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

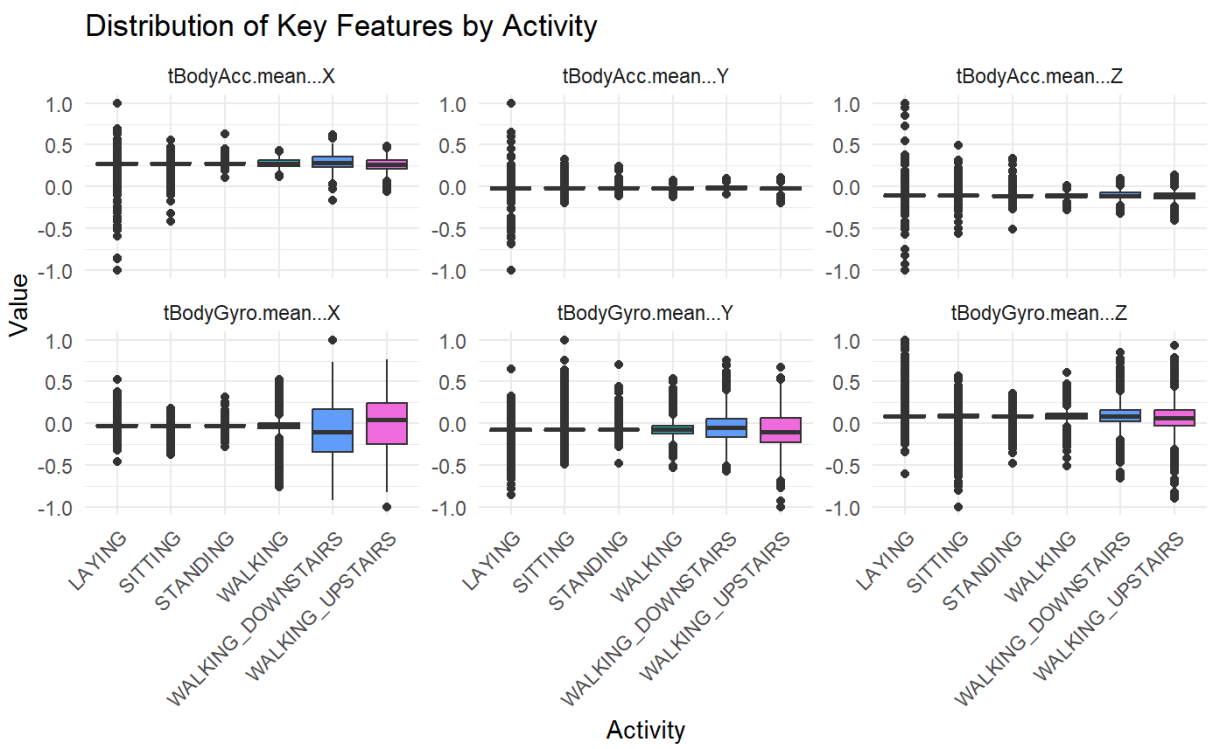
Data Preprocessing

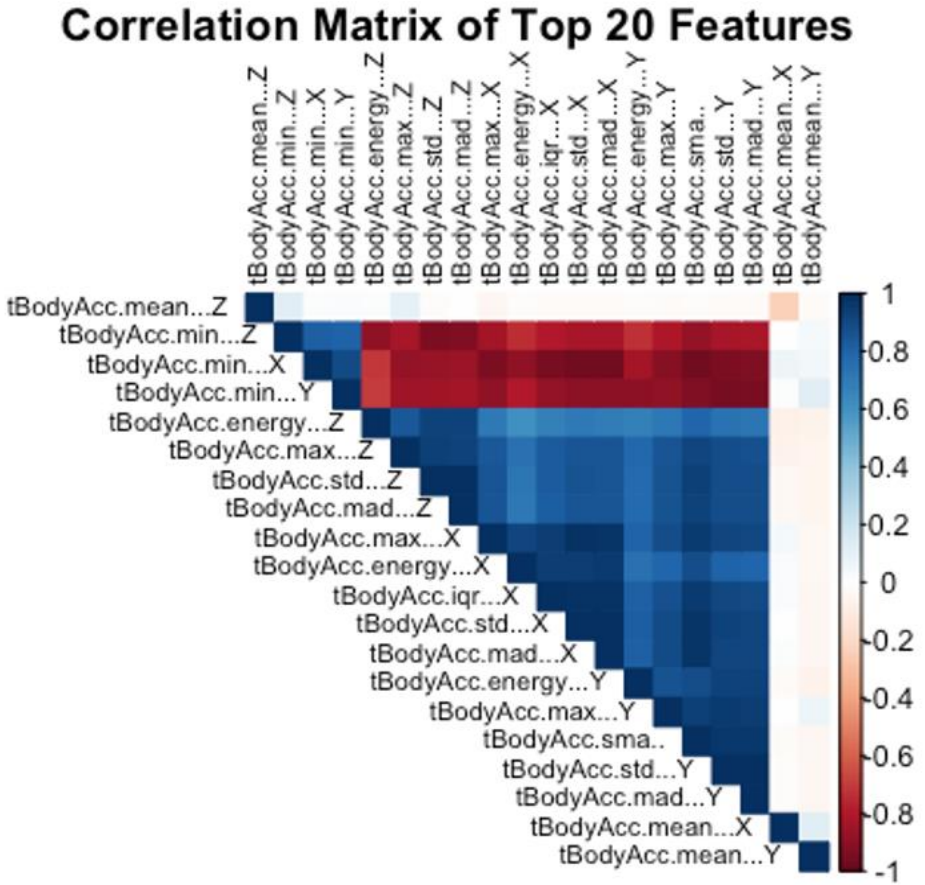
- Loaded feature names and activity labels to assign proper column names
- Combined training data (features, activities, subjects) into a single file
- Did the same for the test data
- Replaced numeric activity codes with descriptive labels
- The dataset was already standardized by its creators. So, all features were scaled and bounded between -1 and 1
- No null values or duplicates
- Features were carefully engineered to have similar scales

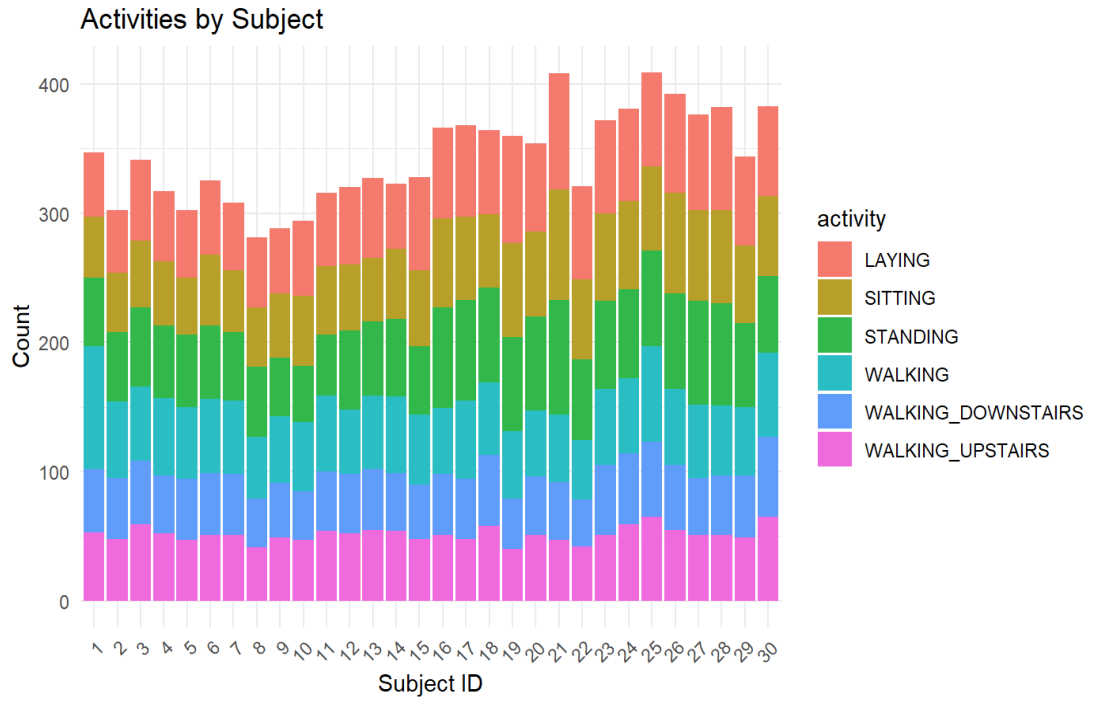
Objective / Goal

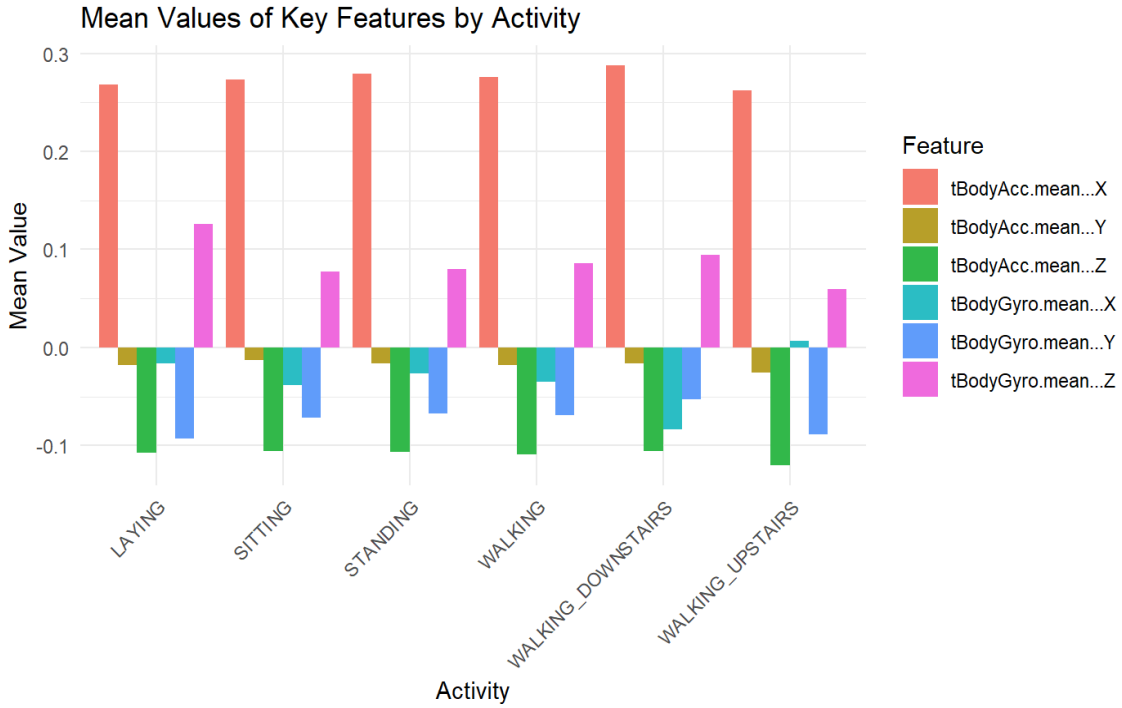
Investigate how dimensionality reduction techniques like PCA and Kernel PCA, combined with classifiers such as multi-logistic regression and linear discriminant analysis, can overcome the curse of dimensionality to accurately predict human activity from high-dimensional smartphone sensor data.

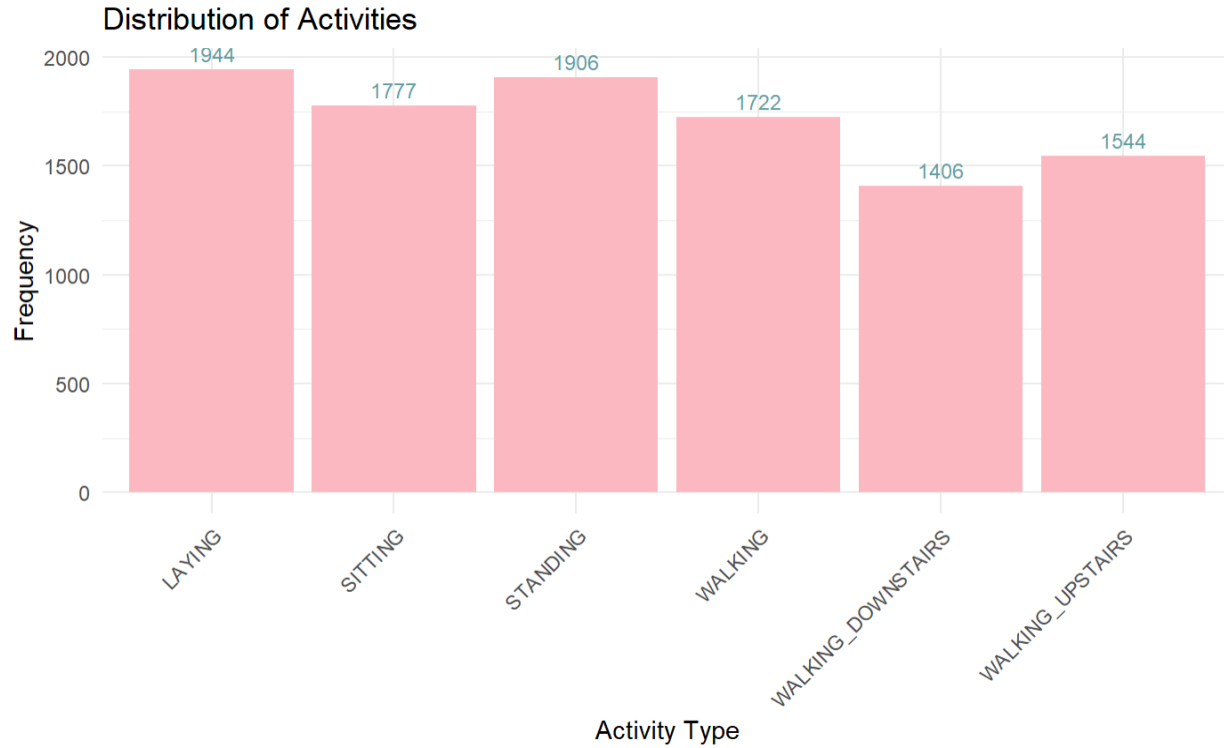
Exploratory Data Analysis



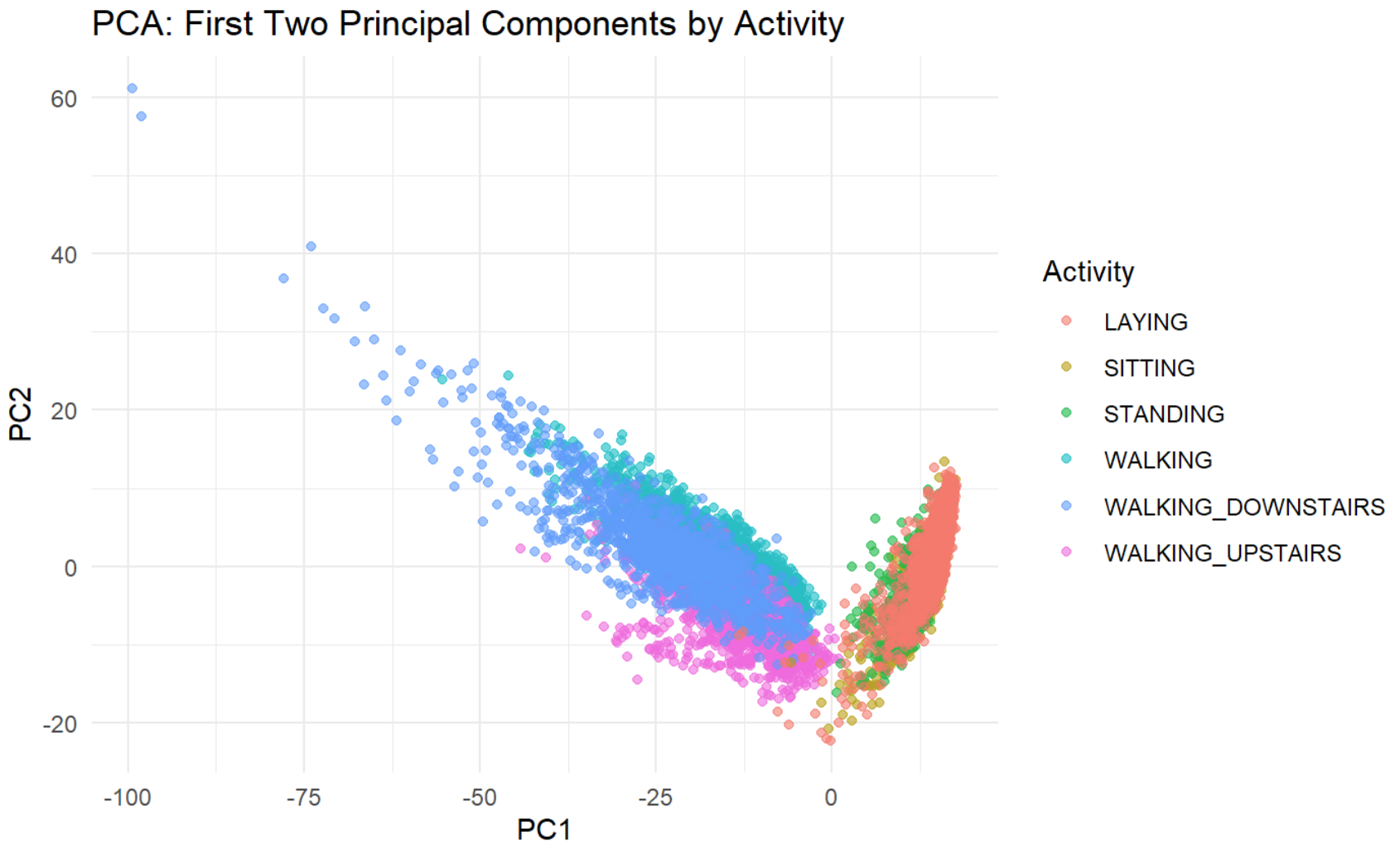



Activities by Subject
Count
Subject ID
activity
LAYING
SITTING
STANDING
WALKING
WALKING_DOWNSTAIRS
WALKING_UPSTAIRS

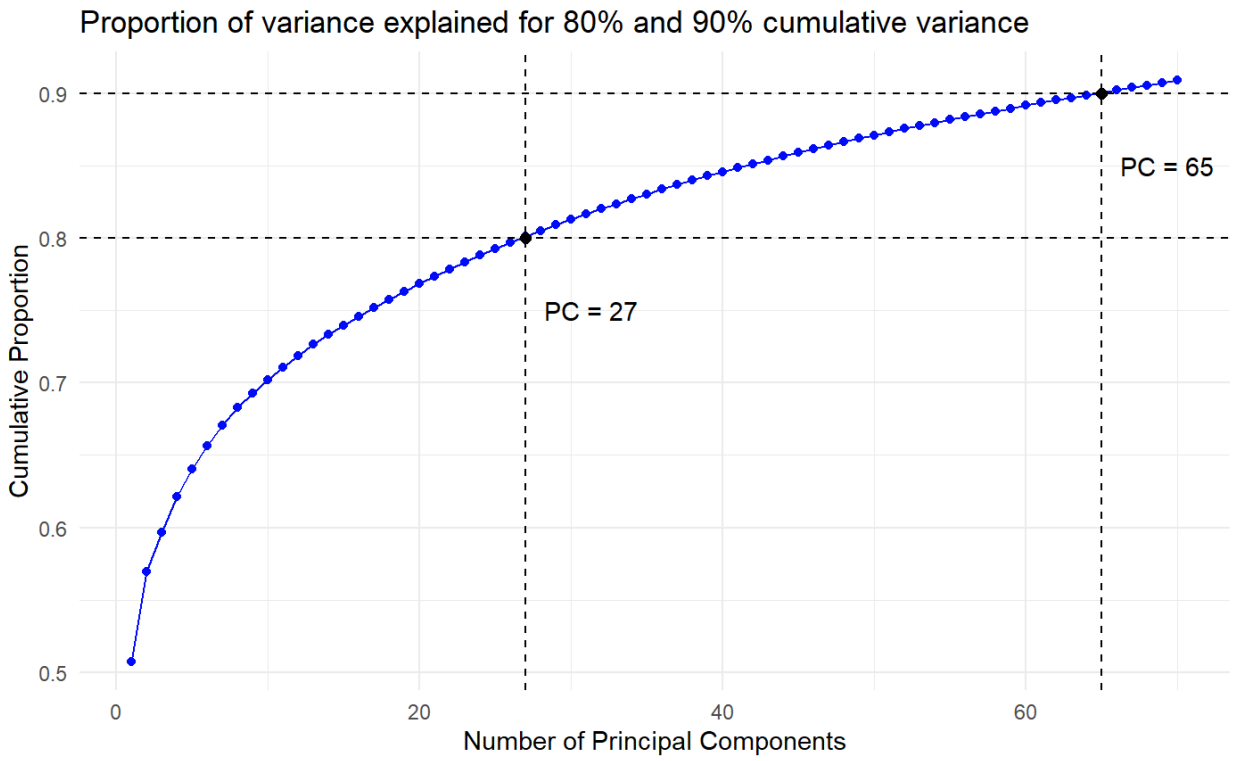

Mean Values of Key Features by Activity
Mean Value
Activity
Feature
tBodyAcc.mean...X
tBodyAcc.mean...Y
tBodyAcc.mean...Z
tBodyGyro.mean...X
tBodyGyro.mean...Y
tBodyGyro.mean...Z
LAYING
SITTING
STANDING
WALKING
WALKING_DOWNSTAIRS
WALKING_UPSTAIRS

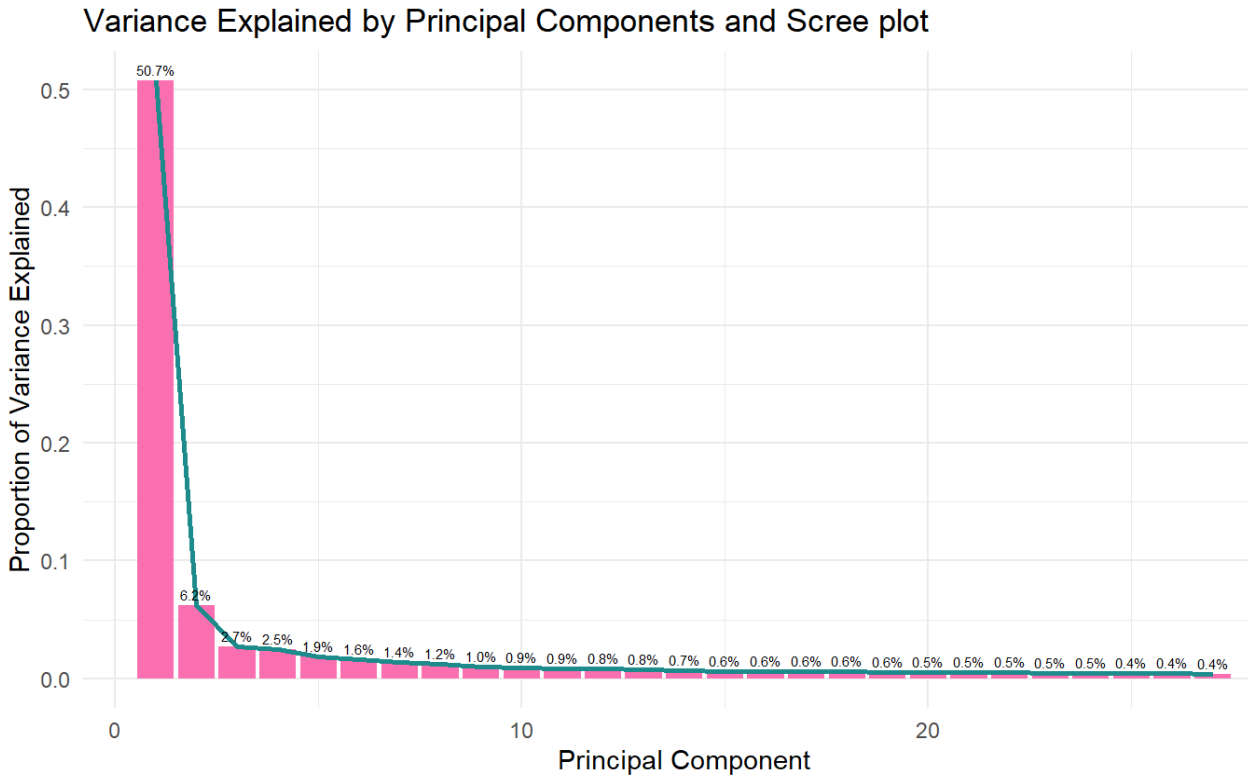

Distribution of Activities
Frequency
Activity Type
1944
1777
1906
1722
1406
1544
LAYING
SITTING
STANDING
WALKING
WALKING_DOWNSTAIRS
WALKING_UPSTAIRS

EDA WITH PCA

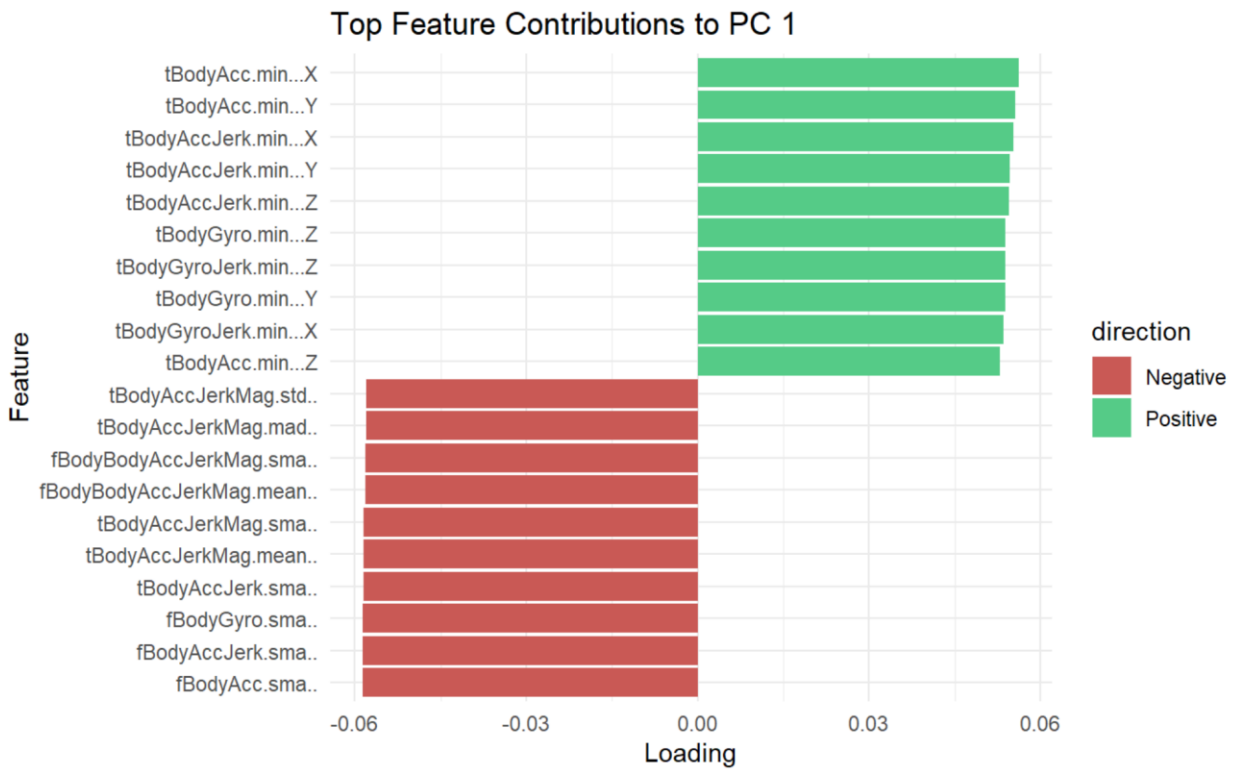

PCA: First Two Principal Components by Activity
PC2
PC1
60
40
20
0
-20
-100
-75
-50
-25
0
Activity
LAYING
SITTING
STANDING
WALKING
WALKING_DOWNSTAIRS
WALKING_UPSTAIRS

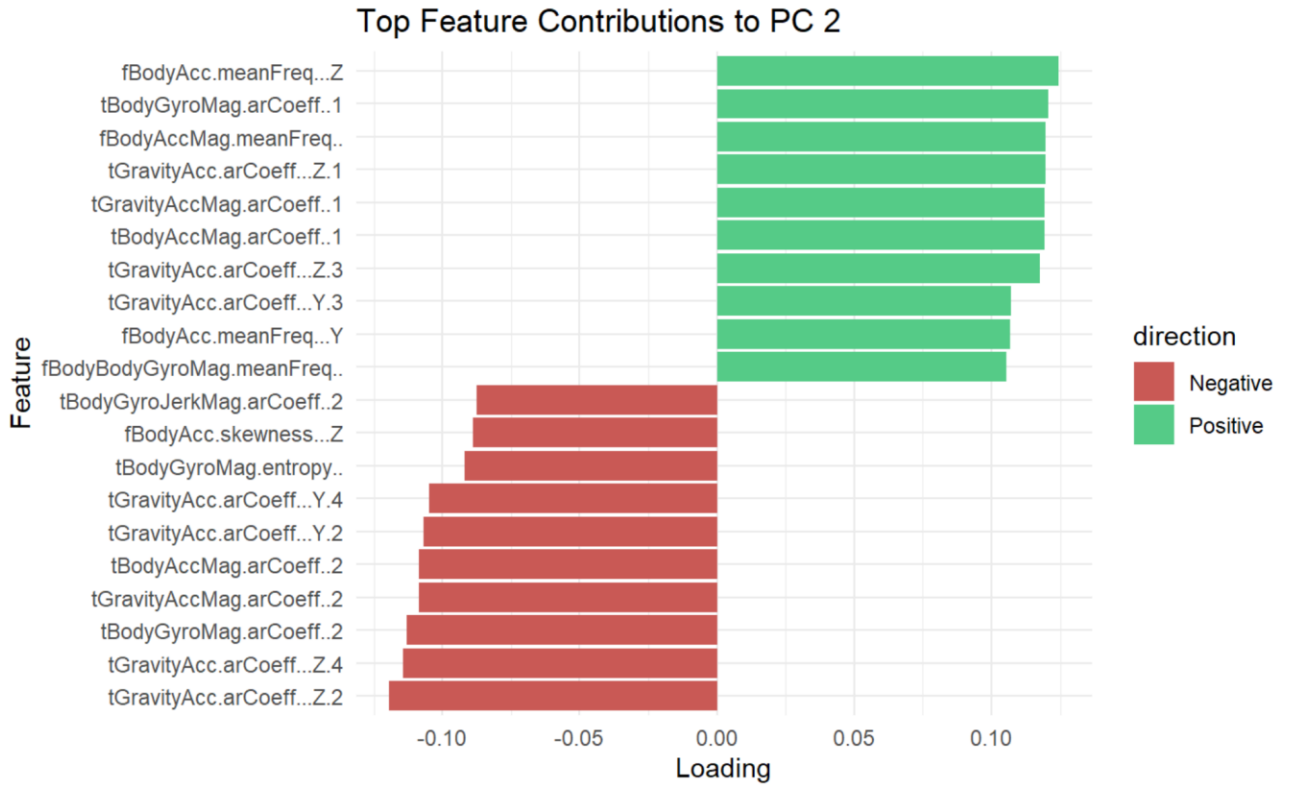
So, how many PCs do we need ?



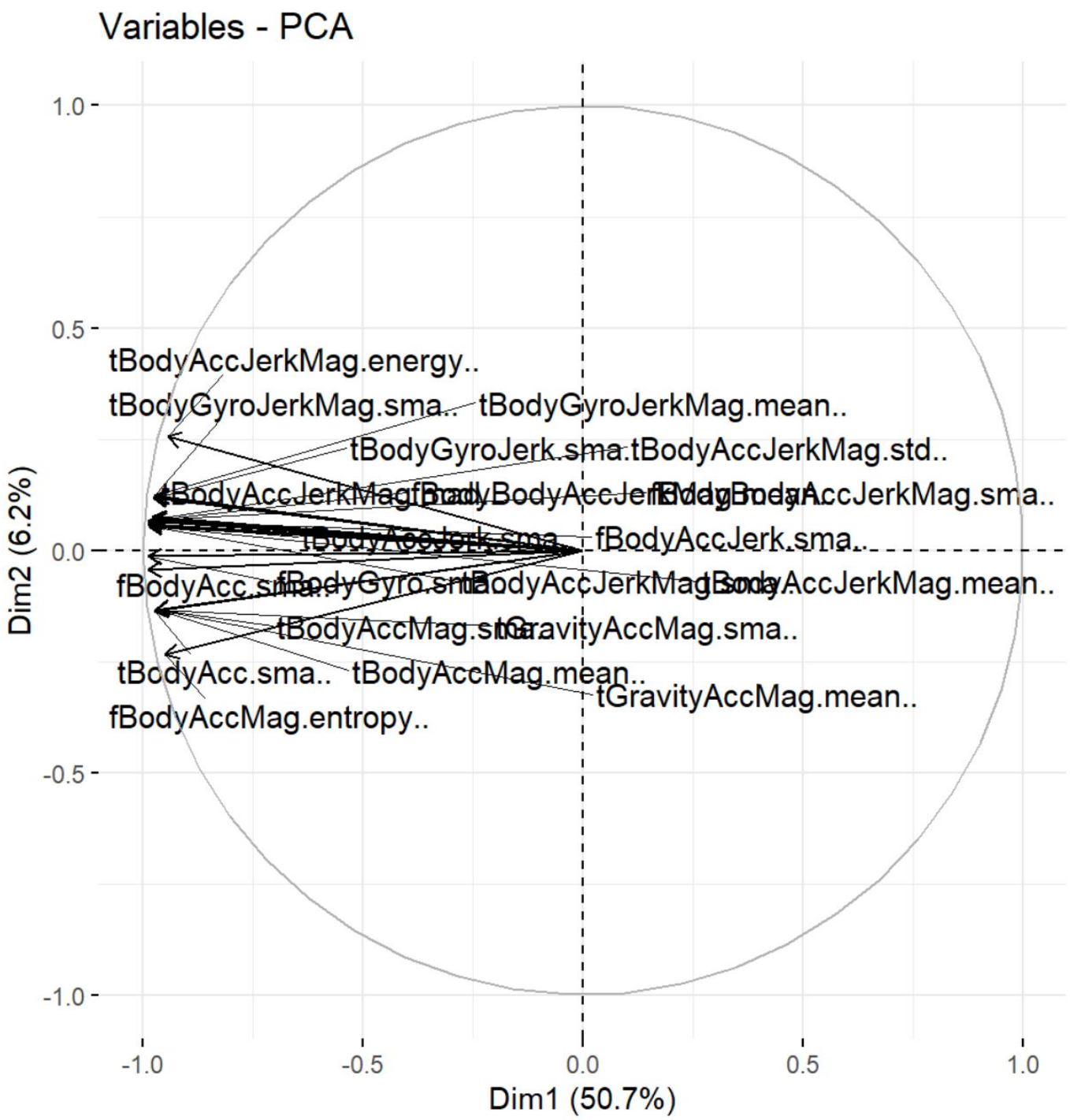


Top feature loadings for the first two PCs

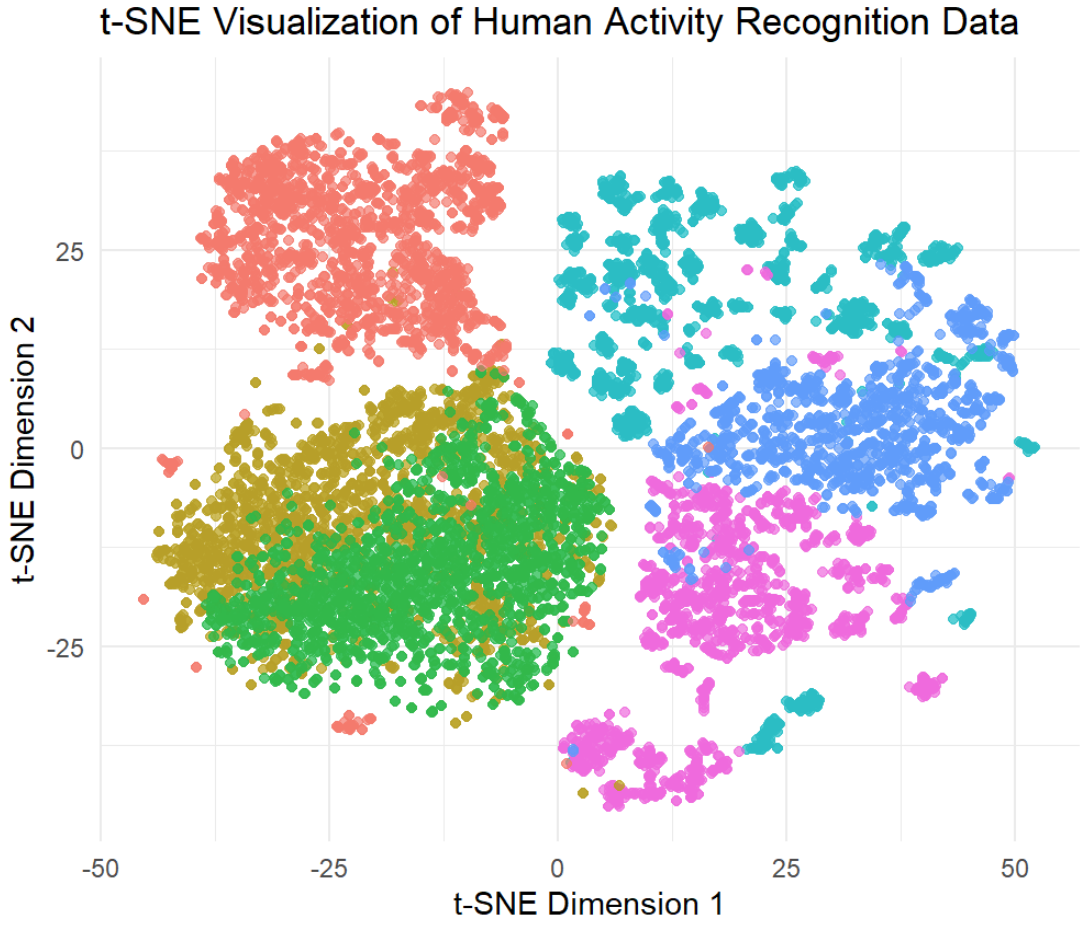


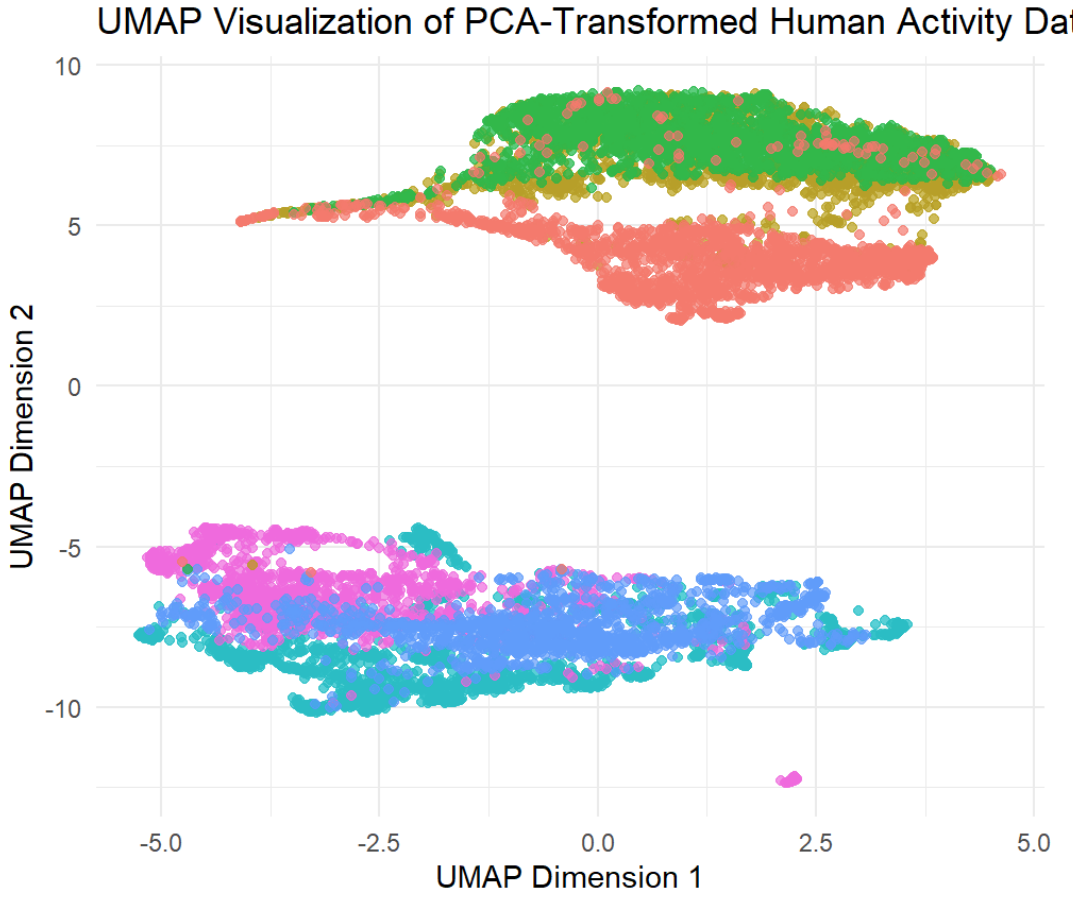


Biplot


Variables - PCA
1.0
0.5
0.0
-0.5
-1.0
Dim2 (6.2%)
tBodyAccJerkMag.energy..
tBodyGyroJerkMag.sma..
tBodyGyroJerkMag.mean..
tBodyGyroJerk.sma..
tBodyAccJerkMag.std..
fBodyAccJerk.sma..
fBodyAcc.sma..
tBodyAccJerkMag.mean..
tBodyAccMag.sma..
tGravityAccMag.sma..
tBodyAcc.sma..
tBodyAccMag.mean..
tGravityAccMag.mean..
fBodyAccMag.entropy..
-1.0
-0.5
0.0
0.5
1.0
Dim1 (50.7%)

UMAP and t-SNE





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

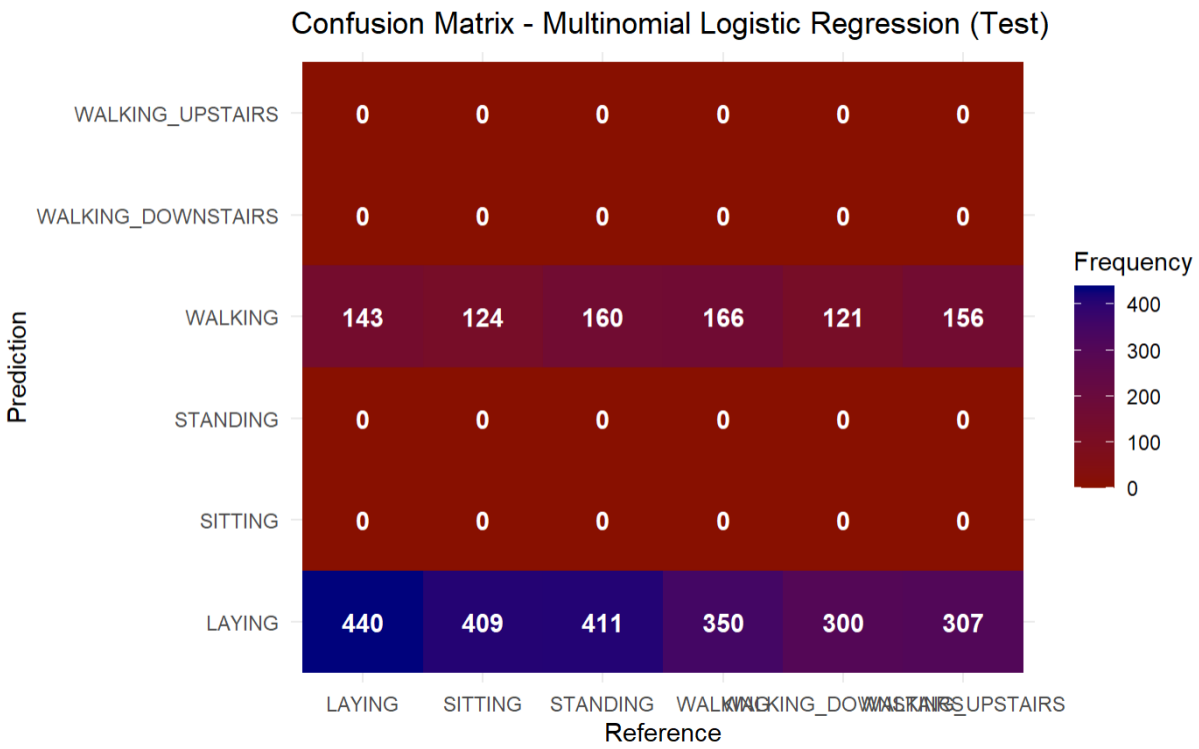
F) Laying

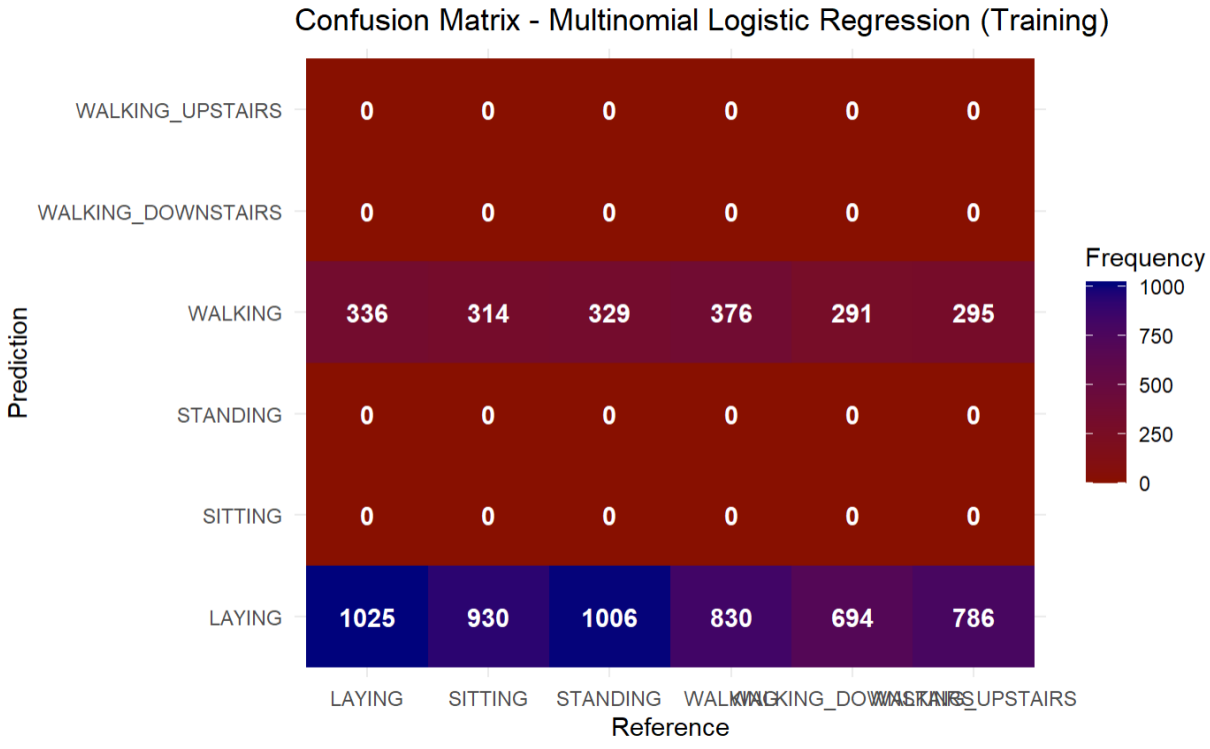
Baseline Classification Model Fitting

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

Confusion Matrices and Performance results

Confusion Matrix for MLR

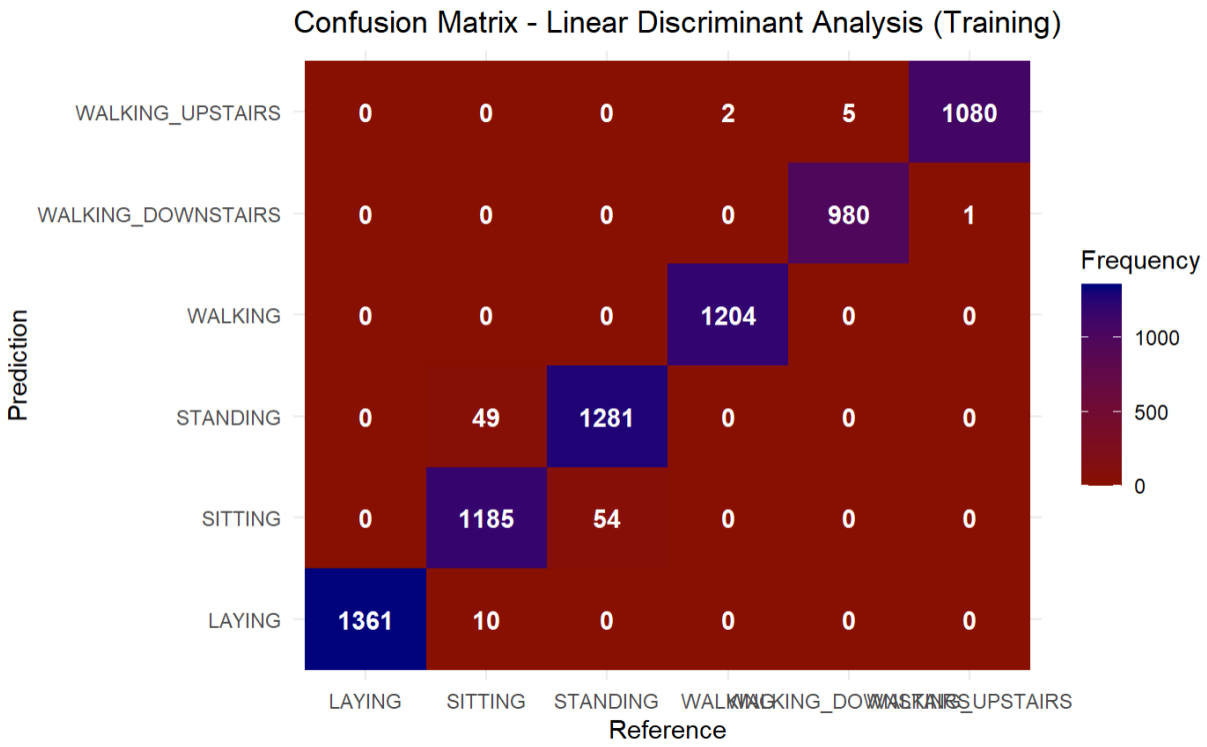


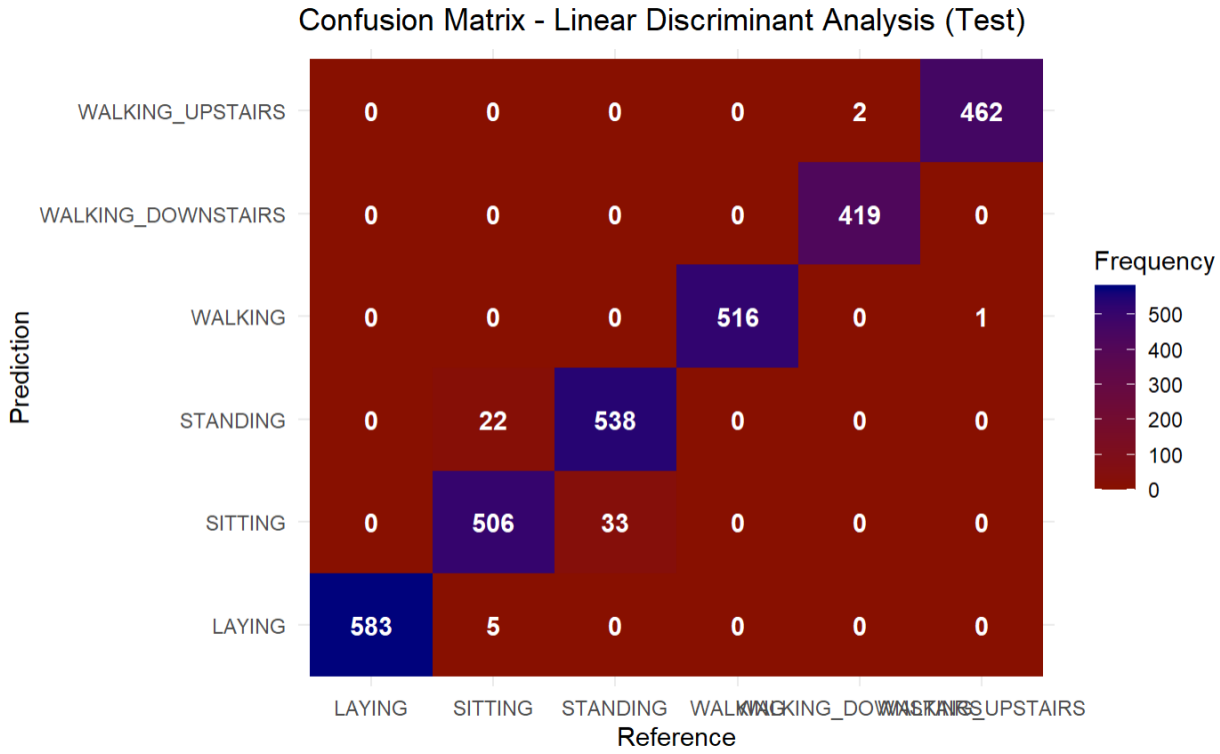


Results of MLR (using a subset of the data)

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





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:

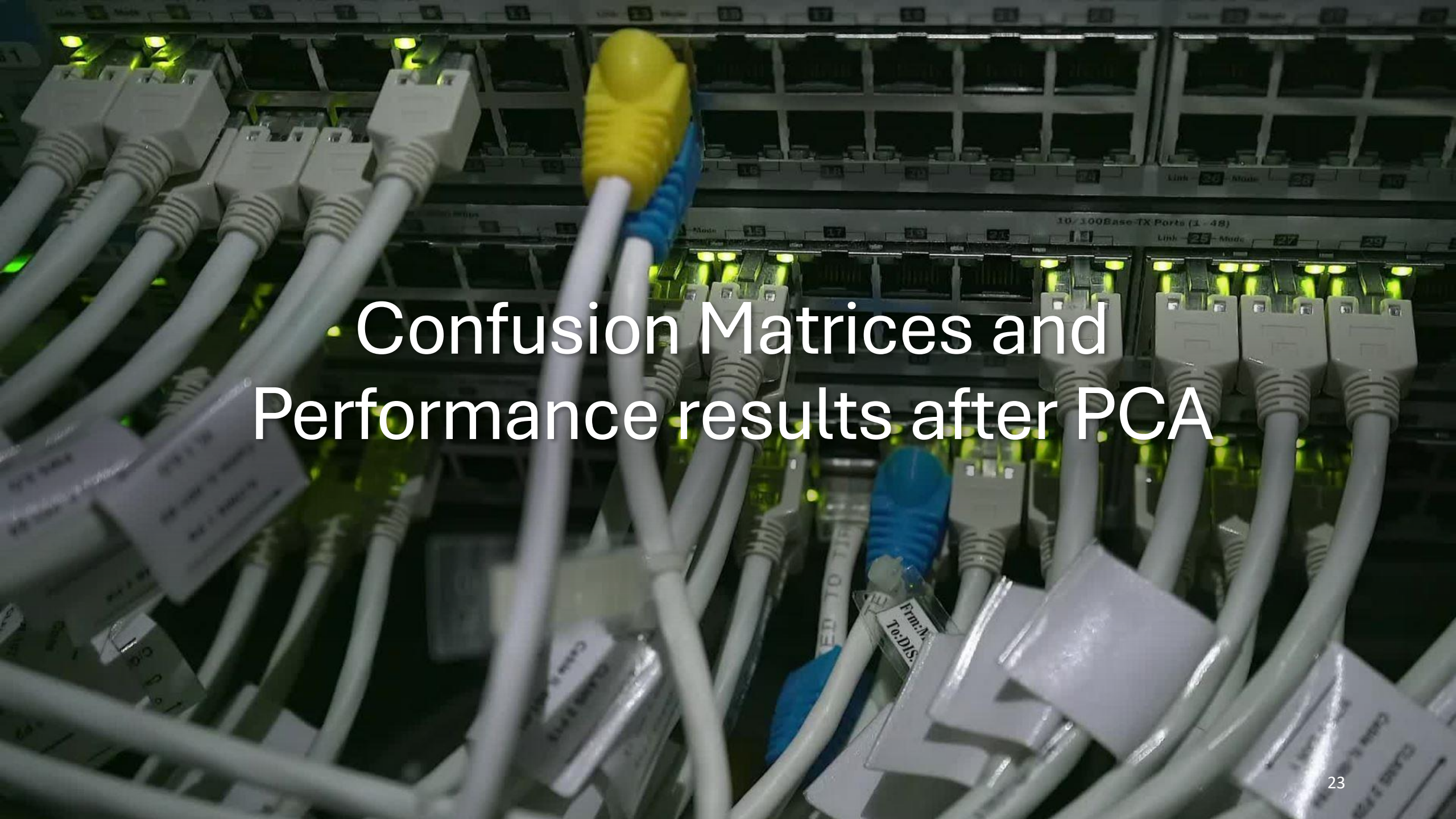
```
Error in nnet.default(X, Y, w, mask = mask, size = 0, skip = TRUE, softmax = TRUE,  :
   too many (3384) weights
```

Linear Discriminant Analysis:

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

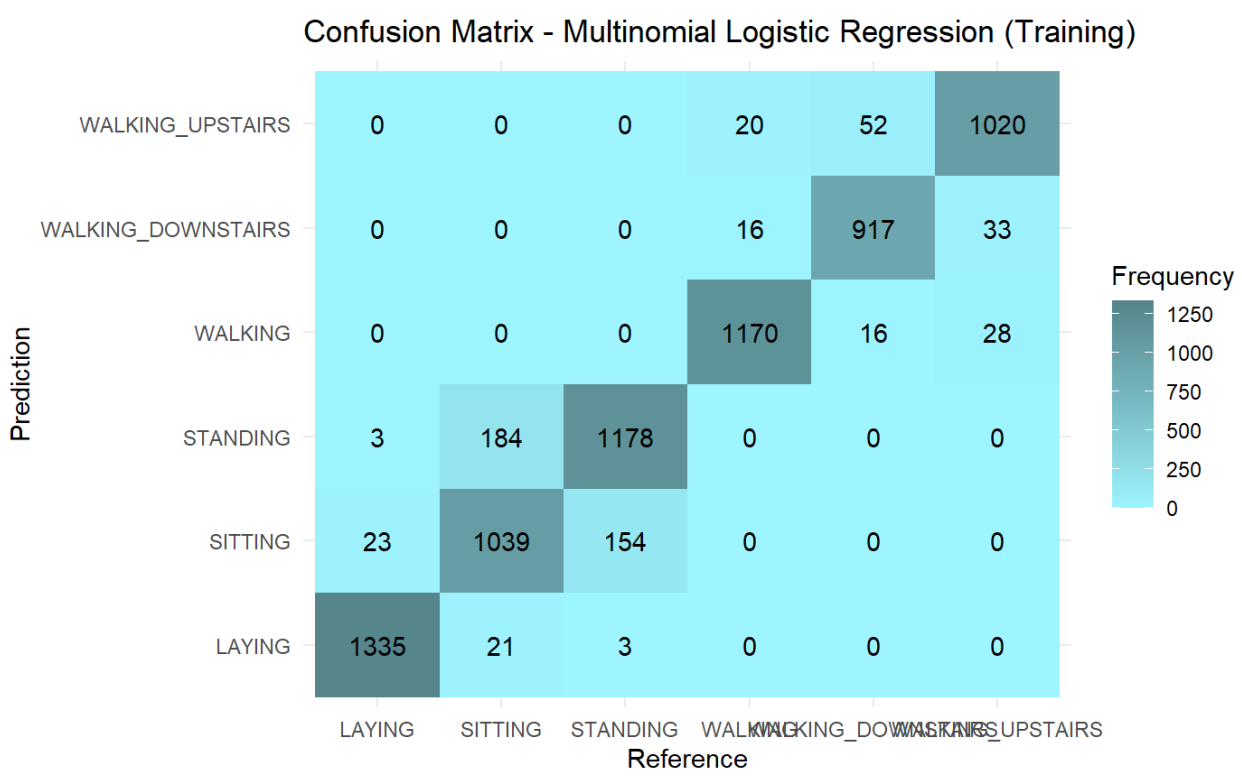
Classification Model Fitting **after** PCA

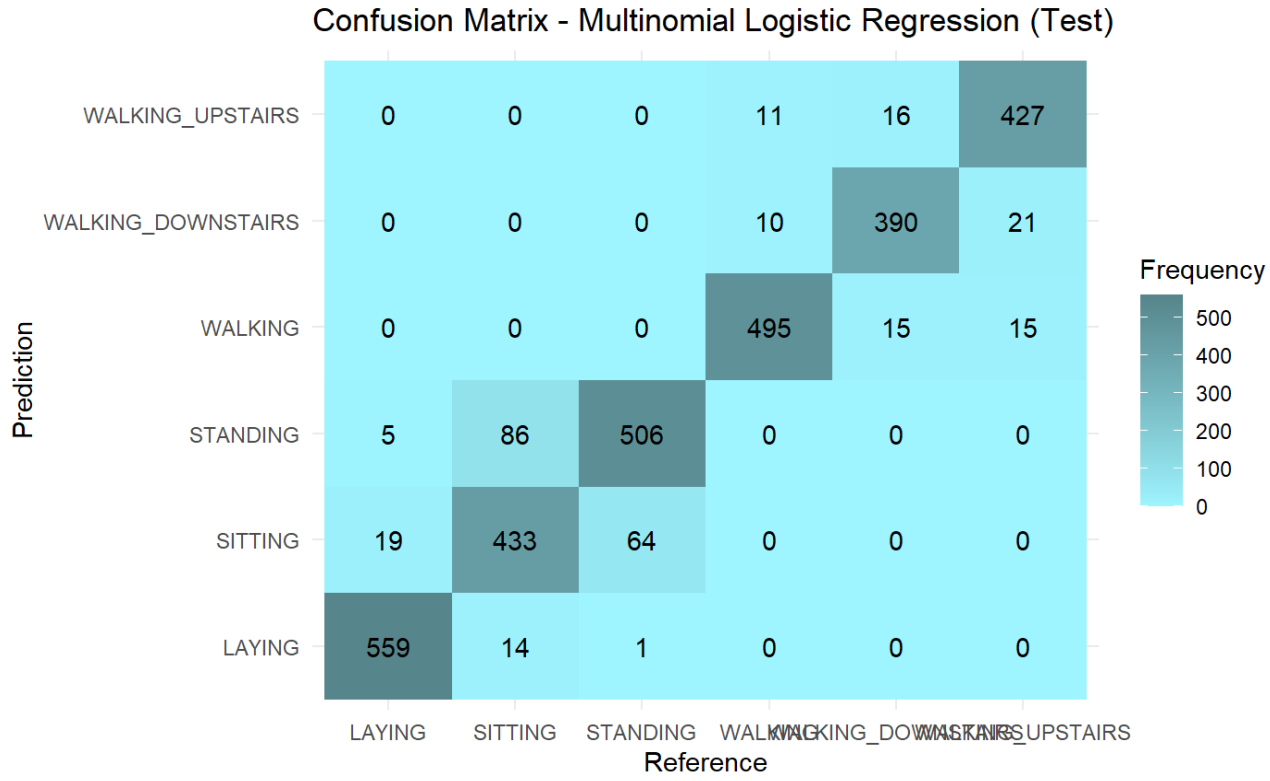
- Multiple logistic regression (MLR)
- Linear Discriminant Analysis (LDA)



Confusion Matrices and Performance results after PCA

Confusion Matrix for MLR with PCA

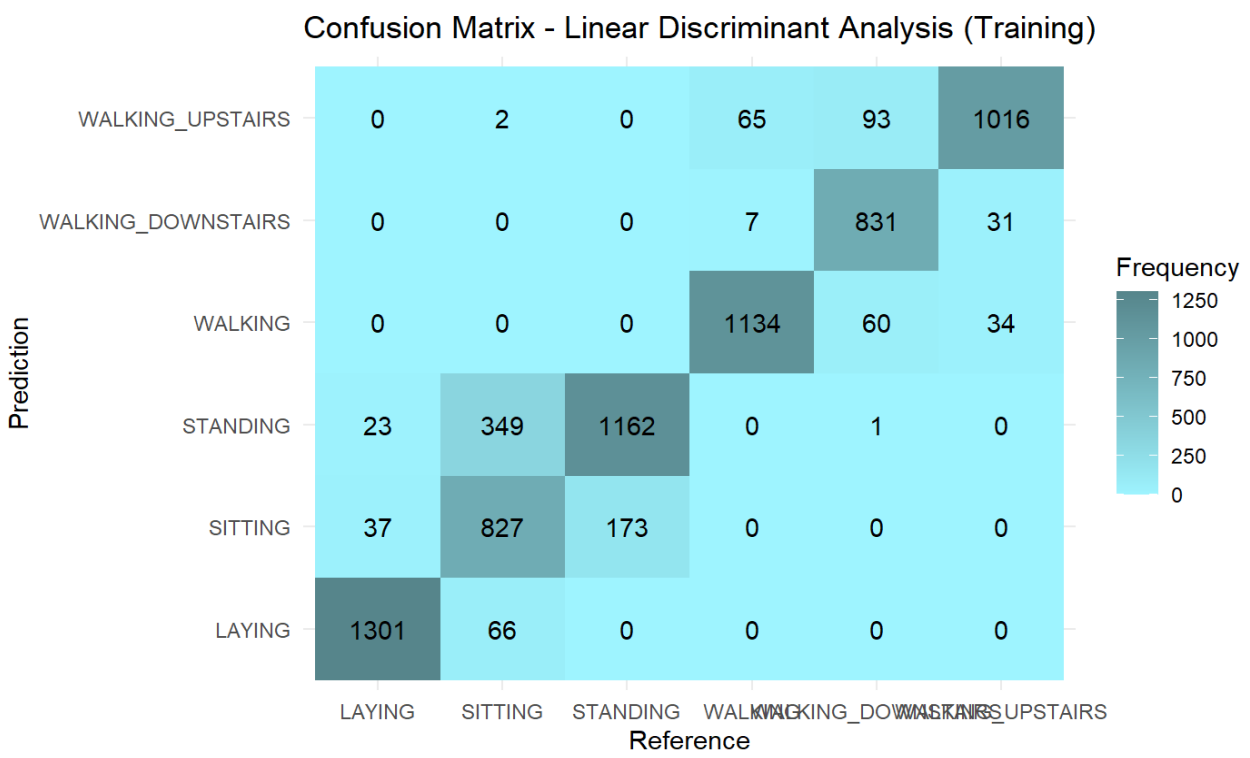


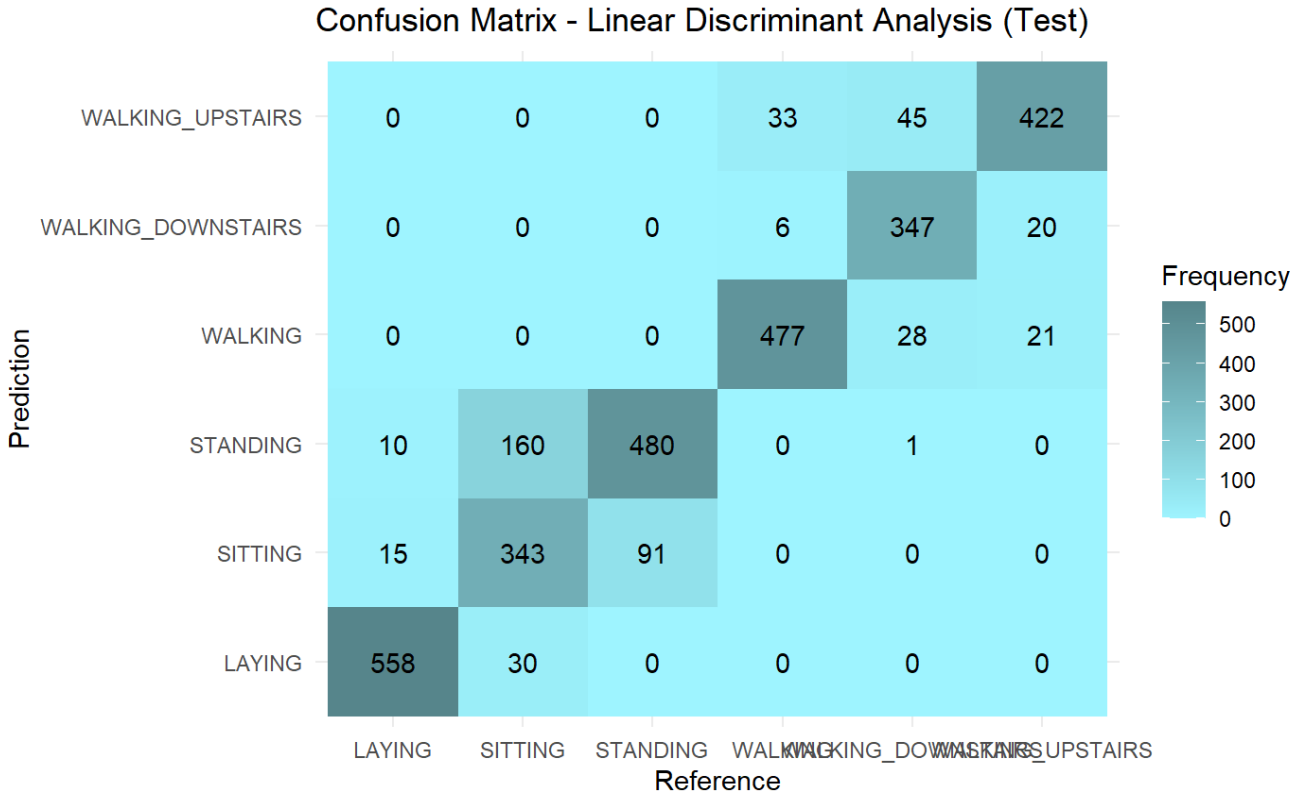


Results of MLR with PCA

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





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

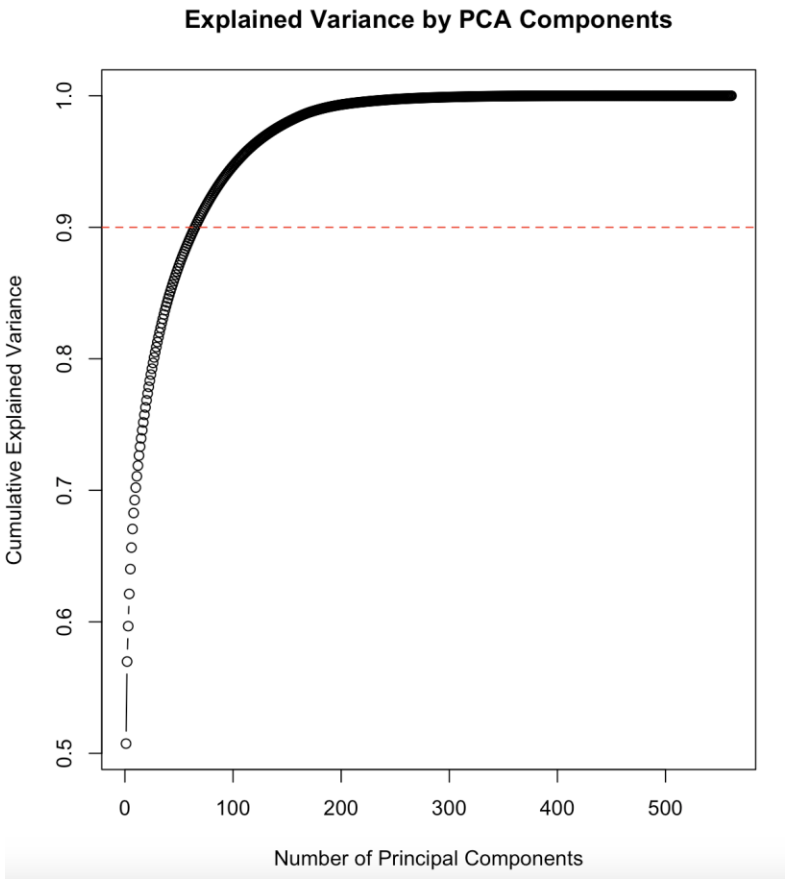
- Exponential increase in data sparsity and computational complexity as the number of features (dimensions) increase.
- Makes models harder to train, generalize, and interpret.

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



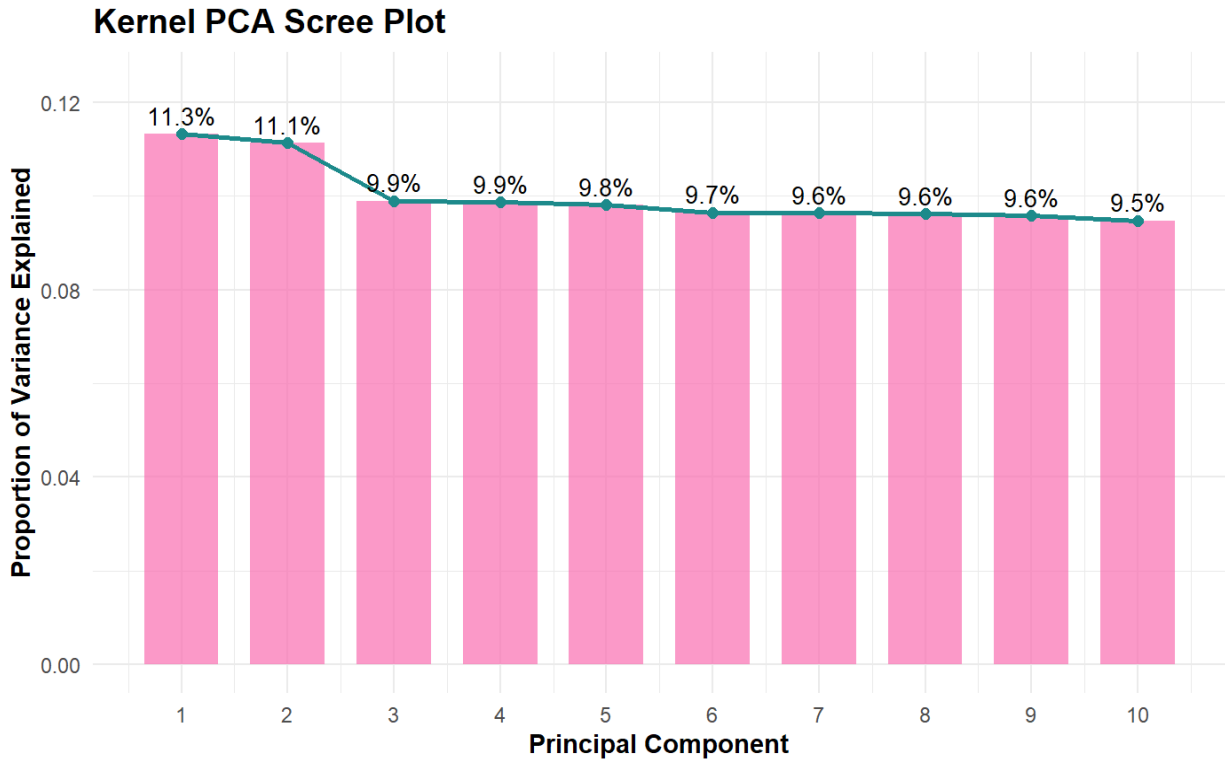
Scree Plot here shows that the first ~50 principal components explain over 90% of the variance. This proves that most features were unnecessary and contributed little unique information.

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

Yes! - Kernel PCA

Why Kernel PCA ?

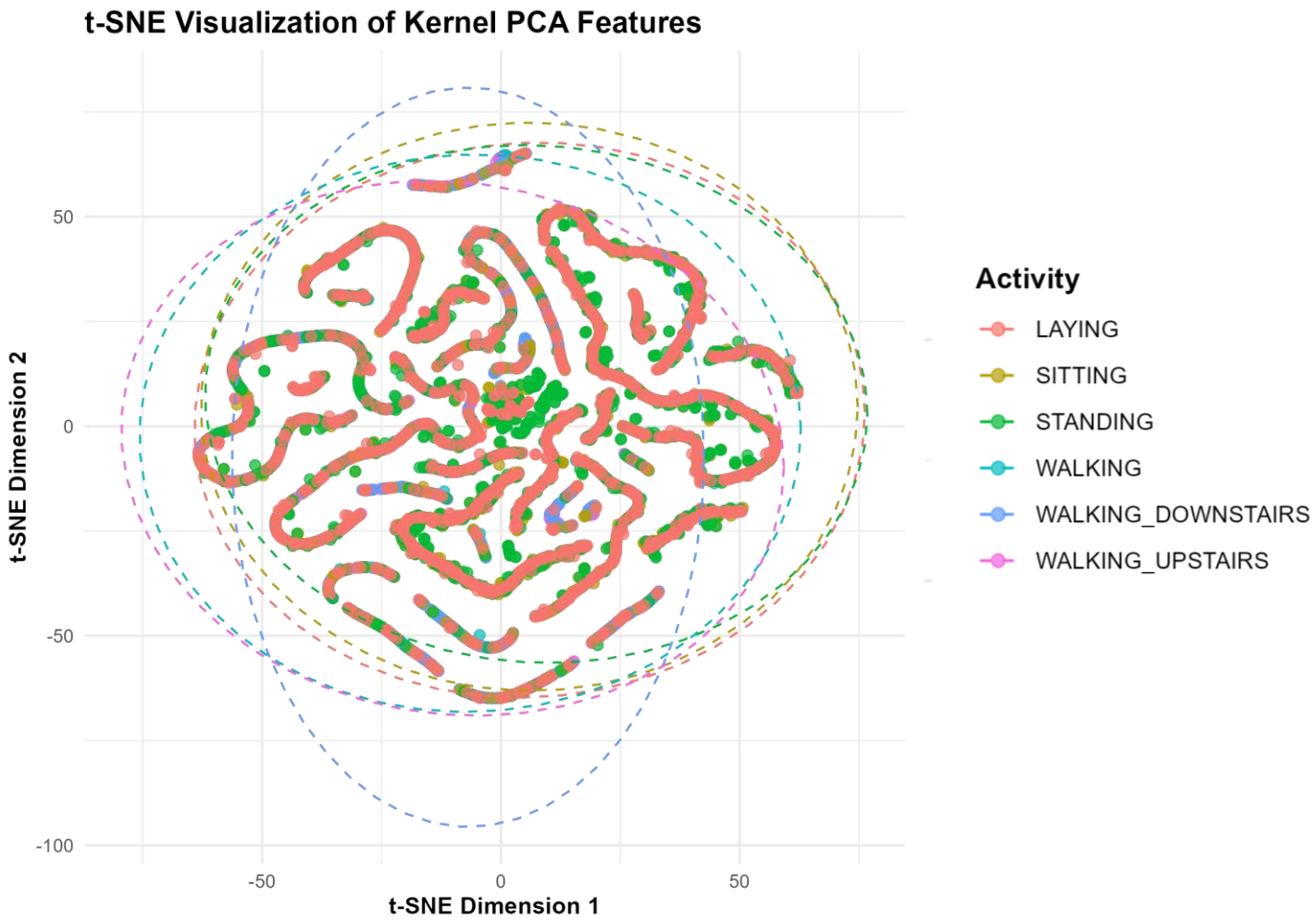
- Primarily captures non – linear relationships.
- Reduces dimension to a manageable number of components while preserving information needed to distinguish between activities – Just like PCA!
- Spreads the variance more evenly compared to PCA.
- About 8 PCs is sufficient to explain 80% of proportion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

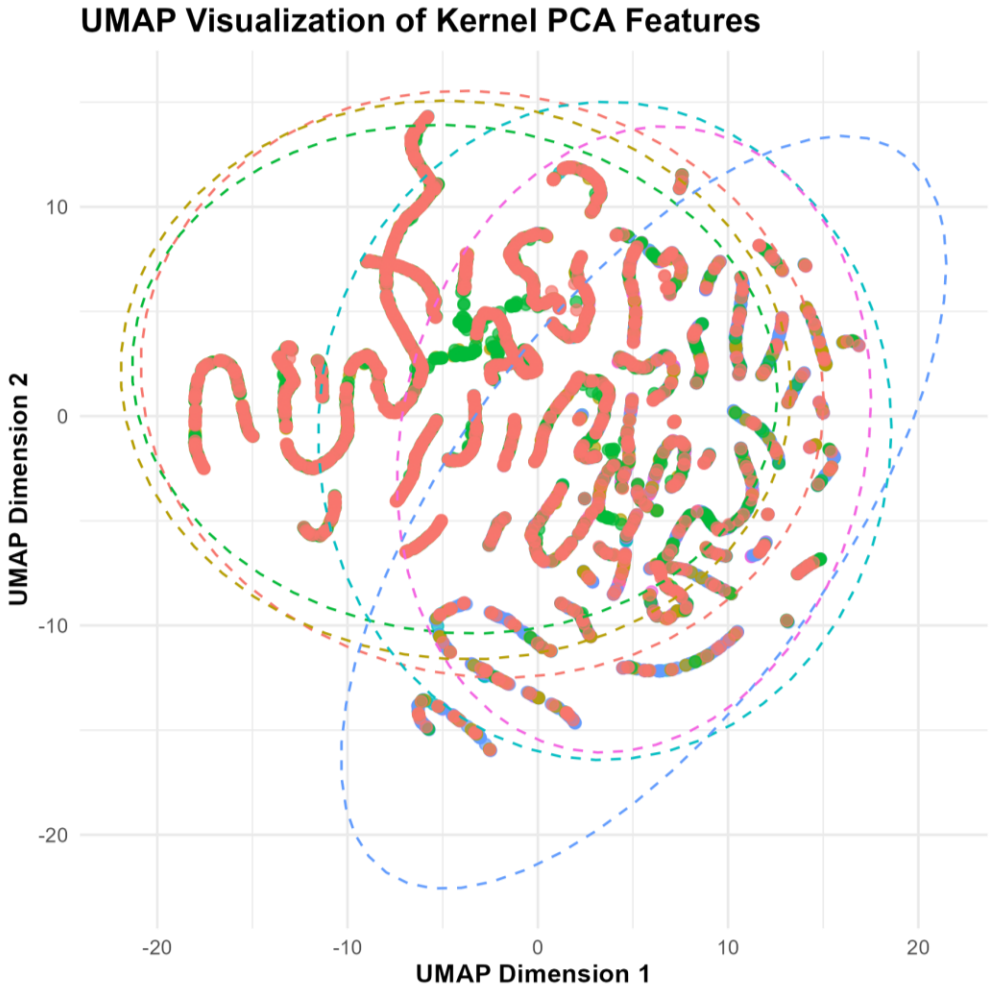


What is Kernel PCA ?

- 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.

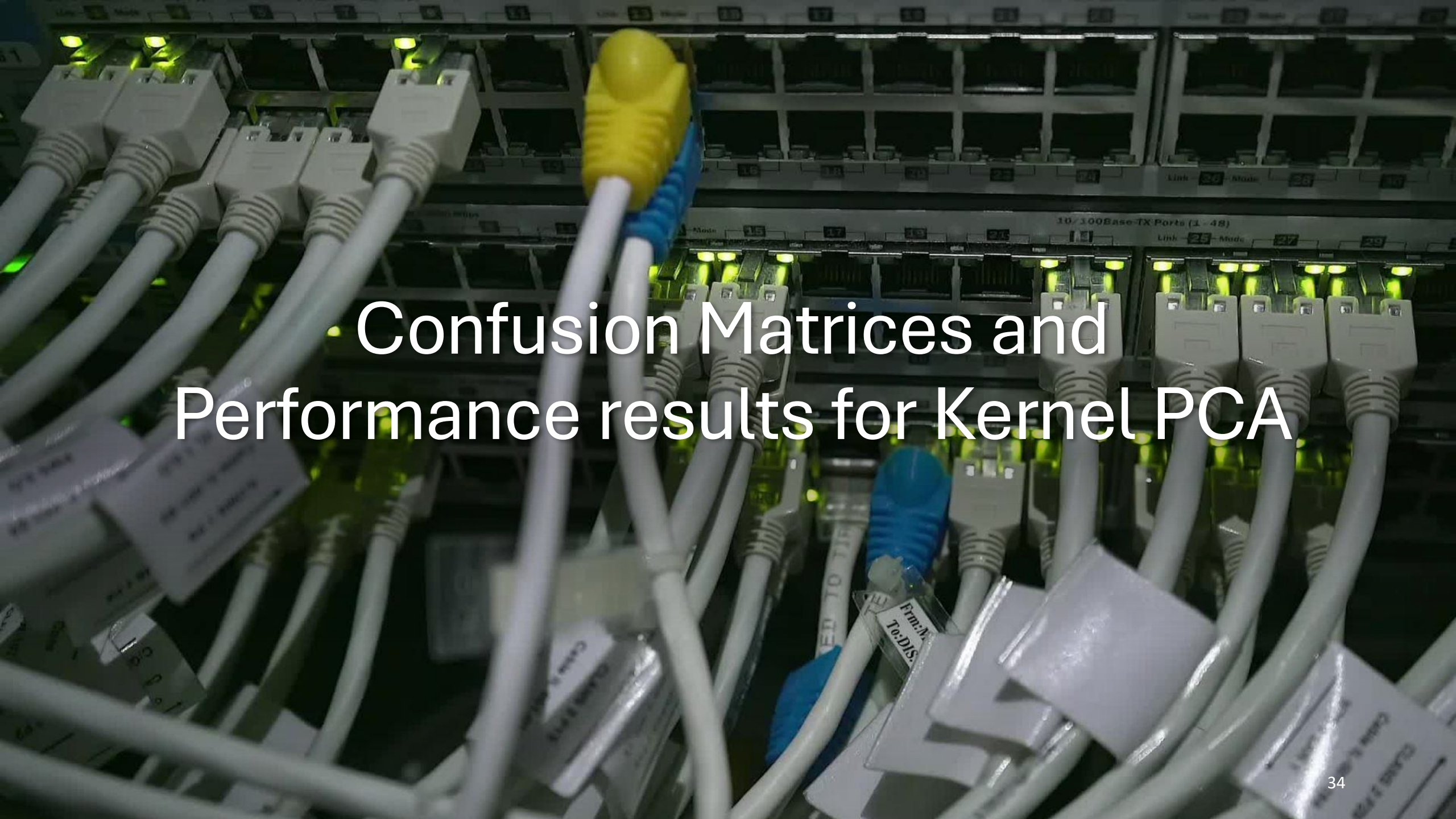
EDA with Kernel PCA





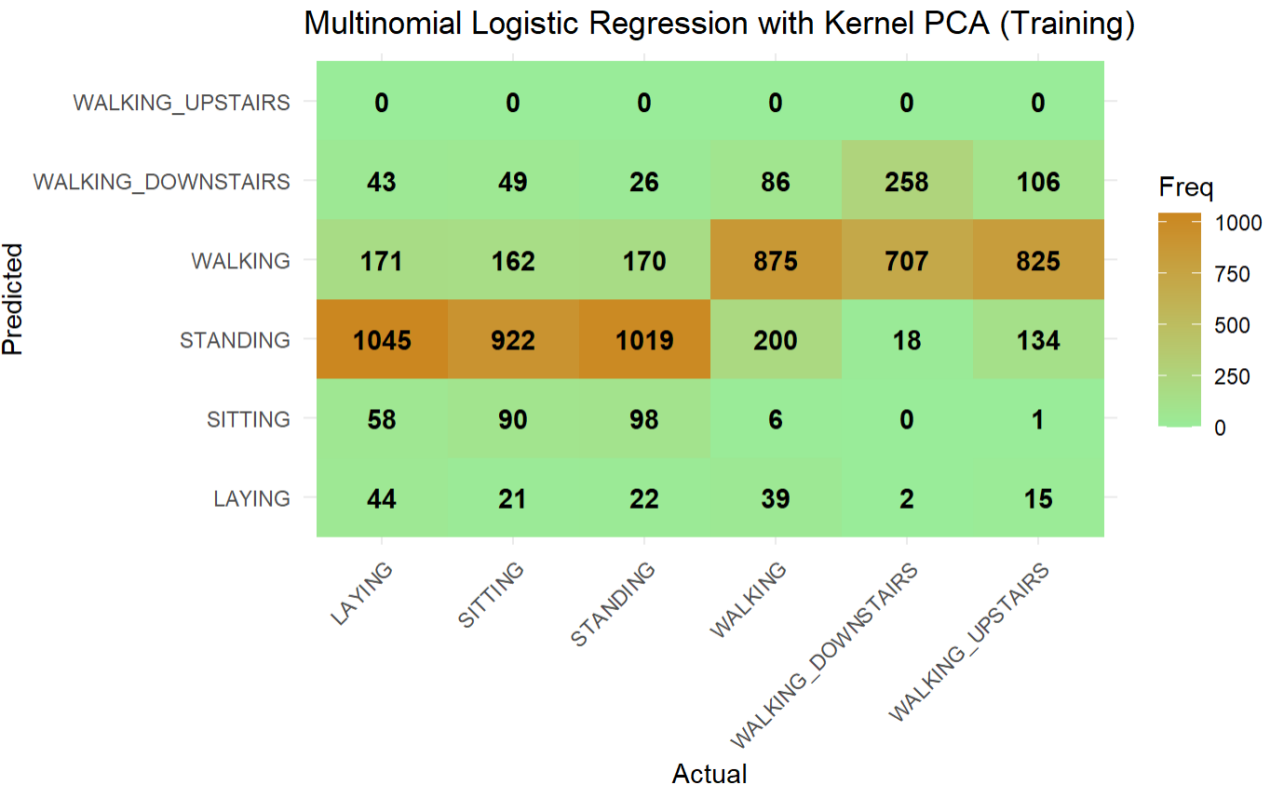
Classification Model Fitting with Kernel PCA

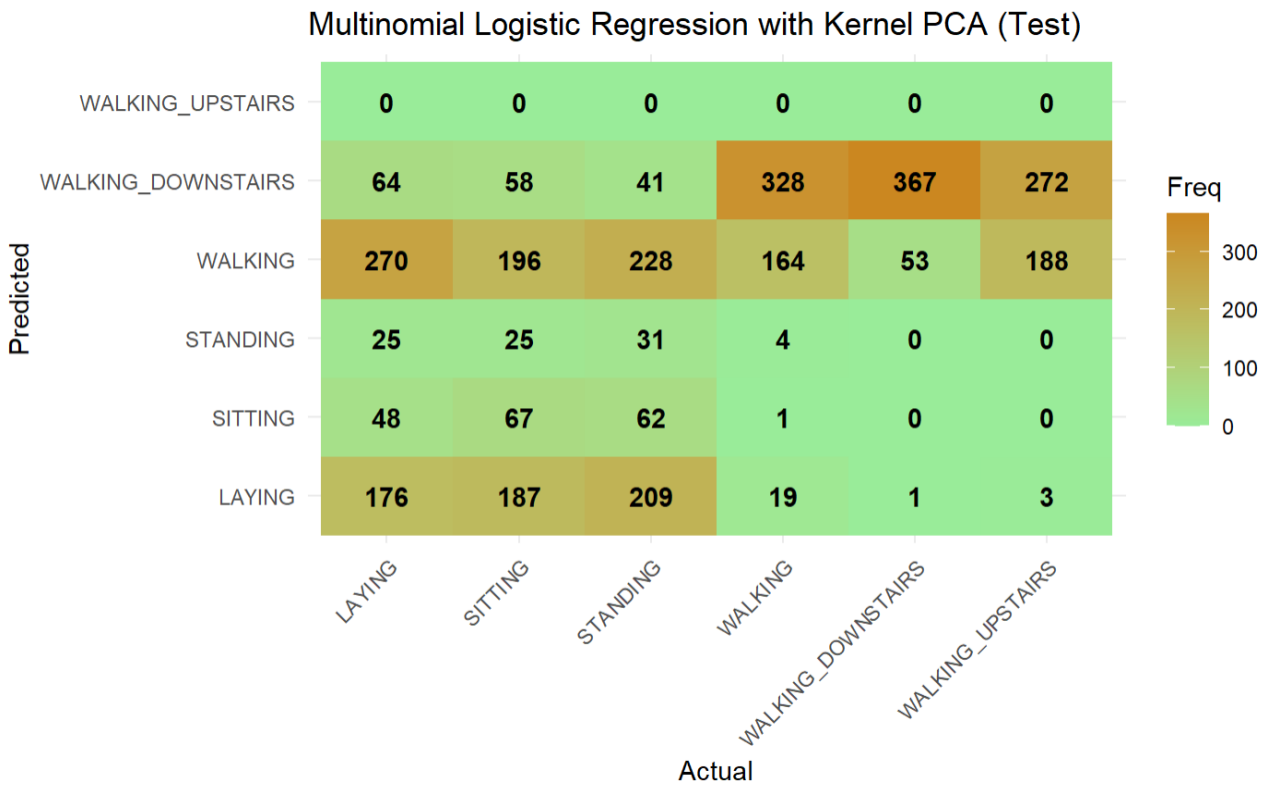
- Multiple logistic regression (MLR)
- Linear Discriminant Analysis (LDA)



Confusion Matrices and Performance results for Kernel PCA

Confusion Matrix for MLR with Kernel PCA

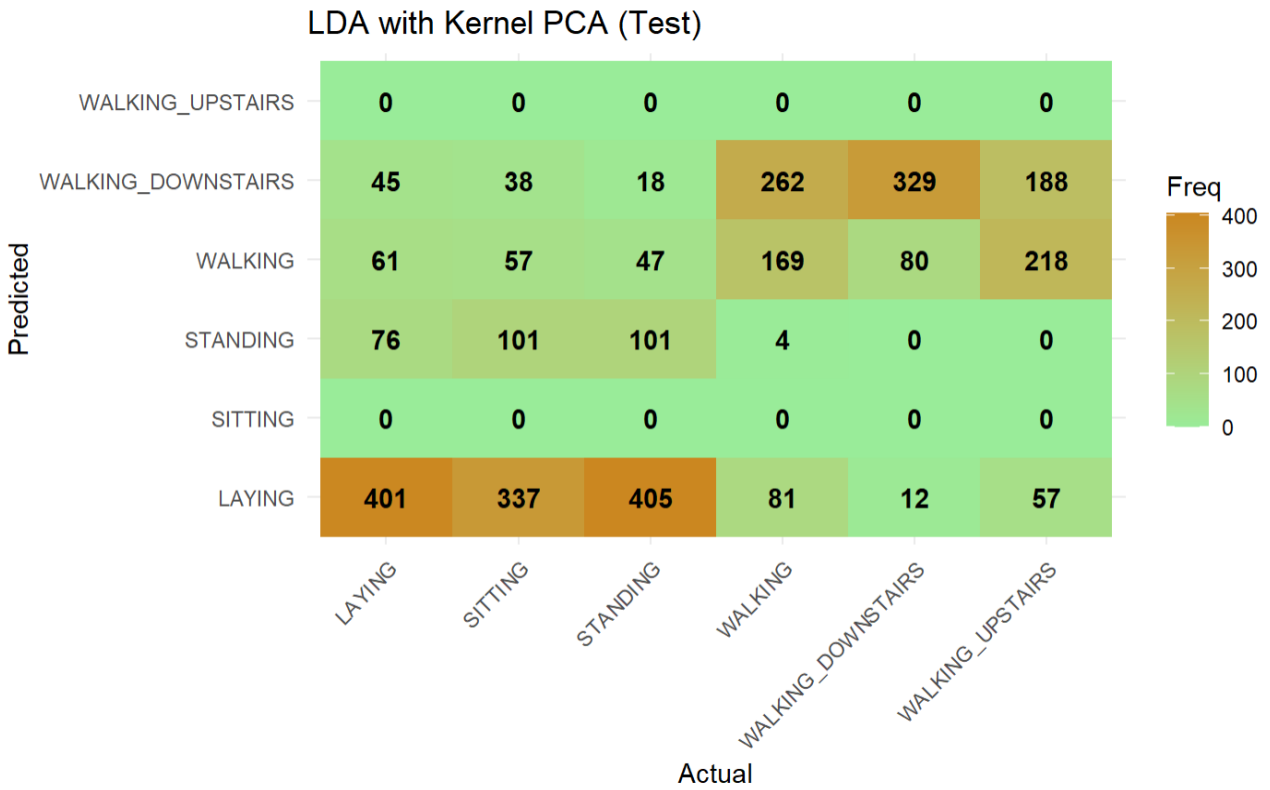


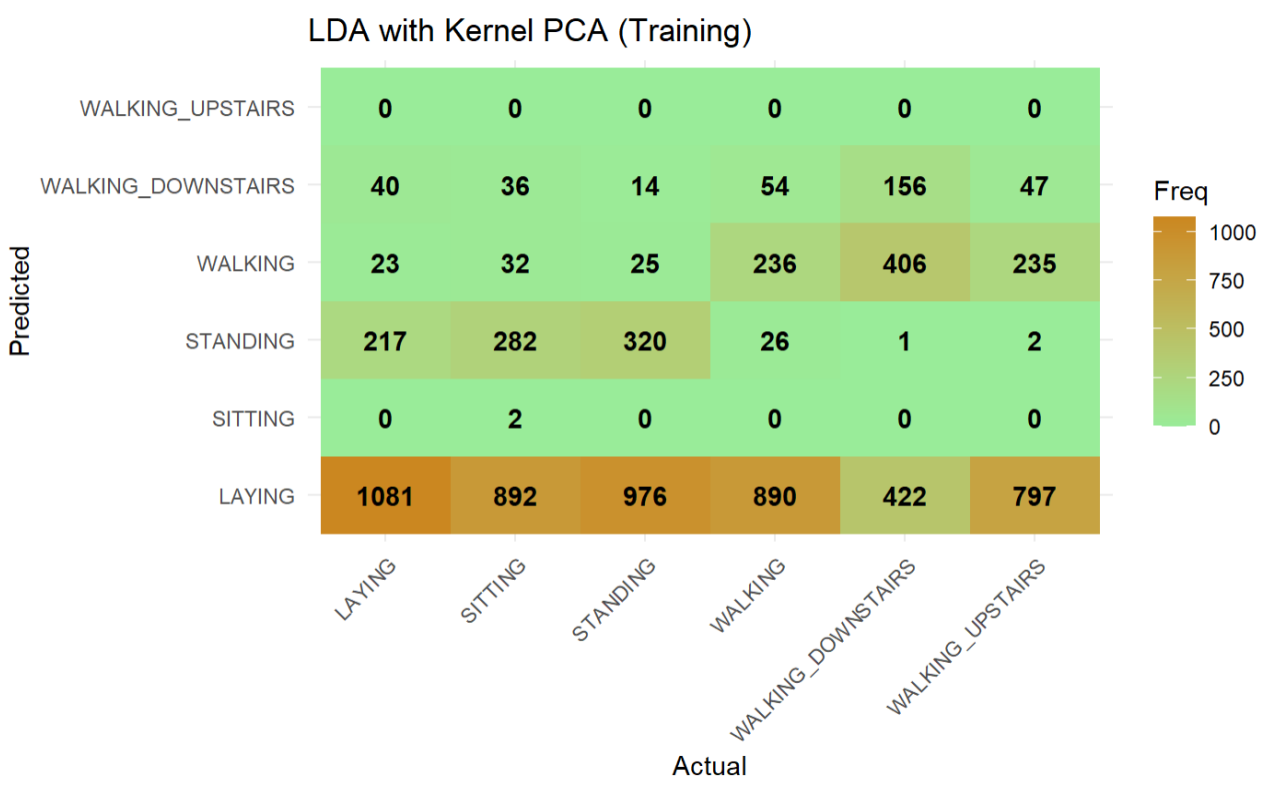


Results of MLR with Kernel PCA

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

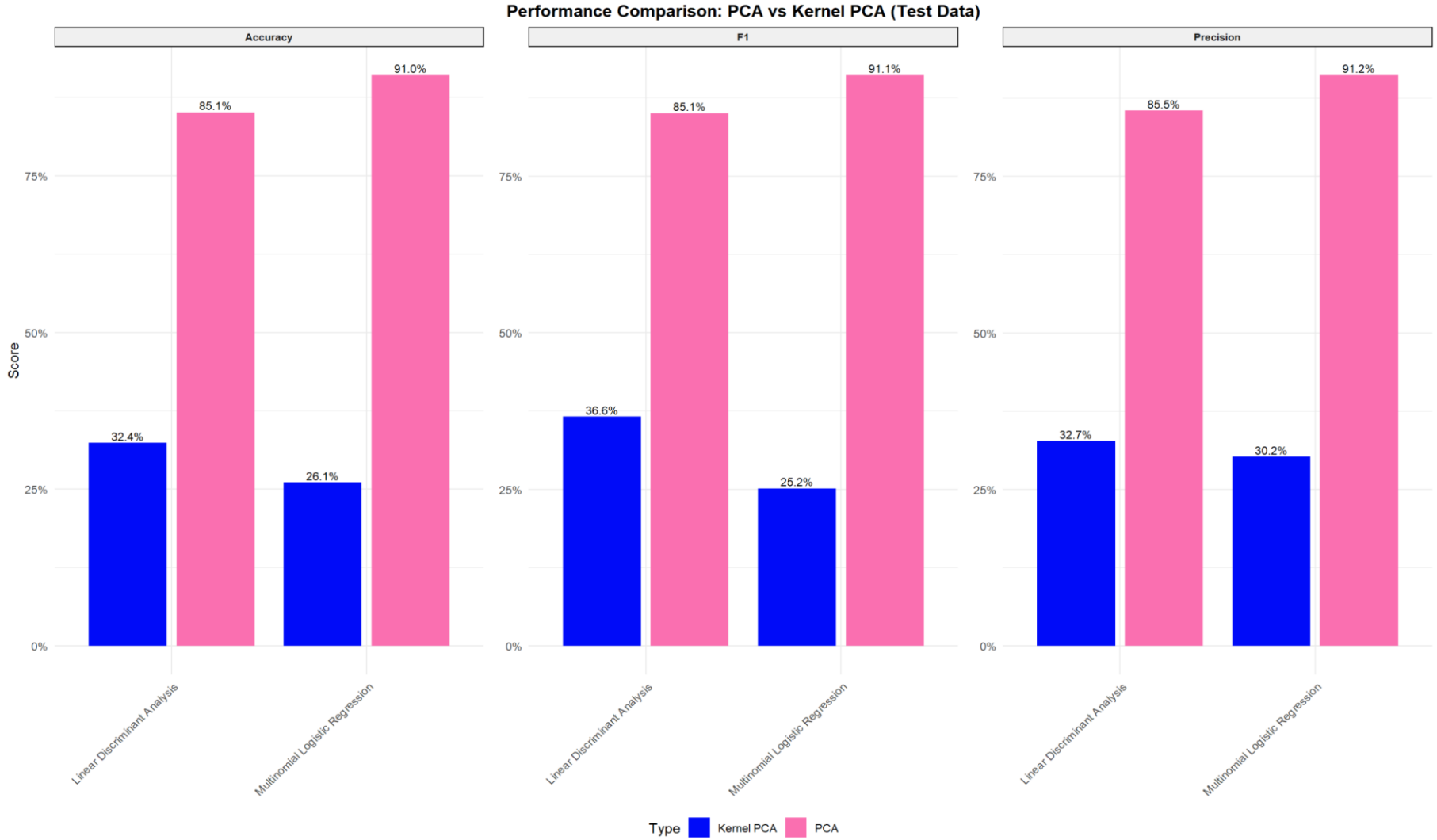




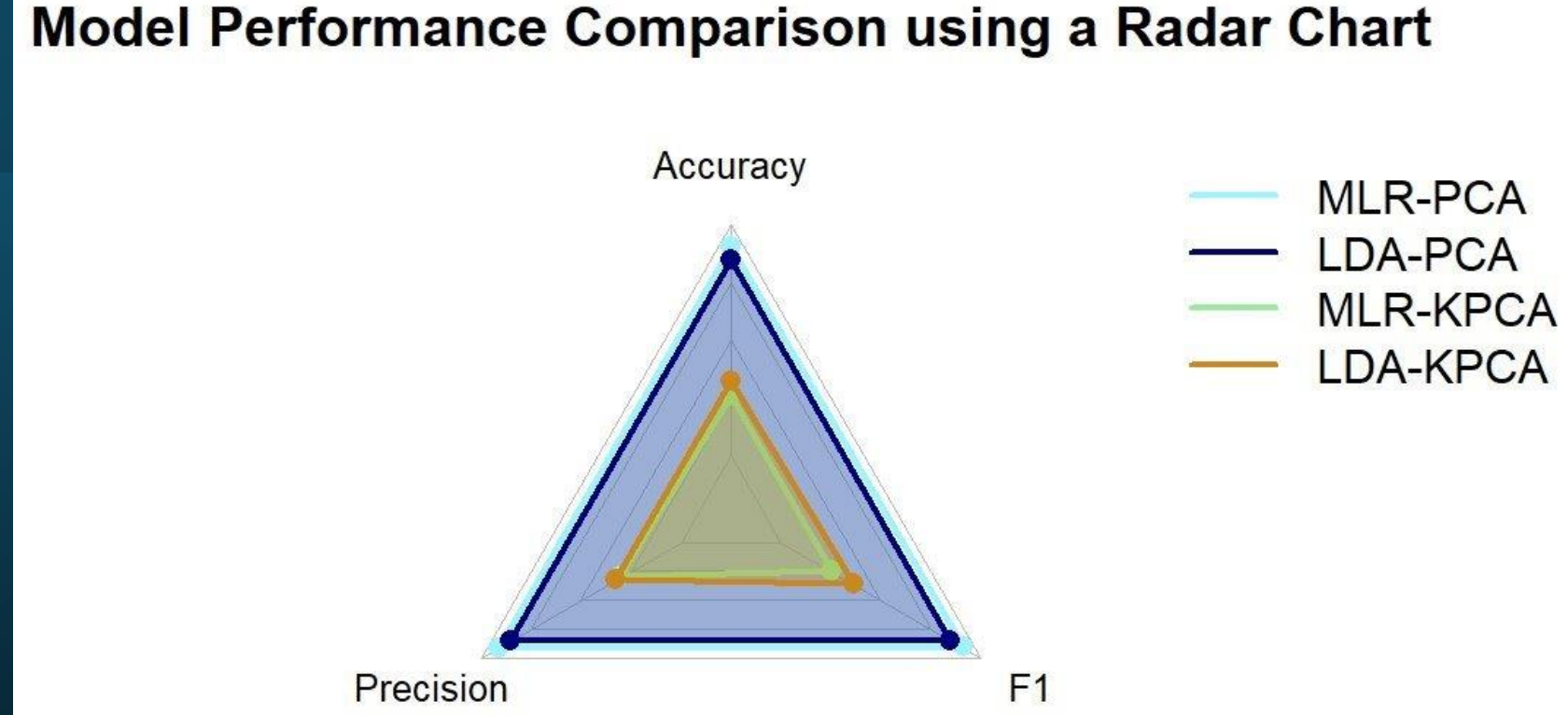
Results of LDA with Kernel PCA

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



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

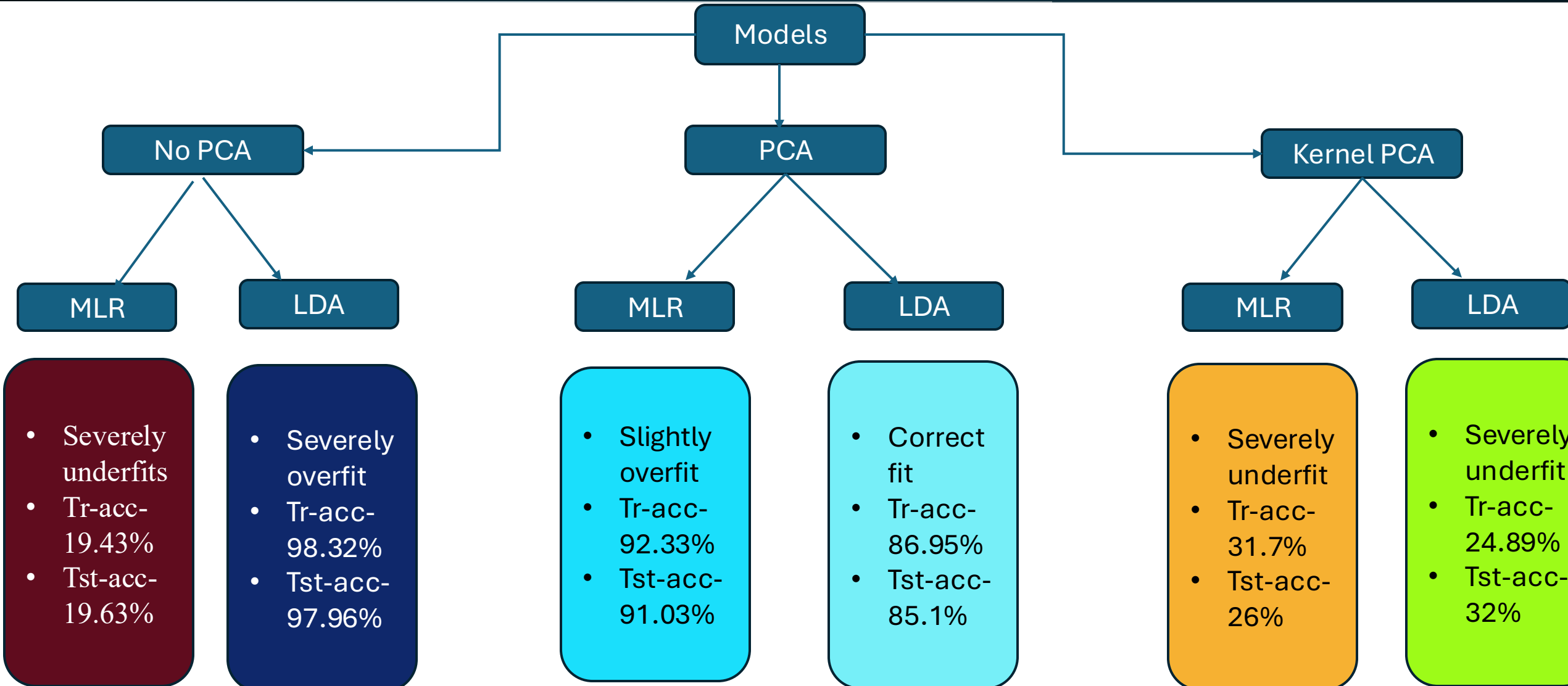
C) MLR with PCA

D) LDA with PCA

E) MLR with Kernel PCA

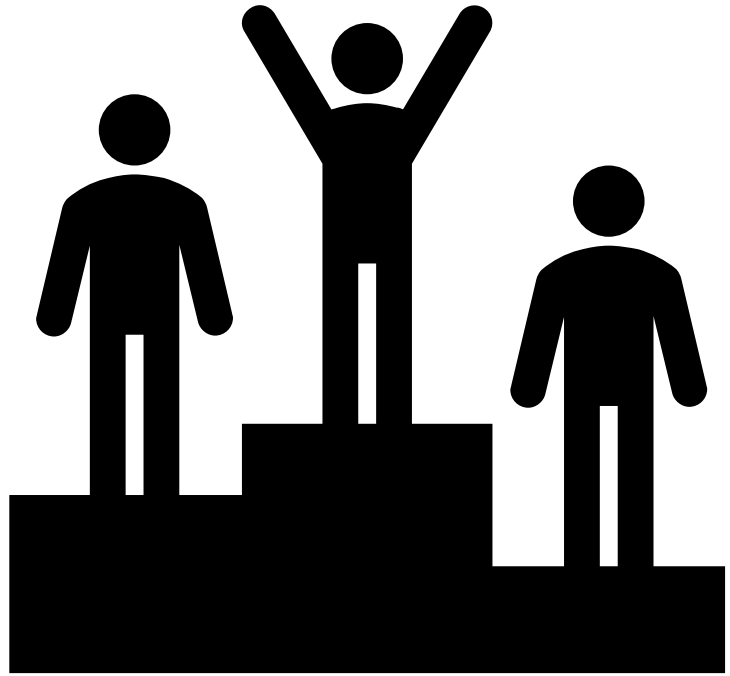
F) LDA with Kernel PCA

Conclusion


Models
No PCA
PCA
Kernel PCA
MLR
LDA
MLR
LDA
MLR
LDA
• Severely underfits
• Tr-acc-19.43%
• Tst-acc-19.63%
• Severely overfit
• Tr-acc-98.32%
• Tst-acc-97.96%
• Slightly overfit
• Tr-acc-92.33%
• Tst-acc-91.03%
• Correct fit
• Tr-acc-86.95%
• Tst-acc-85.1%
• Severely underfit
• Tr-acc-31.7%
• Tst-acc-26%
• Severely underfit
• Tr-acc-24.89%
• Tst-acc-32%

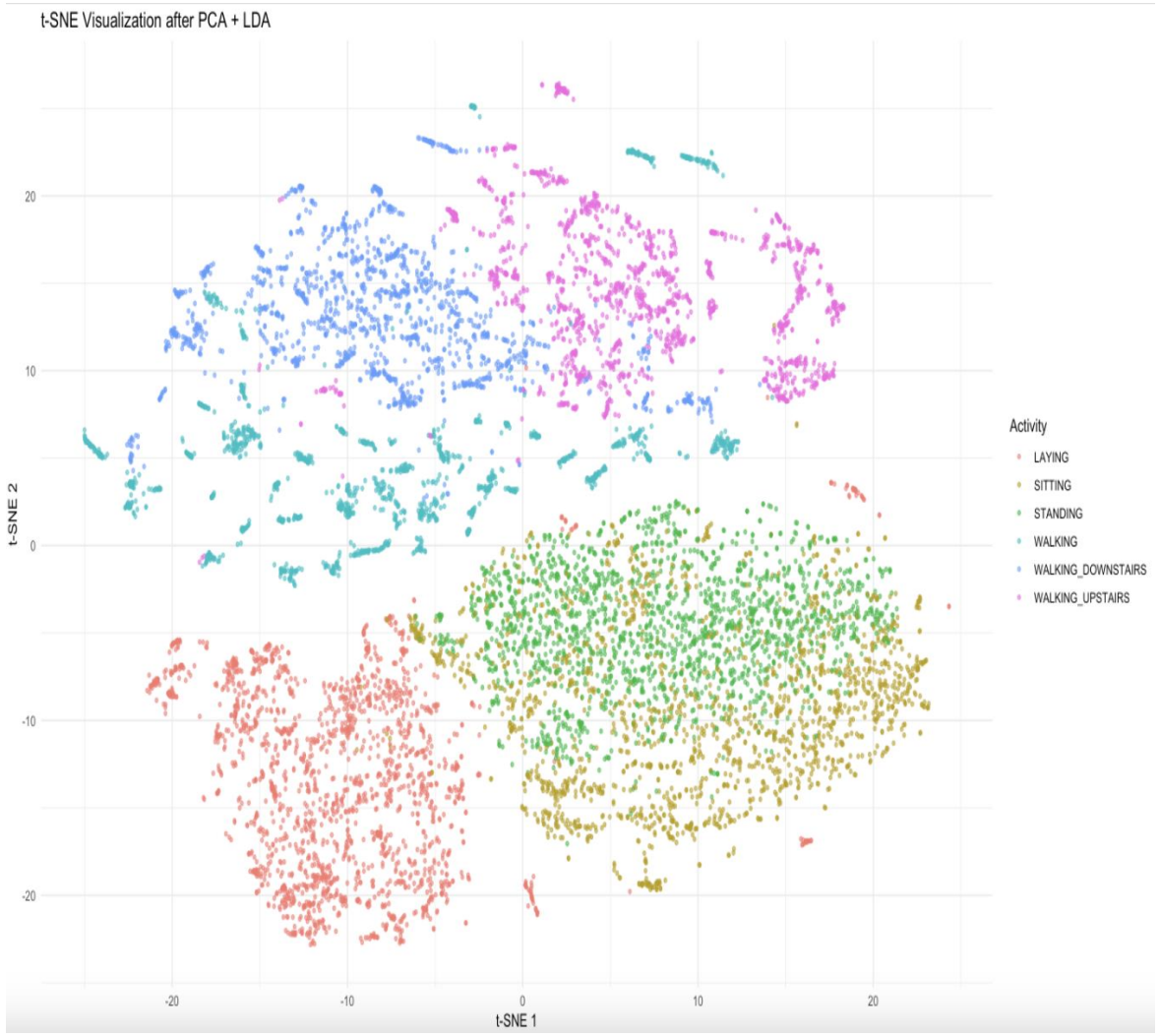
Best Model

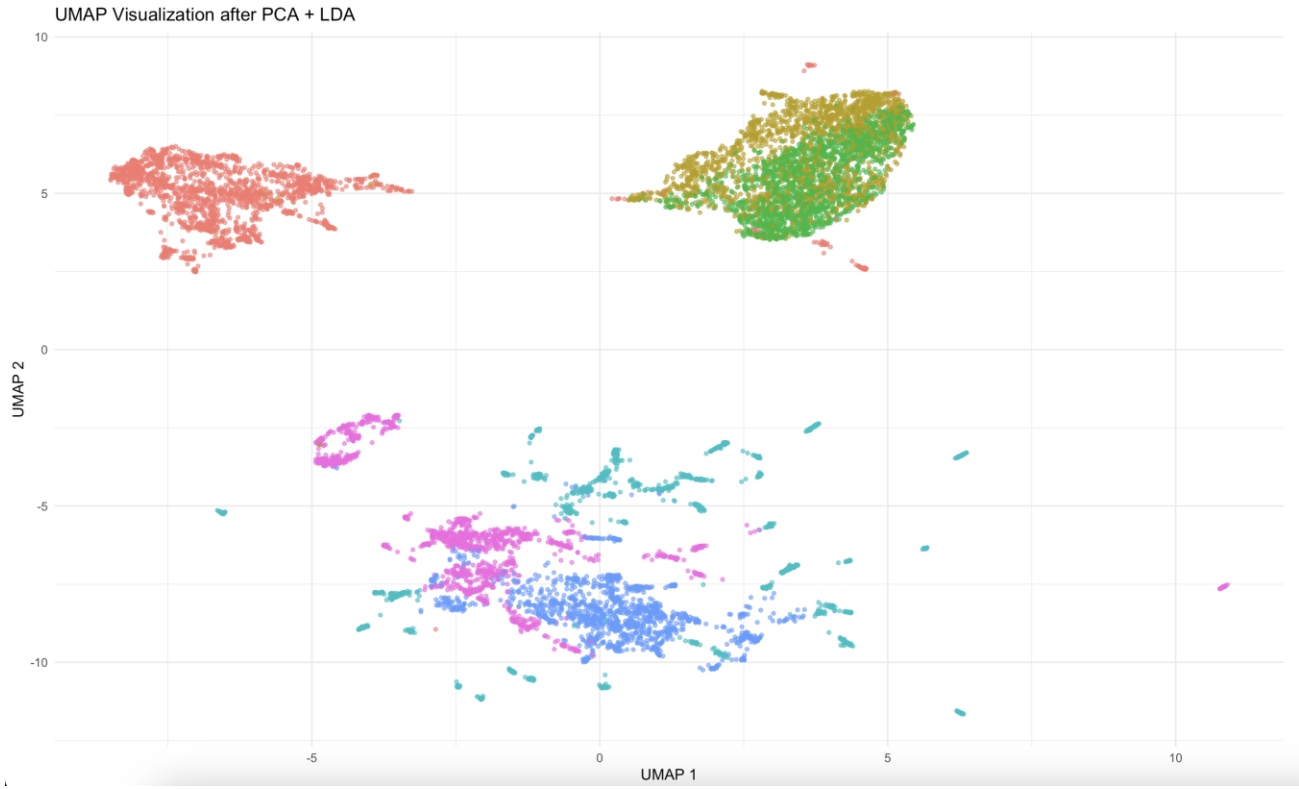
- ✓ **No PCA:** Underfit / Overfit
- ✓ **PCA:** Best performance
- ✓ **PCA + LDA:** Optimal balance
- ✓ Neither underfitting nor overfitting
- ✓ High test accuracy
- ✓ Good generalization without overfitting



PCA with LDA

UMAP & t-SNE for LDA with PCA





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 (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 !