Dimensionality Reduction and Analysis

Name  
*Institute*  
Email

# **Introduction**

Dimensionality Reduction is simply defined as ‘its process of changing high-dimension data into a low dimension without losing the features of original data” [1]. One of the major objectives of dimensionality reduction is to reduce the computation time of processing. It also increases the number of features for a machine learning model which helps us to generate accurate results. High-dimension data also cause overfit problems. Reducing dimensionality can address these issues by simplifying the model and enhancing its overall generalization performance. Two primary methods for dimensionality reduction include feature selection and feature extraction.

# **DATASET**

In our framework, we use the Breast Cancer Wisconsin (Diagnostic) Data Set [4] in which the basic task is to classify the cancer as benign or malignant. The features in this dataset are derived from a digitized image of a fine needle aspirate (FNA) of a breast mass, specifically characterizing the attributes of cell nuclei present in the image. The 3-dimensional space used for description follows the model. The dataset is accessible through the UW CS ftp server and the UCI Machine Learning Repository. The attribute information includes an ID number, diagnosis (M = malignant, B = benign), and ten real-valued features for each cell nucleus, such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. Each image underwent computation of mean, standard error, and "worst" (the largest among the three largest values) across 30 features. All feature values are documented with four significant digits, and no attribute values are missing. The class distribution reveals 357 instances of benign and 212 instances of malignant cases.

# Pre-processing

First of all, we check the frequency distribution [2] of the dataset. We analyze that the malignant class has 357 instances and the benign class has 212. In proportional to target analysis it’s important to note that the malignant class has 62.74% while the benign has 37.26 as illustrated in Figure 1.

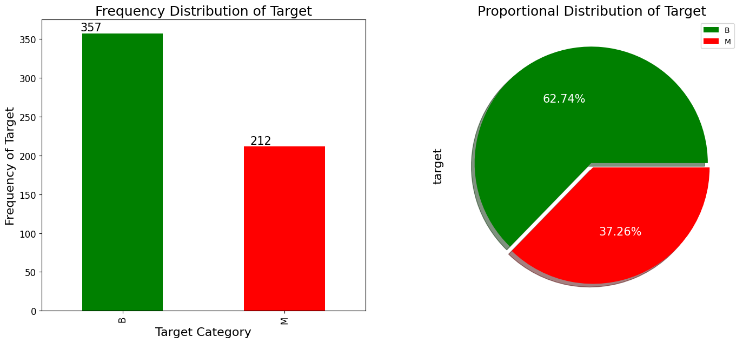


Figure 1: Frequency Distribution

We then normalize the data and used to describe the process of converting numerical data into a standardized scale in the context of data analysis and machine learning. Putting disparate characteristics or variables on a comparable scale allows for more accurate comparisons and enhances the efficiency of specific algorithms. In dataset split is to facilitate the training of a model on one subset of data and subsequently assess its efficacy on an entirely distinct subset that it has not encountered during the training phase. The widely adopted practice involves a partition, commonly expressed as a 70/30 or 80/20 split, where the larger segment serves as the training set, and the smaller segment is reserved for evaluating the model's performance. The training set plays a pivotal role in honing the model's parameters and features to the provided data, allowing it to learn patterns and relationships. In this context, a deliberate decision has been made to execute an 80/20 split, 80% of the dataset is for training purposes and the rest is 20%.

In classification evaluation, common metrics include accuracy (overall correctness), precision (quality of positive predictions), recall (coverage of positive instances), and F1 score (harmonic mean of precision and recall). The classification report summarizes these metrics for each class, providing insights into model performance. The confusion matrix breaks down true positives, false positives, true negatives, and false negatives, offering a detailed view of the model's accuracy and errors.

# Results Without Dimensionality Reduction

We emulate our dataset without dimensionally reduction techniques with different classifiers in which The Logistic Regression model [3] achieved an accuracy of 96.49%, with balanced precision and recall at 95.74%. Support Vector Classifier (SVC) [5] demonstrated a slightly higher accuracy of 97.37%, with comparable precision and an impressive recall of 97.83%. The K-Neighbors Classifier [6] exhibited a slightly lower accuracy of 95.61%, with a precision of 89.36% and a perfect recall of 100.00%. The MLP Classifier [7] achieved a notable accuracy of 97.37%, with high precision (97.87%) and respectable recall (95.83%). Linear SVC, while achieving an accuracy of 94.74%, maintained a good balance between precision (95.74%) and recall (91.84%). The Random Forest [8] Classifier matched the performance of MLP with an accuracy of 97.37%, precision of 97.87%, and recall of 95.83%. It's noteworthy that the training times varied, with the MLP Classifier having the longest training time at 1.51 seconds. The confusion matrix of Logistic Regression and SVC is shown in Figure 2.

The overall table of results is also illustrated below in Table 1.

Table 1: Results on different classifiers without dimensional reduction

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1** | **Time** | **Technique** |
| **Logistic Regression** | 96.46 | 95.74 | 95.74 | 95.74 | 0.043034 | All Features |
| **SVC** | 97.37 | 95.74 | 97.83 | 96.77 | 0.008203 | All Features |
| **K-neighbors** | 95.61 | 89.36 | 100 | 94.38 | 0.001170 | All Features |
| **MLP** | 97.37 | 97.87 | 95.83 | 96.84 | 1.508635 | All Features |
| **Linear SVC** | 94.74 | 95.74 | 91.84 | 93.75 | 0.022814 | All Features |
| **Random Forest** | 97.37 | 97.87 | 95.83 | 96.84 | 0.356251 | All Features |

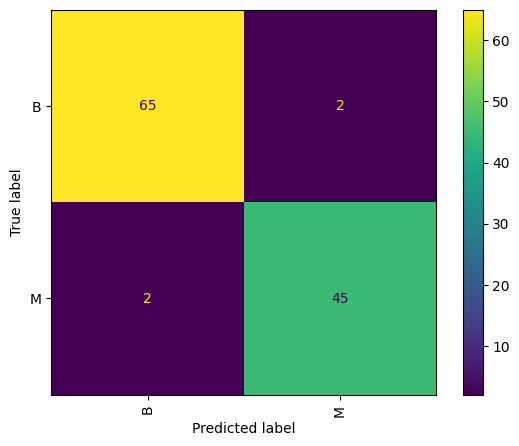
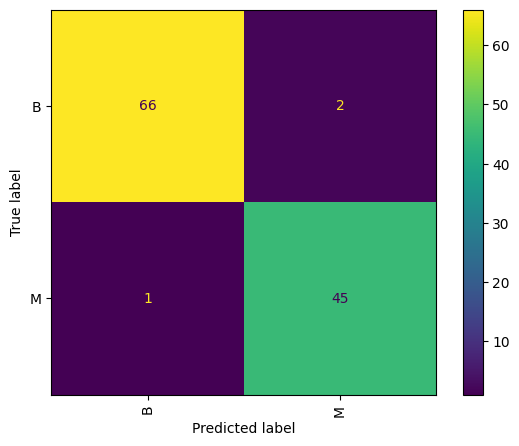
 

Figure 2: Confusion Matrix of Logistic Regression and SVC

# Principal Component Analysis (PCA)

Principal Component Analysis (PCA) [9] is a dimensionality reduction technique widely used in machine learning and data analysis. Its major goal is to transform a dataset into a new coordinate system, where the variance is maximized along the principal components, thereby extracting the most critical information in the data. In simple PCA identifies the directions, or principal components, in which the data varies the most. We project the original data onto these components, PCA allows for a more compact representation while retaining the essential features. This not only aids in visualizing high-dimensional data but also facilitates more efficient and streamlined analysis by focusing on the most influential aspects of the dataset.

After conducting a Principal Component Analysis (PCA) on the Wisconsin dataset, it was determined that a final selection of 9 principal components (features) is sufficient to capture 95% of the variance within the data. This finding underscores the effectiveness of PCA in achieving significant dimensionality reduction while preserving the essential information embedded in the dataset. With just 9 features, the reduced dataset retains a vast majority of the original dataset's variability, making it a practical and efficient choice for subsequent analysis and modeling. PCA Variance of different features is illustrated in Figure 3.

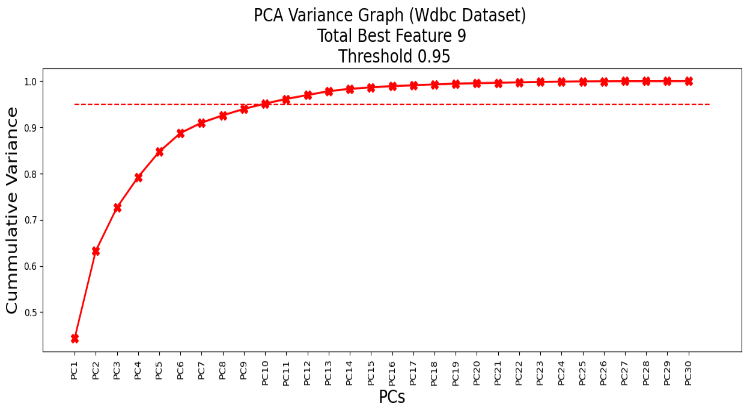


Figure 3: PCA Varience Graph

# Results With PCA

The result presented in this paragraph showcases the performance metrics of various machine learning models applied to a dataset, with a notable emphasis on the use of Principal Component Analysis (PCA) for dimensionality reduction. Table 2 presents the performance metrics of various machine learning models using Principal Component Analysis (PCA) on a dataset. Notably, the Support Vector Classifier (SVC) achieved the highest accuracy at 97.37%, along with strong precision, recall, and F1 scores, indicating its effectiveness in correctly classifying instances. The Multi-Layer Perceptron (MLP) Classifier also demonstrated competitive results, showcasing the robustness of neural network-based approaches. The choice of 9 principal components seems to offer a balanced trade-off between dimensionality reduction and model performance across different algorithms.

Table 2: Results of different classifiers with PCA

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1** | **Time** | **Technique** |
| **Logistic Regression** | 94.74 | 93.62 | 93.62 | 93.62 | 0.019762 | PCA |
| **SVC** | 97.37 | 95.74 | 97.83 | 96.77 | 0.008122 | PCA |
| **K-neighbors** | 95.61 | 89.36 | 100 | 94.38 | 0.003553 | PCA |
| **MLP** | 97.37 | 95.74 | 97.83 | 96.77 | 0.629184 | PCA |
| **Linear SVC** | 93.86 | 93.62 | 91.67 | 92.63 | 0.009651 | PCA |
| **Random Forest** | 93.86 | 89.36 | 95.45 | 92.31 | 0.209389 | PCA |

# The confusion matrix of Logistic regression and SVC is also illustrated below in Figure 4.

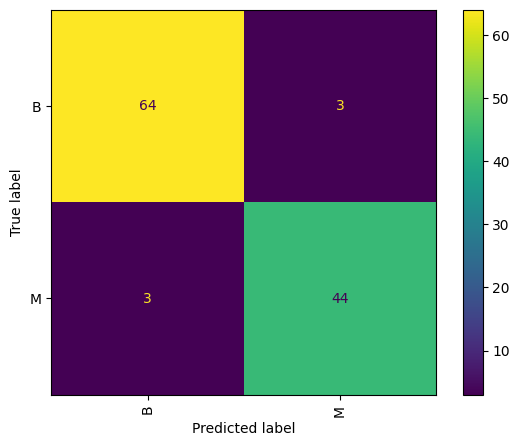
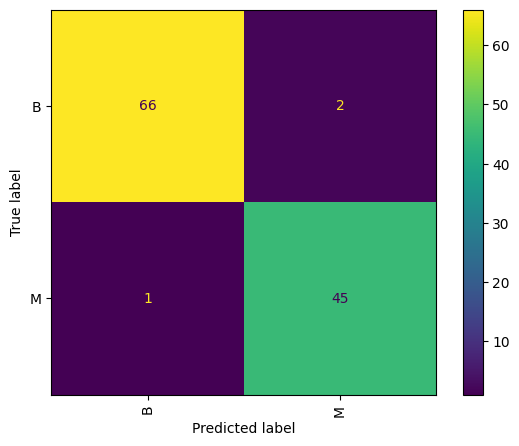
 

Figure 4: Confusion Matric of Logistic Regrsion and SVC with PCA

# Independent Component Analysis (ICA)

Independent Component Analysis (ICA) [10] is a computational technique used in signal processing and data analysis to separate a multivariate signal into additive, independent components. The fundamental idea behind ICA is to find a set of statistically independent components that, when linearly combined, give rise to the observed data. Unlike methods such as Principal Component Analysis (PCA), which focuses on capturing the maximum variance in the data, ICA aims to identify underlying sources that are statistically independent of each other.

On the Wisconsin dataset, the optimal choice for component selection is determined to be 10. This decision is supported by the observation that this number effectively captures the most non-Gaussian and independent features within the dataset. The emphasis on non-Gaussian features aligns with the goal of extracting meaningful information and reducing dimensionality in a manner that is conducive to subsequent analysis. This finding suggests that, based on kurtosis analysis, employing 10 components is a judicious approach (as shown in Figure 5) for enhancing the dataset's representational efficiency and supporting more focused and streamlined analytical processes.

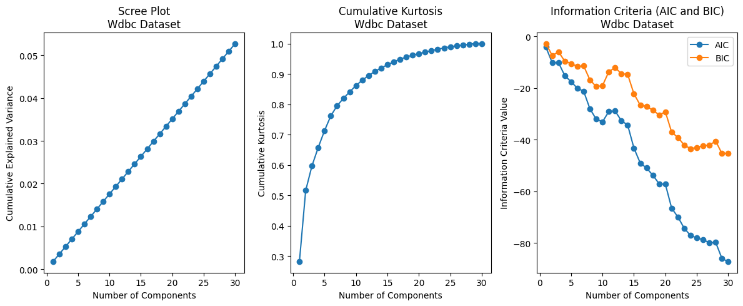


Figure 5: Graph of ICA on WDBC Dataset

# Results With ICA

In terms of ICA logistic regression surprisingly not achieve a good accuracy that is 61.40%. However other models in the table generate good accuracy. Support Vector Classifier (SVC) demonstrated the highest accuracy at 93.86%, emphasizing its effectiveness in making correct predictions. K Neighbors Classifier followed closely with an accuracy of 87.72%, showcasing its capability for inaccurate classification. MLP Classifier and Linear SVC both achieved high accuracies of 93.86%, positioning them as robust models for the given task. Random Forest Classifier also demonstrated strong performance with an accuracy of 92.11%. All results with ICA are shown in Table 3.

Table 3: Result of different classifiers with ICA

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1** | **Time** | **Technique** |
| **Logistic Regression** | 61.40 | 6.38 | 100.0 | 12.00 | 0.006219 | ICA |
| **SVC** | 93.86 | 89.36 | 95.45 | 92.31 | 0.013596 | ICA |
| **K-neighbors** | 87.72 | 74.47 | 94.59 | 83.33 | 0.002717 | ICA |
| **MLP** | 93.86 | 85.11 | 100.0 | 91.95 | 1.703456 | ICA |
| **Linear SVC** | 93.86 | 85.11 | 100.0 | 91.95 | 0.005539 | ICA |
| **Random Forest** | 92.11 | 82.98 | 97.50 | 89.66 | 0.875021 | ICA |

The confusion matrix of SVC and MLP is illustrated below in Figure 6.

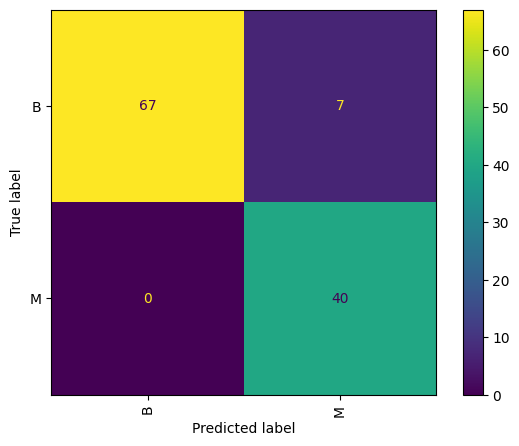
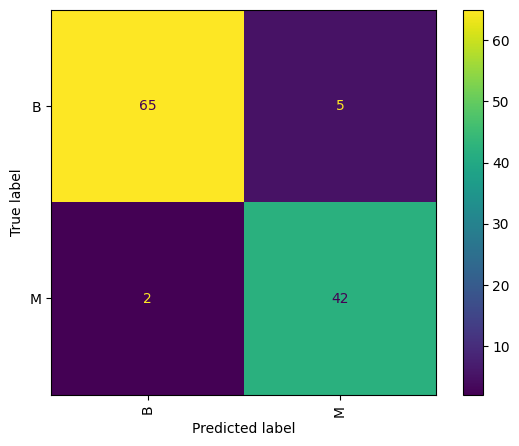


Figure 6: Confusion Matrix of ICA on SVC and MLP Classifier

# Randomized Projections (RP)

Randomized Projections (RP) [11] is a technique used in machine learning and signal processing for dimensionality reduction. The main idea behind RP is to project high-dimensional data into a lower-dimensional space using a random matrix. This random matrix is typically constructed by sampling entries from a probability distribution.

The key advantage of using randomized projections for dimensionality reduction is that it allows for efficient computation of approximations to certain linear algebra operations, such as matrix multiplication.

In the analysis of the Wisconsin dataset, focusing on the reconstruction error measured by Root Mean Squared Error (RMSE), it becomes evident that the optimal number of components is approximately 10. Beyond this point, the reconstruction error plateaus, suggesting that additional components cease to make a substantial contribution to enhancing the reconstruction. Hence, opting for 10 components emerges as a judicious choice for this particular dataset, striking a balance between capturing essential information and avoiding unnecessary complexity in the representation of the data. This observation underscores the importance of careful consideration and analysis when determining the appropriate number of components in dimensionality reduction techniques applied to specific datasets. Figure 6 shows the reconstruction error concerning several components.

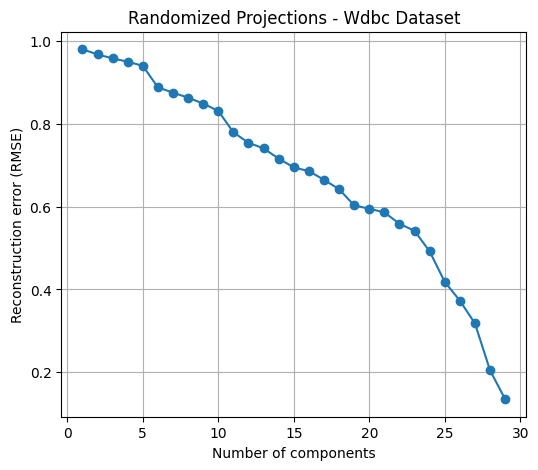


Figure 7: Reconstruction error of RP

# Results With RP

The Wisconsin dataset analysis employing Randomized Projections (RP) for dimensionality reduction reveals distinct model performances. Logistic Regression and Support Vector Classifier (SVC) both achieve 95.61% accuracy, with SVC demonstrating higher recall. K Neighbors Classifier, at 93.86% accuracy, excels in recall (97.62%) but with minimal training time. Multilayer Perceptron (MLP) Classifier shows balanced metrics at 94.74%, while Linear SVC boasts high precision (97.87%). Random Forest achieves 94.74% accuracy. RP proves a common effective preprocessing technique. The model choice depends on the specific application's requirements, considering both prediction accuracy and computational efficiency. A summary of the result for RP is given in Table 4.

Table 4: Summary of results with RP

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1** | **Time** | **Technique** |
| **Logistic Regression** | 95.61 | 95.74 | 93.75 | 94.74 | 0.014628 | RP |
| **SVC** | 95.61 | 93.62 | 95.65 | 94.62 | 0.008935 | RP |
| **K-neighbors** | 93.86 | 87.23 | 97.62 | 92.13 | 0.001889 | RP |
| **MLP** | 94.74 | 95.74 | 91.84 | 93.75 | 1.768658 | RP |
| **Linear SVC** | 95.61 | 97.87 | 92.00 | 94.85 | 0.008339 | RP |
| **Random Forest** | 94.74 | 91.49 | 95.56 | 93.48 | 0.810257 | RP |

# Linear Discriminant Analysis (LDA)

LDA is another dimensionality reduction [12] Like PCA, it involves projecting the data onto axes; however, its goal is to choose axes that maximize class separability and minimize intra-class scatter simultaneously. With a binary-class dataset, the data will be projected onto a line. With a n-class dataset, the data will be projected onto an (n-1)-dimensional-space.

# Results with LDA

Table 5 summarizes the performance of various machine learning models on a dataset with LDA. Logistic Regression and Linear SVC lead in accuracy at 95.61%, with balanced precision and recall. SVC follows closely with 95.61% accuracy, emphasizing recall. K-neighbors excels in recall (97.62%) but lag in precision (87.23%). MLP balances precision (95.74%) and recall (91.84%) at 94.74% accuracy. Random Forest achieves 94.74% accuracy, trading off precision (91.49%) and recall (95.56%). Execution times vary, with K-neighbors being the quickest (0.001889 seconds), Logistic Regression mid-range (0.014628 seconds), and MLP the longest (1.768658 seconds). The "RP" notation suggests preprocessing.

Table 5: Result summary of different classifiers with LDA

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1** | **Time** | **Technique** |
| **Logistic Regression** | 95.61 | 95.74 | 93.75 | 94.74 | 0.014628 | RP |
| **SVC** | 95.61 | 93.62 | 95.65 | 94.62 | 0.008935 | RP |
| **K-neighbors** | 93.86 | 87.23 | 97.62 | 92.13 | 0.001889 | RP |
| **MLP** | 94.74 | 95.74 | 91.84 | 93.75 | 1.768658 | RP |
| **Linear SVC** | 95.61 | 97.87 | 92.00 | 94.85 | 0.008339 | RP |
| **Random Forest** | 94.74 | 91.49 | 95.56 | 93.48 | 0.810257 | RP |

# Comparative analysis

Based on the provided data, let's discuss which reduction technique gives better results. In the baseline scenario without any dimensionality reduction, models utilizing all features demonstrate high accuracy and precision, albeit with varying training times. Principal Component Analysis (PCA) effectively reduces dimensionality by transforming data into uncorrelated principal components, maintaining high performance similar to the All Features approach while reducing training time. Independent Component Analysis (ICA), which separates data into statistically independent subcomponents, exhibits lower accuracy and precision, potentially influenced by dataset characteristics and ICA's independence assumptions. Random Projection (RP) yields mixed results, with some models matching All Features performance and others experiencing a slight decrease. RP, however, offers significantly lower training times, enhancing computational efficiency. Linear Discriminant Analysis (LDA), focused on maximizing class separation, consistently performs well with high accuracy and precision akin to the All Features approach, coupled with relatively low training times.

A complete graph of all techniques is shown below in Figure 7 indicating that LDA gets the highest accuracy with all classifiers as compared to others.

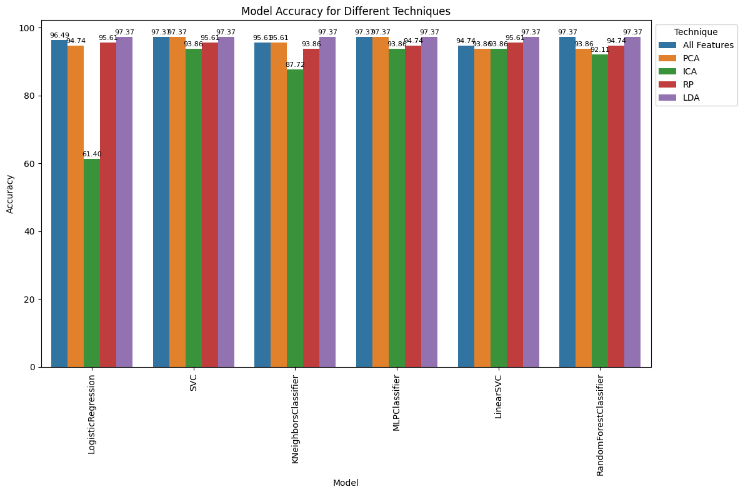


Figure 8: Model Accuracy for Different Techniques

# Conclusion

Overall, based on the provided results, LDA seems to be the most effective dimensionality reduction technique, as it consistently maintains high accuracy and precision across different models. PCA also performs well and significantly reduces the feature space, making it a good option when dimensionality reduction is desired. RP shows mixed results, while ICA performs relatively poorly compared to the other techniques. It's important to note that the performance of dimensionality reduction techniques can vary depending on the dataset and the specific problem at hand. It's recommended to experiment with different techniques and evaluate their performance on specific tasks to determine the most suitable approach.

##### REFERENCES

1. Sorzano, C. O. S., Vargas, J., & Montano, A. P. (2014). A survey of dimensionality reduction techniques. arXiv preprint arXiv:1403.2877.
2. Varpa, K., Iltanen, K., & Juhola, M. (2008). Machine learning method for knowledge discovery experimented with otoneurological data. Computer methods and programs in biomedicine, 91(2), 154-164.
3. LaValley, M. P. (2008). Logistic regression. Circulation, 117(18), 2395-2399.
4. Agarap, A. F. M. (2018, February). On breast cancer detection: an application of machine learning algorithms on the wisconsin diagnostic dataset. In Proceedings of the 2nd international conference on machine learning and soft computing (pp. 5-9)
5. Suthaharan, S., & Suthaharan, S. (2016). Support vector machine. Machine learning models and algorithms for big data classification: thinking with examples for effective learning, 207-235.
6. Viswanath, P., & Sarma, T. H. (2011, September). An improvement to k-nearest neighbor classifier. In 2011 IEEE Recent Advances in Intelligent Computational Systems (pp. 227-231). IEEE.
7. Windeatt, T. (2006). Accuracy/diversity and ensemble MLP classifier design. IEEE Transactions on Neural Networks, 17(5), 1194-1211.
8. Rigatti, S. J. (2017). Random forest. Journal of Insurance Medicine, 47(1), 31-39.
9. Dong, W., Woźniak, M., Wu, J., Li, W., & Bai, Z. (2022). Denoising aggregation of graph neural networks by using principal component analysis. IEEE Transactions on Industrial Informatics, 19(3), 2385-2394.
10. Garcia-Bracamonte, J. E., Ramirez-Cortes, J. M., de Jesus Rangel-Magdaleno, J., Gomez-Gil, P., Peregrina-Barreto, H., & Alarcon-Aquino, V. (2019). An approach on MCSA-based fault detection using independent component analysis and neural networks. IEEE Transactions on Instrumentation and Measurement, 68(5), 1353-1361.
11. Gastaldo, P., Zunino, R., Cambria, E., & Decherchi, S. (2013). Combining ELM with random projections. IEEE intelligent systems, 28(6), 46-48.
12. Nie, F., Wang, Z., Wang, R., Wang, Z., & Li, X. (2020). Adaptive local linear discriminant analysis. ACM Transactions on Knowledge Discovery from Data (TKDD), 14(1), 1-19.