Exploring the Impact of Dimensionality Reduction Techniques on Dry Bean Classification in Machine Learning: A Pattern Recognition Approach

Han Chau - 20012654
Department of Mathematical Sciences
Stevens Institute of Technology
Hoboken, NJ, U.S.
hchau@stevens.edu

1. Abstract:

This project delves into the realm of bean classification, leveraging a dataset encompassing seven distinct registered dry beans. Recognized for their nutritional richness and health benefits, beans play a vital role in providing protein, fiber, iron, and essential vitamins. [1] The project addresses the critical need for effective seed classification in agriculture, emphasizing its significance for sustainable farming and marketing. With a focus on optimizing machine learning models, such as SVM, Logistic Regression, Random Forest, Gradient Boosting Classifer, Naive Bayes, and Neural Network the study employs Principal Component Analysis (PCA) to reduce dimensionality. By comparing the performance and training efficiency of models on the original and dimensionality-reduced datasets, the project seeks to identify the most effective dimensions for accurate bean classification.

2. Introduction:

Beans, as a versatile and nutrient-dense food source, offer a myriad of health benefits, ranging from heart health to tissue regeneration. The immense genetic diversity within the realm of dry beans, particularly the most widely produced among edible legumes, underscores their pivotal role in global agriculture. [2] The quality of bean seeds emerges as a linchpin in crop production, necessitating robust classification methodologies for both marketing and sustainable agricultural systems.

In addressing the imperative of accurate seed classification, this project leverages a dataset capturing the nuances of seven distinct registered dry beans. Curated through high-resolution imaging and feature extraction, the dataset comprises 16 features, including 12 dimensions and 4 shape forms. A conventional approach to bean classification involves the application of machine learning models like SVM and Random Forest, yet the efficiency of these models in terms of memory and time utilization remains a crucial consideration.

With this backdrop, the project introduces an innovative approach by incorporating Principal Component Analysis (PCA) to reduce the dimensionality of the dataset. The overarching objective is to scrutinize the impact of dimensionality reduction on machine learning models' performance and training times. By identifying the most informative dimensions for accurate classification, the project aims to contribute insights that can enhance both the efficiency and effectiveness of bean classification models in real-world agricultural applications.

3. Method:

The research methodology for this project entails a systematic approach to optimize the classification of seven distinct registered dry beans utilizing machine learning models. The dataset, having undergone prior feature extraction, encompasses 16 relevant features, including 12 dimensions and 4 shape forms, acquired through high-resolution imaging and segmentation. The primary objective is to compare the performance and training efficiency of machine learning models on both the original dataset and a dimensionality-reduced dataset obtained through Principal Component Analysis (PCA).

4. Tool:

This project harnessed the power of several Python libraries to seamlessly navigate through data exploration, preprocessing, and machine learning tasks. The dataset manipulation and analysis were efficiently handled using the Pandas library, allowing for versatile data structures and operations. NumPy complemented this functionality, providing essential support for numerical operations. Seaborn and Matplotlib were indispensable for data visualization, generating clear and insightful graphical representations.

The scikit-learn library played a crucial role in model training and evaluation, offering a diverse set of tools for machine learning tasks. The train_test_split function facilitated the partitioning of the dataset into training and test sets. Evaluation metrics such as confusion matrices, classification reports, and accuracy scores were efficiently computed using scikit-learn's metrics module. Label encoding was performed using the LabelEncoder from scikit-learn to convert categorical labels into a numerical format suitable for machine learning models.

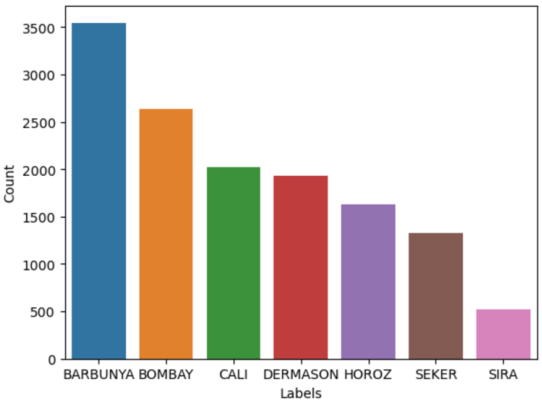
An ensemble of machine learning models was employed for classification tasks. RandomForestClassifier, Support Vector Machines (SVC), GradientBoostingClassifier, LogisticRegression, GaussianNB, and MLPClassifier from scikit-learn were selected for their diverse capabilities in handling various data patterns. The time library was utilized to measure and compare the training times of different models. To enhance the robustness of the analysis, warnings were muted using the warnings library, and a seed was set using random to ensure reproducibility.

Incorporating a comprehensive suite of libraries, this project executed a well-rounded and efficient approach to explore, preprocess, and model the dataset, facilitating a thorough examination of dry bean classification.

5. Approach:

a) Data description:

The dataset consists 13,611 grains of 7 different registered dry beans (Seker, Barbunya, Bombay, Cali, Dermosan, Horoz, and Sira). This dataset has undergone meticulous curation, extracting morphological features from high-resolution images of seven distinct registered dry beans. The features encapsulate key dimensions and shape characteristics, providing a comprehensive representation of the beans' structural attributes. The dataset comprises 16 features, including 12 dimensions and 4 shape forms, offering a detailed exploration of the beans' geometry.



b) Data preprocessing:

To prepare the dataset for training, we used the LabelEncoder() method to convert the string values into the numerical value. The final dataset for implementing the machine learning model and PCA algorithm was obtained by concatenating the matrix of label code with the other attributes. Lastly, we split the data into training and testing sets using an 80:20 ratio.

c) Model Selection:

❖ Random Forest Classifier:

The Random Forest Classifier, a robust ensemble learning method, was chosen for its ability to handle complex relationships and capture non-linear patterns. Comprising multiple decision trees, Random Forests are adept at mitigating overfitting and providing reliable classification results. In this implementation, the RandomForestClassifier is utilized with specific parameters. The max_depth is set to 5, limiting the depth of each decision tree, and n_estimators is set to 10, specifying the number of trees in the forest.

❖ Support Vector Machines (SVM):

SVM, a powerful algorithm for both classification and regression tasks, was employed due to its effectiveness in handling high-dimensional data. Its ability to identify optimal hyperplanes for classification and flexibility in accommodating different kernel functions made it a suitable choice. This implementation utilizes the SVC (Support Vector Classification) class with a linear kernel (kernel='linear'). The C parameter, set to 1.0, controls the regularization strength. SVM aims to find the hyperplane that best separates different classes, maximizing the margin between them.

❖ Gradient Boosting Classifier:

The Gradient Boosting Classifier was chosen for its sequential training approach, where weak learners are incrementally added to the ensemble. This technique often results in high accuracy and has the advantage of adaptability to different data distributions. The GradientBoostingClassifier is employed in this implementation with n_estimators set to 50, determining the number of boosting stages.

❖ Logistic Regression:

Logistic Regression, a fundamental statistical model, was included for its simplicity and interpretability. Despite its simplicity, Logistic Regression can often perform exceptionally well, especially in scenarios with linear separability. In this implementation, the LogisticRegression

class is employed with specific parameters. The max_iter parameter is set to 1000, determining the maximum number of iterations for the solver to converge.

- ❖ Gaussian Naive Bayes:

Gaussian Naive Bayes, based on Bayes' theorem, was selected for its simplicity and efficiency in handling high-dimensional data. This probabilistic model is particularly effective in scenarios where feature independence assumptions hold.

- ❖ Multi-layer Perceptron (MLP) Classifier:

The inclusion of an MLP Classifier addressed the need for a neural network-based approach. Neural networks, particularly multi-layer perceptrons, excel at capturing complex relationships in data and adapting to intricate patterns.

d) Dimensionality Reduction - PCA:

Principal Component Analysis (PCA) was employed to diminish the dimensionality of the dataset, aiming to evaluate its impact on model performance and training efficiency. The implementation of PCA in scikit-learn facilitated this exploration, providing a nuanced understanding of how different dimensions contribute to the classification process.

The Dimensionality Reduction process using PCA unfolded in the following manner. Before delving into PCA, data normalization was essential to mitigate the influence of features with different scales on the resulting principal components. This preprocessing step ensured the reliability of the subsequent dimensionality reduction. [7]

To achieve a meaningful reduction in dimensions, eigenvalues and their corresponding eigenvectors were computed and then sorted in descending order. This sorting procedure was crucial for identifying the most significant principal components, which would contribute significantly to the variance in the dataset.

Subsequently, a Scree Plot was generated to visualize the cumulative variance ratio across different dimensions. The objective was to pinpoint the best dimension, where the change in explained variance became negligible. The determination of this optimal dimension involved identifying the point where the rate of change in explained variance dropped below a predefined threshold (in this case, 10^{-3}).

Once the optimal dimension was identified, a function was created to transform the original data into a new dataset with reduced dimensions. This function systematically applied the PCA transformation based on the identified best dimension, preparing the data for subsequent stages in the project.

e) Model Training and Evaluation:

In the experimental phase, a suite of machine learning models, including Logistic Regression, Support Vector Machines (SVM), Random Forest, Gradient Boosting Classifier, Naive Bayes, and Neural Network, were applied to assess their performance on both the original and dimensionality-reduced datasets obtained through Principal Component Analysis (PCA). For each model, two training sets were considered: the original dataset and the dataset with reduced dimensions using PCA. The evaluation metrics, accuracy, and execution time were recorded for each model on both training sets. The models were trained and tested using the standardized experimental framework.

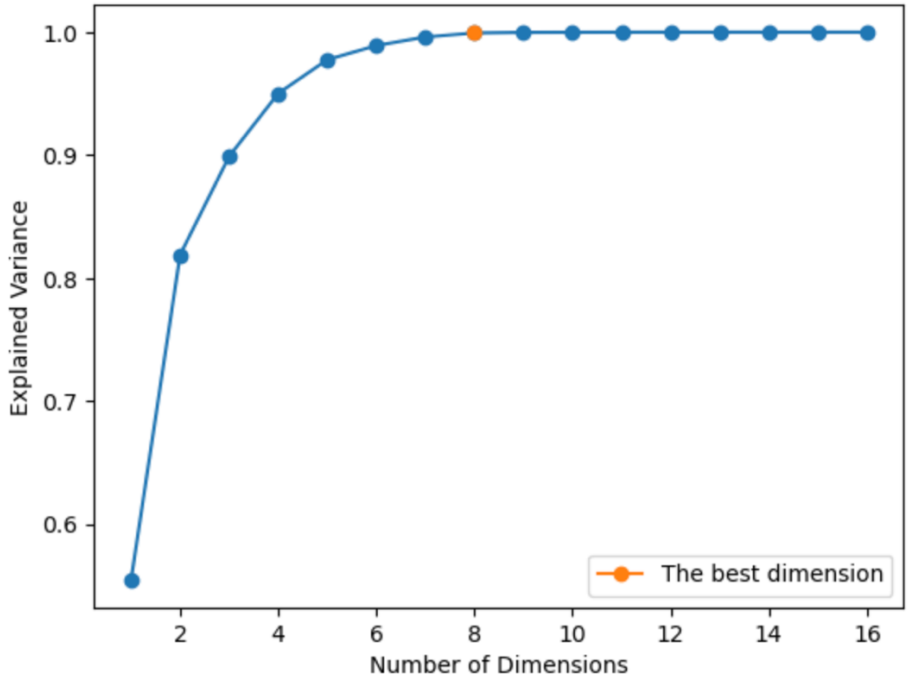
The results, presented in tabular form, provide insights into the comparative performance and computational efficiency of each model under varying dimensionality conditions. This systematic evaluation contributes to the understanding of the trade-offs associated with dimensionality reduction in the context of dry bean classification.

f) Prediction:

The predict_label function is designed to predict the label of a specific instance and subsequently translate it into the corresponding class using the predefined 'labels' dictionary. Following this, the predicted class is printed, and a verification step is executed to check if the predicted class aligns with the true class label. The ultimate output reveals whether the prediction is accurate or not.

6. Result:

The plot illustrates the variance ratio across dimensions, revealing a notable increase as the dimensionality rises. Beginning at one dimension, the variance ratio stands at 0.56, demonstrating an upward trend with higher dimensions. However, beyond eight dimensions, there is minimal discernible difference compared to the ninth dimension, as indicated by the threshold of 0.001. Consequently, the analysis identifies eight dimensions as the optimal choice for dimensionality in the dataset.



The results obtained from the machine learning models applied to both the original dataset and the reduced-dimension dataset using Principal Component Analysis (PCA) reveal interesting insights into the trade-offs between accuracy and computational efficiency.

Machine Learning algorithm: Logistic Regression

	Original dataset	Reduced dimension dateset with PCA
Accuracy (%)	89.28	92.51
Time (s)	1.81	0.33

Machine Learning algorithm: SVM

	Original dataset	Reduced dimension dateset with PCA
Accuracy (%)	91.92	92.91
Time (s)	36.67	0.46

Machine Learning algorithm: Random Forest

	Original dataset	Reduced dimension dateset with PCA
Accuracy (%)	88.03	90.08
Time (s)	0.15	0.09

Machine Learning algorithm: Gradient Boosting Classifer

	Original dataset	Reduced dimension dateset with PCA
Accuracy (%)	92.51	92.14
Time (s)	15.52	8.35

Machine Learning algorithm: Naive Bayes

	Original dataset	Reduced dimension dateset with PCA
Accuracy (%)	75.80	89.94
Time (s)	0.01	0.00

Machine Learning algorithm: Neural Network

	Original dataset	Reduced dimension dateset with PCA
Accuracy (%)	42.27	93.21
Time (s)	0.60	3.82

- ❖ Logistic Regression:

Logistic Regression exhibits a substantial improvement in accuracy on the reduced-dimension dataset (92.51%) compared to the original dataset (89.28%). Simultaneously, there is a significant reduction in training time on the reduced-dimension dataset (0.31 seconds) compared to the original dataset (1.77 seconds). This underscores the positive impact of dimensionality reduction on both accuracy and training efficiency for Logistic Regression.

- ❖ Support Vector Machines (SVM):

SVM demonstrates a marginal improvement in accuracy on the reduced-dimension dataset (92.91%) compared to the original dataset (91.92%). Notably, there is a significant reduction in training time on the reduced-dimension dataset (0.50 seconds) compared to the original dataset (42.92 seconds). This emphasizes the effectiveness of dimensionality reduction in enhancing accuracy and training efficiency for SVM.

- ❖ Random Forest:

Random Forest exhibits a slight improvement in accuracy on the reduced-dimension dataset (90.08%) compared to the original dataset (88.03%). Importantly, there is a substantial reduction in training time on the reduced-dimension dataset (0.09 seconds) compared to the original dataset (0.16 seconds). This highlights the efficiency gains achieved through dimensionality reduction without sacrificing accuracy.

- ❖ Gradient Boosting Classifier:

Gradient Boosting shows a minimal decrease in accuracy on the reduced-dimension dataset (92.14%) compared to the original dataset (92.51%). Additionally, there is a substantial reduction

in training time on the reduced-dimension dataset (8.35 seconds) compared to the original dataset (15.52 seconds). The results suggest a positive impact of dimensionality reduction on training efficiency, with a marginal trade-off in accuracy.

- ❖ Naive Bayes:

Naive Bayes exhibits a significant improvement in accuracy on the reduced-dimension dataset (89.94%) compared to the original dataset (75.80%). Remarkably, training time is extremely minimal for both datasets, underscoring the efficiency of Naive Bayes, particularly on the reduced-dimension dataset.

- ❖ Neural Network:

The Neural Network demonstrates a substantial improvement in accuracy on the reduced-dimension dataset (93.21%) compared to the original dataset (42.27%). However, there is an increase in training time on the reduced-dimension dataset (3.82 seconds) compared to the original dataset (0.60 seconds), indicating a trade-off between accuracy and training time. The results highlight the significant positive impact of dimensionality reduction on Neural Network accuracy, though with a moderate impact on training time.

- ❖ Discussion and Comparison of Confusion Matrices:

----Logistic Regression Model-----

	precision	recall	f1-score	support
0	0.89	0.83	0.86	261
1	0.99	1.00	1.00	117
2	0.89	0.93	0.91	317
3	0.88	0.89	0.89	671
4	0.95	0.92	0.93	408
5	0.94	0.93	0.93	413
6	0.82	0.84	0.83	536
accuracy			0.89	2723
macro avg	0.91	0.90	0.91	2723
weighted avg	0.89	0.89	0.89	2723

----Logistic Regression Model Using PCA-----

	precision	recall	f1-score	support
0	0.92	0.91	0.92	261
1	1.00	1.00	1.00	117
2	0.96	0.93	0.94	317
3	0.91	0.92	0.91	671
4	0.97	0.96	0.97	408
5	0.97	0.94	0.95	413
6	0.85	0.89	0.87	536
accuracy			0.93	2723
macro avg	0.94	0.93	0.94	2723
weighted avg	0.93	0.93	0.93	2723

The model trained on the original dataset shows the overall accuracy of 89%. Particularly noteworthy is the perfect precision and recall achieved for Class 1, indicating the model's accuracy in identifying instances of this class. The Logistic Regression model, trained on the dataset reduced through PCA, demonstrates notable improvements in various performance metrics. Precision, recall, and F1-score values maintain or increase for most classes, showcasing the positive impact of dimensionality reduction on the model's discriminative capabilities. The overall accuracy increases to 93%, indicating enhanced classification performance with the reduced-dimensional dataset. The comparison between the two models reveals that the Logistic Regression model using PCA outperforms its counterpart trained on the original dataset. The reduced-dimensional representation contributes to improved precision, recall, and F1-score metrics, resulting in a more accurate and efficient classification. Class 1 consistently exhibits exceptional performance, emphasizing the model's proficiency in identifying instances of this class.

7. Compile and run the code:

The process of compiling and executing the submitted code involves several crucial steps to guarantee a successful analysis of the dry bean dataset. Initially, it is imperative to establish a

suitable Python environment, and it is recommended to utilize virtual environments for improved dependency management. Ensure the presence of essential libraries such as Pandas, NumPy, Seaborn, Matplotlib, and scikit-learn by executing the provided pip install command in the terminal or command prompt.

Additionally, the dataset, named 'Dry_Bean_Dataset.xlsx,' must reside in the same directory as the code. Carefully verify the dataset's location, and if necessary, modify the file path within the code accordingly. Subsequently, open the Python program in either a Jupyter Notebook or Google Colab and proceed to execute the code blocks in sequential order. This systematic approach ensures the correct flow of operations and facilitates a comprehensive analysis of the dry bean dataset.

8. Conclusion:

The dry bean dataset is strategically simplified by the PCA dimensionality reduction with a dimension of eight. The objective of this reduction is to use the advantages of computational performance while preserving an adequate degree of information integrity. In light of the larger machine learning analysis, the discussion sheds light on the considerations related to choosing this particular PCA dimension.

This project involved dimensionality reduction, model training, and performance evaluation in the investigation of machine learning models for the classification of dry bean types. The main conclusions and ramifications of this project highlight how important it is to choose models carefully and use dimensionality reduction strategies to maximize accuracy and efficiency.

Subsequent versions of this project could investigate other methods of reducing dimensionality and adjust hyperparameters to improve the performance of the model. Incorporating more sophisticated models or group techniques may also provide insights on pushing the limits of classification accuracy.

In conclusion, this effort shed light on the complex interactions that exist between machine learning models and dimensionality reduction in the categorization of dry bean types. The effective use of PCA in conjunction with a thorough model assessment paves the way for future developments in the application of machine learning for purposes other than crop classification.

References

[1] K. W. Warwick, "Medical News Today," 16 November 2023. [Online]. Available: https://www.medicalnewstoday.com/articles/320192. [Accessed 12 2023].

[2] M. Koklu and I. A. Ozkan, "Multiclass classification of dry beans using computer vision and machine learning techniques," *Computers and Electronics in Agriculture,* vol. 174, no. https://doi.org/10.1016/j.compag.2020.105507, 2020.

[3] A. Governorate, "Kaggle," 2021. [Online]. Available: https://www.kaggle.com/code/aboudaladdin/dry-beans-starter-eda. [Accessed December 2023].

[4] G. Słowiński, "Dry Beans Classification Using Machine Learning," https://ceur-ws.org/Vol-2951/paper3.pdf, Poland.

[5] C. Thrampoulidis, S. Oymak and M. Soltanolkotabi, "Theoretical Insights Into Multiclass Classification: A High-dimensional Asymptotic View," https://par.nsf.gov/servlets/purl/10206601.

[6] B. M. Alsafy, Z. . M. Aydam and W. K. Mutlag, "Multiclass Classification Methods: A Review," https://www.researchgate.net/publication/347327472_Multiclass_Classification_Methods_A_Review, 2019.

[7] S. Ulsha, "Medium," 12 October 2019. [Online]. Available: https://medium.com/analytics-vidhya/principal-component-analysis-pca-dive-deep-411db0f9ee10. [Accessed December 2023].