

# High-Dimensional Classification with PCA

## Learning Objectives

- Understand the challenges of high-dimensional feature spaces in classification.
- Apply Principal Component Analysis (PCA) for dimensionality reduction.
- Compare classification performance before and after dimensionality reduction.
- Implement and evaluate models using R.

## Assignment Details

**Dataset:** Human Activity Recognition Using Smartphones Dataset

Available at: <https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones>

**Training Data Dimensions:** 7352 observations, 561 numeric features

**Classes:** Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing, Laying

The dataset consists of motion sensor readings (accelerometer and gyroscope) from smartphones worn by 30 subjects during daily activities. Each instance has 561 features extracted using time and frequency domain techniques. Due to high dimensionality, traditional classifiers may overfit, making PCA-based dimensionality reduction a key preprocessing step.

## Suggested Steps to Follow

### Step 1: Data Loading and Exploration

1. Load the dataset into R.
2. Perform some exploratory data analysis: check feature dimensions, class labels, and summary statistics.
3. Standardize the features if necessary.

### Step 2: Baseline Classification

1. Train a baseline model (e.g., logistic regression or LDA) using all original features.
2. Evaluate performance using accuracy, precision, recall, F1-score, and confusion matrix.
3. Report and interpret the results. reflect on *curse of dimensionality*.

### Step 3: Dimensionality Reduction with PCA

1. Apply PCA to reduce dimensionality of the feature set.
2. Determine the number of components to retain (e.g., using explained variance).
3. Transform the original dataset using the selected components.

### Step 4: Classification on PCA-Reduced Data

1. Train the same model (as in Step 2) on the PCA-reduced features.
2. Evaluate and compare the model performance.
3. Discuss any changes in metrics and computational efficiency.

### Step 5: Model Comparison and Analysis

1. Compare original vs. PCA-based models on all evaluation metrics.
2. Discuss trade-offs and suitability of PCA in high-dimensional classification tasks.
3. Suggest scenarios where PCA may or may not be beneficial.

## Suggested R Packages and Resources

#### Useful R Packages:

```
library(caret)
library(e1071)
library(ggplot2)
library(tidyverse)
library(factoextra)
```

#### References:

- [PCA in R \(UC Business Analytics\)](#)
- [PCA Tutorial on DataCamp](#)

# High Dimensional Classification with PCA

**Harshul Shah**  
**Lakshmipriya Narayanan**  
**Priyasri Sankaran**



# Content



- 1) Dataset
- 2) Data Pre-Processing
- 3) Exploratory Data Analysis
- 4) Model Fitting
- 5) PCA Model Fitting
- 6) Curse of Dimensionality
- 7) Kernel PCA
- 8) Comparison of the models and Conclusion
- 9) Future scope



# All About Our Data



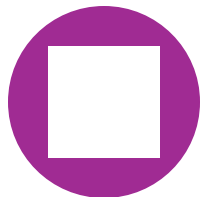
Human Activity Recognition Using  
[Smartphones Dataset](#)



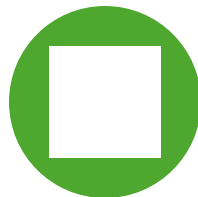
Sensor signals/sensor readings  
(accelerometer and gyroscope)  
from smartphones



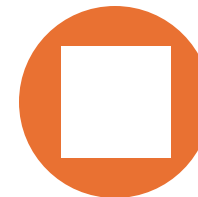
30 participants during daily  
activities. Between the age of 19-48



Using time and frequency domain  
techniques. Due to high  
dimensionality, traditional  
classifiers may overfit, making  
PCA-based dimensionality  
reduction a key preprocessing step.



563 Features and about 10,000  
observations



70% training and 30% test



# Data Preprocessing

Loaded feature names and activity labels to assign proper column names

Combined training data (features, activities, subjects) into a single file

Did the same for the test data

Replaced numeric activity codes with descriptive labels

The dataset was already standardized by its creators. So, all features were scaled and bounded between -1 and 1

No null values or duplicates

Features were carefully engineered to have similar scales

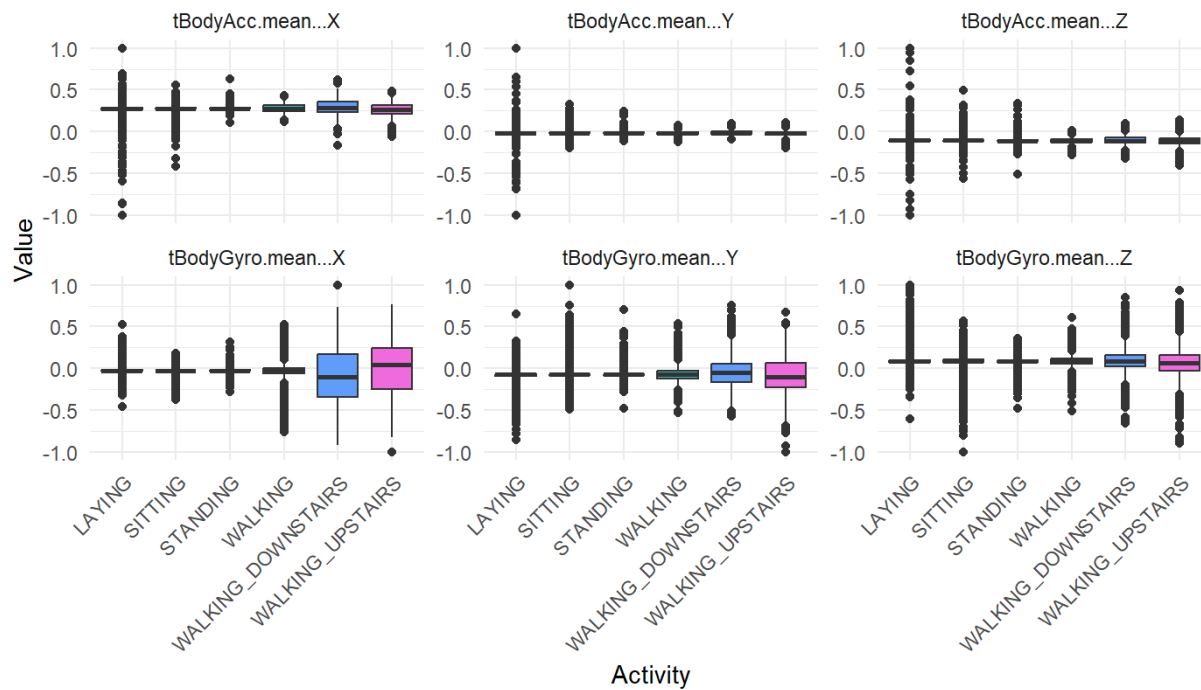
# Objective / Goal

Investigate how dimensionality reduction techniques like PCA and Kernel PCA, combined with classifiers such as multi-logistic regression and linear discriminant analysis, can overcome the curse of dimensionality to accurately predict human activity from high-dimensional smartphone sensor data.

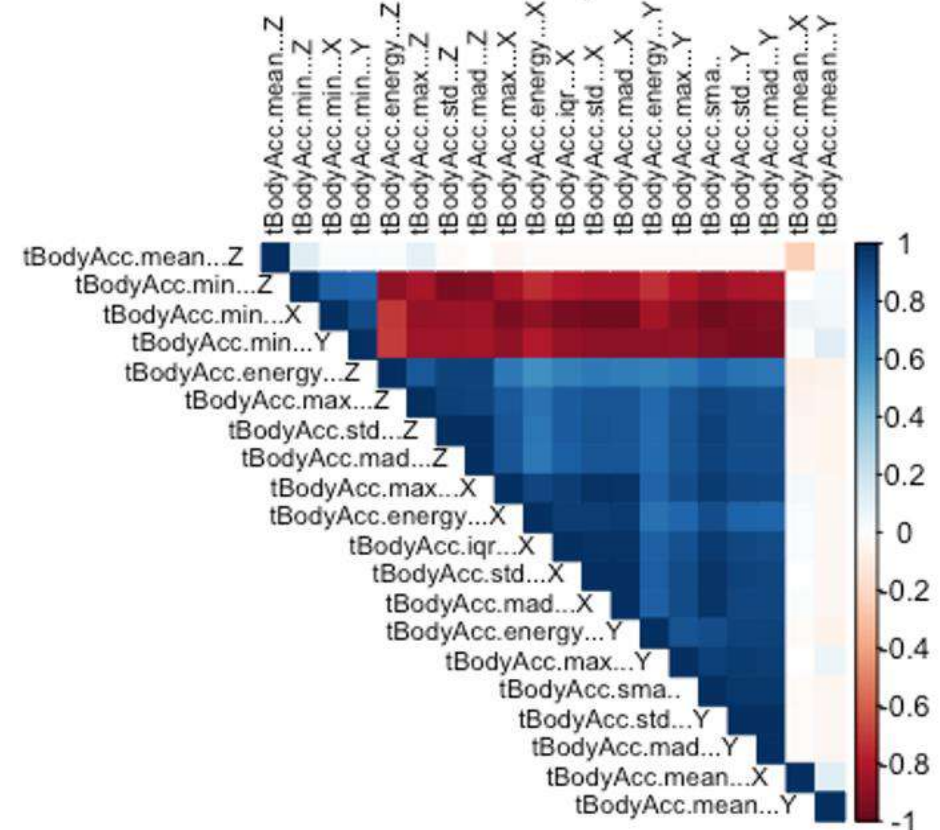


# Exploratory Data Analysis

Distribution of Key Features by Activity

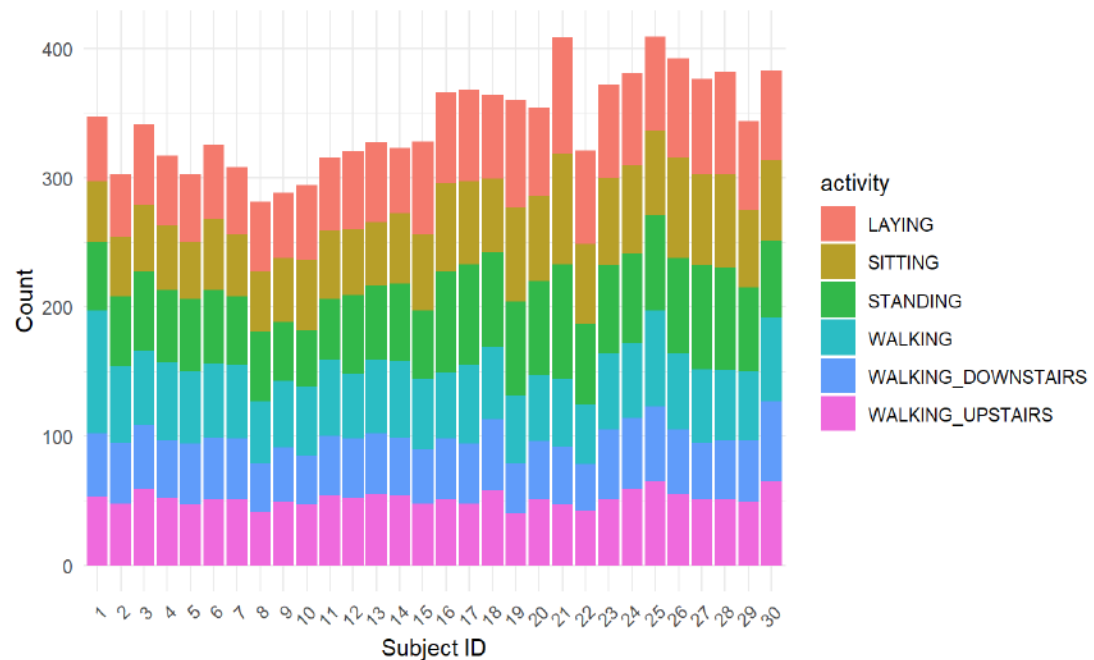


Correlation Matrix of Top 20 Features

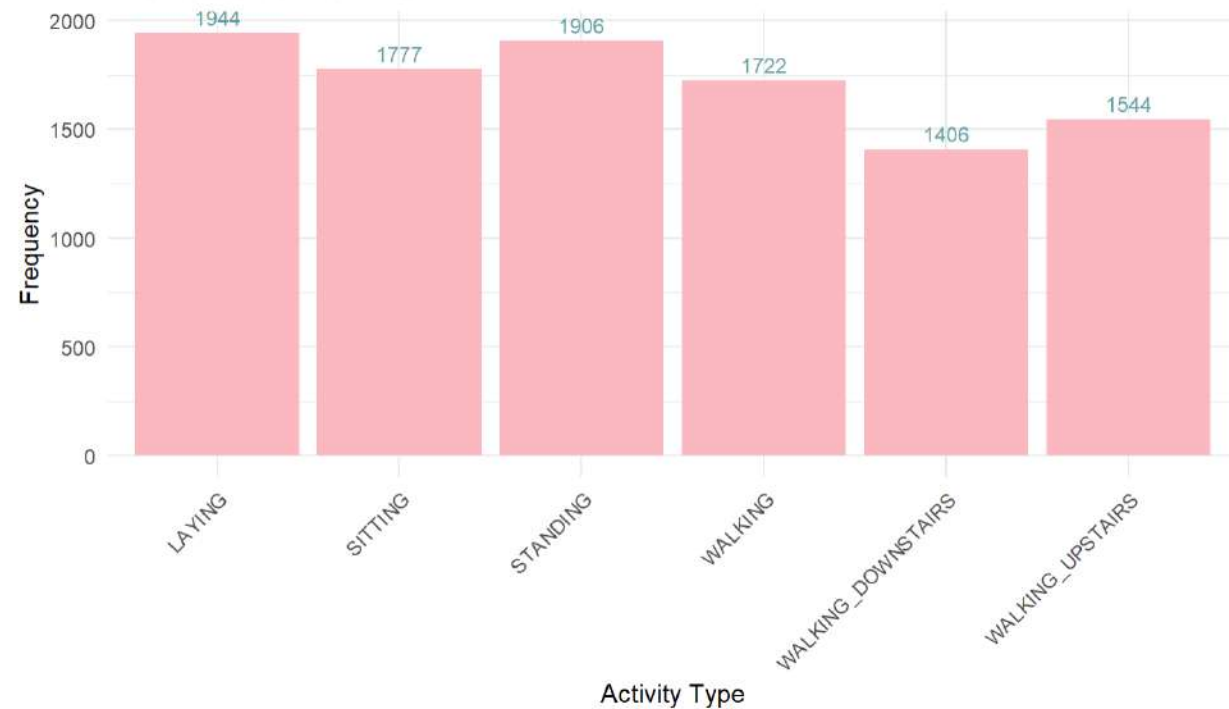




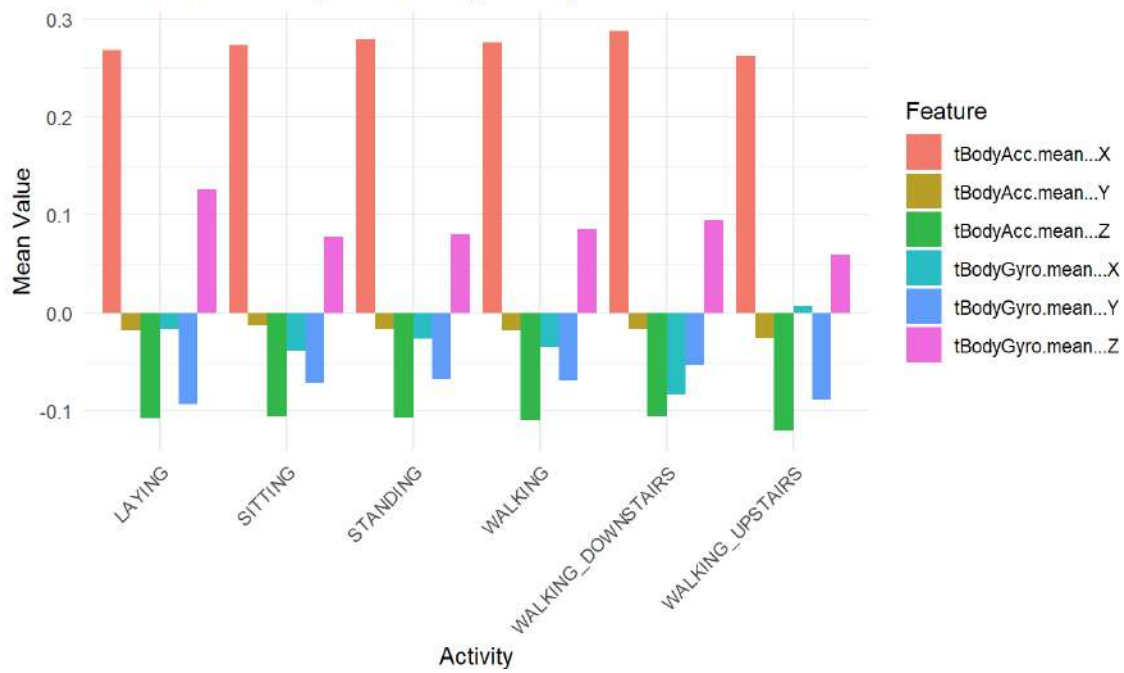
Activities by Subject



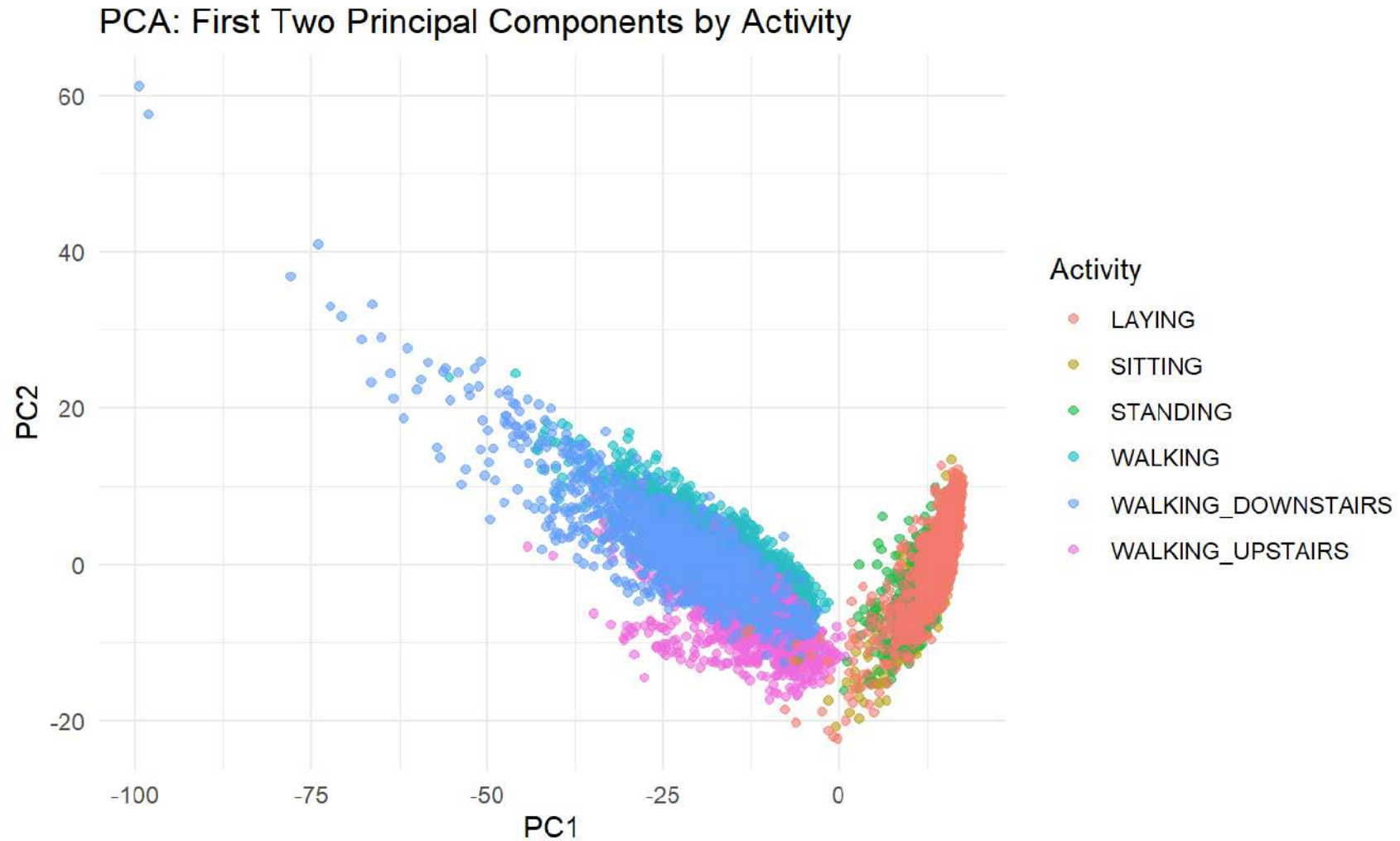
Distribution of Activities



Mean Values of Key Features by Activity

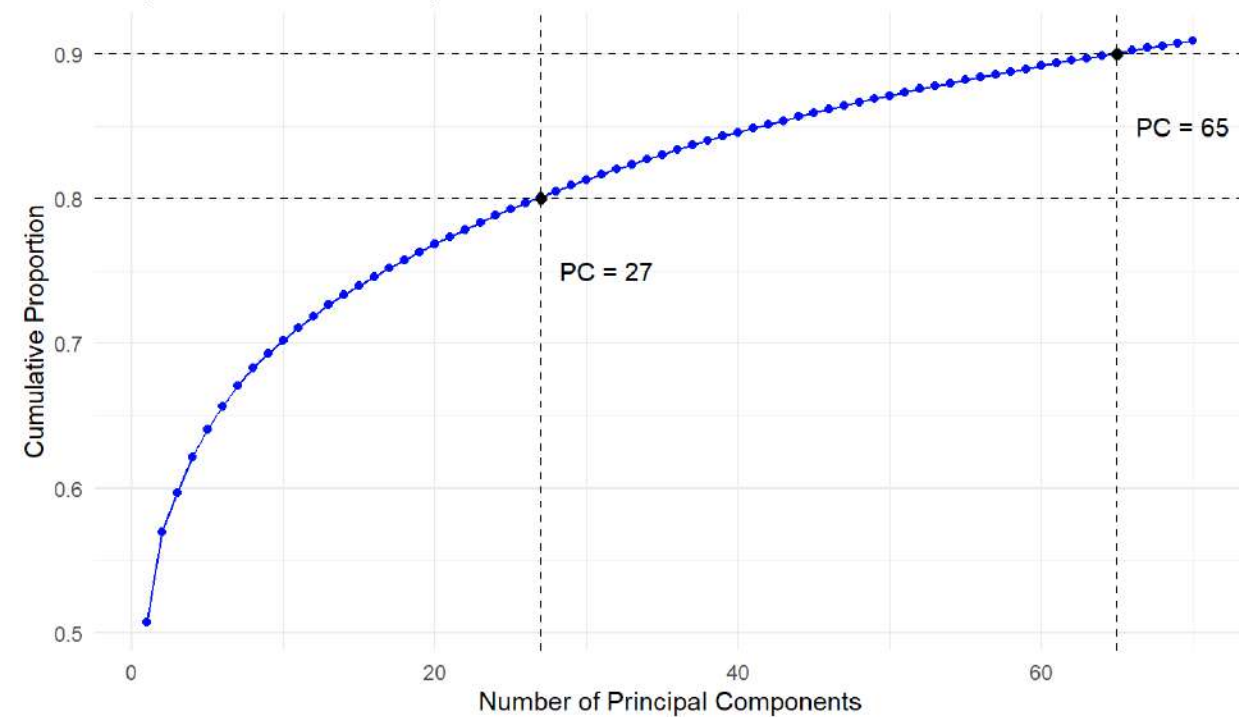


# EDA WITH PCA

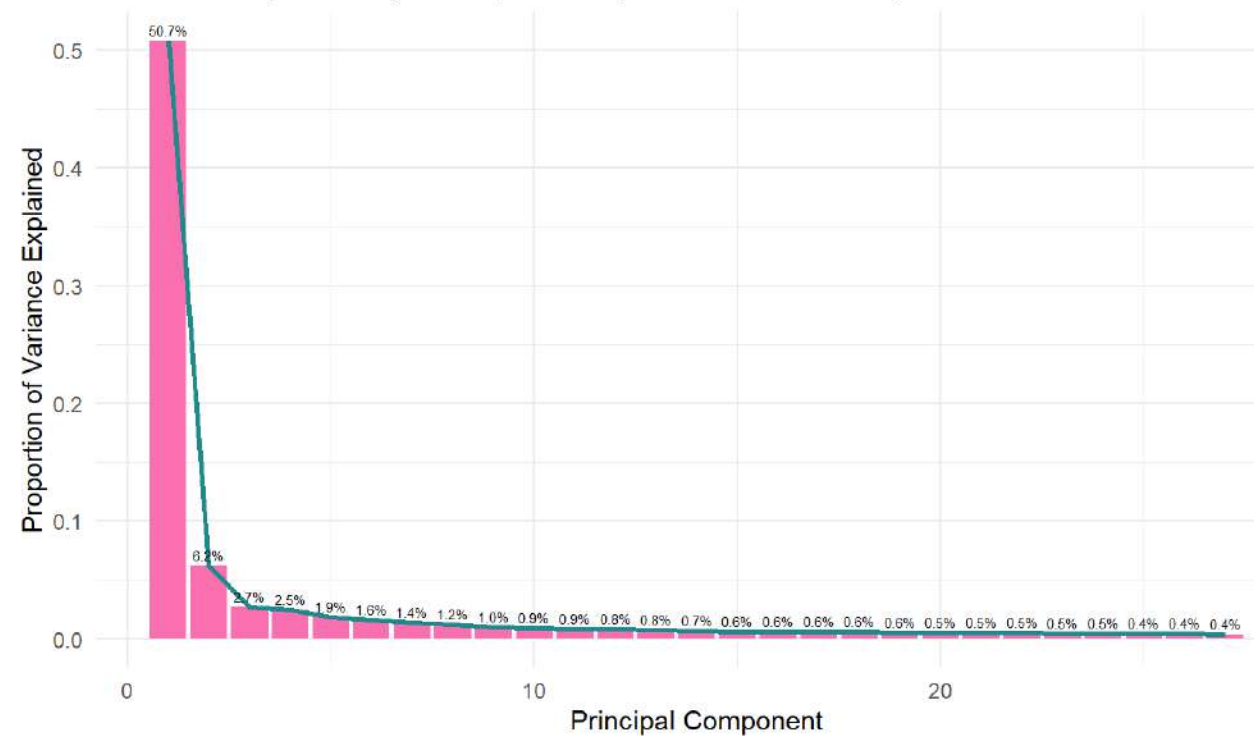


# So, how many PCs do we need ?

Proportion of variance explained for 80% and 90% cumulative variance

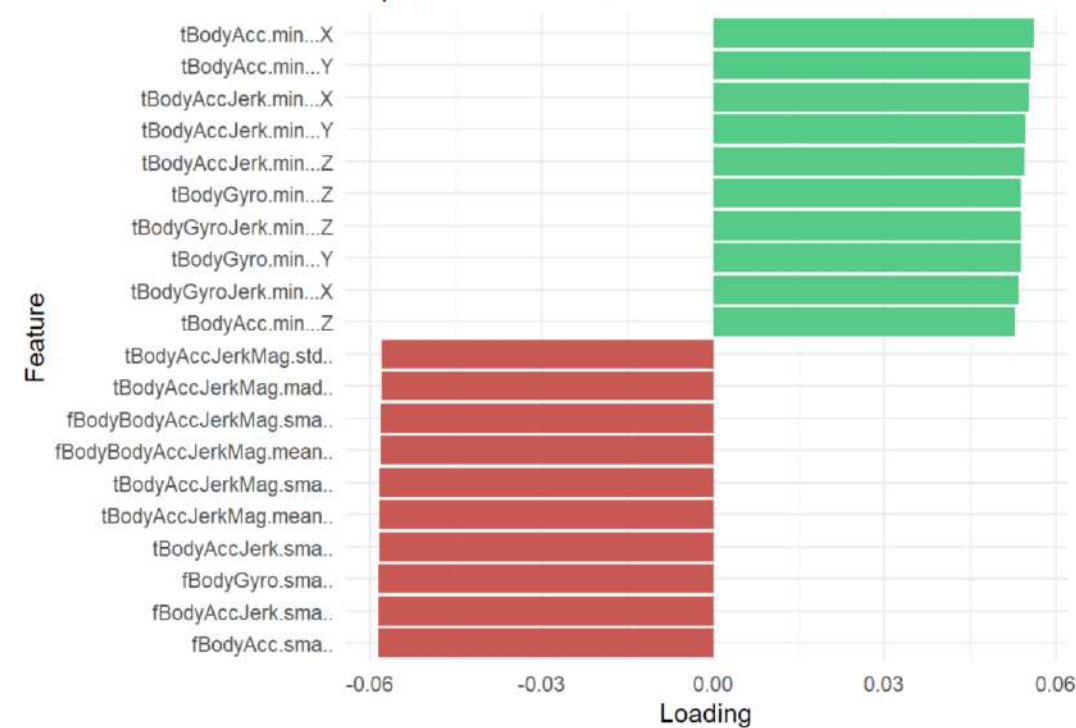


Variance Explained by Principal Components and Scree plot

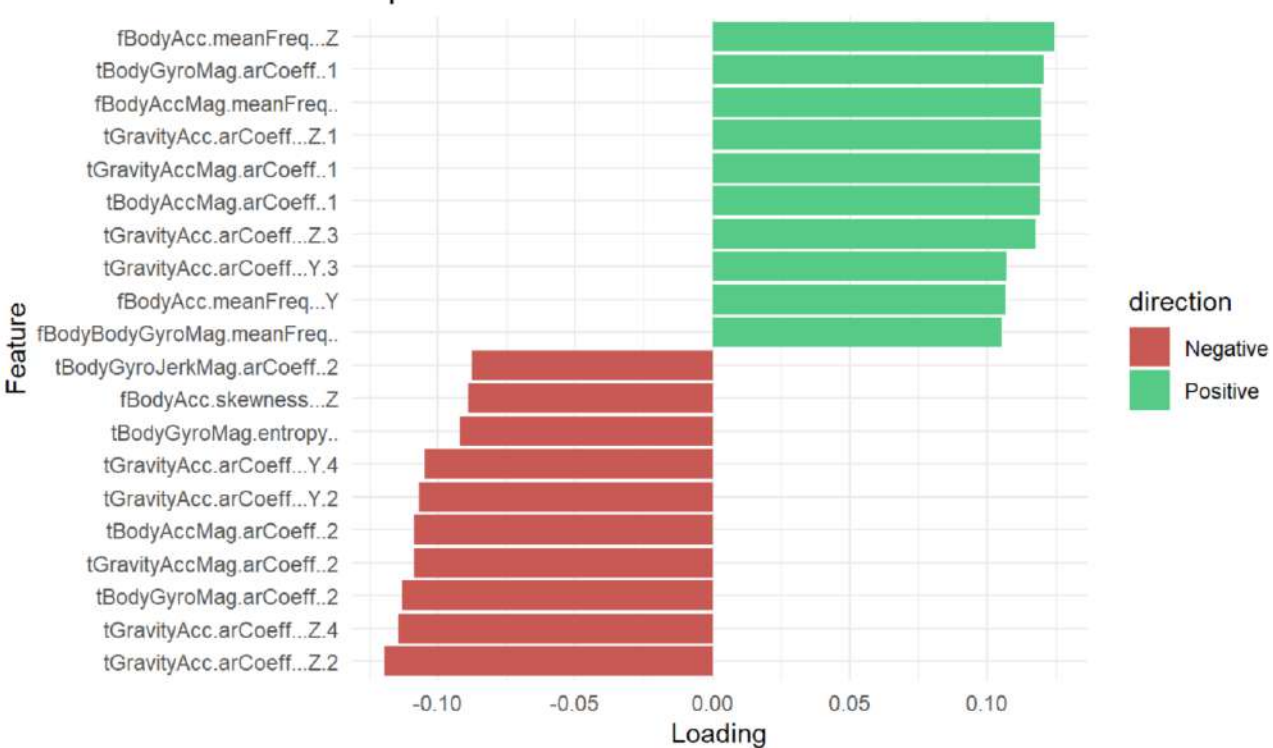


# Top feature loadings for the first two PCs

Top Feature Contributions to PC 1

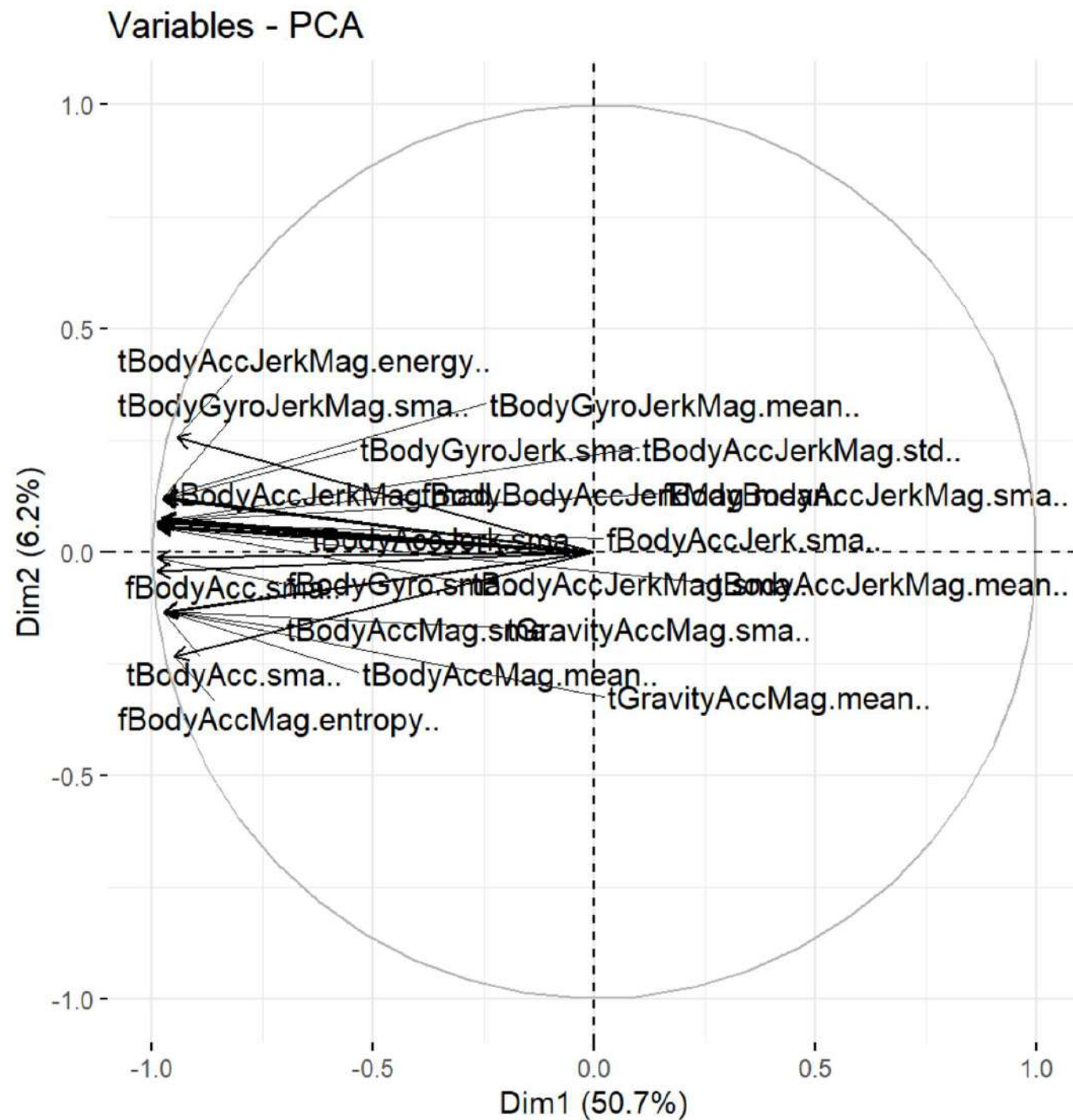


Top Feature Contributions to PC 2



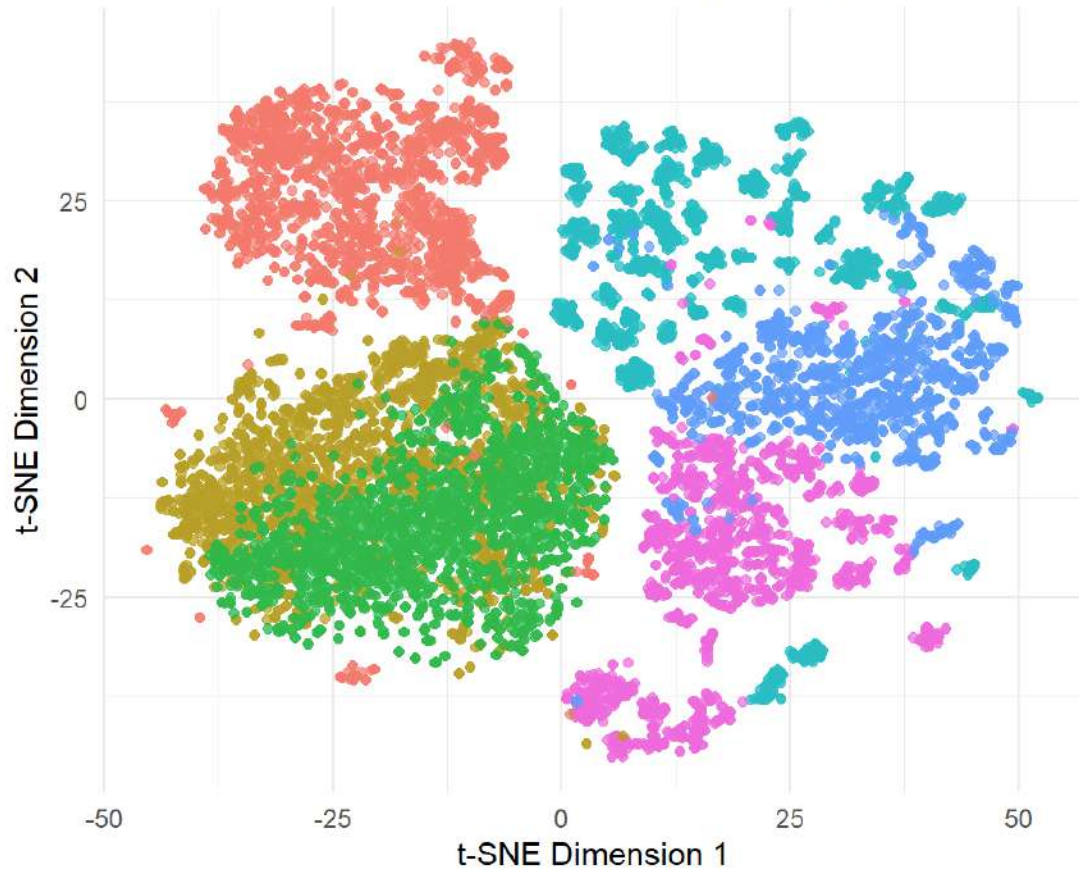


# Biplot

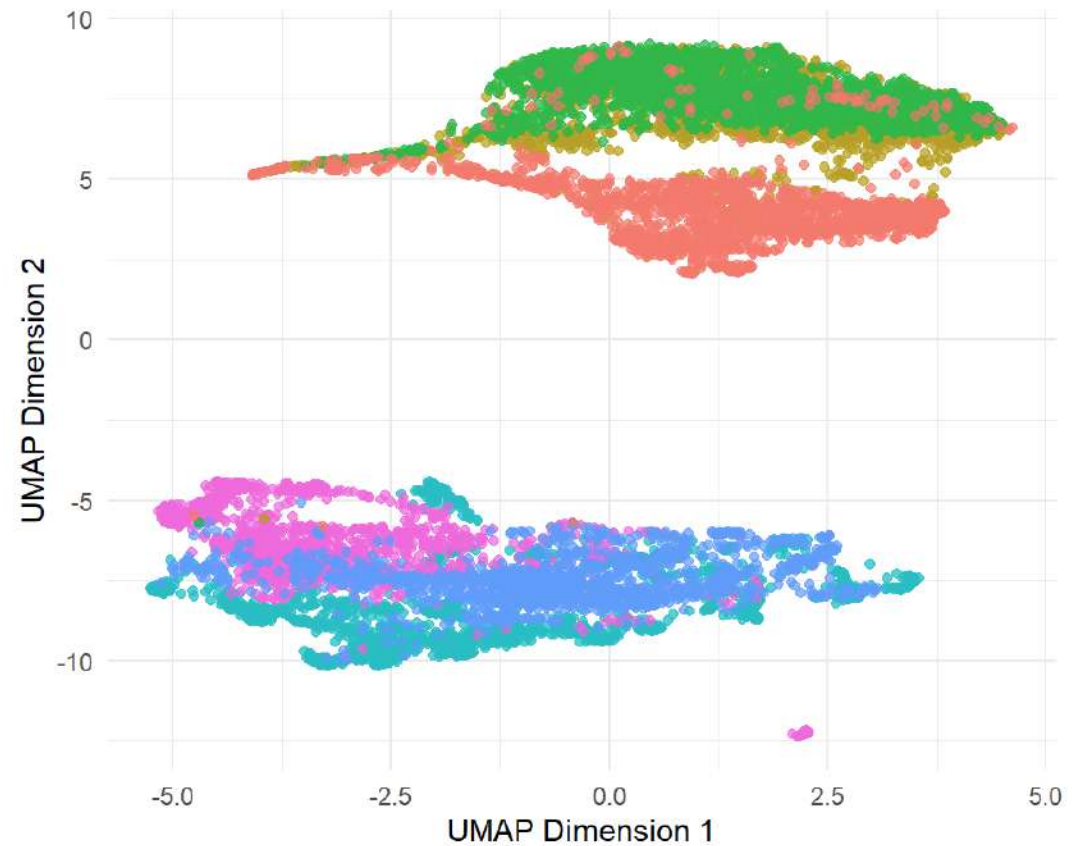


# UMAP and t-SNE

t-SNE Visualization of Human Activity Recognition Data



UMAP Visualization of PCA-Transformed Human Activity Data



● LAYING ● STANDING ● WALKING\_DOWNSTAIRS ● SITTING ● WALKING ● WALKING\_UPSTAIRS



Which activity do you think appears most distinctly separated from the others?

- A) Walking
- B) Walking Upstairs
- C) Walking Downstairs
- D) Sitting
- E) Standing
- F) Laying



Which activity do you think appears most distinctly separated from the others?

- A) Walking
- B) Walking Upstairs
- C) Walking Downstairs
- D) Sitting
- E) Standing
- F) Laying



# Baseline Classification Model Fitting

- ↙ Multiple logistic regression (MLR)
- ↙ Linear Discriminant Analysis (LDA)

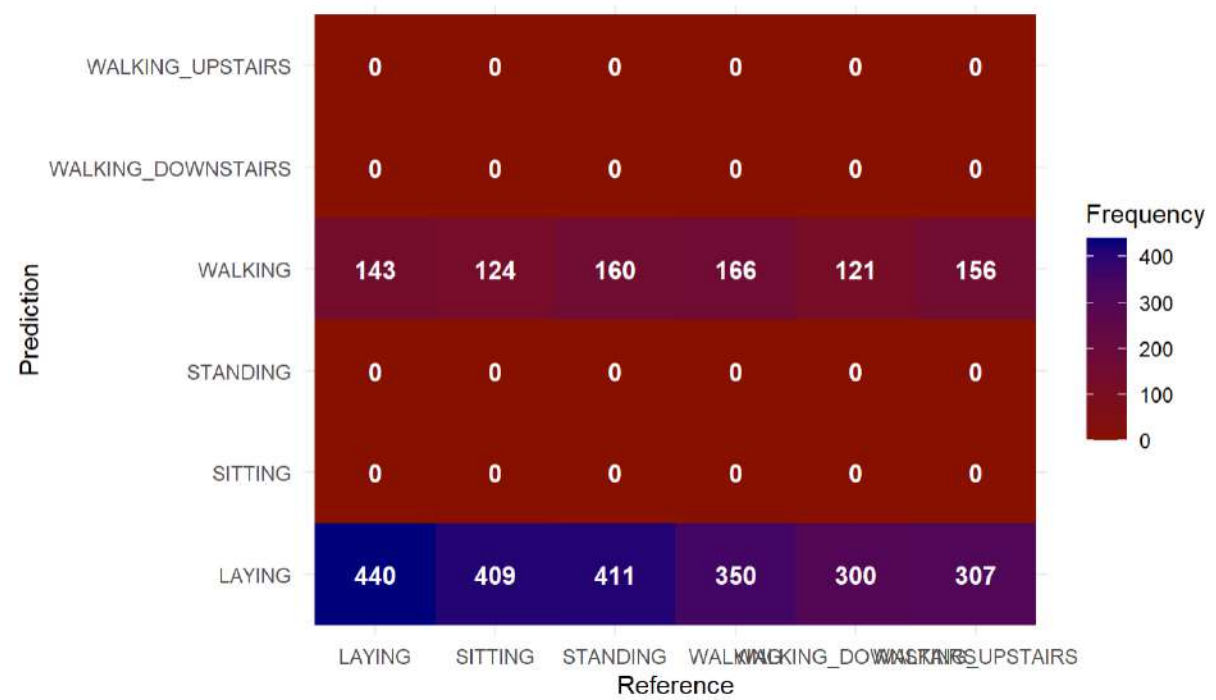




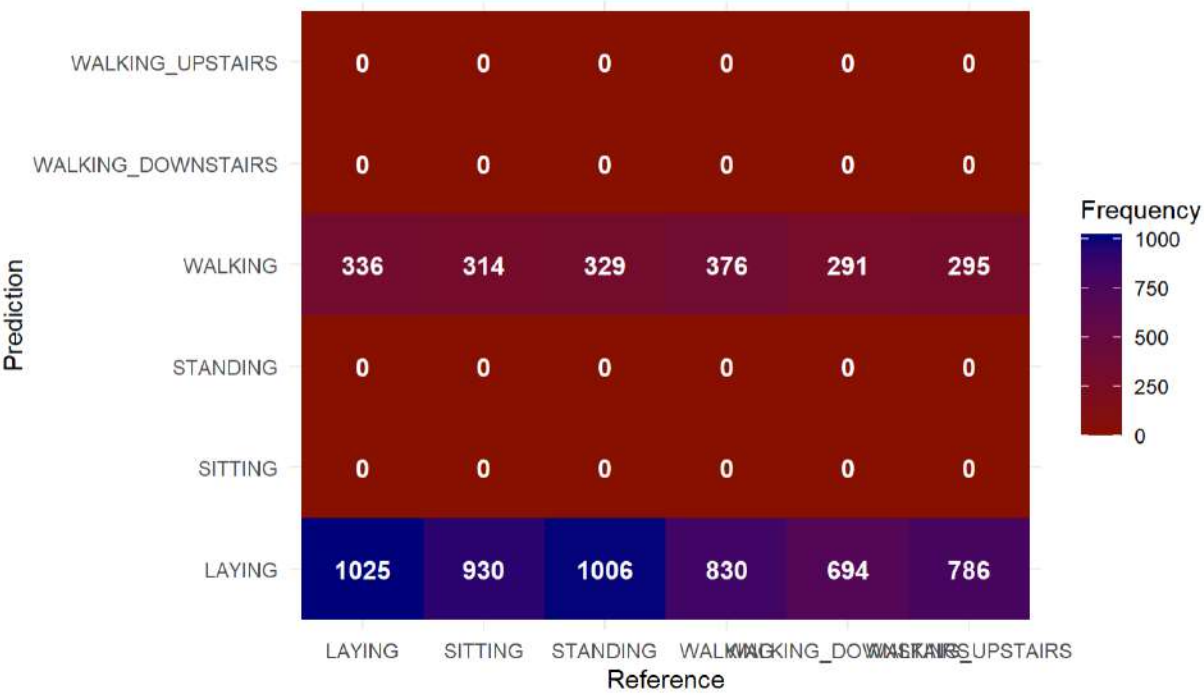
# Confusion Matrices and Performance results

# Confusion Matrix for MLR

Confusion Matrix - Multinomial Logistic Regression (Test)



Confusion Matrix - Multinomial Logistic Regression (Training)





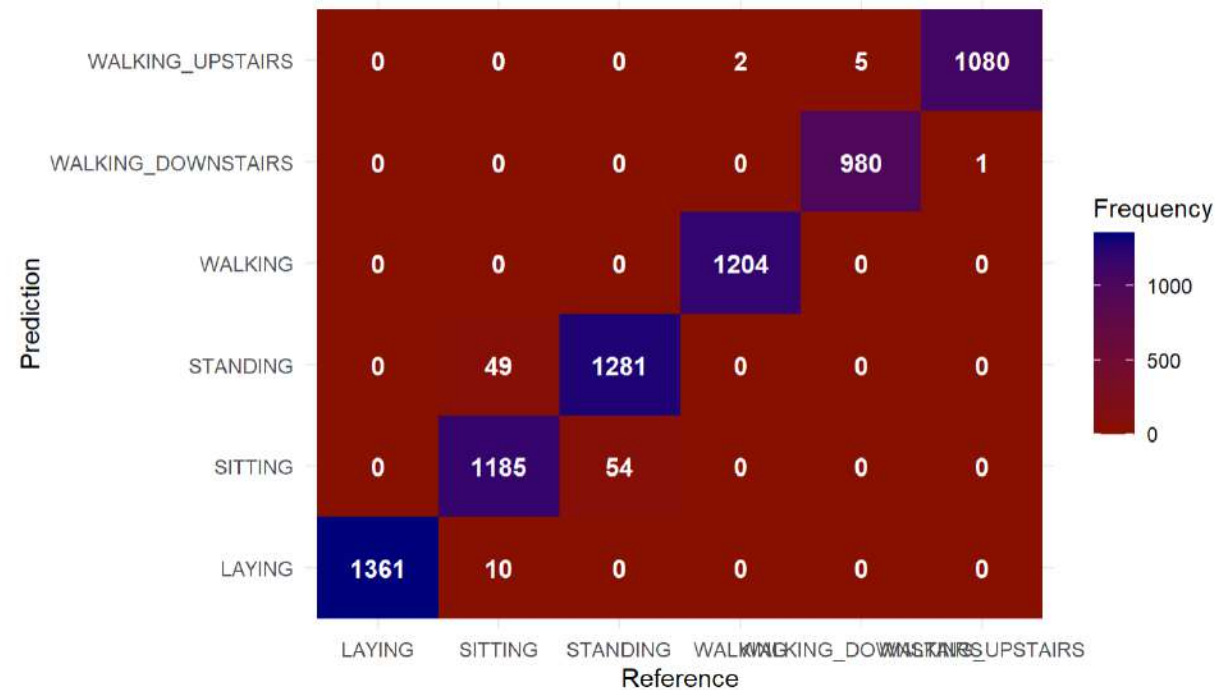
# Results of MLR (using a subset of the data)

Activity	Test Set Evaluation Metrics			Training Set Evaluation Metrics		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Laying	0.1984	0.754	0.314	0.1944	0.753	0.309
Sitting	NaN	0	NaN	NaN	0	NaN
Standing	NaN	0	NaN	NaN	0	NaN
Walking	0.1908	0.321	0.239	0.193	0.311	0.238
Walking Downstairs	NaN	0	NaN	NaN	0	NaN
Walking Upstairs	NaN	0	NaN	NaN	0	NaN
Overall Accuracy	19.63 %			19.43 %		

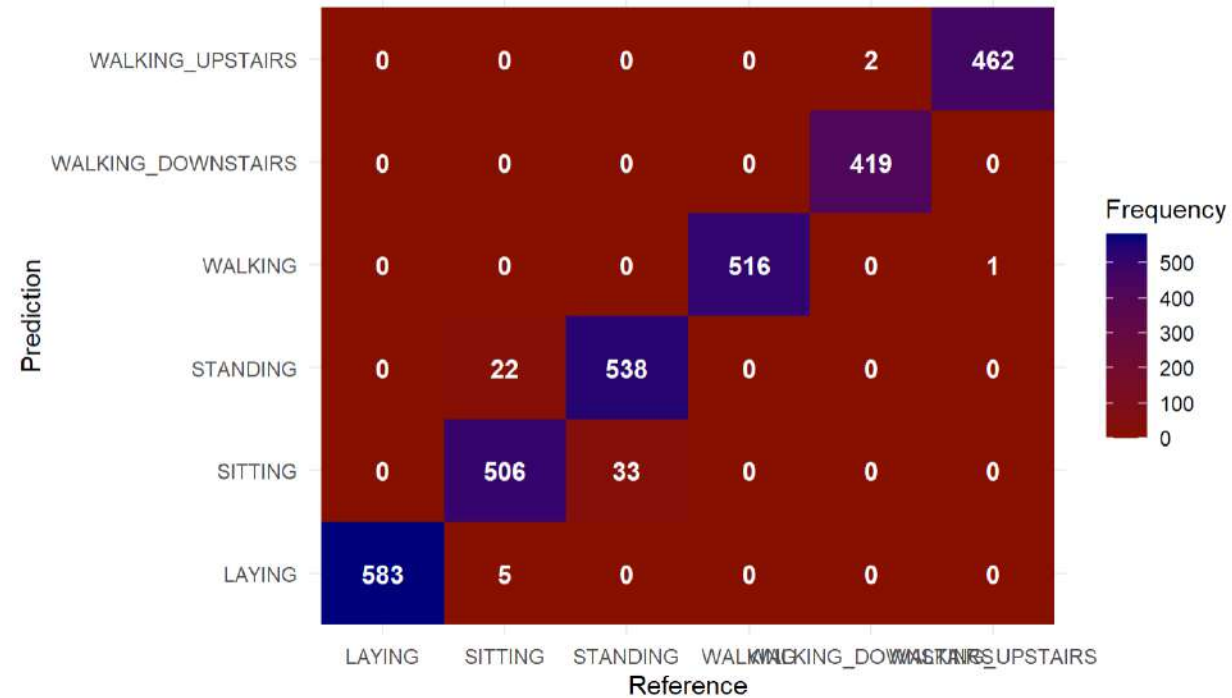


# Confusion Matrix for LDA

Confusion Matrix - Linear Discriminant Analysis (Training)



Confusion Matrix - Linear Discriminant Analysis (Test)



# Results of LDA

Activity	Test Set Evaluation Metrics			Training Set Evaluation Metrics		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Laying	0.991	1	0.995	0.993	1	0.996
Sitting	0.938	0.949	0.944	0.958	0.955	0.956
Standing	0.960	0.942	0.951	0.965	0.961	0.963
Walking	0.998	1	0.999	1	0.999	0.999
Walking Downstairs	1	0.995	0.997	0.998	0.996	0.997
Walking Upstairs	0.995	0.997	0.996	0.996	0.999	0.997
Overall Accuracy	97.96 %			98.32 %		

# Problems encountered while fitting basic model

## Multiple Logistic Regression:

```
Error in nnet.default(X, Y, w, mask = mask, size = 0, skip = TRUE, softmax = TRUE,  :  
  too many (3384) weights
```

## Linear Discriminant Analysis:

```
Warning in lda.default(x, grouping, ...) : variables are collinear
```

## Classification Model Fitting after PCA

- ↙ Multiple logistic regression (MLR)
- ↙ Linear Discriminant Analysis (LDA)



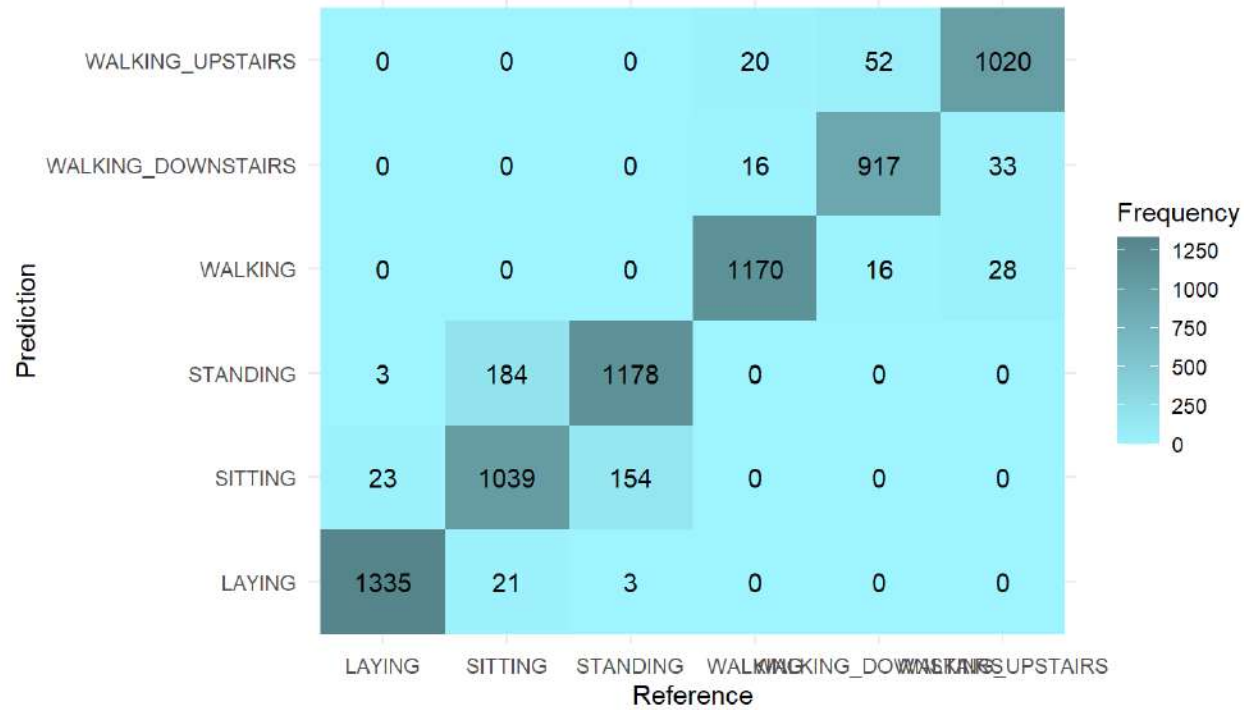




# Confusion Matrices and Performance results after PCA

# Confusion Matrix for MLR with PCA

Confusion Matrix - Multinomial Logistic Regression (Training)



Confusion Matrix - Multinomial Logistic Regression (Test)



# Results of MLR with PCA

Activity	Test Set Evaluation Metrics			Training Set Evaluation Metrics		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Laying	0.97	0.95	0.96	0.98	0.98	0.98
Sitting	0.83	0.81	0.82	0.85	0.83	0.84
Standing	0.84	0.88	0.86	0.86	0.88	0.87
Walking	0.94	0.95	0.95	0.96	0.97	0.96
Walking Downstairs	0.92	0.92	0.92	0.94	0.93	0.94
Walking Upstairs	0.94	0.92	0.93	0.93	0.94	0.93
Overall Accuracy	91.03 %			92.33 %		



# Confusion Matrix for LDA with PCA

Confusion Matrix - Linear Discriminant Analysis (Training)



Confusion Matrix - Linear Discriminant Analysis (Test)





# Results of LDA with PCA

Activity	Test Set Evaluation Metrics			Training Set Evaluation Metrics		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Laying	0.98	0.99	0.99	0.98	0.99	0.99
Sitting	0.86	0.79	0.82	0.87	0.79	0.83
Standing	0.83	0.89	0.863	0.83	0.89	0.86
Walking	0.96	0.96	0.962	0.95	0.96	0.96
Walking Downstairs	0.97	0.94	0.959	0.98	0.92	0.95
Walking Upstairs	0.92	0.95	0.938	0.91	0.95	0.93
Overall Accuracy	85.1 %			86.95 %		

# Curse of Dimensionality

The Curse of Dimensionality refers to the

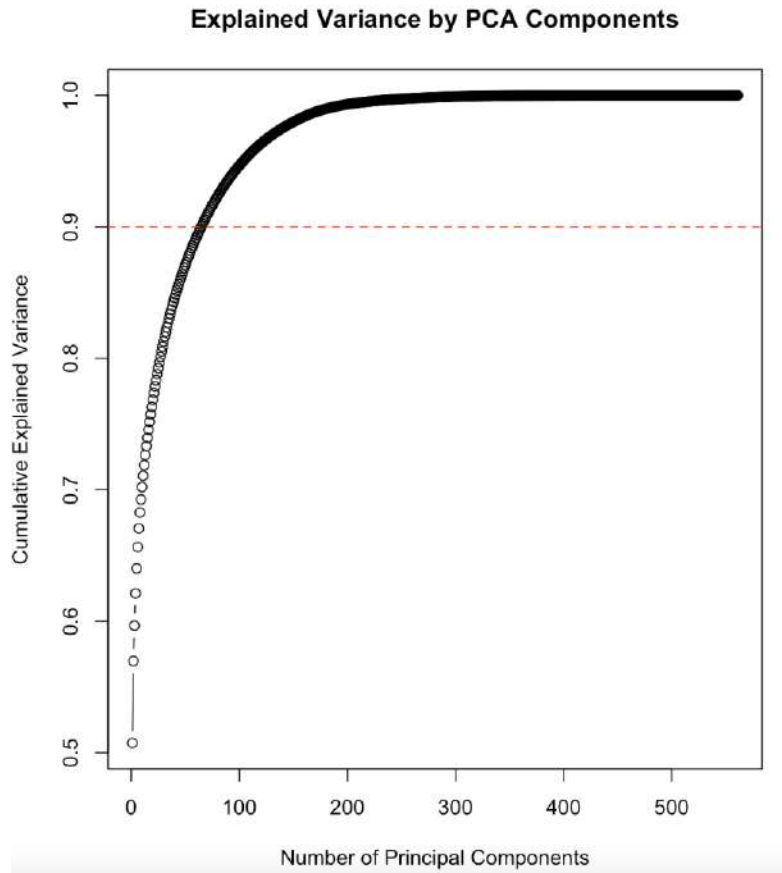
- Exponential increase in data sparsity and computational complexity as the number of features (dimensions) increase.
- Makes models harder to train, generalize, and interpret.

It's a Problem because:

- It increases overfitting risk
- Slows down training time due to evaluation of redundant features
- Reduces accuracy and interpretability to be resolved by LDA or other techniques.

In the dataset, during LDA classification, R issued a collinearity warning, showing that many features were redundant or linearly dependent. This impacted model performance and interpretability.

# Curse of Dimensionality



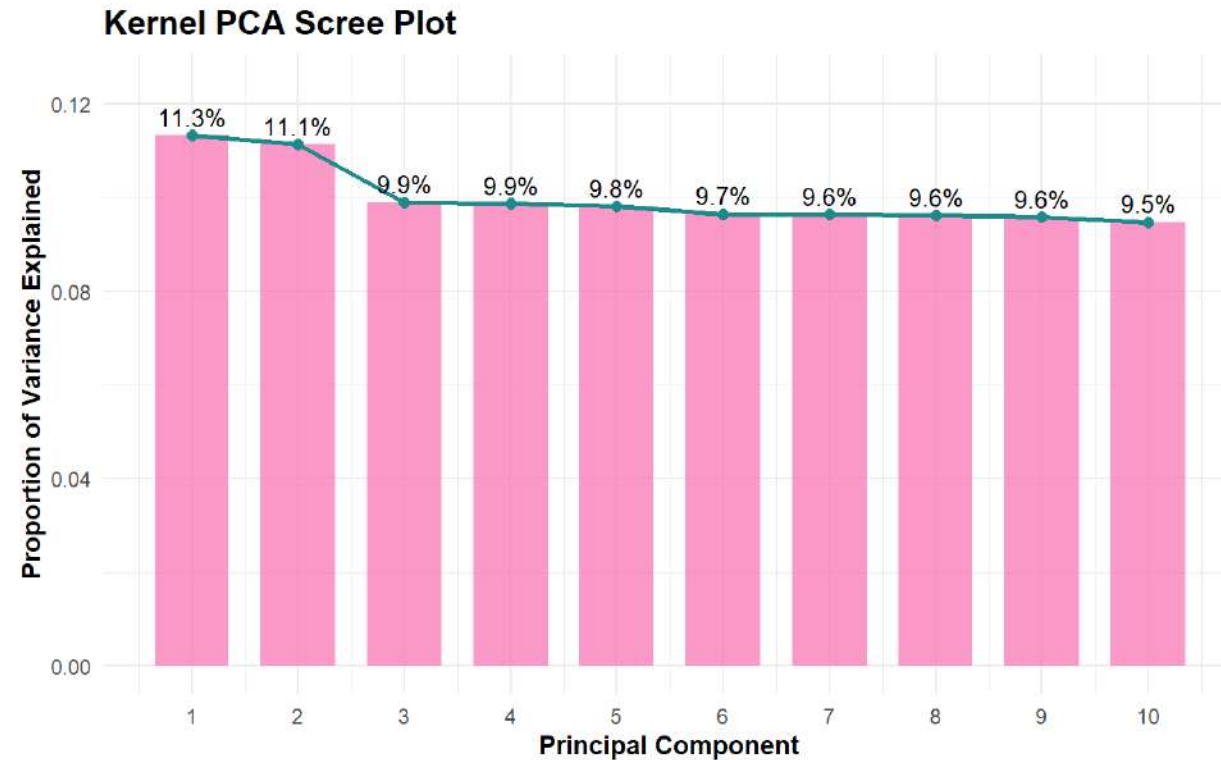
Scree Plot here shows that the first ~50 principal components explain over 90% of the variance. This proves that most features were unnecessary and contributed little unique information.

**Is it possible to simplify the dataset without losing valuable information?**

**Yes! - Kernel PCA**

# Why Kernel PCA ?

- Primarily captures non – linear relationships.
- Reduces dimension to a manageable number of components while preserving information needed to distinguish between activities – Just like PCA!
- Spreads the variance more evenly compared to PCA.
- About 8 PCs is sufficient to explain 80% of proportion of variance.



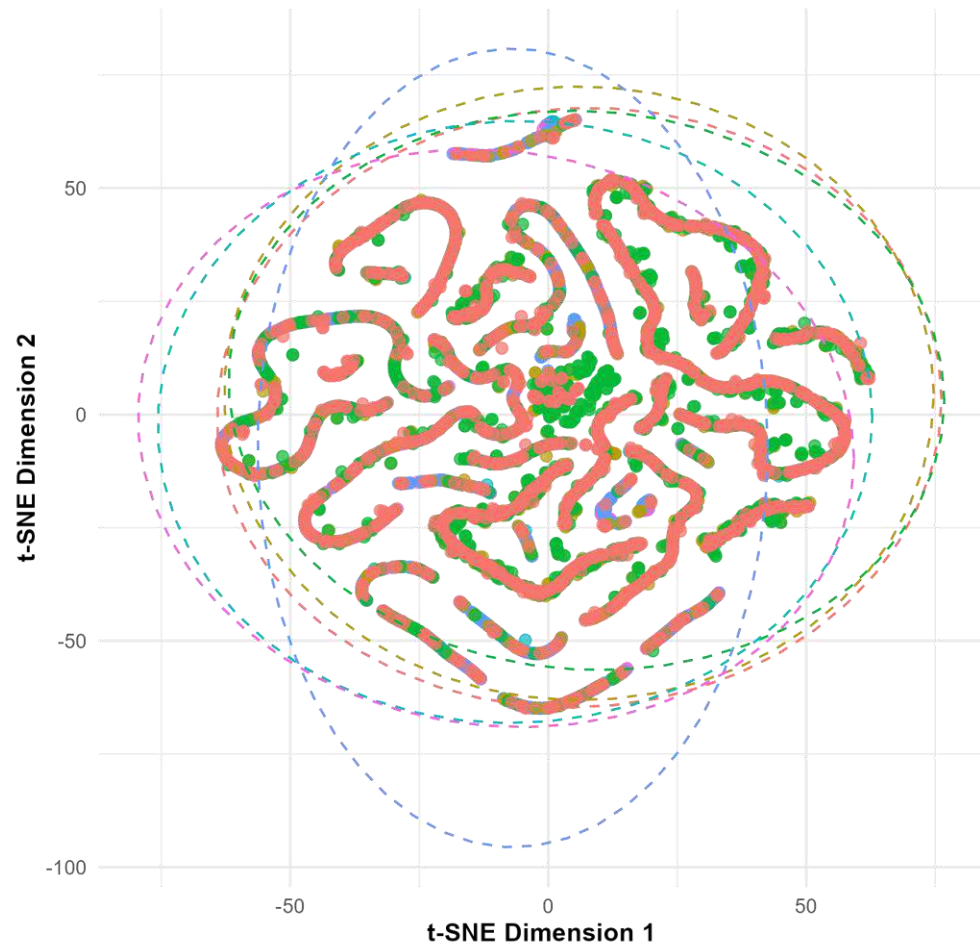


# What is Kernel PCA ?

- Smarter version of regular PCA that helps us deal with non-linear data.
- Kernel PCA lifts the data into a higher dimensional space – like unfolding a crumpled paper.
- Hidden or messy patterns become clear and easier to separate.
- We use the RBF (Radial Basis Function) kernel for this dataset to capture non-linear relationships.
- This method measures similarity between two data points based on their distance.

# EDA with Kernel PCA

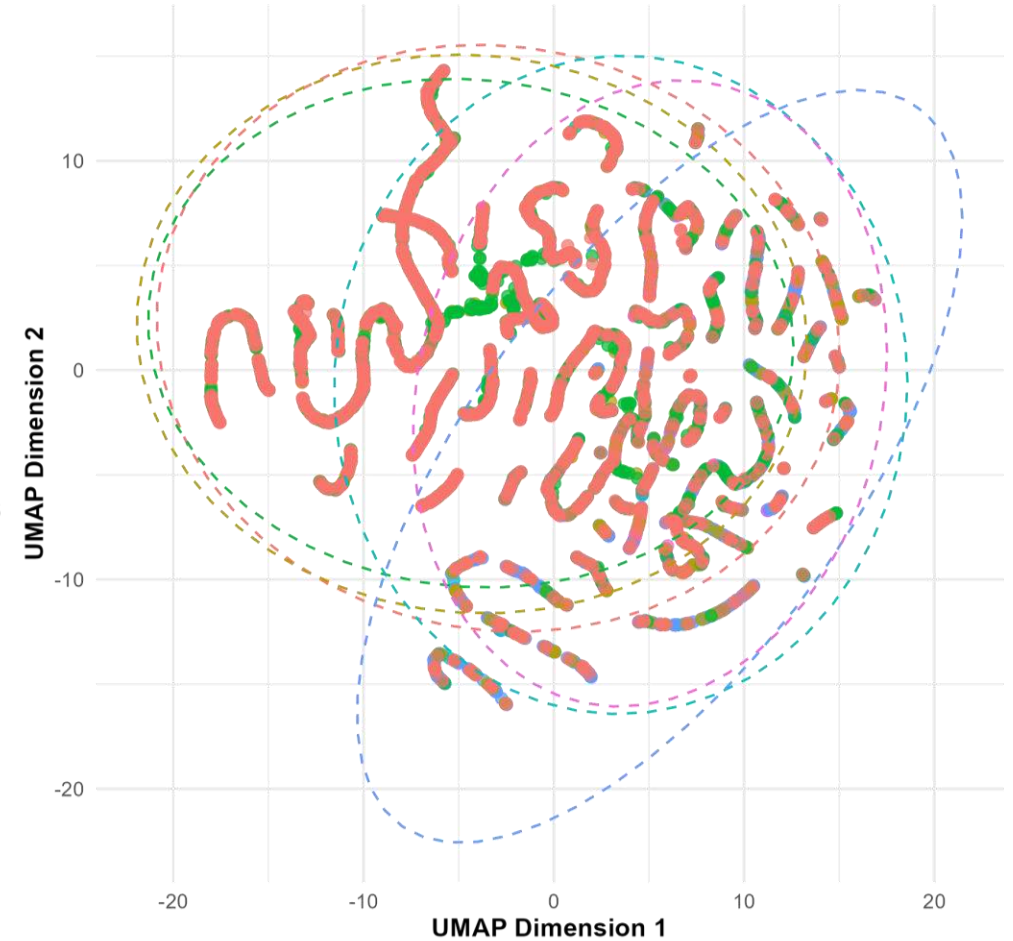
t-SNE Visualization of Kernel PCA Features



## Activity

- LAYING
- SITTING
- STANDING
- WALKING
- WALKING\_DOWNSTAIRS
- WALKING\_UPSTAIRS

UMAP Visualization of Kernel PCA Features



# Classification Model Fitting with Kernel PCA

- ↙ Multiple logistic regression (MLR)
- ↙ Linear Discriminant Analysis (LDA)

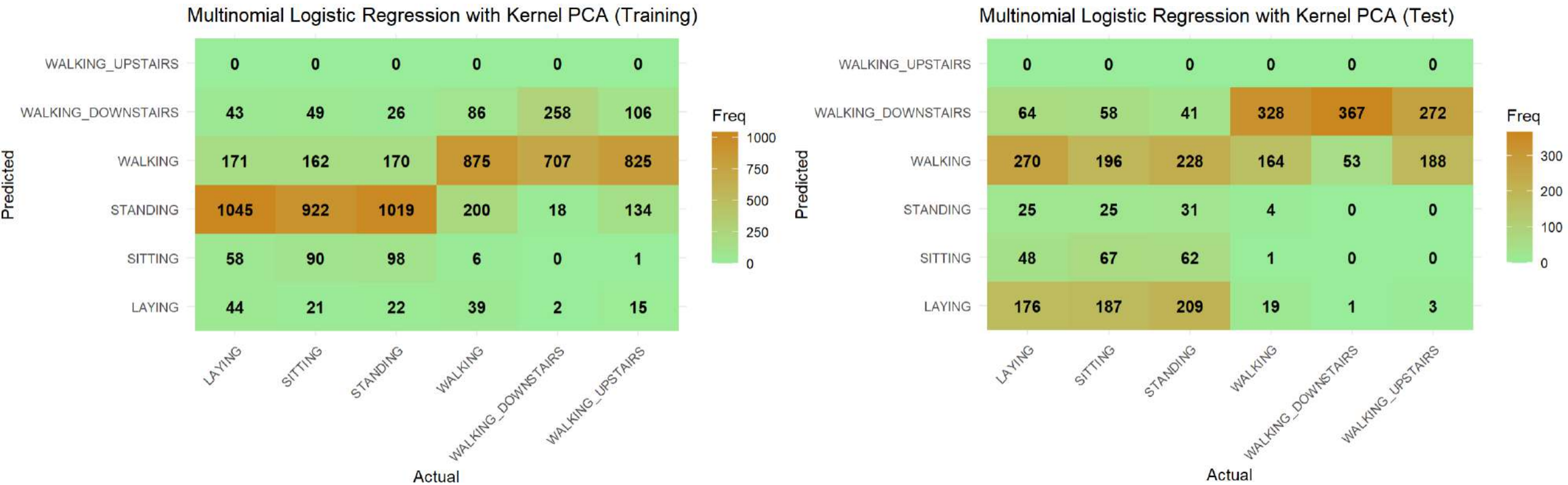






# Confusion Matrices and Performance results for Kernel PCA

# Confusion Matrix for MLR with Kernel PCA



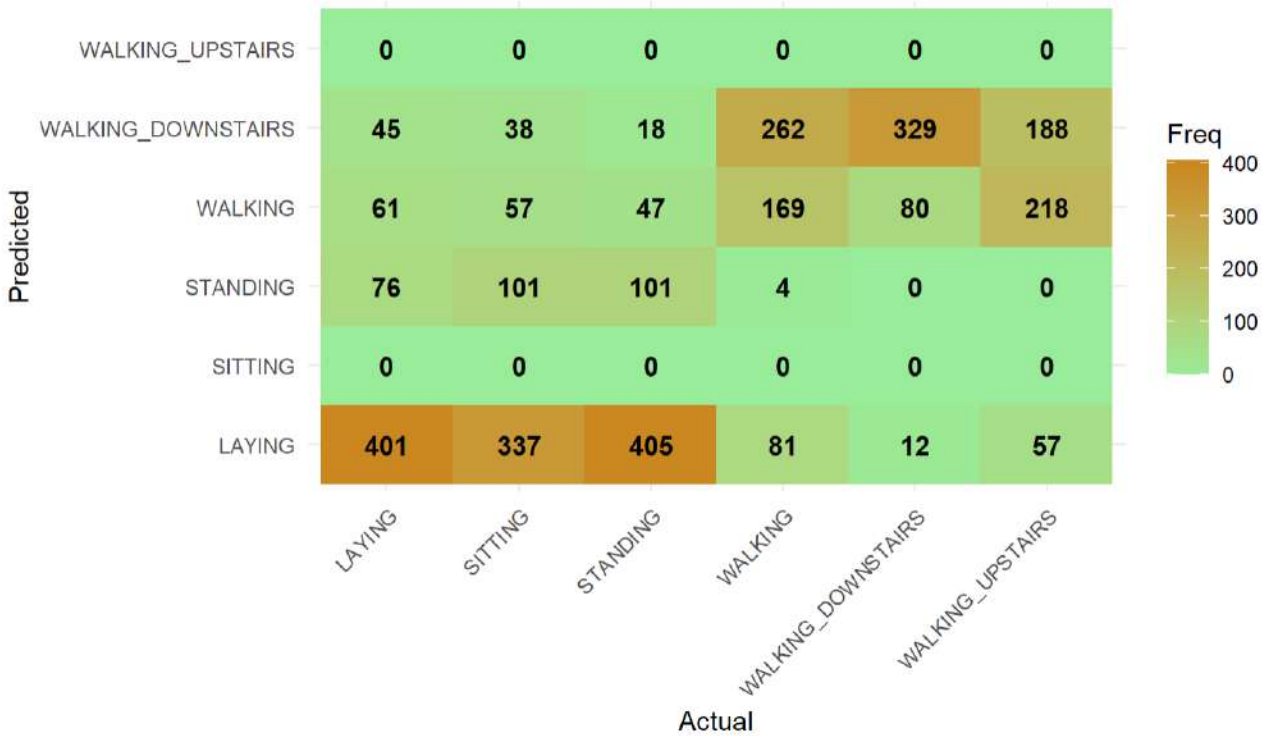


# Results of MLR with Kernel PCA

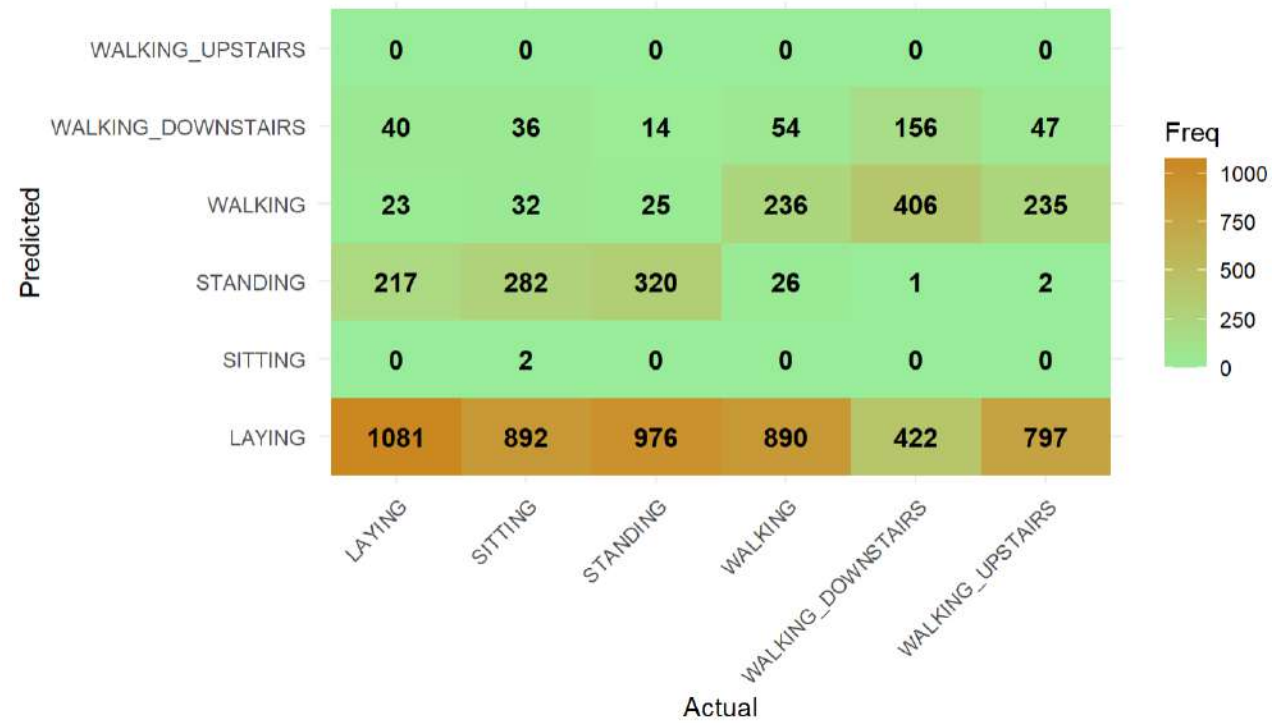
Activity	Test Set Evaluation Metrics			Training Set Evaluation Metrics		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Laying	0.30	0.28	0.29	0.29	0.59	0.39
Sitting	0	0	NaN	1	0	1
Standing	0.34	0.21	0.26	0.36	0.41	0.37
Walking	0.16	0.37	0.22	0.30	0.63	0.24
Walking Downstairs	0.34	0.81	0.48	0.45	0.20	0.45
Walking Upstairs	NA	0	NA	NA	0	NA
Overall Accuracy	26 %			31.7 %		

# Confusion Matrix for LDA with Kernel PCA

LDA with Kernel PCA (Test)



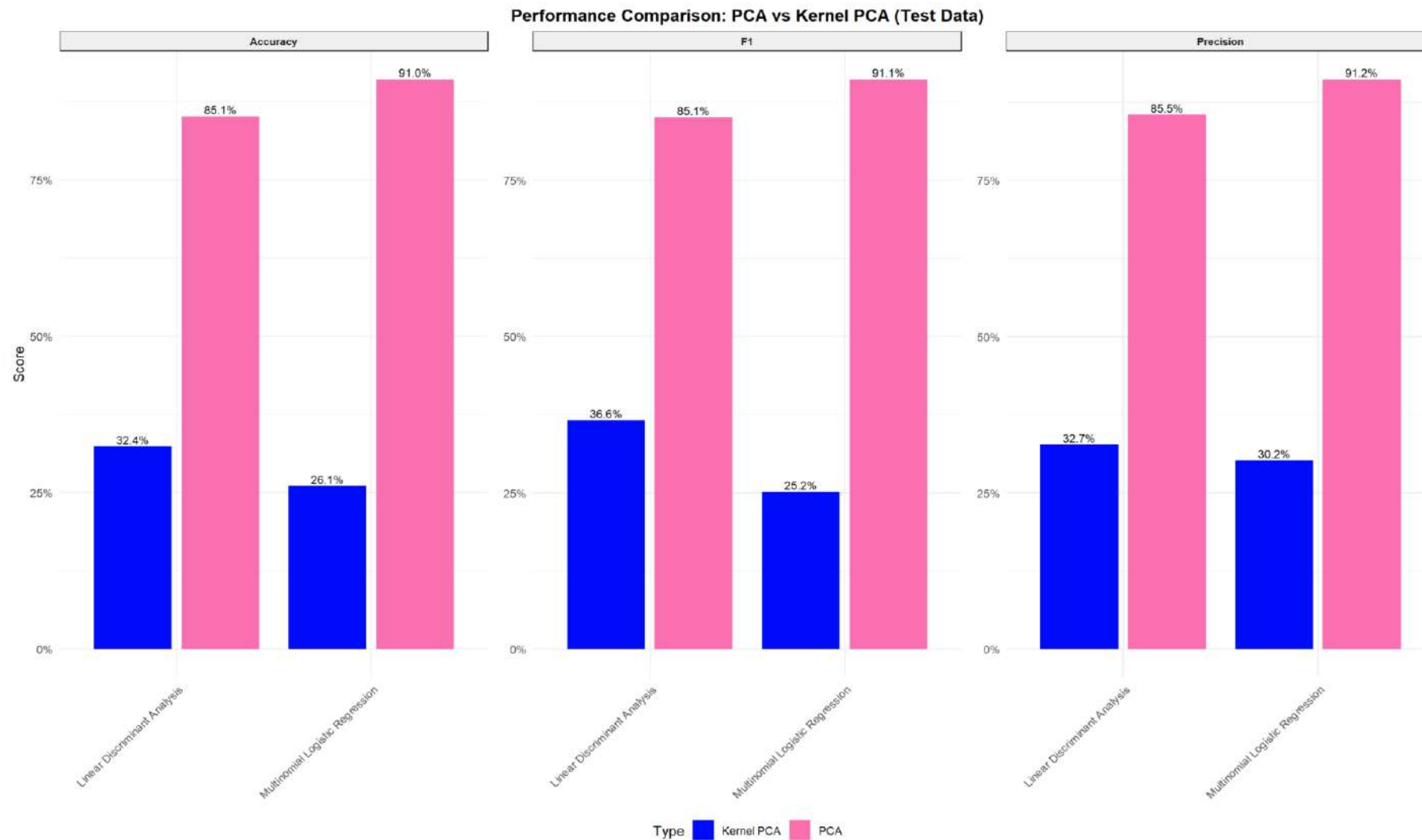
LDA with Kernel PCA (Training)



# Results of LDA with Kernel PCA

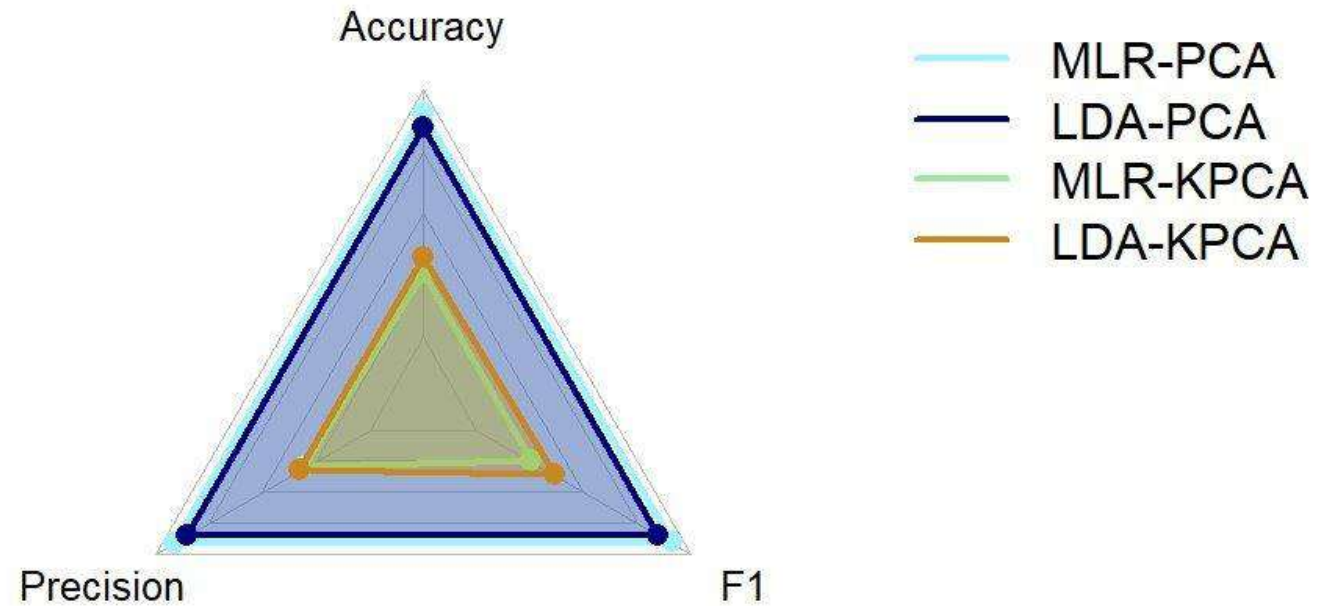
Activity	Test Set Evaluation Metrics			Training Set Evaluation Metrics		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Laying	0.30	0.69	0.42	0.21	0.80	0.33
Sitting	NA	0	NA	1	0	0
Standing	0.35	0.15	0.21	0.37	0.21	0.27
Walking	0.27	0.34	0.30	0.24	0.18	0.21
Walking Downstairs	0.37	0.76	0.50	0.45	0.15	0.22
Walking Upstairs	NA	0	NA	NA	0	NA
Overall Accuracy	32 %			24.89 %		

# So, which is better?



# Comparative performance of the four models across three key metrics

## Model Performance Comparison using a Radar Chart



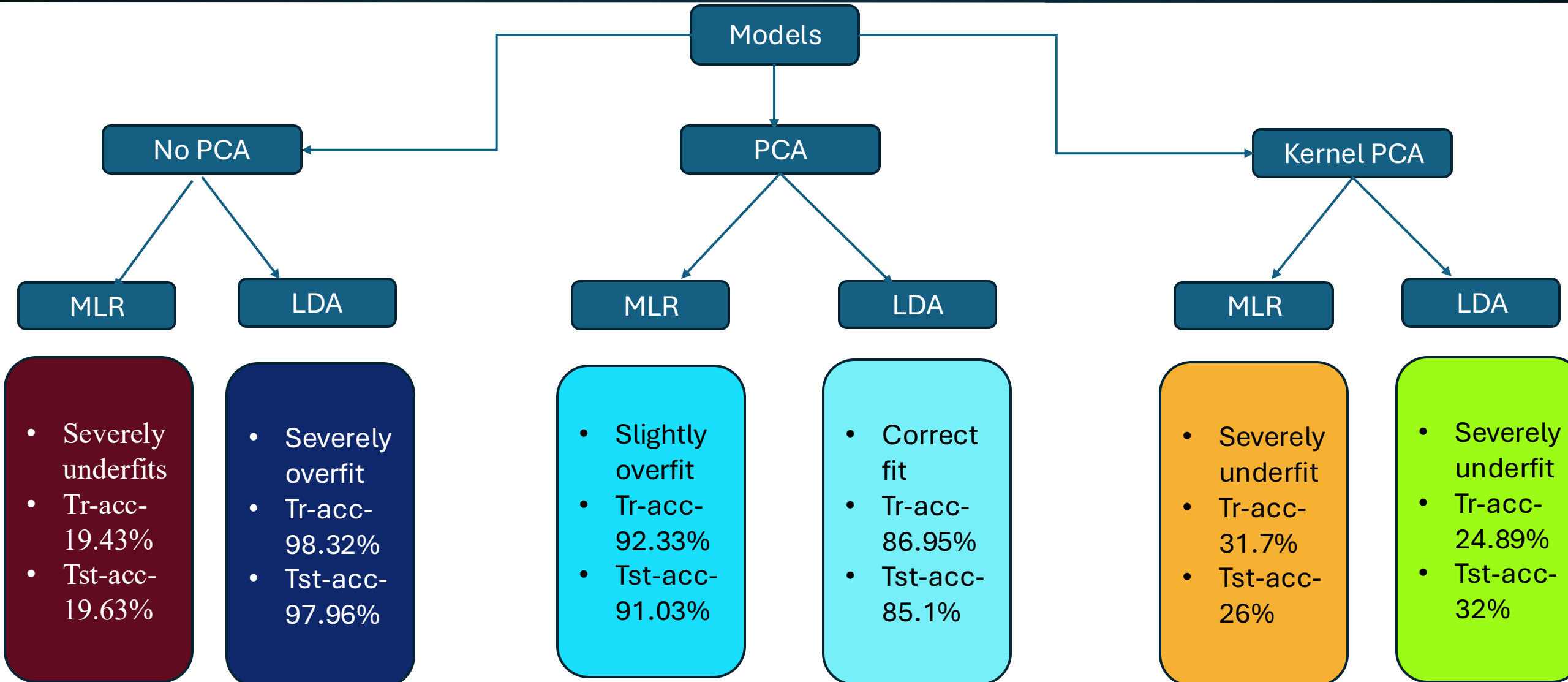




Based on the presentation results,  
which classification approach  
performed best on the test data  
according to you?

- A) MLR without dimensionality reduction
- B) LDA without dimensionality reduction
- C) MLR with PCA
- D) LDA with PCA
- E) MLR with Kernel PCA
- F) LDA with Kernel PCA

# Conclusion





# Best Model

- ✓ **No PCA:** Underfit / Overfit
- ✓ **PCA:** Best performance
- ✓ **PCA + LDA:** Optimal balance
- ✓ Neither underfitting nor overfitting
- ✓ High test accuracy
- ✓ Good generalization without overfitting



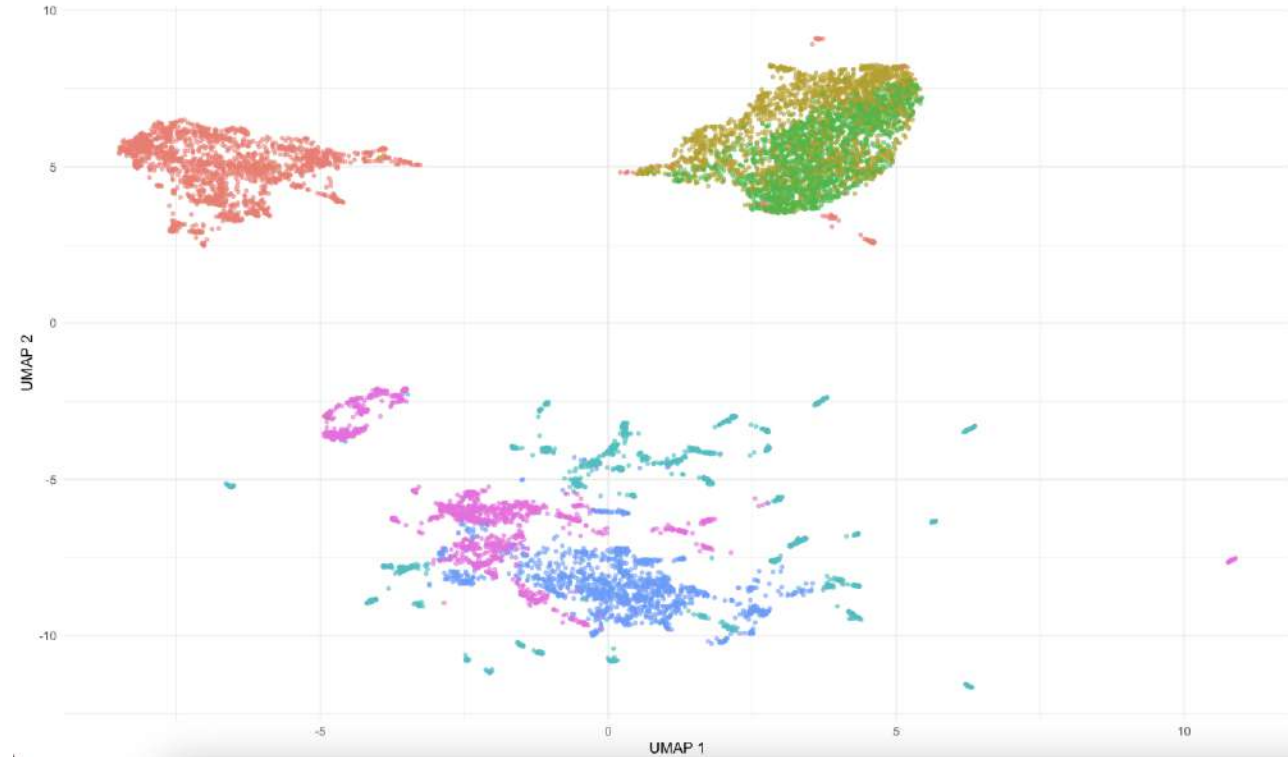
**PCA with LDA**

# UMAP & t-SNE for LDA with PCA

t-SNE Visualization after PCA + LDA



UMAP Visualization after PCA + LDA





# Future Scope

- The current feature representation, even after kernel PCA doesn't provide clean separation. This suggests that more sophisticated kernels could be explored like polynomial kernel, neural network kernel, etc.
- Trying better models like XGBoost to improve accuracy and reduce overfitting.
  - Great for multiclass classification with large feature sets
  - Built-in feature importance and regularization to reduce overfitting
- Include dynamic activities like running or jumping to make the model more robust and applicable in real-world fitness or health tracking.



Which of these future research directions would you find most valuable to explore next with this dataset?

- A) Applying deep learning models
- B) Exploring other dimensionality reduction techniques
- C) Investigating real-time activity classification for mobile devices
- D) Personalizing models for individual users

# References

- License
- <https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones>
- This dataset is licensed under a [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) (CC BY 4.0) license.
- <https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones>
- <https://uc-r.github.io/pca>
- <https://www.datacamp.com/tutorial/pca-analysis-r>
- <https://www.geeksforgeeks.org/curse-of-dimensionality-in-machine-learning>
- [Sponsored Search Acution Design Via Machine Learning \(Kernel PCA\)](#)

*Thank You!*

