# Principal Component Analysis (PCA) in Breast Cancer Malignancy Prediction

This study systematically analyses how dimensionality reduction using Principal Component Analysis (PCA) affects machine learning classifiers prediction of breast cancer malignancy. PCA reduces multicollinearity, and noise. Logistic Regression, Support Vector Machine (SVM), Random Forest, and Multi-layer Perceptron (MLP) classifiers were trained and evaluated across all possible PCA-components (1–30). For each PCA dimension, the models hyperparameter were fine-tuned via grid search. This study optimises for F2-score, due to the clinical necessity to minimise potentially fatal false negatives. Logistic regression performed the best, benefiting from PCA through increased linear separability. SVM showed expected diminishing returns with increasing PCA-components. Random Forest classifiers performed poorly. MLP classifiers performance varied, showing high sensitivity to dimensionality. Although the best models achieved high F2-scores (0.981), high recall (97–98%) and perfect precision with zero false positives, none detected all malignancies. This highlights their role as complementing a physician in a diagnostic support system. Due to the small dataset size (569 samples), all results are sensitive to the initial random seed when splitting the data set. This limits their statistical significance. Future research should replicate and average this experiment often (e.g., 100 repetitions) or employ larger datasets.

## 1. Introduction

Breast cancer is a leading cause of mortality for women. Early, and accurate diagnosis is critical for effective treatment. Predictive modelling helps reduce misdiagnosis. Higher predictive performance directly impacts early detection, reduces misdiagnosis, and improves patient survival rates. Yet, the binary classification problem of predicting the malignancy of breast tumours, high-dimensional data can introduce noise and redundant information, which impedes predictive performance.

This study attempts to answer the question of how dimensionality reduction using Principle Component Analysis (PCA) affects the predictive performance of machine learning classifiers in distinguishing between benign and malignant breast cancer cases. The goal is to quantify the effects of PCA on the predictive performance of classifiers in breast cancer diagnosis. By optimising diagnostic models through dimensionality reduction, the study could lead to better clinical decision support systems.

This study utilises the widely-used Breast Cancer Wisconsin dataset (Wolberg et al., 1993) to predict breast tumour malignancy from fine needle aspiration (FNA) biopsy features. The dataset contains 569 instances with 30 numerical features—including tumour radius, texture, perimeter, and smoothness. Each instance is labelled as benign (non-cancerous) or malignant (cancerous).

To solve the problem of noise and redundant information this study uses Principal Component Analysis (PCA). PCA reduces the number of dimensions of a data set. It reduces multicollinearity, which can prevent overfitting, improve computational efficiency and interpretability, reduces noise, and can enhance model generalisation.

Many previous studies have worked on improving predictive performance on the Breast Cancer Wisconsin dataset; several ones have also explored the effects of dimensionality reduction. Esen et al. (2024) use 5 PCA-components to achieve impressive perfect performance, but fail to show clearly that these results are due to PCA. Jamal et al. (2018) train two different ML-algorithms using PCA. Ibrahim et al. (2021) explore the use of PCA in an ensemble of different models. While achieving an accuracy of 98.24% and precision (99.29%), their recall value of 95.89% is poorer.

These studies all do not clearly show the effect of PCA over different numbers of dimensions, and for a variety of different models. Furthermore, this study argues, that the above studies optimise their models for a clinically wrong evaluation metric, which might

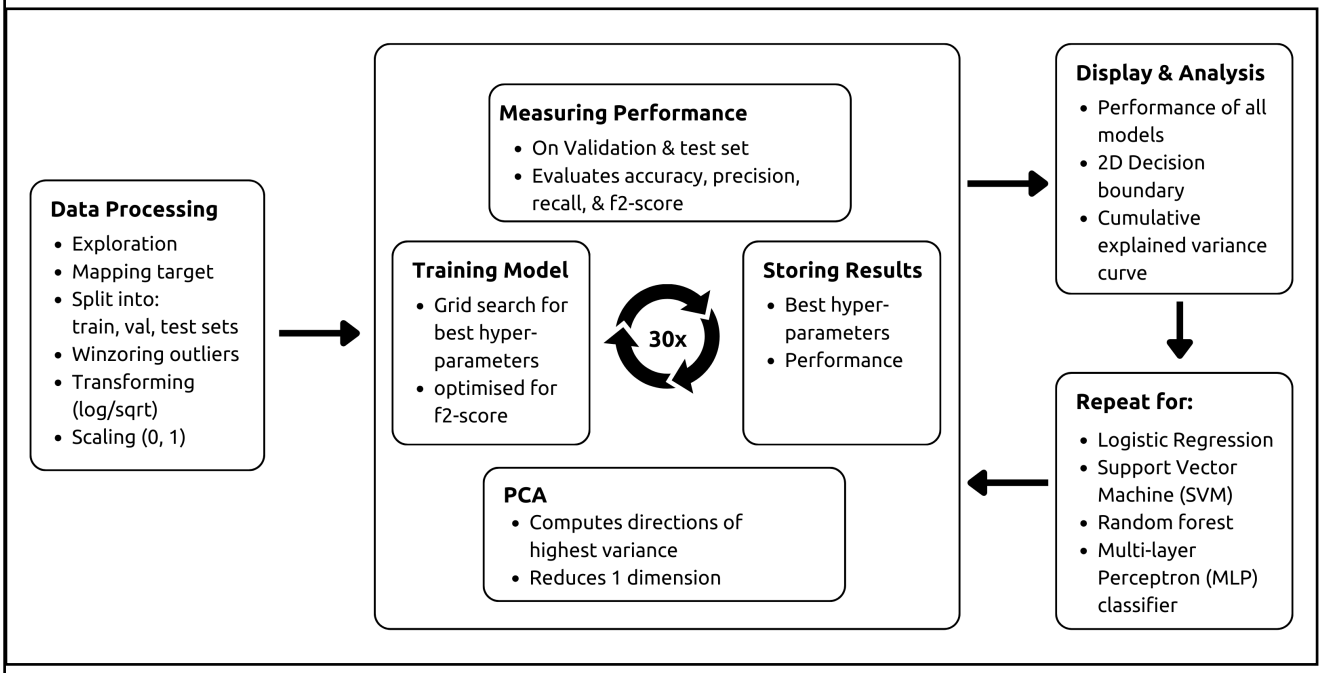


***Figure 1.*** *Flow-diagram for this study*

prove fatal in clinical practice. This study aims to explore these three research gaps by fine-tuning 4 different model architectures on every possible number of PCA-components (1-30) and measuring the performance of all resulting 120 different models.

# 2. Methods

## Data Exploration and Processing

Figure 1 shows this study's flow diagram. The Breast Cancer Wisconsin Diagnostic dataset contains 569 instances with 30 numerical features, and binary class labels. As is the standard in ML-projects the data set was split into training (60%), validation (20%) and test (20%) sets. The split was stratified on the target. Beyond mapping the target to 0 and 1, no further data cleaning was required.

Exploring the distributions of the features revealed that several features had outliers. This is a problem since PCA performs worse with outliers (Jolliffe et al. 2016). That is why this project chose to winzorise outliers. The function "winzorize" clips all values outside of 3 standard deviations from the mean, to exactly these bounds. This applies to approximately 0.3 percent of the data. The winsorisation bounds are learned only from the train set and then applied to validation and test sets.

The distributions of features also showed several not normally distributed features, which further impedes the performance of machine learning algorithms. Transforming these features is especially important for PCA to work. The functions classify_skewness gives a list of features that need heavy transformation (skewness >1), need mild transformation (skewness 0.5-1), and need no transformation (skewness < 0.5). These are then normalised using log for heavy transformation and square root for mild transformations. Lastly, many ML algorithms do not deal well with data at different scales. That is why the function scale_features scales all features between 0 and 1 using Sklearn's MinMaxScaler (Pedregosa et al., 2011). The scaler is fit only on the training set, and only then applies the same transformations to the validation and test sets.

## Principle Component Analysis (PCA)

PCA is a statistical method to reduce the dimensionality of data while retaining as much of its variability and information as possible (Jolliffe et al. 2016). It finds the most useful directions of maximum variation in the data and then shows that distance on principle components (Figure 2.). In essence, PCA turns dimensional data into their most useful directions. This technique has been used through the scikit-learn library. It was this project's aim to analyse the performance of different classifiers for every single possible number of PCA components. This is why the function run_dimensionality_experiment starts at 30 dimensions, as there are 30 features in data set, and then in a loop takes away the PCA direction with the least variation one by one.

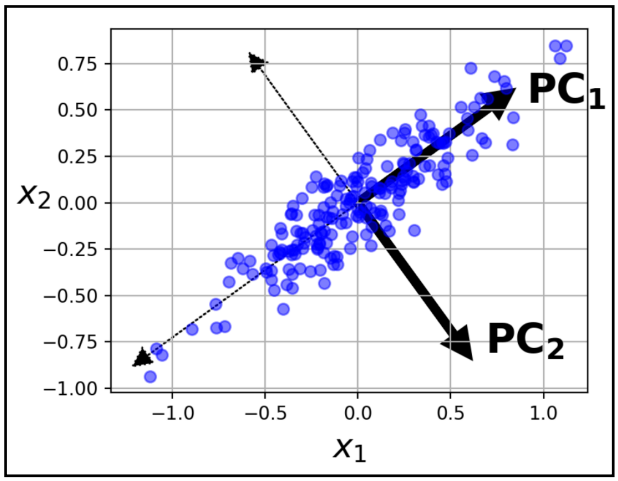


***Figure 2.*** *Principle Component Analysis visualisation, taken from Lecture by Peter Macgregor for ID5059*

## Model Training

For each number of PCA-components, run_dimensionality_experiment then trains a specified classification model. This study has chosen to repeat this process for four different models. They represent diverse modelling approaches, which allows for a good comparison of PCA's effects across different algorithm types.

First, a Logistic Regression model is trained and evaluated. This model estimates the probability of a sample belonging to a certain class by fitting a logistic function. Second, a Support Vector Machine (SVM) is trained. This model calculates linear decision boundaries optimised with stochastic gradient descent. Third, a Random Forest classifier is trained and evaluated. This ensemble method builds many decision trees and aggregates their predictions, where each tree's "vote" "elects" the final prediction. Fourth, a Multi-layer Perceptron (MLP) is trained and evaluated. This is a type of feed-forward neural network with an input layer, hidden layers, and an output layer. Each neuron uses non-linear activation functions to capture complex patterns in the data. The network learns by adjusting the weights between neurons. All models are created and train through the scikit-learn library (Pedregosa et al., 2011).

As the complexity of the data varies greatly within the loop of going through all possible numbers of dimensions, one cannot just define the model's hyperparameter once. This is why at each step of the loop, for each number of PCA-components, the function runs a grid search, looking for the best possible hyper-parameters within a predefined range (see Appendix A). A randomised grid search fits individual models for a predefined number of combinations of hyper-parameters and picks the combination that scores the highest on a predefined evaluation metric. For most cases in this study the function ran a completely grid search, exhausting all possible combinations. This all happens only on the training set.

## Evaluation

The grid search requires an evaluation metric to optimise for, a choice which goes to the heart of what this study aims to achieve. Clinically, as this model will not be used alone, but rather in complement to a physician, the most important aspect for this model is to pick up as many malignant tumours as possible. Its task should be to flag malignancies that the physician might have missed. Thus, it is most important to reduce false negatives, as they are potentially fatal. This important insight is overlooked by many of the previous studies on this dataset (Esen et al., 2024; Jamal et al., 2018; Ibrahim et al., 2021). As a secondary priority, it is also relevant to minimise false positive to some extent, as they might cause over-diagnosis, over-treatments, and the accompanying dangers, costs and psychological strain.

Accuracy shows the share of correct predictions. Precision is the percentage of the model's predicted successes that are actually successes. Recall describes the share of all true predicted successes, which, as described, is most important for this study. The F2 Score balances the above, but more heavily weights recall. This minimises false negatives, while still keeping false positives in check. F2 aligns most with this project's objectives of flagging as many malignancies as possible, and is what this project and grid search optimises for.

The function then runs the model also on the validation and test set and computes all evaluation metrics for all sets. As the models have already been fine-tuned and run on the test set, they will not be changed further.

## Storing and displaying results

run_dimensionality_experiment returns a results dictionary, which stores the PCA dimension as keys and also the corresponding evaluation metrics and best parameters. The function plot_results then loops over the keys and evaluation metrics in the results

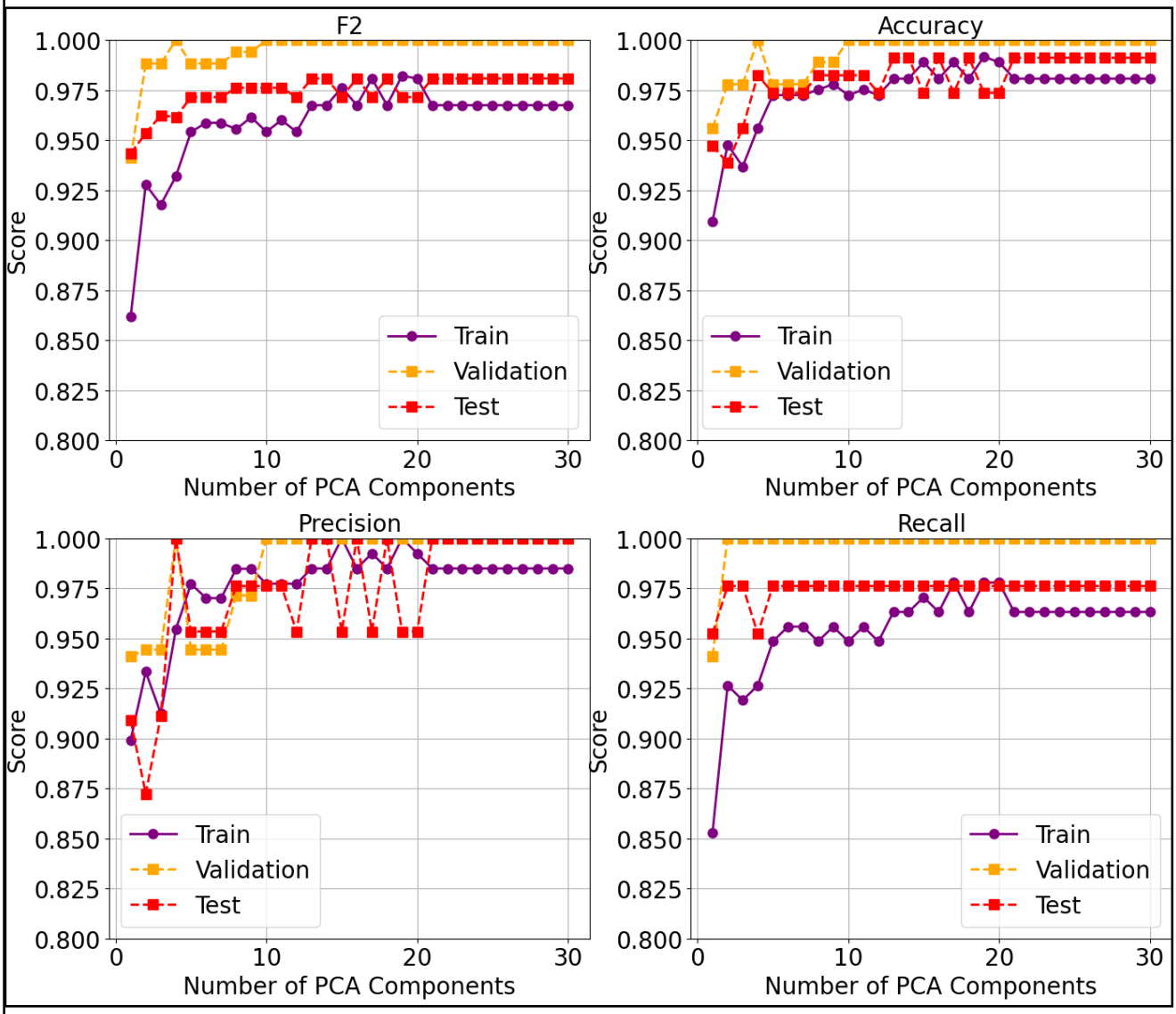


***Figure 3.*** *Performance of Logistic Regression model over PCA components*

dictionary. It creates a subplot for each evaluation metric (accuracy, recall, precision, F2-score) over the number of PCA-components. Each subplot contains the metrics for the training, validation, and test sets. The function rank_results then adds the four different result dictionaries, ranks all models performance based on F2-score, and puts them in a table. Thereby, these two functions shows the primary results of this study.

Two additional functions accompany this analysis. First, decision_boundary is build on the best models for 2 PCA components and plots all four models decision boundaries (see figure 7) with the data of the test set. It uses the DecisionBoundaryDisplay from scikit-learn. A second function then plots the cumulative explained variance of the PCA-components.

# 3. Results

An initial exploration of the data showed extremely high linear correlations of the target with many features. 15 features have a linear correlation with ‘y’ higher than 0.5. 8 features have a linear correlation with ‘y’ higher than 0.7. This further underscores the importance of dimensionality reduction.

Figures 3 to 6 show the Logistic Regression model’s, SVM’s, random forest’s, and MLP’s performance over all possible number of PCA-components. See Appendix B for the best performing models. In general, all models perform very well. The top 26 models perform equally well and have an F2-Score of 0.981. Furthermore these 26 models have an accuracy of 0.991 as well as perfect precision of 1. This results in a false positive rate of 0. The 60 best models all find 97.6% (recall) of malignant cancers in the test set, meaning

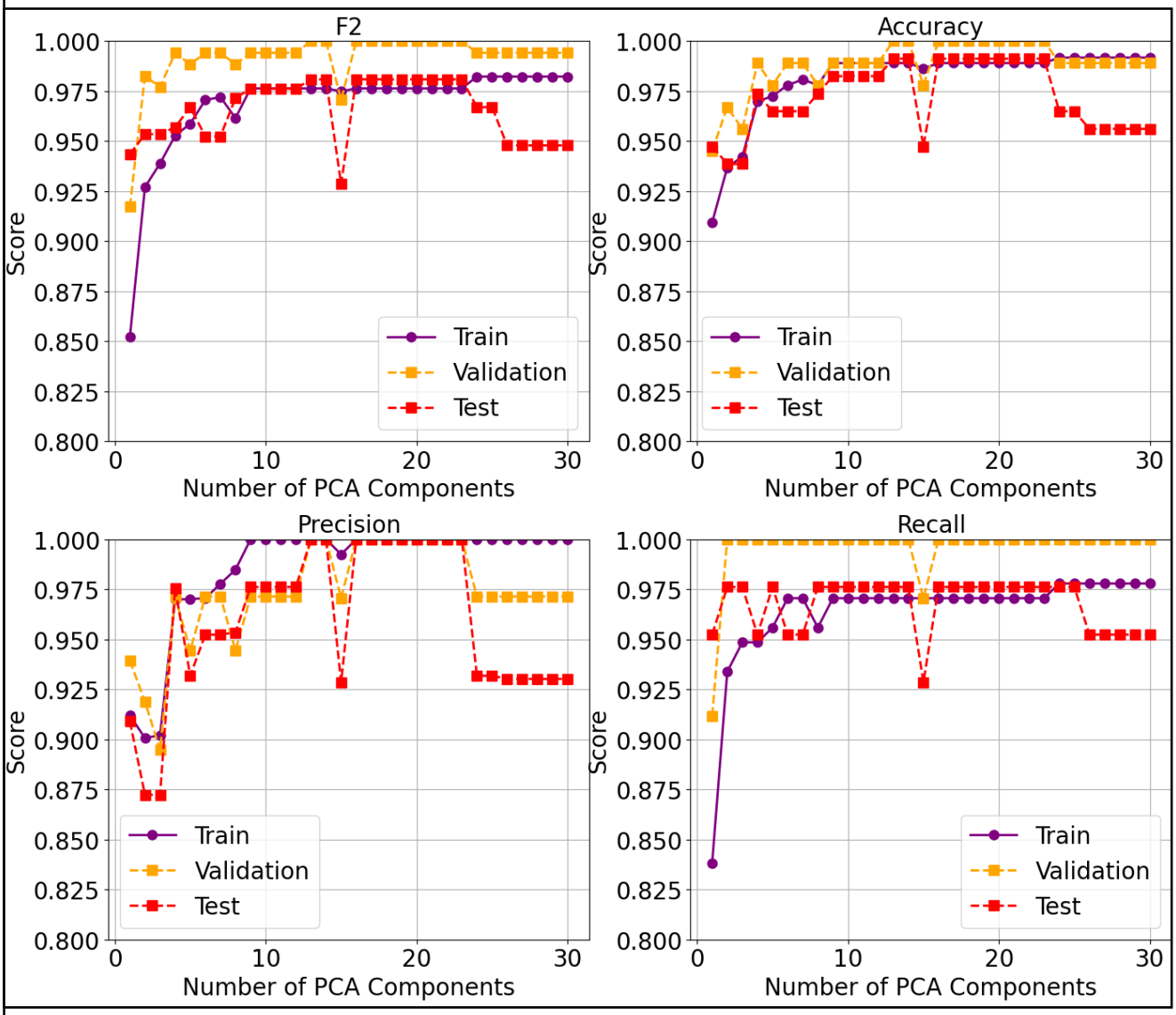


***Figure 4.*** *Performance of SVM over PCA components*

they have found all but one. Figure 7 visualises the decision boundaries of all 4 models under 2 PCA-components, and shows well how one malignant cancer in the test set is very difficult to classify.

Overall, logistic regression performed the best, closely followed by SVM and MLP. The random forest classifier performs the worst. Logistic regression, SVM, and MLP all perform better with only one PCA-component (F2: 0.943), than most random forest models.

Most of the best performing models operate with a higher number of PCA-components (>10). But at more than around 10 PCA-components, there seems to be very little to no systematic change in performance. This is aligned with the results of the a cumulative explained variance analysis, which shows that 9 PCA-components capture 95% of the variance in the data set (Figure 8.).

## Logistic Regression

The logistic regression models overall perform the best out of all four different classifiers. Almost half (14/30) of the top 30 performing models are logistic regression models (F2: 0.981, see Appendix B). The best performing models are at 13, 14, 16, 18, and 21-30 PCA Components. The performance of logistic regression increases with the number of PCA components, both in raw performance, as well as in consistency. While with over 10 PCA-components the logistic regression models sometimes achieves its best results, it only does so above 20 PCA-components for every number of PCA-components (Figure 3.).

## Support Vector Machine (SVM)

The performance of SVM, while still very high, seems to show diminishing returns with the number of PCA-components (Figure 4.). First, it increases as the dimensions rise, then it plateaus at around 13 PCA-components, until

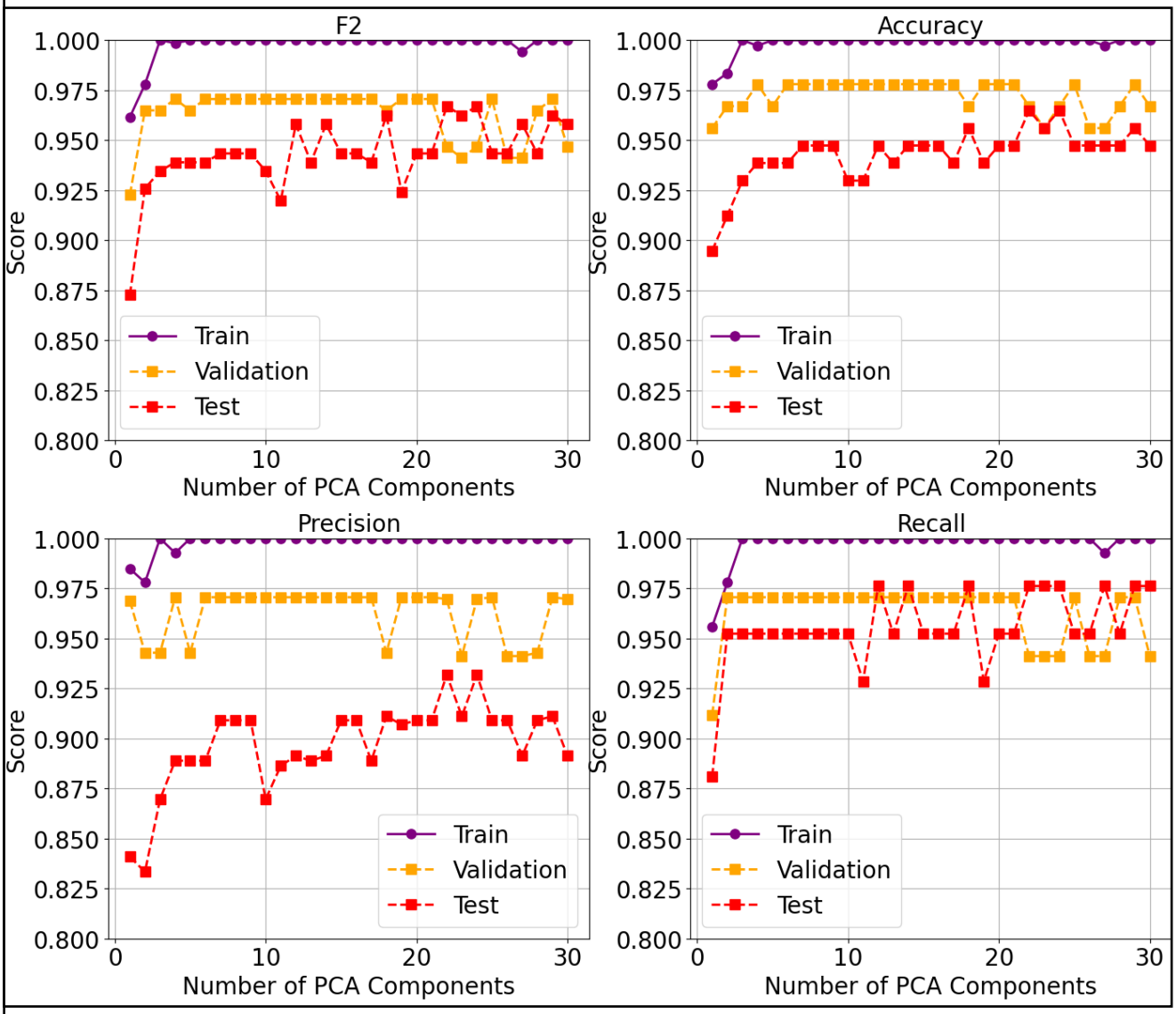


***Figure 5.*** *Performance of Random Forest classifier over PCA components*

it falls off again after 23 components. The highest performing SVM models with an F2-score of 0.981 are at 13-14, and 16-23 PCA-components.

### Random Forest Classifier

The random forest classifiers performed the worst. 19 of the worst performing 25 models were random forests. This is likely due to overfitting, as the models performed almost perfectly on the training set, but relatively poorly on the validation and test sets (Figure 5.). The model's overfitting can be nicely seen in its complicated 2D decision boundary (Figure 7.). On the test set the random forests performance tends to increase with the number of PCA-components. On the test set it achieved its highest F2-score of 0.967 with 22 and 24 components. While these two F2 scores are slightly lower than that of the other models, their recall is equal to the best performing models at 0.976.

### Multi-layer Perceptron (MLP) classifier

The MLP classifier had the most varied results. Its performance on the test set rises around 5-12 components, then drops, and then rises again between 23-30 components (Figure 6.). The best performing MLP models trained on 23 and 25 PCA-components with an F2-score of 0.981, while MLP on 6, 9, and 12 components achieved a high F2 of 0.976.

## 4. Discussion

### Limitations due to Data and Scope

The data set is so small with only 569 instances that variations in performance cannot be excluded to be due to the initial random seed in data splitting. In this study for example, all models consistently perform better on the validation set than on the test set, even though there has been no fine-tuning on the validation set. This difference

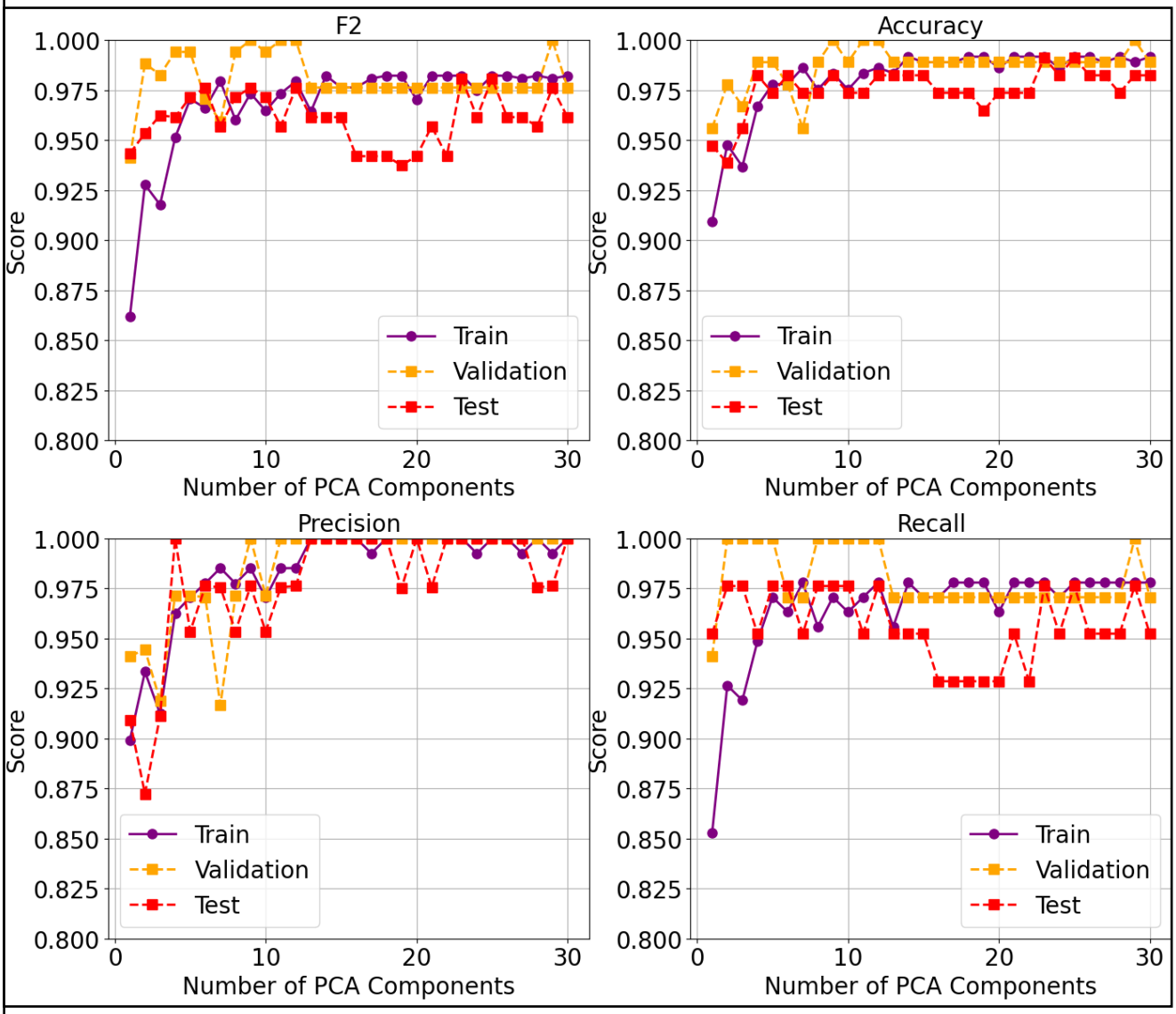


***Figure 6.*** *Performance of MLP-classifier over PCA components*

can only be explained by random effects, due to the size of the data set. These effects are further amplified by the model's general high performance. In these cases just one outlier malignant tumour can skew results. K-fold validation would have mitigated the random effects on the validation set, but not on the test set. A perfect predictive performance (such as Esen et al. (2024)) could be explained by a lucky random seed, in which there are only usual malignant tumours in the test set. All this is a limitation on the significance of the results of this study, especially when it comes to marginal improvements.

A possible solution would be to run the entire experiment of this study 100 or even 1000 times with different random seeds and average the results. This would show smoother trends and give more statistically significant results. Unfortunately this goes beyond the scope of this study. Only afterwards can be decided which model and number of PCA components is best for clinical use.

### High Logistic Regression Performance

In this data where many features are highly linearly correlated with the target, the first few principle components capture large parts of the variance. Here, the transformed data becomes nearly linearly separable. The linear nature of logistic regression benefits greatly from the improved linear separability provided by PCA. Its simplicity also protects it from overfitting. These two aspects might explain why logistic regression consistently performed the best over a range of numbers of PCA-components.

### Expected SVM Performance

With very few PCA components, all models tend to underfit as they lack sufficient information. As the number of components

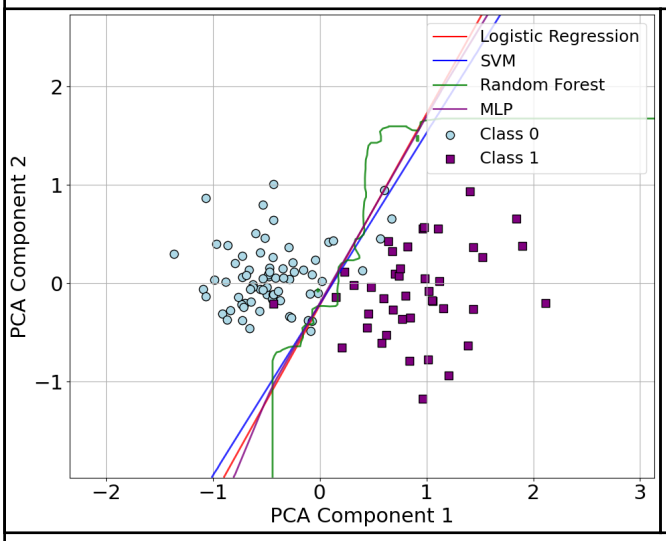


***Figure 7.*** *Decision Boundaries at 2 PCA-components learned on training set. Here displayed with data points from test set.*

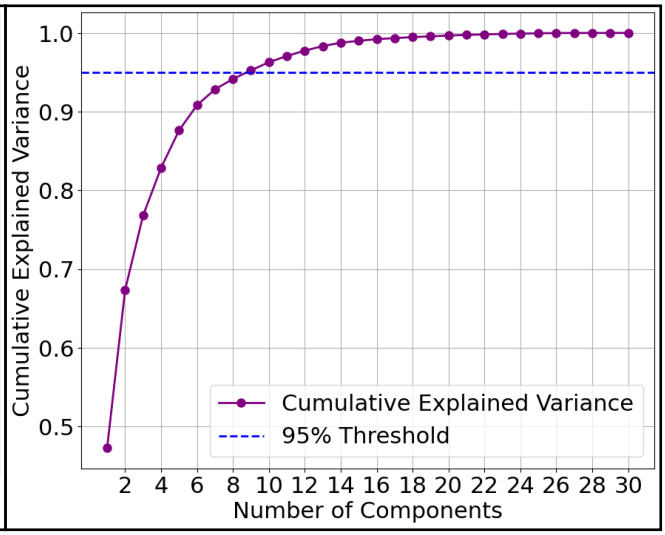


***Figure 8.*** *Cumulative explaining variance of PCA-components for training set*

increases, performance improves. But adding extra components yields diminishing returns. After a certain point, it may even introduce noise, which can slightly impede performance. The results for the small vector machine (SVM) are the results that best show this expected distribution of performance (Figure 4.). Such a curve can also be seen, for example, by the logistic regression on the training set (Figure 3.). Most performance distributions show hints of this type of diminishing returns distribution.

**Low Random Forest Performance**

Usually random forests are robust to overfitting. However, PCA transforms the data into a set of linear combinations, which could make individual trees more similar. Thereby it is weakening the variance reduction benefit of averaging over many trees, which may lead to overfitting.

**Varied MLP Performance**

MLPs are sensitive to the dimensionality of the input data. With very few PCA components, the network may underfit as it lacks enough information to model the complexity of the problem. Moreover, the optimal network architecture, found through grid search, can change substantially with different input dimensionalities, resulting in more varied performance across different numbers of PCA components than for other models.

### Effectiveness of PCA in Breast Cancer malignancy prediction

Principle component analysis has in this study show itself to be effective in improving the performance of logistic regression models as well as SVM models in predicting the malignancy of tumours. It has not shown consistent improvements for MLP classifiers and it may have contributed to poor performance of random forest classifiers.

The performance of 26 here trained classifiers in predicting the malignancy of breast cancer tumours is very high. They predict between 97-98% of all malignant tumours. They do so while having perfect precision and keeping the false positive rate on the test set at 0.

Despite this high performance, the models still miss about 2-3% of malignant cancers. That is why they should never be used on their own for breast cancer diagnosis. They should rather compliment a physician and flag potentially malignant tumours. If either the physician or the model suspects a malignancy, the tumour should be further investigated. Thereby the here generated models constitute a great diagnostic support system.

## 5. Conclusion

This study systematically analysed how dimensionality reduction using Principal Component Analysis (PCA) affects machine learning classifiers prediction of breast cancer malignancy. It trained and evaluated Logistic Regression, Support Vector Machine (SVM), Random Forest, and Multi-layer Perceptron (MLP) classifiers on all possible PCA-components (1–30.

Overall, dimensionality reduction via PCA enhanced the predictive performance for logistic regression and SVM classifiers.

Logistic regression models performed best overall, as it may have benefited from PCA due to increased linear separability. SVM performance followed the expected trend of diminishing returns as dimensionality increases. Conversely, PCA may have negatively impacted Random Forest performance, and caused overfitting. MLP performance highly varied across PCA dimensions, which shows the model's sensitivity to dimensionality changes.

Many here trained models demonstrated high predictive ability, consistently achieving F2-scores of 0.981, recall of 97–98%, 99% accuracy, perfect precision and zero false-positive rates. However, no model detected all malignant tumours. This underscores that these classifiers should only be used to supplement clinical diagnosis and flagging missed malignancies.

The results were highly sensitive to the initial random seed at the splitting of the data due to the small size of the dataset (569 instances). This limits the statistical significance and generalisability of these findings. In the future studies should repeat this experiment numerous times (e.g., 1000 repetitions) with different random seeds and average the results. Moreover, this study should also be replicated on a larger dataset. Only then can be decided which model and number of PCA components is best for clinical use.

# References


1. Wolberg, W., Mangasarian, O., Street, N., & Street, W. (1993). Breast Cancer Wisconsin (Diagnostic) [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5DW2B.

2. Esen, G., Altaibek, A., Amankulov, J., Matkerim, B., & Nurtas, M. (2024). Enhancing breast cancer detection with dimensionality reduction techniques: A study using PCA and LDA on Wisconsin breast cancer data. Procedia Computer Science, 251(C), 414–421. https://doi.org/10.1016/j.procs.2024.11.128

3. Jamal, A., Handayani, A., Septiandri, A. A., Ripmiatin, E., & Effendi, Y. (2018). Dimensionality reduction using PCA and k-means clustering for breast cancer prediction. Lontar Komput. J. Ilm. Teknol. Inf, 9(3), 192–201.

4. Ibrahim, S., Nazir, S., & Velastin, S. A. (2021). Feature selection using correlation analysis and principal component analysis for accurate breast cancer diagnosis. Journal of Imaging, 7(11), 225. https://doi.org/10.3390/jimaging7110225

5. Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, *374*(2065), 20150202.

6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

7. Macgregor, P. (2024). ID5059 - Knowledge Discovery and Data Mining. 07: Dimensionality Reduction


# Appendix A: Hyperparameters

**Random Seed Train-, val-, test-split**
5302

**Logistic Regression**
model__C: [0.1, 1, 10, 100]

**SVM**
model__C: [0.1, 1, 10, 100]
model__kernel: ['linear','rbf']
model__gamma: ['scale', 'auto']

**Random Forest**
random_state=1
model__n_estimators: [10, 25, 50, 100]
model__max_depth: [None, 5, 10, 20]
model__min_samples_split: [2, 5, 10]

**MLPClassifier**
random_state=1
max_iter=3000, early_stopping=False)
model__hidden_layer_sizes: [(5,), (10,), (20,), (10, 10)]
model__activation: ['relu', 'tanh']
model__alpha: [0.001, 0.01, 0.1]
model__learning_rate_init: [0.001, 0.005, 0.01]

# Appendix B: Top Results

| Rank | Model | PCA-Comp. | F2-Score | Recall | Accuracy | Precision |
|---|---|---|---|---|---|---|
| 1 | Log. Regression | 30 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 2 | Log. Regression | 14 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 3 | MLP | 23 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 4 | MLP | 25 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 5 | SVM | 21 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 6 | SVM | 20 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 7 | SVM | 19 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 8 | Log. Regression | 29 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 9 | SVM | 18 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 10 | SVM | 17 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 11 | SVM | 16 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 12 | SVM | 14 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 13 | SVM | 22 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 14 | SVM | 13 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 15 | Log. Regression | 13 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 16 | SVM | 23 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 17 | Log. Regression | 22 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 18 | Log. Regression | 23 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 19 | Log. Regression | 28 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 20 | Log. Regression | 18 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 21 | Log. Regression | 27 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 22 | Log. Regression | 26 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 23 | Log. Regression | 25 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 24 | Log. Regression | 24 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 25 | Log. Regression | 21 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 26 | Log. Regression | 16 | 98.100 % | 97.600 % | 99.122 % | 100.000 % |
| 27 | MLP | 9 | 97.600 % | 97.600 % | 98.245 % | 97.619 % |
| 28 | MLP | 6 | 97.600 % | 97.600 % | 98.245 % | 97.619 % |
| 29 | SVM | 12 | 97.600 % | 97.600 % | 98.245 % | 97.619 % |
| 30 | MLP | 29 | 97.600 % | 97.600 % | 98.245 % | 97.619 % |