Fashion MNIST Classification

Project 4

Trey Schulman (treys3)

12/11/2025

Table of Contents

1. Introduction

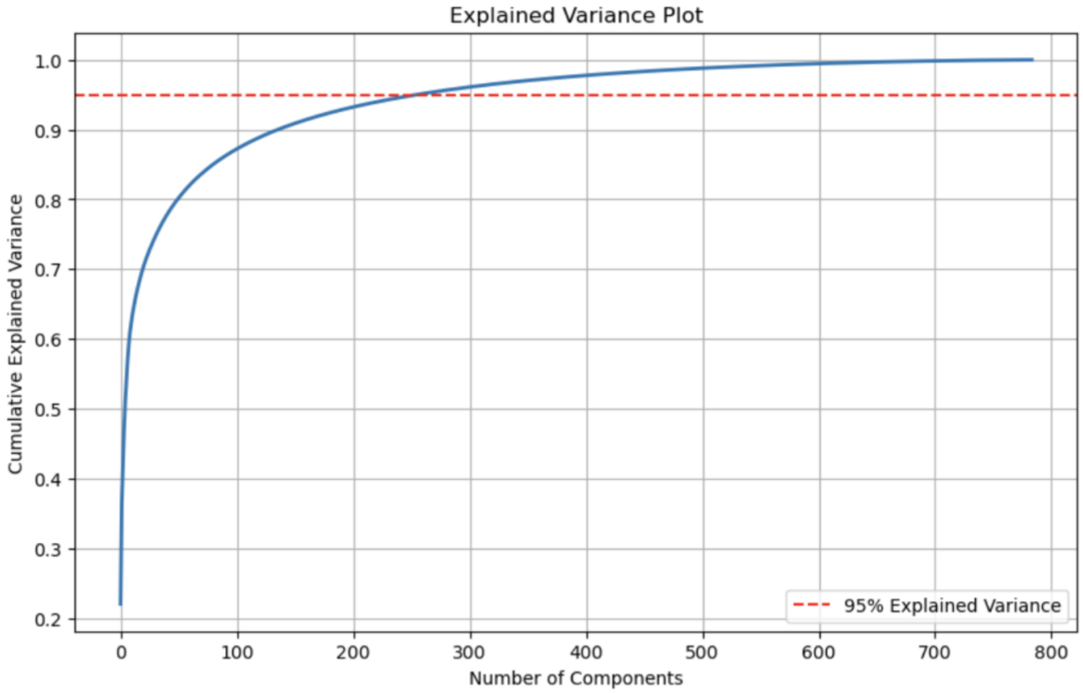
The objective of this project is to perform multi-class classification and unsupervised analysis on the Fashion-MNIST dataset. Unlike standard MNIST digit recognition, this dataset consists of grayscale images of clothing items, which creates a more complex challenge due to the strong similarity between classes (e.g., "Pullover" and "Coat"). The project is divided into two phases of unsupervised learning to uncover natural groupings within the data, and supervised learning to predict the specific clothing label. The clustering was performed through K-Means and Hierarchical Clustering algorithms to detect differences between class groups. I then implemented Logistic Regression, Support Vector Machines (SVM), Random Forests, and Gradient Boosting and evaluated them based on classification accuracy.

2. Data and Preprocessing

The dataset consists of 70,000 grayscale images (60,000 training, 10,000 testing), each represented as a 28x28 pixel grid. The labels correspond to 10 distinct classes: T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot. I analyzed the distribution of the outcome variable and found the dataset to be perfectly balanced. Both the training set (6,000 samples per class) and testing set (1,000 samples per class) exhibited a uniform distribution. This balance ensured that accuracy would be a reliable metric for evaluation, and no class imbalance correction would be required.

Preprocessing and Dimensionality Reduction

The raw image data consists of 784 features per sample, which can be computationally expensive for some clustering and classification tasks, To prepare the data for modeling, I standardized the pixel values to have a mean of 0 and a variance of 1. Because algorithms like SVM and K-Means struggle with high dimensional data, I applied Principal Component Analysis (PCA) to reduce the feature space.



As shown in the cumulative variance plot, PCA retained 95% of the variance with 256 components. This reduced dataset was used for the distance-based algorithms (SVM, Logistic Regression, K-Means), while the raw scaled features were retained for the tree-based algorithms (Random Forest, Gradient Boosting) which handle high-dimensional raw data more effectively.

3. Unsupervised Learning

I applied two clustering algorithms to the training data to determine if the mathematical properties of the images naturally aligned with the assigned labels.

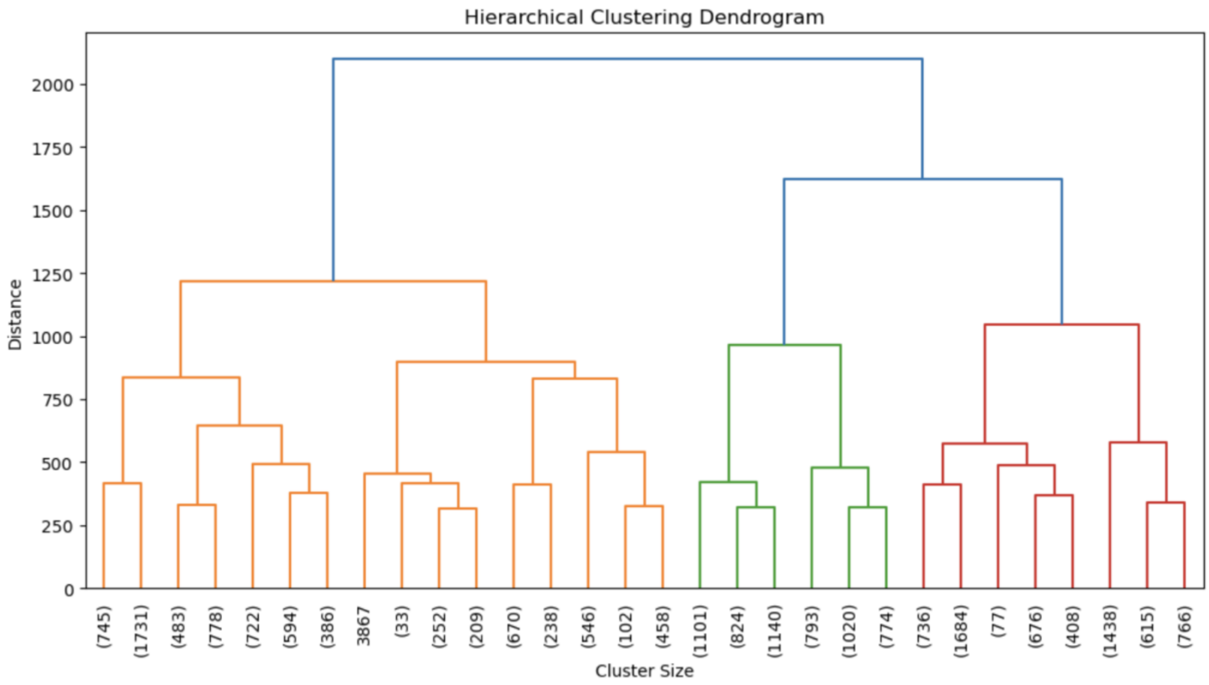
3.1 K-Means Clustering

I utilized the K-Means algorithm on the PCA-reduced data and determined the optimal number of clusters through a systematic search using the Silhouette score for k’s 8, 10, 12, and 14. The Silhouette score peaked at 10 clusters with a score of approximately 0.13, which reinforced the presence of 10 distinct classes.

The resulting clusters showed mixed purity. Distinct items like "Bags" (Cluster 3, 93.8%) and "Ankle boots" (Cluster 5, 87.4%) formed clean clusters, but "Shirt" was poorly clustered with the dominant cluster for shirts capturing only 25.6% of the true labels. This suggests K-Means struggled to separate visually similar upper-body garments.

3.2 Hierarchical Clustering

Initially, I attempted Spectral Clustering, but the algorithm failed to converge on the high-dimensional data and produced graph connectivity warnings even on reduced subsets. Therefore, I pivoted to Hierarchical Clustering with Ward linkage. I ran the algorithm on a subset of 20,000 samples to accommodate memory constraints while ensuring solid representation.



The dendrogram revealed a clear structure where by cutting the tree at a height of approximately 600 there would be 10 isolated clusters.

The hierarchical approach achieved slightly cleaner separation than K-Means for certain categories. For example, Cluster 4 was 98.6% "Bag" and Cluster 5 was 93.6% "Ankle boot". However, similarly to K-Means, it struggled to distinguish "Coat" from "Pullover" and grouped them into mixed clusters due to their similarities.

4. Classification Methodology

I trained four multi-class classifiers all using a 3-fold cross-validation framework to tune hyperparameters and optimize performance.

4.1 Logistic Regression

We implemented a multinomial Logistic Regression using the lbfgs solver because other methods failed to converge. This linear model served as a quality baseline for further comparisons. The model used the PCA features for computational efficiency. Grid search was used to select the optimal regularization parameter C of 0.1.

4.2 Support Vector Machine (SVM)

I then utilized an SVM with a Radial Basis Function (RBF) kernel to capture nonlinear relationships. Due to the computational complexity of SVMs, this model also utilized the 256 PCA components. Cross validation tuned C and gamma to identifying C=10 and gamma='scale' as the best configuration.

4.3 Random Forest

I trained the Random Forest ensemble on the raw scaled data, since decision trees inherently perform feature selection and handle high dimensions well. Cross validation tuned the number of estimators and tree depth and found that 200 trees with no maximum depth provided the best results. Feature importance analysis confirmed that tree-based methods could isolate relevant pixels without PCA.

4.4 Gradient Boosting

Finally, I implemented Gradient Boosting using the XGBoost classifier. Similar to the Random Forest, this was trained on raw scaled data. I utilized the histogram-based tree method for speed. The optimal hyperparameters were found to be a learning rate of 0.2, a max depth of 6, and 200 estimators.

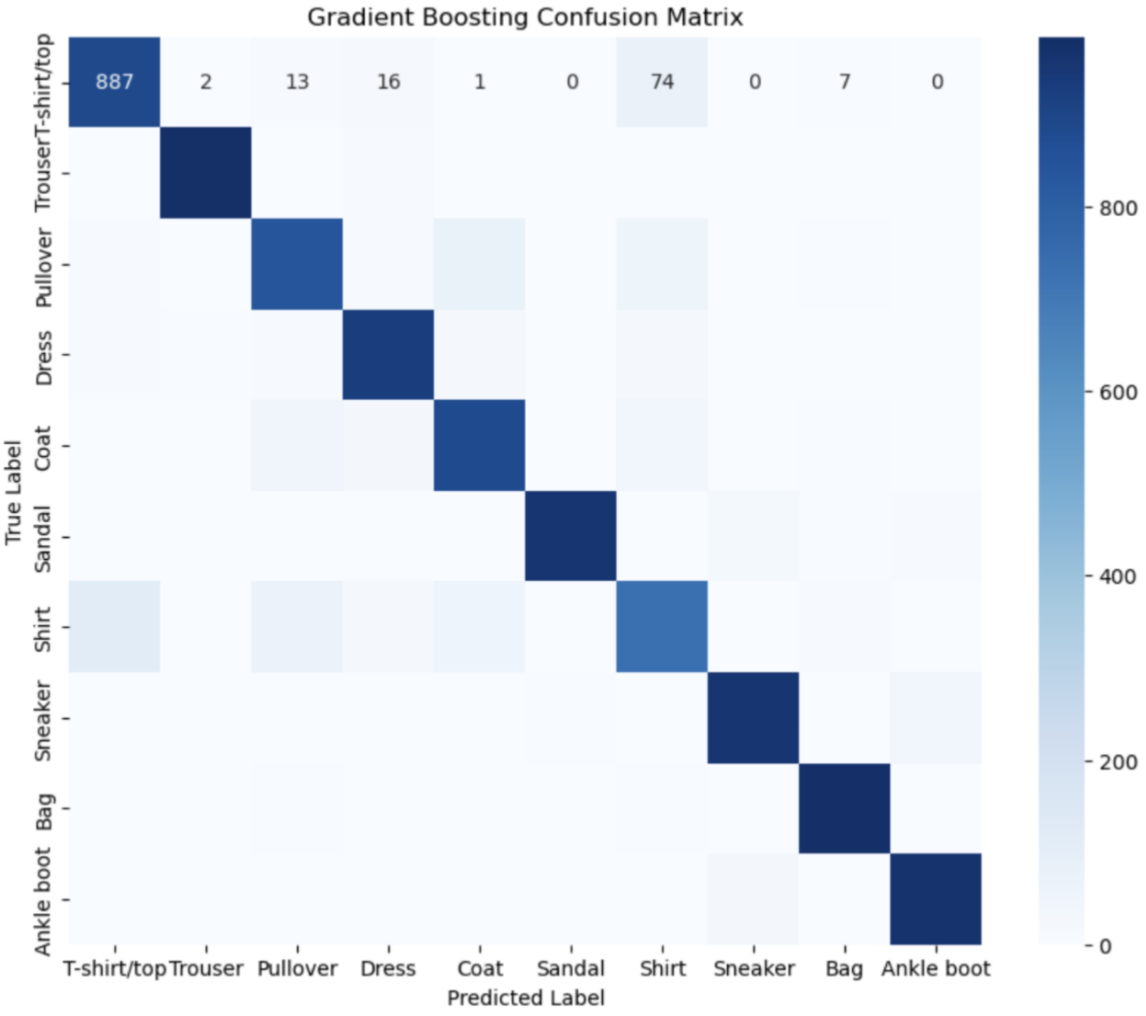
5. Results

Table 1 summarizes the final accuracy on the 10,000 record test set. Gradient Boosting achieved the highest performance, followed closely by the SVM.

Table 1: Model Performance Comparison

Model	**Test Accuracy**
Gradient Boosting	91.22%
SVM	90.70%
`Random Forest	88.44%
Logistic Regression	85.79%

The Gradient Boosting model achieved a test accuracy of 91.22%. The confusion matrix below illustrates where the model succeeded and failed.



The model achieved near-perfect classification for distinct items like "Trouser" (98.4% recall) and "Bag" (97.8% recall). The primary source of error was confusion between "Shirt" and "T-shirt/top" where 74 shirts were misclassified as T-shirts.

6. Discussion

The results demonstrate that nonlinear models significantly outperform linear baselines for this image classification task. The Logistic Regression model, limited to linear decision boundaries, achieved the lowest accuracy (85.79%). In contrast, both SVM and Gradient Boosting achieved >90% accuracy, which demonstrates how the relationship between pixel intensity and clothing category is highly nonlinear.

Gradient Boosting was likely the superior method (91.22%) because it sequentially corrects errors on difficult-to-classify examples (such as the Shirt/Coat boundary). It is worth noting that the SVM performed surprisingly well (90.70%) using only the PCA-reduced features, which suggests that the top 95% of variance captures the vast majority of the signal required for classification.

The unsupervised analysis mirrored the supervised results. The algorithms easily clustered footwear and trousers but struggled to differentiate between the upper-body garments. This implies that the pixel-wise Euclidean distance between a "Shirt" and a "Coat" is small, and supervised algorithms require complex nonlinear boundaries to distinguish them effectively. Future improvements could involve using Convolutional Neural Networks (CNNs) to capture spatial hierarchies that algorithms like Random Forests and SVMs miss.