PCA and SVM Analysis of Diabetes Datasets (No Activity vs Physical Activity)

Introduction

This work analyzes the impact of physical activity on diabetes based on two independent data sets: one for the inactive and another for the active. Both sets are provided with health as well as diabetes measurements. Exploratory visualization, Principal Component Analysis (PCA), and training Support Vector Machine (SVM) classifiers for both activity classes are my methodology. I then estimate the performance of each model for the other group to realize the generalization capacity of the patterns.

Dataset Overview

There are four CSV files in the analysis, each containing:

A target variable (Diabetes_binary: 1=diabetic, 0=non-diabetic)

Seven indicators of health: BMI, GenHlth (self-reported health), MentHlth (bad mental health days/month), PhysHlth (bad physical health days/month), Age (categorical grouping), Education (ordinal level), and Income (ordinal level)

The datasets are balanced with:

No-Activity group: 700 training samples (352 diabetic/348 non-diabetic) and 300 test samples

Physical Activity group: Same sample distribution

I imported all the data sets using pandas and then split the seven health features (X matrices) from the diabetes outcome variable (y vectors).

Exploratory Data Analysis

Scatter Plot of BMI vs. Age

To get an initial sense of the data, I plotted **BMI** against **Age** for each dataset (using BMI and Age as two high-variance features). I created a 2x2 grid of scatter plots for: No-Activity Training, No-Activity Test, Phys-Activity Training, and Phys-Activity Test. In each plot, points are colored blue for non-diabetic (y=0) and orange for diabetic (y=1).

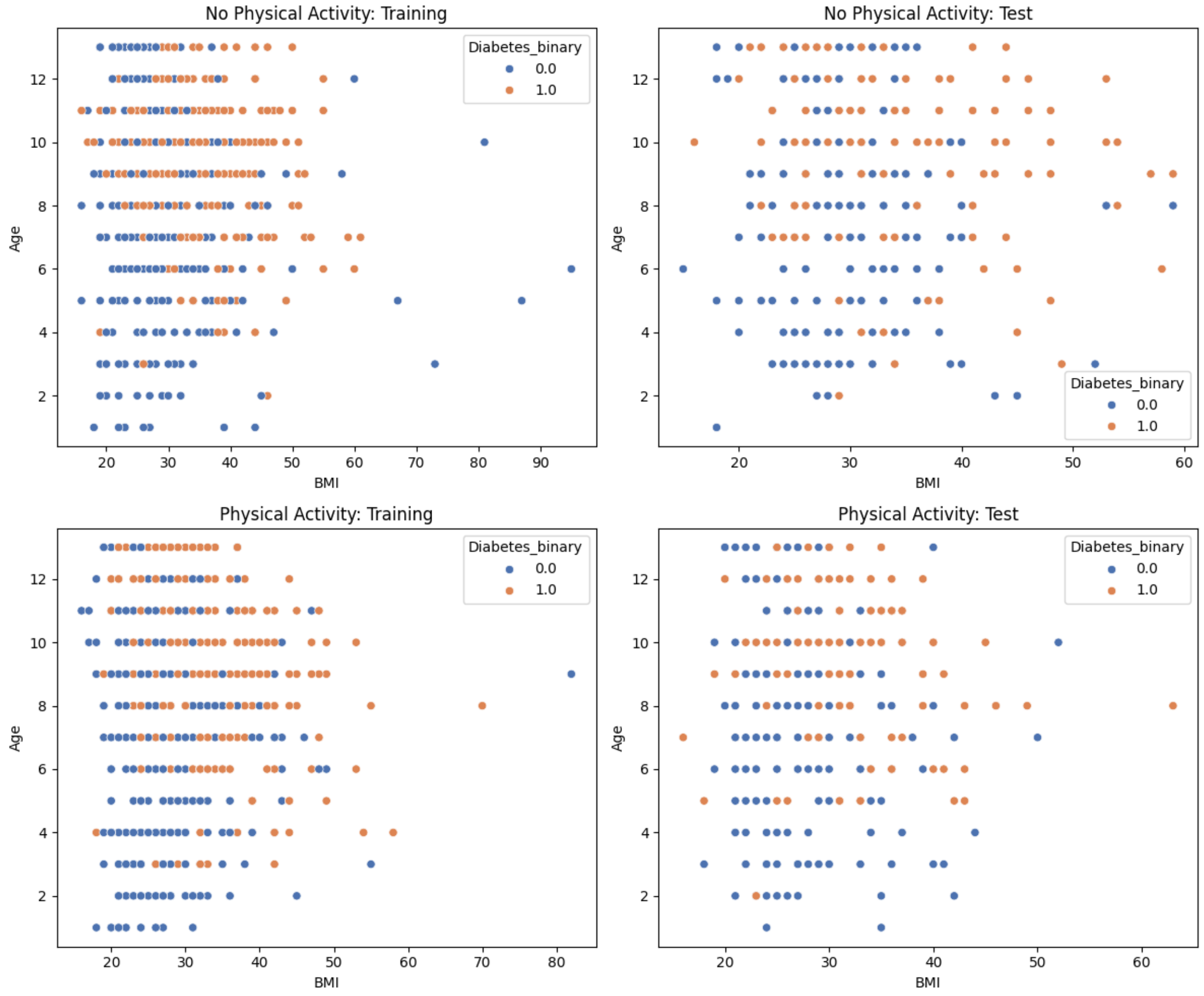

Figure 1: Scatter plots of BMI vs. Age for the four datasets (top row: No Physical Activity group; bottom row: Physical Activity group; left: training set; right: test set). Blue points denote non-diabetic individuals and orange points denote diabetic individuals. I observe substantial overlap between orange and blue points in all plots, indicating that diabetes cannot be cleanly separated by BMI and Age alone. The age feature is categorical (1–13), so points align vertically; diabetics and non-diabetics appear in all age groups. The Physical Activity group (bottom) shows a slight leftward shift in BMI compared to the No-Activity group (top), suggesting that physically active individuals tend to have slightly lower BMI on average (e.g., many Phys-Activity points are concentrated at BMI 20–35, whereas No-Activity has more individuals with BMI above 35).

Data Normalization

Before further analysis, I normalized the features in each dataset using standardization (zero mean, unit variance). For each group, the StandardScaler was fit on the training set features and then applied to **both** the training and test features of that group. This procedure maintains comparability while ensuring the test data remains unseen during scaling. After scaling, each feature in the training sets has mean 0 and standard deviation 1 by construction. The test sets, when transformed using the training scalers, end up approximately standardized. For example, in the No-Activity test set, the first feature (BMI) has mean ≈ 0.051 and std ≈ 0.932 after scaling, and in the Phys-Activity test, BMI has mean ≈ –0.043 and std ≈ 0.874. These are close to 0 and 1, respectively. The slight deviations reflect that the test distribution isn’t identical to training—e.g. the Phys-Activity test BMI average is slightly lower than the Phys-Activity training BMI average (resulting in a small negative mean after scaling).

Principal Component Analysis (PCA)

I applied PCA to each training set after standardization, in order to investigate the variance structure and to visualize the data in a lower-dimensional space. I retained all 7 principal components for analysis. **Table 1** below shows the variance explained by each principal component for the two groups:

Table 1: Explained variance ratio per principal component for the No-Activity and Phys-Activity training data. (Each value represents the fraction of total variance in the dataset captured by that component.)

Principal Component	No-Activity Variance %	Phys-Activity Variance %
PC1	31.3%	30.4%
PC2	17.2%	17.5%
PC3	15.2%	14.5%
PC4	13.7%	13.8%
PC5	8.9%	9.3%
PC6	8.3%	8.3%
PC7	5.4%	6.1%

Both groups exhibit a very similar pattern: the first principal component (PC1) accounts for about 30–31% of variance, the second for ~17–18%, and the first two together around 48% of the variance. Subsequent components each contribute progressively less (PC3 ~15%, PC4 ~14%, etc.). This suggests that the overall data structure (in terms of variance distribution among features) is alike for those with and without physical activity. I can infer that no single feature completely dominates the variance; rather, multiple components are needed to explain the data spread.

I also visualized the training data in the space of the first two principal components (which together capture roughly half the variance):

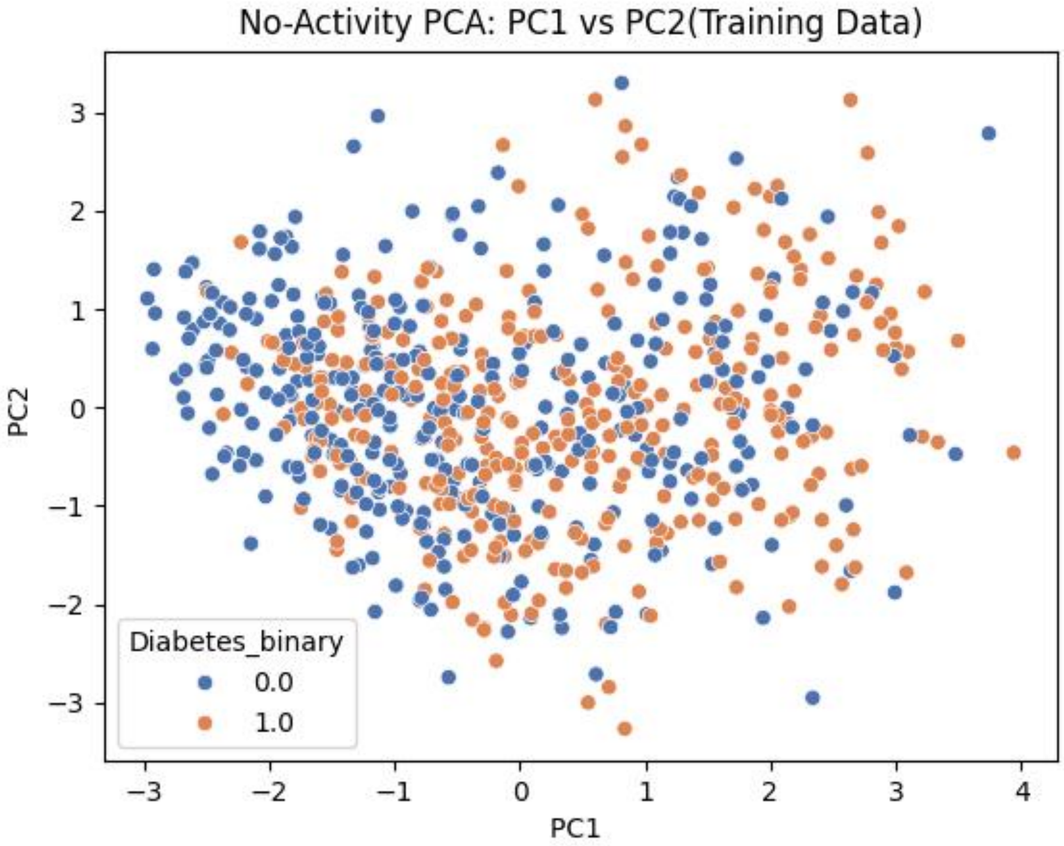

Figure 2: PCA scatter plot for the No-Activity training data, showing individuals in the plane of principal component 1 (PC1) vs principal component 2 (PC2). Orange points are diabetic cases and blue points are non-diabetic. PC1 and PC2 explain ~31% and ~17% of variance respectively. I see that the two classes are largely intermixed in this 2D projection. There is no clear boundary that separates orange and blue points, indicating that a simple linear classifier on these components would struggle. Some regions (e.g., the upper-right) have a slightly higher concentration of orange points, but overall the overlap is significant. This implies that if a linear combination of features can separate the classes, it likely involves higher-order components or non-linear relations, which motivates the use of a more complex classifier (like SVM with an RBF kernel).

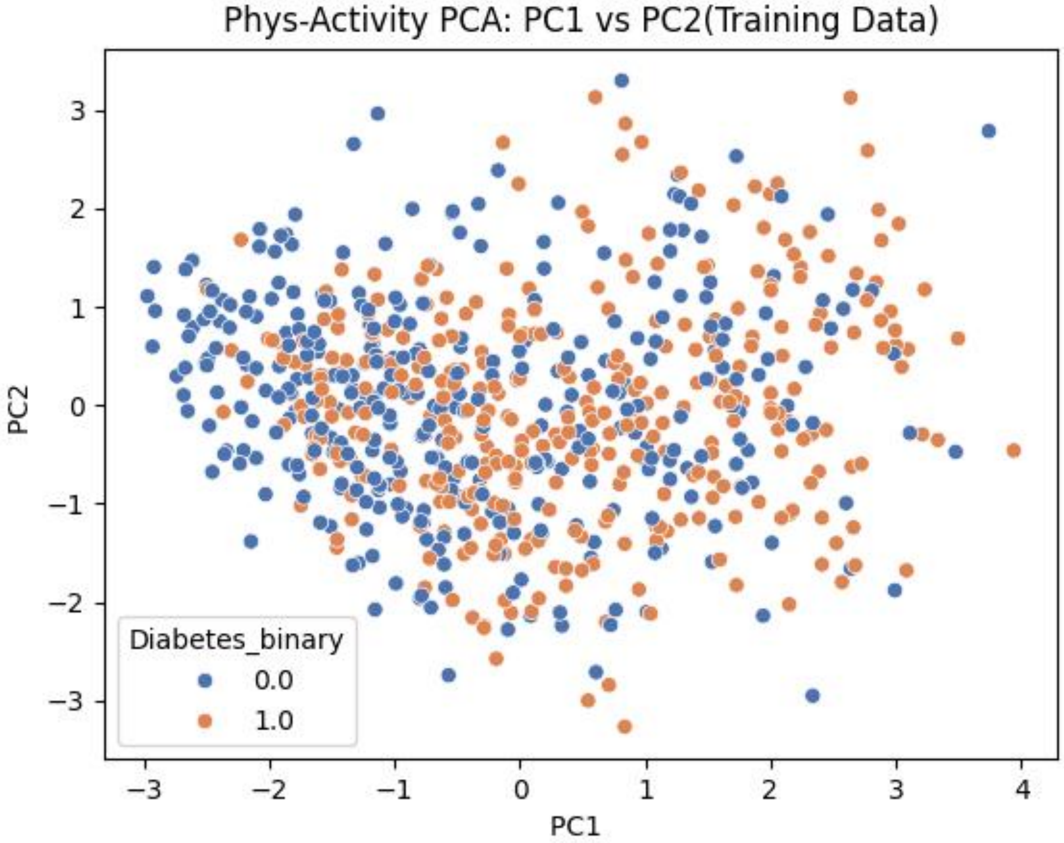

Figure 3: PCA scatter plot for the Phys-Activity training data (PC1 vs PC2), colored by class (blue: no diabetes, orange: diabetes). The pattern is very similar to Figure 2. Diabetic and non-diabetic individuals overlap throughout the plot, without distinct clustering by class. Thus, even though PC1 for this group also captures ~30% of variance (and PC2 ~17%), these top components do not correspond to an obvious diabetic vs non-diabetic axis. Comparing Figure 2 and 3, the distribution of points is analogous, which reinforces that both groups have comparable internal variance structure. Any subtle differences are not enough to linearly distinguish the classes in the first two principal components space.

SVM Classification – No-Activity Group

Training/Validation Split and Scaling

For the No-Physical-Activity group, I used the provided training set (700 samples) and further split it into an 80% **training subset** (560 samples) and a 20% **validation subset** (140 samples). I stratified this split to maintain the class balance (ensuring roughly equal proportion of diabetics in both subsets). I then re-applied standardization: a new StandardScaler was fit on the 560-sample training subset and used to transform both the training and validation subset features. This scaling was done independently of the earlier full-dataset PCA scaling and is part of the modeling pipeline. After scaling, the training subset's feature means are 0 and std devs 1. The validation subset's features came out close to standardized as well; for instance, the mean of scaled BMI in the validation subset was ~0.05 with std ~0.90, indicating the validation data distribution is fairly similar to the training subset distribution. I did not observe large discrepancies in any feature's scale between train and validation, so no significant distribution shift is present within the NoActivity data split. This setup ensures that our SVM training and validation occur on the same scaled feature space, and the validation performance will guide model selection.

Hyperparameter Tuning on Validation Set

I trained SVM classifiers (with an RBF kernel) on the 560-sample training subset using three different hyperparameter combinations provided:

- **Model 1:** C = 1, γ = 1
- **Model 2:** C = 5, γ = 0.5
- **Model 3:** C = 0.5, γ = 0.05

Here C is the regularization parameter (larger C means less regularization, allowing more complex decision boundaries) and γ (gamma) is the RBF kernel coefficient (higher γ means a more wiggly, localized decision boundary). I evaluated each trained model on the 140-sample validation set and recorded the accuracy:

- C=1, γ=1 → **Validation Accuracy = 67.9%**
- C=5, γ=0.5 → **Validation Accuracy = 66.4%**
- C=0.5, γ=0.05 → **Validation Accuracy = 70.0%**

Model 3 (C=0.5, γ=0.05) performed the best on the validation set with 70.0% accuracy, slightly outperforming the other two. I selected this parameter combination as the best for the NoActivity group. It's interesting to note that the model with the lowest C (more regularization) and lowest γ (less complex kernel) did best, suggesting that a simpler decision boundary was most appropriate for this training subset – the higher-capacity models might have started to overfit the training data or captured noise, leading to lower validation accuracy.

Test Set Evaluation (No-Activity Model)

After choosing the best hyperparameters, I took the SVM model trained with C=0.5, γ=0.05 (on the 560 training subset) and evaluated it on the No-Activity **test set** (300 samples that were held out from the beginning). Before predicting, I scaled the test set features using the same scaler fit on the 560 training subset (this is critical – I apply the exact training feature transformations to the test data). The model's performance on the test set was as follows:

- **Test Accuracy (No-Activity model on No-Activity test)** = **70.0%** (210/300 correct classifications).

To better understand the results, **Table 2** presents the confusion matrix of the predictions on the test set:

Table 2: Confusion matrix for the No-Activity SVM model on its own test set (300 samples). The rows correspond to actual class and columns to predicted class.

Actual \ Predicted	No Diabetes (Pred=0)	Diabetes (Pred=1)
No Diabetes (actual 152)	95 (TN)	57 (FP)

Actual \ Predicted	No Diabetes (Pred=0)	Diabetes (Pred=1)
Diabetes (actual 148)	33 (FN)	115 (TP)

From the confusion matrix: out of 152 truly non-diabetic individuals, 95 were correctly classified as non-diabetic (true negatives) and 57 were incorrectly classified as diabetic (false positives). Out of 148 actual diabetics, 115 were correctly identified (true positives) while 33 were missed (false negatives). This implies:

- **Sensitivity** (recall for diabetic class) ≈ 115/148 ≈ 77.7%. The model detects roughly 78% of actual diabetes cases.
- **Specificity** (recall for non-diabetic class) ≈ 95/152 ≈ 62.5%. It correctly rules out diabetes in ~62% of actual non-cases, meaning it has about a 37.5% false positive rate.

SVM Classification – Physical Activity Group

I repeated an analogous process for the Physically Active group's data.

Train-Validation Split and Scaling

The Phys-Activity training set of 700 samples was split into 80% (560 samples) for training and 20% (140 samples) for validation, again stratified by class (ensuring both subsets have roughly equal class proportions, which were 352 diabetic vs 348 non-diabetic in the full set). I scaled the features of this group separately: fitting a StandardScaler on the 560 Phys-Activity training subset and using it to transform both training and validation features for this group. The scaling yielded training subset features with mean 0, std 1; the validation subset's scaled features had means close to 0 and stds ~1 (for example, scaled BMI in the Phys-Activity validation had mean –0.019 and std 0.902, very similar to the training distribution). There were no large anomalies in feature distribution between train and validation, indicating a stable split.

Hyperparameter Tuning on Validation Set

Using the Phys-Activity training subset, I trained SVMs with the same three hyperparameter combinations:

- C=1, γ=1
- C=5, γ=0.5
- C=0.5, γ=0.05

and evaluated on the 140-sample Phys-Activity validation subset. The validation accuracies obtained were:

- C=1, γ=1 → **72.1%**
- C=5, γ=0.5 → **73.6%**
- C=0.5, γ=0.05 → **74.3%**

All three models performed somewhat better here than in the NoActivity case, with the highest accuracy again from C=0.5, γ=0.05 (74.3%). This was only slightly above the second model's 73.6%, but I chose **C=0.5, γ=0.05** as the best for consistency (it gave the top result). Interestingly, the same combination was optimal for both groups, suggesting that a simpler, more regularized model generalized best in both subsets. Possibly the features and class patterns in both groups benefit from a smoother decision boundary (lower γ) and more regularization (lower C) to avoid overfitting noise.

Test Set Evaluation (Phys-Activity Model)

I took the SVM model trained on the Phys-Activity subset with C=0.5, γ=0.05 and evaluated it on the 300-sample Phys-Activity test set. Again, the test features were scaled using the Phys-Activity training subset's scaler prior to prediction. The model achieved:

- **Test Accuracy (Phys-Activity model on Phys-Activity test)** = **76.33%** (approximately, 229 out of 300 correct).

The confusion matrix for this model's test predictions is shown in **Table 3**:

Table 3: Confusion matrix for the Phys-Activity SVM model on its test set (300 samples).

Actual \ Predicted	No Diabetes (Pred=0)	Diabetes (Pred=1)
No Diabetes (actual 152)	106 (TN)	46 (FP)
Diabetes (actual 148)	25 (FN)	123 (TP)

Out of 152 non-diabetic individuals in the Phys-Activity test set, 106 were correctly classified (TN) and 46 were false positives. For 148 diabetics, 123 were identified (TP) and 25 were missed (FN). In terms of performance metrics:

- **Sensitivity** ≈ 123/148 ≈ 83.1%, meaning the model caught over 83% of actual diabetic cases in the physically active group (higher recall than the NoActivity model had).
- **Specificity** ≈ 106/152 ≈ 69.7%, so about 30.3% false positives (also better than the NoActivity model's ~37.5% false positive rate).

This model thus performed better on both precision and recall compared to the NoActivity model. A 76% accuracy is a notable improvement over 70%. The Phys-Activity model seems to achieve a more balanced and higher performance, suggesting that the diabetes vs. no-diabetes classification might be slightly easier for the physically active population with these features.

Cross-Group Model Evaluation

Having trained separate models for each group, I next examined how well each model generalizes to the other group's data. This is a **cross-model check** to see if a model trained on one physical activity cohort can be used on the other, or if the differences in distributions and feature relevance degrade performance.

No-Activity Model on Phys-Activity Test Data

I took the SVM model trained on the No-Activity group (C=0.5, γ=0.05) and applied it to the Phys-Activity test set. A crucial step was **normalizing the Phys-Activity test features using the No-Activity scaler** (i.e., using the mean and std from the NoActivity training data). I do this because the model was trained on features scaled in the NoActivity feature space; to get meaningful predictions, I must represent the PhysActivity individuals in that same space. (If I instead scaled PhysActivity data with its own mean/std, the values would not be comparable to the model's training, potentially leading to erroneous results.) After transforming the PhysActivity test set with the NoActivity scaler, I fed it into the NoActivity SVM model.

The NoActivity-trained model achieved **72.0% accuracy** on the PhysActivity test set. This is quite close to the model's original accuracy on its own test (70%), indicating some degree of generalization. However, the pattern of errors differed notably. The confusion matrix for this cross-application is:

- **NoActivity model on PhysActivity test:** 124 TN, 28 FP, 56 FN, 92 TP (out of 152+148 samples).

In a clearer form: the model correctly identified **124 out of 152** non-diabetics in the PhysActive test (so specificity ~81.6%, much higher than the ~62.5% it had on NoActivity test), and it identified **92 out of 148** diabetics (sensitivity ~62.2%, much lower than the ~77.7% it had originally). In other words, on the Phys-Activity population, the No-Activity model became more **conservative**: it labeled far fewer people as diabetic (only 120 predicted positives vs 148 actual positives), resulting in fewer false positives but many more false negatives. I can interpret this shift as follows: physically active individuals tend to have lower BMI and perhaps generally better health metrics, so the NoActivity model (trained on a sedentary population with generally higher risk factor levels) finds many of the PhysActive diabetics not sufficiently "extreme" to cross its decision threshold. Thus it misses some diabetics (56) that it would likely catch if they had the higher-risk profile typical in the no-activity group. On the flip side, it had little trouble recognizing most non-diabetics in the PhysActive group (since many of them probably have even healthier profiles), hence the drop in false positives. The net accuracy (72%) didn't suffer compared to original, but the **error balance changed** dramatically – the model traded off sensitivity for higher specificity in this new group.

Phys-Activity Model on No-Activity Test Data

Conversely, I tested the SVM model trained on the Phys-Activity group (C=0.5, γ=0.05) on the No-Activity test set. I **normalized the No-Activity test features using the PhysActivity scaler** (from the Phys training data) to put them in the correct scale for the Phys model. The accuracy in this case was **68.33%**, a bit lower than the Phys model's original 76.3% on its own test and also slightly lower than the NoActivity model's 70% on this data. The confusion matrix for the Phys-model on NoActivity test was:

- **PhysActivity model on NoActivity test:** 70 TN, 82 FP, 13 FN, 135 TP.

This model correctly identified **135 of 148 diabetics** in the no-activity group – an impressive sensitivity of ~91.2% (much higher than its original 83% on Phys group). However, it only correctly identified **70 of 152 non-diabetics**, with **82 false positives** (!), yielding a specificity of just ~46%. In summary, the Phys-trained model was very **liberal** in assigning the diabetes label when applied to the no-activity population. It essentially flagged most individuals as diabetic, catching nearly all true diabetics but also falsely alarming more than half of the non-diabetics. This is the opposite error profile of the previous cross-test. The likely reason is that the No-Activity group has generally higher risk factor values (e.g., higher average BMI, more health issues). The PhysActivity model, having been trained on a healthier cohort, interprets many of these no-activity individuals (even some who are actually non-diabetic) as high risk and thus predicts them as diabetic. For example, a person with BMI 30 in the Phys-active group might have been near the higher end of that distribution and possibly diabetic, whereas in the no-activity group BMI 30 is fairly common even for non-diabetics.

Comparison and Discussion of Cross-Model Results

The cross-model evaluation highlights how the relationship between features and the diabetes outcome is **context-dependent on the physical activity group**. Each model performed best on the population it was trained on. When applied to the other population, both models still achieved around 68–72% accuracy, but they did so by shifting the balance of sensitivity and specificity:

- The **No-Activity model** on the PhysActive group favored higher specificity (few false positives) at the cost of missing many diabetics (low sensitivity).
- The **Phys-Activity model** on the NoActivity group favored higher sensitivity (catching most diabetics) at the cost of flagging too many non-diabetics (low specificity).

Neither cross-application is as robust as using the model on its native group. These differences likely stem from distributional shifts: the No-Activity group has overall poorer health metrics (higher BMIs, more days of ill health, etc.), which changes the baseline risk level. A model trained on that group will have a higher threshold for what it considers "diabetic" risk (since even non-diabetics can have moderately high BMI or some health issues), whereas a model trained on a healthier, active group will have learned a lower threshold (since in that group, even moderate risk factors might strongly indicate diabetes). When swapping, the thresholds no longer align optimally with the new group's reality.

In practical terms, this suggests that a diabetes prediction model might need to account for physical activity status directly (perhaps as an input feature, which was absent here because I split the data by that attribute). If one were to deploy a single model across all individuals, one should incorporate the **"physical activity" factor** or train on a combined dataset covering the full spectrum, so that the model can adjust its decision boundary accordingly.

Conclusion

In this experiment, I analyzed two subsets of a diabetes dataset grouped by physical activity. I found that both groups exhibit similar variance patterns in their features (PCA results were alike, with no single component explaining a majority of variance). Using an RBF SVM, I achieved moderate success in classifying diabetes: about 70% accuracy for sedentary individuals and 76% for physically active individuals. The no-activity model was too conservative on active folks, and the active model too aggressive on sedentary folks.

Overall, the SVM models did capture some signal (exceeding baseline random accuracy of 50% by a good margin), but there is room for improvement. More sophisticated approaches could include incorporating the activity variable into a unified model, trying additional features or algorithms, or performing cross-validation more extensively for hyperparameter tuning. Nonetheless, this analysis provided insights into how these factors play out.