A Survey of the Effects of PCA on Classifier Performance

Alexander Cooper

Abstract

PCA is an important and useful tool for reducing complexity of machine learning models and for data visualization. It is used to project data down into a lower dimension while preserving the most important characteristics. But how much does PCA effect a machine learning classifier's ability to solve a classification problem?

1 Introduction

Principle Component Analysis (PCA) is a popular method for decreasing the dimensionality of data for visualization or for training machine learning models Zhao et al. [2022]. Reducing the dimensions of a dataset for machine learning is desirable because it decreases the complexity of the model, allowing for more confident and computationally efficient predictions. This survey aims to uncover which classification models are most robust while using PCA to reduce a feature space to one of lower dimensions. Four classification models are compared: Decision Tree, K Nearest Neighbors, Perceptron, Logistic Regression. It is important to mention that Support Vector Machine was also meant to be included in this study, but unfortunatly it took too long to train on our dataset and we had to drop it to meet time constraints. We hypothesize that K Nearest Neighbors (KNN) will benefit the most from PCA, as it is well known that performance of KNN drops in high dimensions, and Decision Tree will benefit the least from PCA, as such low dimensions will not allow a Decision Tree to make enough decisions to predict accurately. The code for the experimentation is stored in a GitHub repo [1].

2 Background

The dataset used to train and evaluate the models is the Diabetes Health Indicators Dataset which is a consolidated and cleaned version of CDC's BRFSS 2015 dataset for Disease Control and Prevention. The reason why this dataset was chosen is because many problems in the medical sector of machine learning suffer from the curse of dimensionality, and PCA is often used to mitigate the issue Verleysen and François [2005]. This dataset has three files, one is imbalanced with 3 classes (0 for no diabetes, 1 for prediabetes, and 2 for diabetes), 21 features, and over 250,000 samples. The second is balanced with 2 classes (0 for no diabetes and 1 for prediabetes or diabetes), 21 features, and over 70,000 samples. The third is similar to the first, but combines prediabetes and diabetes into one class like the second. This report uses the second dataset to keep things simple for the classificationn task, as it has less samples, is a binary classification problem, and has balanced classes.

Another reason why this dataset was chosen was because it has 21 features. Since we are measuring the effects of PCA, we need a dataset with a large number of dimensions to observe the performance of the models on a wide range of dimensionality.

An additional plus for this dataset, which will be explored in more detail in the results section, is that this dataset is not linearly separable, so it is easier to see how the models perform on the classification

[1] `https://github.com/oberon-uofa/CSC580_FinalProject`

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

task in each dimension, as the models are not able to predict the correct label very accurately and so their performance changes a lot as dimensionality is reduced.

3 Methodology

3.1 Applying PCA to the dataset

First, the dataset is split into the standard 80% training data and 20% test data. We do this before the PCA splits so that for each dimension, each model is trained and tested on the same data to avoid deviations in the model performance that do not come from the true independent variable. Then the class labels are pulled out of the data so they are not included in the PCA. The Python package scikit-learn data preprocessing packages are used for the train test split and for PCA Pedregosa et al. [2011].

Then we create a new dataset for each dimension from 1 to 21, applying PCA to the original dataset each time to produce a new data in each lower dimension. This process is done for both the train and test set, as the same dimension of data that was used to train the model must be used to test the model.

Next we interpret the effects of PCA on the dataset by plotting the total explained variance for each dimension, as well as plotting the data produced from using PCA to reduce to two dimensions to see how the data is distributed.

3.2 Training the models on each dimension

Four classification models are trained on each PCA split of the dataset: Decision Tree, K Nearest Neighbors, Perceptron, Logistic Regression. The scikit-learn implementations of the models are used Pedregosa et al. [2011]. Each of the models have a hyperparameter that is tuned using 5-fold cross validation. Decision Tree has max depth (the max decisions to make before making a prediction), K Nearest Neighbors has k (the number of neighbors to consider), Perceptron has α (the learning rate), and Logistic Regression has C (regularization strength).

The models are trained on each dimension of the dataset, then predict class labels on the test set of the same dimension. The predictions are stored for measuring model performance.

3.3 Measuring the model's performance

Two scoring methods are used to analyze the models performance. Zero-one loss is used because the dataset is binary and because it is easy to understand and reason about. It shows exactly how right each classifier is when testing, because the model either predicted right or wrong. F1-score is used because it is a straightforward way to measure the accuracy of a model beyond it's loss. It is expected that loss and f1-score would follow an inverse trend, when loss increases, f1-score is expected to decrease.

4 Results

4.1 Interpreting the effects of PCA

Figure 1 shows the explained variance of each dataset after applying PCA. Around 5 dimensions the explained variance starts to drop, suggesting that there are 5 dimensions out of the original 21 that hold most of the relevant information for this classification task. After observing these results, we expect that in the model performance phase, we should see a drastic drop in performance around 5 dimensions, mirroring this plot.

Figure 2 shows the scatter plot of the data after using PCA to reduce to two dimensions. This plot shows that the data is not linearly separable and thus is a good dataset to compare classifiers with. It is obvious that two dimensions does not represent the variance in the data very well as this figure and Figure 1 1 show.

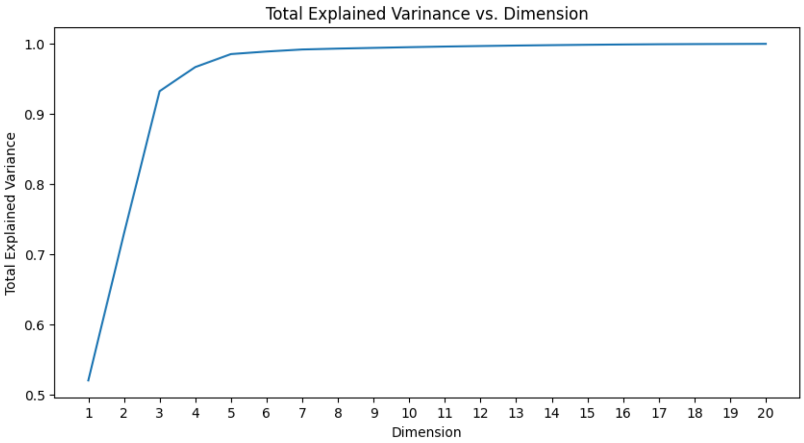

Figure 1: The total explained variation of the data after PCA is applied to reduce to each dimension.

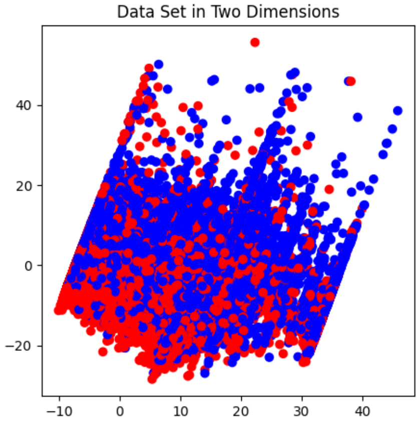

Figure 2: Visualizing the data after reducing to two dimensions with PCA.

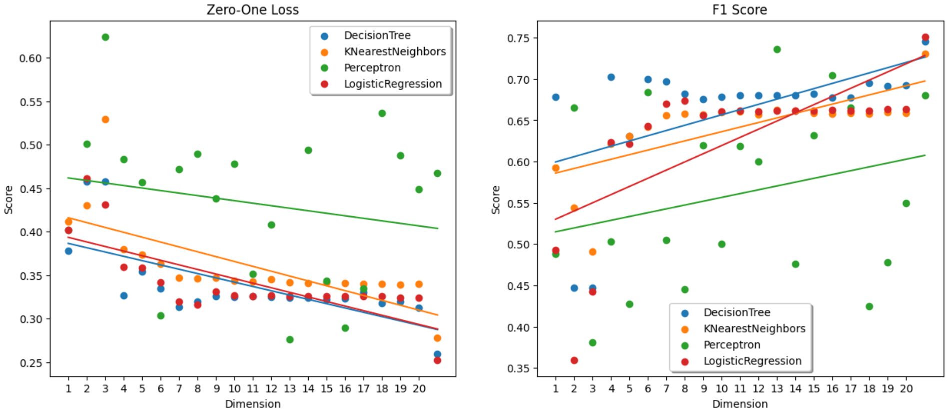

Figure 3: Comparing the performance of each model in each dimension.

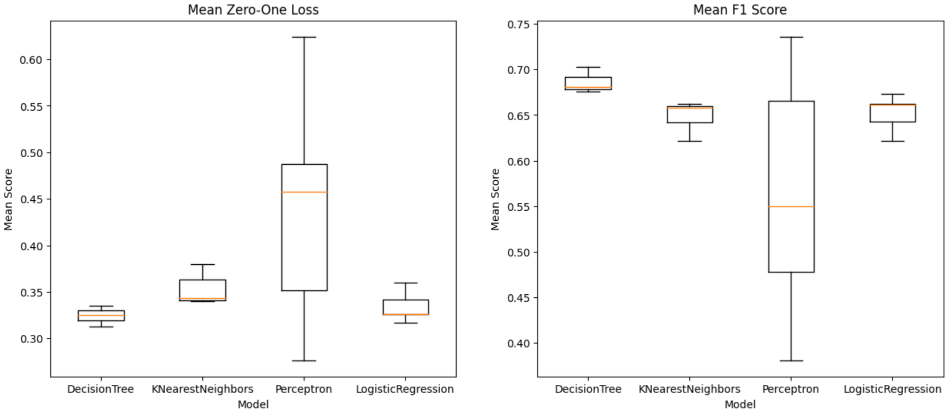

Figure 4: Mean performance of each model across dimensions.

4.2 Analyzing the model's performance

Figure 3 shows the results from training the models in each dimension and measuring the zero-one loss and f1 score. As might be expected, around 5 dimensions the zero-one loss starts increasing and f1 score starts dropping. The lines in the plot are lines that have been fit to each model's scatter plot. Decision Tree consistently has the lowest loss and highest f1-score. Logistic Regression's loss is very close to Decision Tree but it's f1-score is lower for lower dimensions. K Nearest Neighbors has a lower loss than Decision Tree and Logistic Regression, but it's f1-score has a less steep slope from Logistic Regression, suggesting that it is slightly more robust to PCA than Logistic Regression, but Logistic Regression still performs better. Perceptron performs the worst across all dimensions.

Figure 4 shows the box plot of zero-one loss and f1-score for each model across dimensions. This plot shows that Decision Tree has both the lowest mean zero-one loss and the highest mean f1-score, as well as the lowest variance. Overall Decision Tree was the most robust model to PCA. K Nearest Neighbors and Logistic Regression are about equal, and just slightly worse than Decision Tree. Perceptron was the least robust to PCA, as the plot shows it's mean zero-one loss and f1-score were by far the highest/lowest respectively, and Perceptron's variance was much higher than the other models. As figure 3 shows, Perceptron's scores are all over the place, we believe that this is because of Perceptron's random weight initialization that then leads to a very different model each time it is trained.

Figure 3 and figure 4 together do meet our expectations after observing figure 1.

We conclude that out of the four models surveyed in this study, Decision Tree is the best classifier to use when using PCA to reduce dimensionality of a dataset. It is the most accurate and robust of the four models.

The results of the experimentation did not support our hypothesis, as we predicted that Decision Tree would perform the worst. Some potential reasons why Decision Tree performed so unexpectedly well is because it is a nonlinear model and can fit the nonlinear data better than Perceptron or Logistic Regression. K Nearest Neighbors is also a nonlinear model, and performs almost as well as Decision Tree. It's possible that the polygonal nature of Decision Tree's decision boundaries are better at classifying this specific dataset than K Nearest Neighbor's spherical decision boundaries.

5 Related Work

Saringat et al. [2019] compares many of the models we compare in this study, and although it does not focus only on the models performance with PCA, it was a good guide for our comparisons, and we used their idea to use accuracy, precision, and recall as scoring metrics to guide our decision to use f1-score as a scoring metric when comparing models.

Gárate-Escamila et al. [2020] uses PCA and a couple other dimension reducing algorithms and compares them. Uses a confusion matrix an f1 score to evaluate. We also used their idea to use f1-score as a scoring metric to influence our decision to use it.

Bruni et al. [2022] is about using Minimum Description Length (MDL) to perform model selection. It also shows how MDL can be used to increase the usefulness of PCA in image classification, and provides some methods of interpreting principle components which influenced our methods doing the same.

6 Limitations

A major limitation of this study is that only one dataset was used to compare the models. Using multiple datasets would yield more statistically sound results, and would be of major benefit.

Another limitation is that only four classifiers were compared. Adding more classifiers to the study would increase it's usefulness to machine learning engineers and potentially give better insight into the benefits or limitations of using PCA to reduce dimensionality of a training dataset.

References

Vittoria Bruni, Maria Lucia Cardinali, and Domenico Vitulano. A short review on minimum description length: An application to dimension reduction in pca. *Entropy*, 24(2), 2022. ISSN 1099-4300. doi: 10.3390/e24020269. URL `https://www.mdpi.com/1099-4300/24/2/269`.

Centers for Disease Control and Prevention. Behavioral risk factor surveillance system. https://www.kaggle.com/datasets/cdc/behavioral-risk-factor-surveillance-system.

Anna Karen Gárate-Escamila, Amir Hajjam El Hassani, and Emmanuel Andrès. Classification models for heart disease prediction using feature selection and pca. *Informatics in Medicine Unlocked*, 19:100330, 2020. ISSN 2352-9148. doi: https://doi.org/10.1016/j.imu.2020.100330. URL `https://www.sciencedirect.com/science/article/pii/S2352914820300125`.

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

Zainuri Saringat, Aida Mustapha, R. D. Rohmat Saedudin, and Noor Samsudin. Comparative analysis of classification algorithms for chronic kidney disease diagnosis. *Bulletin of Electrical Engineering and Informatics*, 8(4):1496–1501, 2019. ISSN 2302-9285. doi: 10.11591/eei.v8i4.1621. URL `https://www.beei.org/index.php/EEI/article/view/1621`.

Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In Joan Cabestany, Alberto Prieto, and Francisco Sandoval, editors, *Computational Intelligence and Bioinspired Systems*, pages 758–770, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-32106-4.

Baiting Zhao, Xiao Dong, Yongcun Guo, Xiaofen Jia, and Yourui Huang. PCA dimensionality reduction method for image classification. *Neural Process. Lett.*, 54(1):347–368, February 2022.