0.52       201
   macro avg       0.64      0.52      0.40       201
weighted avg       0.64      0.52      0.40       201

Mean Squared Error (MSE)   : 0.48
Root Mean Squared Error    : 0.69
R² Score (Goodness of Fit) : -0.91
```

Results - IMU

- Inaccurate (52%) prediction using IMU data (random forest)
  - Always labeled as low risk
  - Indiscriminate accelerometer/gyro magnitudes between High and Low Risk
- Potential improvements to model
  - change window size and rolling window
  - allow better differentiation between two test subjects
  - preprocess and remove outlier

Discussion

- Muscle strain / fatigue is directly linked to the amount of weight carried
  - Explains accurate prediction using EMG data
- Posture and acceleration may only differentiate when close to failure
  - Inaccurate prediction using IMU data
  - Better to capture data from beginning and end of unlimited reps to failure
- Focused on recall score
  - false negative(missed High Risk case) could risk potential injuries
  - false positive may cause extra caution but no harm

Results & Discussion

- Recall was higher for HighRisk in the random forest EMG model
    - critical for injury prevention
    - EMG recall = 0.93
    - In contrast logistic regression recall was 0.7
- IMU features with current setup lack discriminative power
    - accelerometer and gyroscope magnitudes were too similar between High and Low Risk
    - model default to predicting LowRisk for almost all sample
        - could lead to potential high risk of injury
- Consider combining EMG + IMU features to enhance classification

Conclusion

It is possible to accurately predict the risk of WMSDs using a machine learning model, but muscle strain / fatigue (EMG) is a better indicator than posture / acceleration (IMU).

References

1. Hudgins, B., Parker, P., & Scott, R. N. (2000). *A new strategy for multifunction myoelectric control.* IEEE Transactions on Biomedical Engineering. jneuroengrehab.biomedcentral.com

2. Phinyomark, A., Khushaba, R. N., & Scheme, E. (2018). *Feature extraction and selection for myoelectric control.* IEEE Reviews in Biomedical Engineering. researchgate.net

3. Hume, P. A., & Groll, J. H. (2025). *Decision Trees and IMU Sensors for Risk Prediction in Manual Material Handling.* Springer. link.springer.com

4. Hossain, S., et al. (2023). *Hybrid Learning Models for IMU-Based HAR with Feature Analysis.* Sensors, 23(18), 7802. mdpi.com

5. Di Nardo, F., et al. (2024). *Biomechanical Risk Classification in Repetitive Lifting Using Multi-Sensor Data.* Biosensors, 15(2), 84. mdpi.com