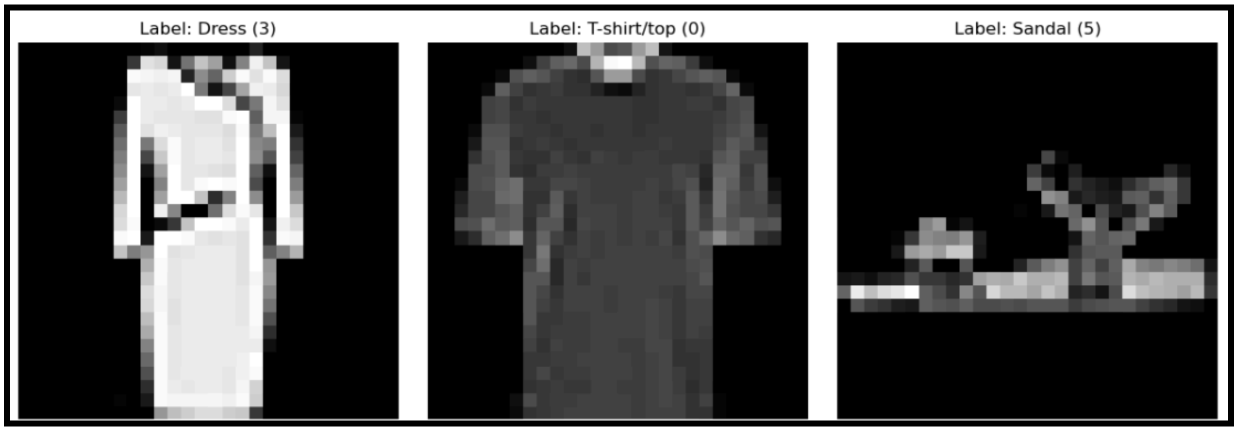
Main objective of the analysis:

- The main objective of my analysis will be to create a basic multilayer perceptron neural network that can accurately predict articles of clothing in 28x28 grayscale images using Principal Component Analysis of 3 different amounts of principal components.
- A group of people that can benefit from this analysis could be fashion company employees. Creating an automated model that can classify clothing can enable employees to save time with classifying clothing on their own, allowing them to save time for other work that needs attention.
- On the other end of the spectrum, consumers could benefit from the company having an image classification model because it can recommend clothing to consumers, making shopping more convenient for them.
- The analysis will contain PCA of 3 different components:
    - 10 principal components
    - 25 principal components
    - 100 principal components

Brief Description of the Dataset:

- The dataset I used was the MNIST-Fashion dataset that was meant to mirror he original MNIST digits data, but slightly more complex, as clothing shapes can tend to be harder to distinguish than numbers
- The data has 60,000 training data, and 10,000 test data.
- Each image is a 28x28 grayscale pixelated image of one of 10 articles of clothing:
    - T-shirt/top
    - Trouser
    - Pullover
    - Dress
    - Coat
    - Sandal
    - Shirt
    - Sneaker
    - Bag
    - Ankle boot
- All classes are balanced (6000 images per class)
- Shown below are some sample images



Brief summary of data exploration and actions taken for data cleaning and feature engineering:

- Something interesting I did was **cut off the first 15 data points of the test data** so that it was never used in fitting the model.
- I did this because I wanted to use the first 15 to actually see how each of the 3 models did. If I had included them in the test data, the models would have gotten all of them correct, which would not really accurately reflect each of the model's performance.
- It is OK for me to take the first 15, as these are random images and not all the same class.

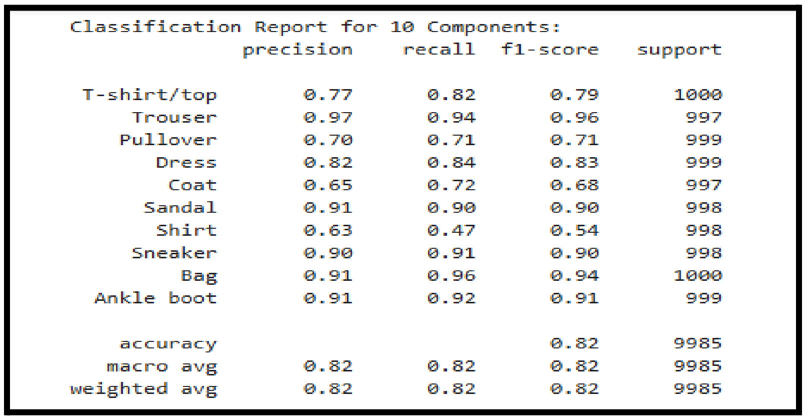
Multilayer Perceptron Neural Network:

- Before I get into each of the different PCA models, I want to first define the MLP Neural NEtwork that I used across all 3 variations.
- The MLP will have the same attributes across all 3 in order to only test how PCA affects the accuracy and prediction strength
- All layers use sigmoid activation function
- **3 Hidden Layers**
- 1st Hidden Dense Layer: 128 neurons
- 2nd Hidden Dense Layer: 64 neurons
- 3rd Hidden Dense Layer: 32 neurons
- Dropout Layer (.3)
- **Output layer: 10 neurons (Softmax for each of the 10 classes)**
- Adam optimizer utilized
- Sparse Categorical Crossentropy loss function utilized
- **20 Epochs(Iterated through training data 20 times)**

- Batch size = 64 samples

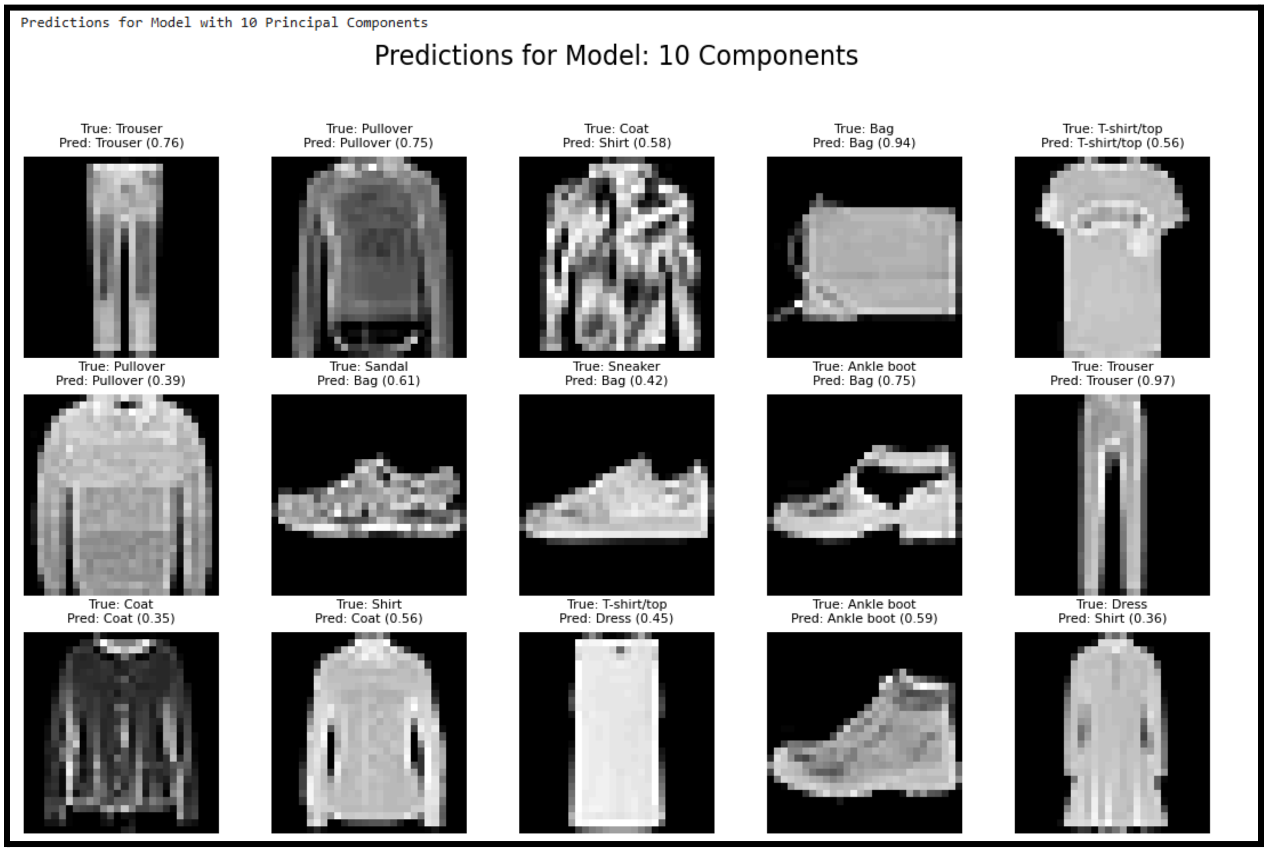
Summary of training MLP with 3 different Principal Component Models (10,25,100):

- **Model with 10 components:**
  - **Classification Report:**

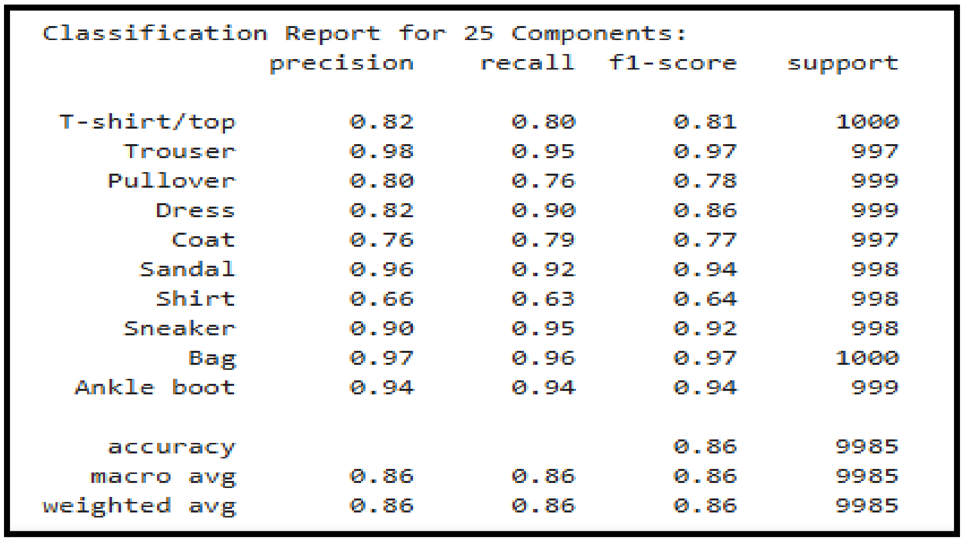


```
Classification Report for 10 Components:
              precision    recall  f1-score   support

 T-shirt/top       0.77      0.82      0.79      1000
     Trouser       0.97      0.94      0.96       997
    Pullover       0.70      0.71      0.71       999
       Dress       0.82      0.84      0.83       999
        Coat       0.65      0.72      0.68       997
      Sandal       0.91      0.90      0.90       998
       Shirt       0.63      0.47      0.54       998
     Sneaker       0.90      0.91      0.90       998
         Bag       0.91      0.96      0.94      1000
  Ankle boot       0.91      0.92      0.91       999

    accuracy                           0.82      9985
   macro avg       0.82      0.82      0.82      9985
weighted avg       0.82      0.82      0.82      9985
```

  - **Performance of model on 15 samples not used in model creation (It got 7 wrong)**

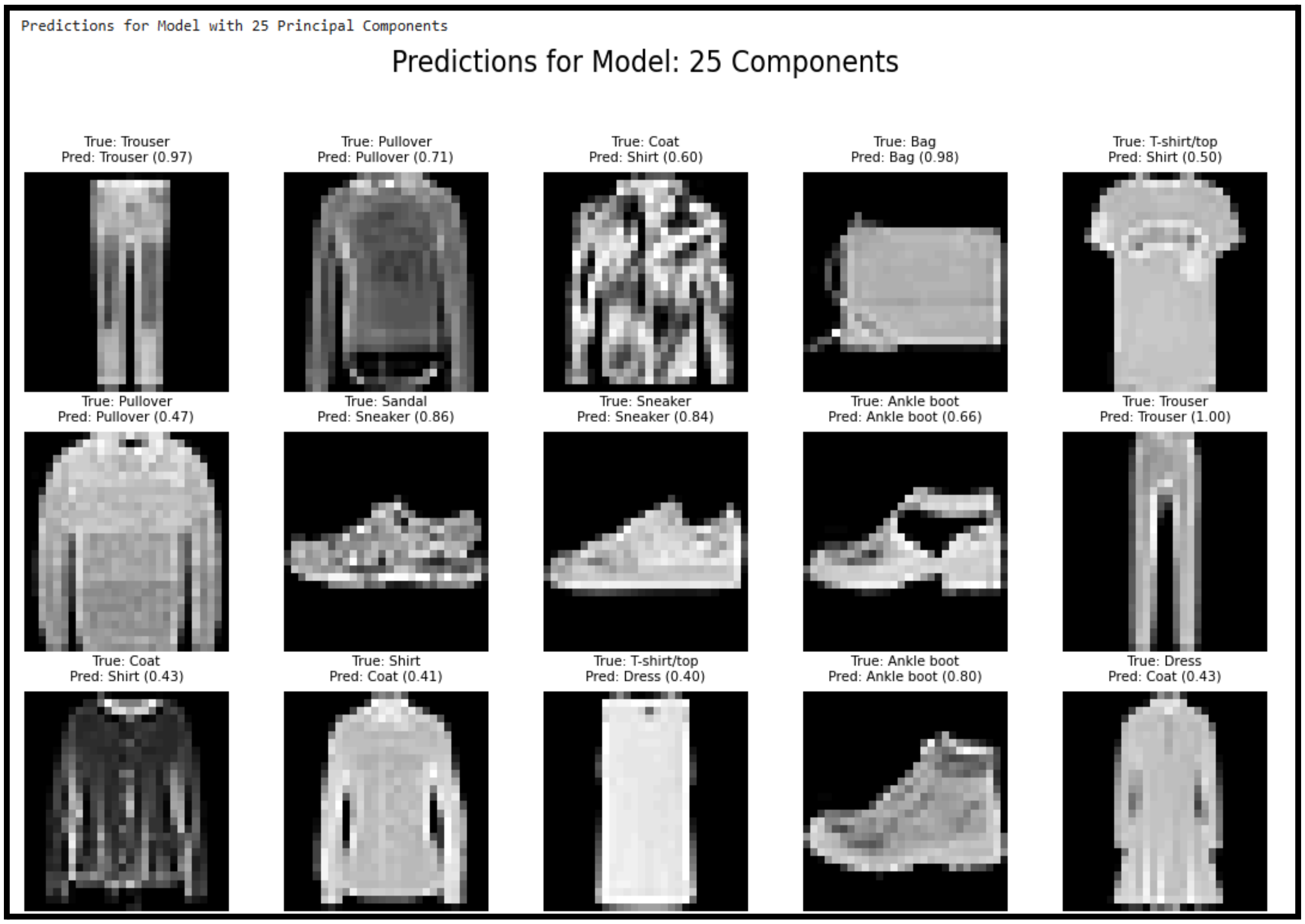

Predictions for Model with 10 Principal Components
Predictions for Model: 10 Components
True: Trouser Pred: Trouser (0.76)
True: Pullover Pred: Pullover (0.75)
True: Coat Pred: Shirt (0.58)
True: Bag Pred: Bag (0.94)
True: T-shirt/top Pred: T-shirt/top (0.56)
True: Pullover Pred: Pullover (0.39)
True: Sandal Pred: Bag (0.61)
True: Sneaker Pred: Bag (0.42)
True: Ankle boot Pred: Bag (0.75)
True: Trouser Pred: Trouser (0.97)
True: Coat Pred: Coat (0.35)
True: Shirt Pred: Coat (0.56)
True: T-shirt/top Pred: Dress (0.45)
True: Ankle boot Pred: Ankle boot (0.59)
True: Dress Pred: Shirt (0.36)

- **Model with 25 components:**
    - **Classification Report:**

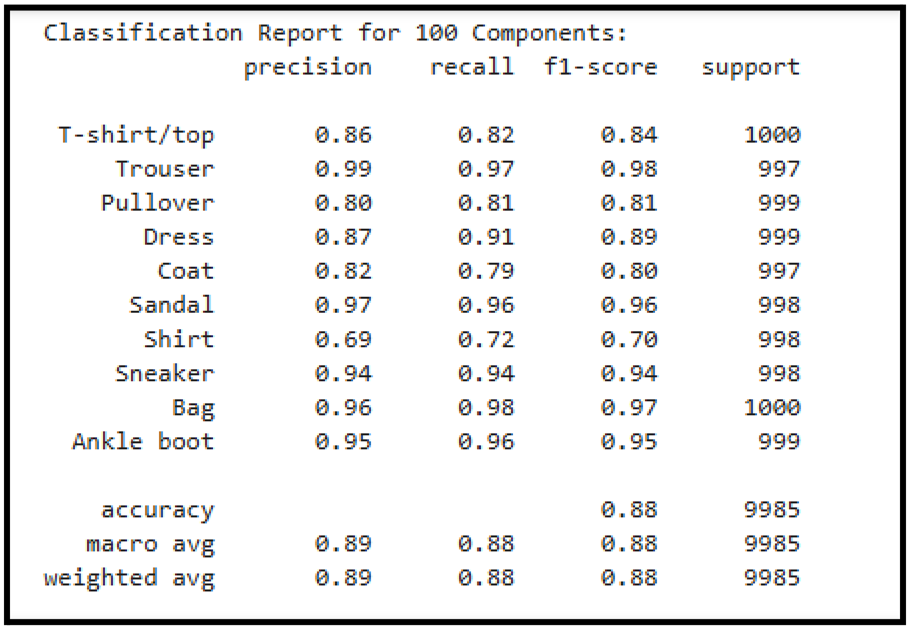


```
Classification Report for 25 Components:
              precision    recall  f1-score   support

 T-shirt/top       0.82      0.80      0.81      1000
     Trouser       0.98      0.95      0.97       997
    Pullover       0.80      0.76      0.78       999
       Dress       0.82      0.90      0.86       999
        Coat       0.76      0.79      0.77       997
      Sandal       0.96      0.92      0.94       998
       Shirt       0.66      0.63      0.64       998
     Sneaker       0.90      0.95      0.92       998
         Bag       0.97      0.96      0.97      1000
  Ankle boot       0.94      0.94      0.94       999

    accuracy                           0.86      9985
   macro avg       0.86      0.86      0.86      9985
weighted avg       0.86      0.86      0.86      9985
```

    - **Performance of model on 15 samples not used in model creation (It got 7 wrong also, but better probabilities)**

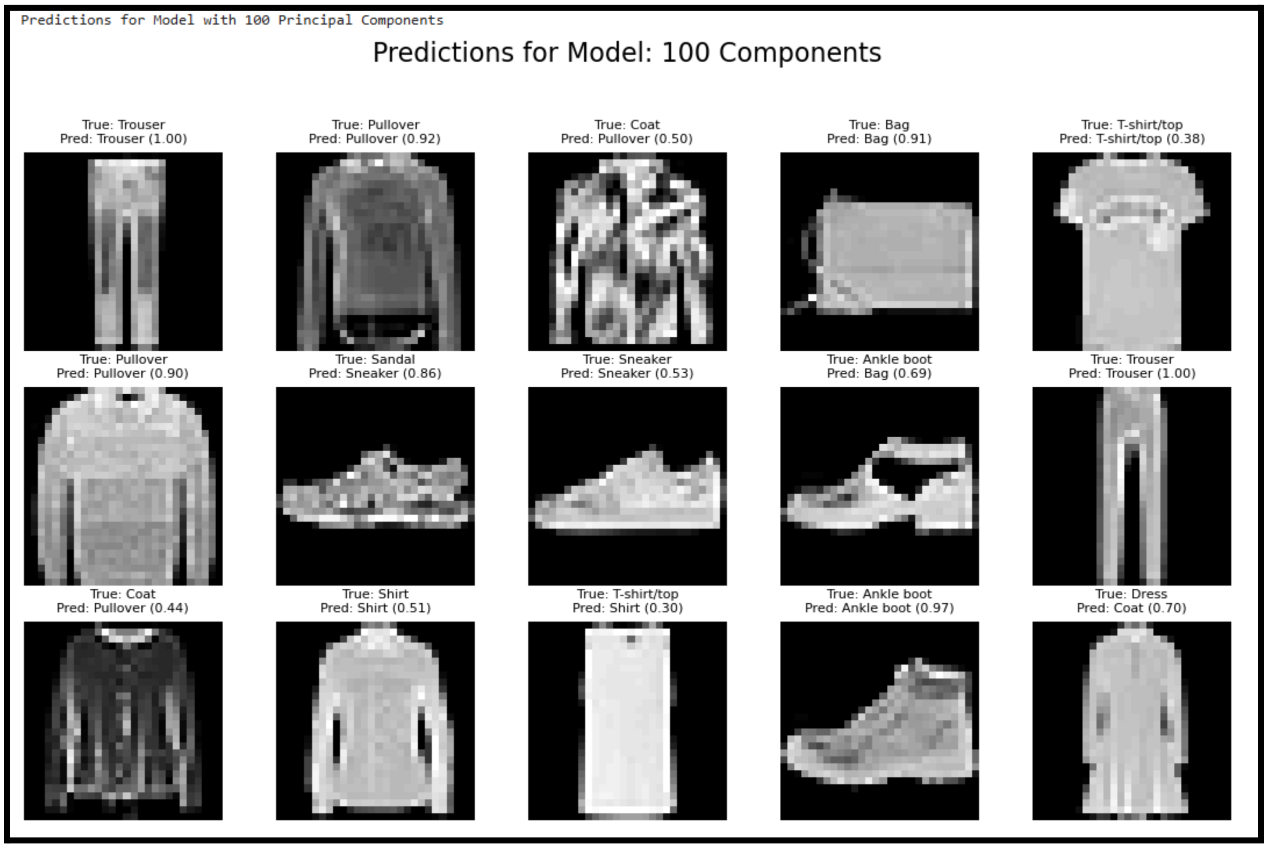

Predictions for Model with 25 Principal Components
Predictions for Model: 25 Components
True: Trouser Pred: Trouser (0.97)
True: Pullover Pred: Pullover (0.71)
True: Coat Pred: Shirt (0.60)
True: Bag Pred: Bag (0.98)
True: T-shirt/top Pred: Shirt (0.50)
True: Pullover Pred: Pullover (0.47)
True: Sandal Pred: Sneaker (0.86)
True: Sneaker Pred: Sneaker (0.84)
True: Ankle boot Pred: Ankle boot (0.66)
True: Trouser Pred: Trouser (1.00)
True: Coat Pred: Shirt (0.43)
True: Shirt Pred: Coat (0.41)
True: T-shirt/top Pred: Dress (0.40)
True: Ankle boot Pred: Ankle boot (0.80)
True: Dress Pred: Coat (0.43)

- **Model with 100 components:**
  - **Classification Report:**



```
Classification Report for 100 Components:
              precision    recall  f1-score   support

 T-shirt/top       0.86      0.82      0.84      1000
     Trouser       0.99      0.97      0.98       997
    Pullover       0.80      0.81      0.81       999
       Dress       0.87      0.91      0.89       999
        Coat       0.82      0.79      0.80       997
      Sandal       0.97      0.96      0.96       998
       Shirt       0.69      0.72      0.70       998
     Sneaker       0.94      0.94      0.94       998
         Bag       0.96      0.98      0.97      1000
  Ankle boot       0.95      0.96      0.95       999

    accuracy                           0.88      9985
   macro avg       0.89      0.88      0.88      9985
weighted avg       0.89      0.88      0.88      9985
```

  - **Performance of model on 15 samples not used in model creation (It got 6 wrong, and better overall probabilities)**

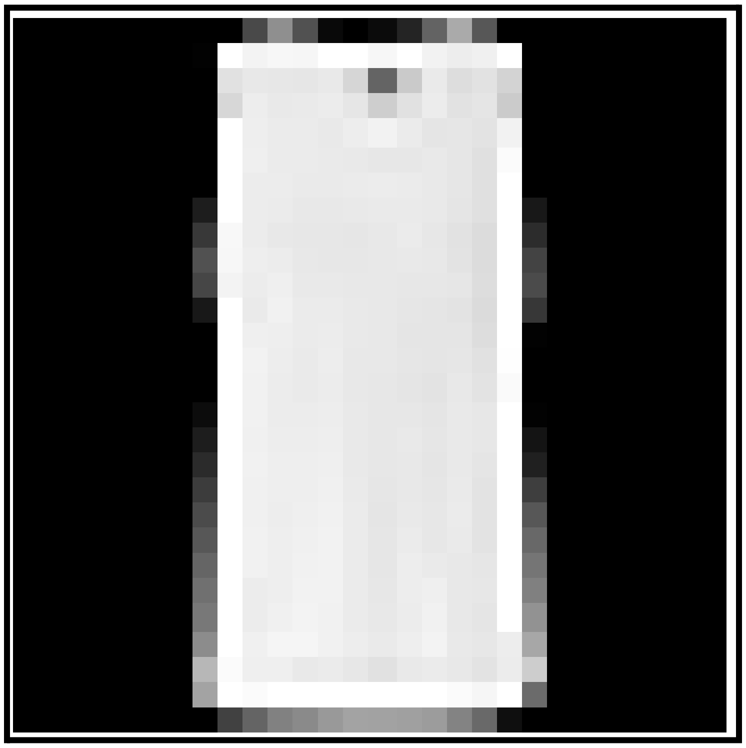


Final Recommendation:

- Out of all the models, **I recommend the 100 layer principal component MLP.** This is because it has a higher accuracy score by far and did noticeably better than the other two models in predicting more correct, and also having better probabilities overall.

Summary Key Findings and Insights:

- I found something really nice while comparing the three performances:
- Specifically, let's look at the 13th sample out of the 15 samples in the holdout set I used.



- This is a T-shirt/top. Even for me, I had no clue what this was. But something really cool I found was that **the more components we added, the better the model got at recognizing that It was not confident in predicting this**. Here's what I mean:
    - 10 Components: Predicted that this was a dress with 45% confidence
    - 25 Components: Predicted that this was a dress with 40% confidence
    - 100 Components: Predicted that this was a shirt with 30% confidence

- Clearly, the more components we added, it recognized that it was not able to predict it well, and it actually got closer to the true t/shirt value by switching its guess from a dress to a shirt.

Suggestions for next steps in analyzing this data:

- Here are some things I left off the table that could be done in future analyses that may improve overall performance
    - **Try different numbers of principal components**
        - My analysis is somewhat limited because I am only testing 3 different amounts of principal components. The optimal number might be between 25 and 100, but I would not know this because I only tested few values.
    - **Try different images for evaluating performance**
        - I only used actual MNIST pictures for evaluating performance. If I used real images online, that would generate a more real-world evaluation that isn't specific to the MNIST data
    - **Use a different neural network or classification model**
        - A better option might(would definitely) be to use a Convolutional neural network, which is especially good at dealing with images like the MNIST data. I know this would make the prediction significantly better regardless of the number of principal components.