

September 5, 2021

Quinn E Knudsen

IST 718

Syracuse University

Modeling the data science way

computer vision for fashion image recognition

**Table of Contents**

**Introduction**……………………………………………………………………………......pg 2

Areas of Focus …………………………………………………………….……......pg 2

**About the Data..**…………………………………………………………………..….….…pg 2

Data Overview ...……………………………………………………………….…...pg 2

**Exploratory Data Analysis & Correlations**……………………………………..….….…pg 2

Class Balance……..…………………………………………………………….…...pg 3

**Modeling**....………………………………………………………………………..………..pg 3

Accuracy…………………...……………………………………………….…....….pg 3

Gaussian Naïve Bayes……….………………………………………….…...….…..pg 4

CatBoost…………….……….………………………………………….…...….…..pg 4

RandomForest……………….………………………………………….…...….…..pg 5

XGBoost…………………….………………………………………….…...….…..pg 6

KNN………………..……….………………………………………….…...….…..pg 7

Neural Network Models…….………………………………………….…...….…..pg 8

Tradeoffs……….………………………………………….……………......….…..pg 11

**Conclusion & Future Directions..**………………………………………..………….…...pg 12

**Introduction**

Fashion forward is the name of the game and model accuracy is the beast to tame. For this task, several models are to be trained and tested in order to correctly classify visual images of clothing. The key questions to be addressed are the following:

* What is the accuracy of each method?
* What are the trade-offs of each approach?
* What is the compute performance of each approach?

**About the Data**

The initial dataset contained 60,000 train samples and 10,000 test samples with each row representing an image and each column representing a pixel of the image. The values in each column range from 0 (white) to 255 (black). The large number of rows and columns, like many computer vision datasets, makes the computational limits a key factor to consider when training and testing models.

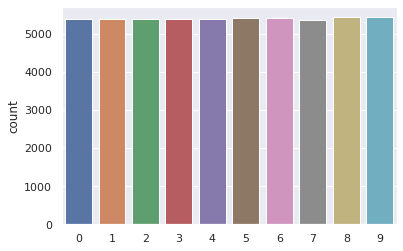
Table 1 below provides an overview of the data labels that the models will attempt to correctly classify.

|  |  |
| --- | --- |
| **Label** | **Class** |
| 0 | T-shirt/top |
| 1 | Trouser |
| 2 | Pullover |
| 3 | Dress |
| 4 | Coat |
| 5 | Sandal |
| 6 | Shirt |
| 7 | Sneaker |
| 8 | Bag |
| 9 | Ankle boot |

*Table 1.* A list of labels and their class for the classification problem.

**Exploratory Data Analysis**

Some exploratory analysis was conducted to understand the class balance of dataset. As demonstrated in Figure 1, there was near perfect class balance amongst the 10 fashion images eliminating the need to do any sampling modifications during the data preprocessing.



*Figure 1.* Class balance for the 10 fashion labels.

The data shape represented 28 x 28 pixels and were scaled prior to neural networking modeling by dividing the pixel value by 255 to ensure each column value value fell between 0 and 1. Below is an example of the labeled data the model will attempt to classify.

****

*Figure 2.* Labeled data for model classification.

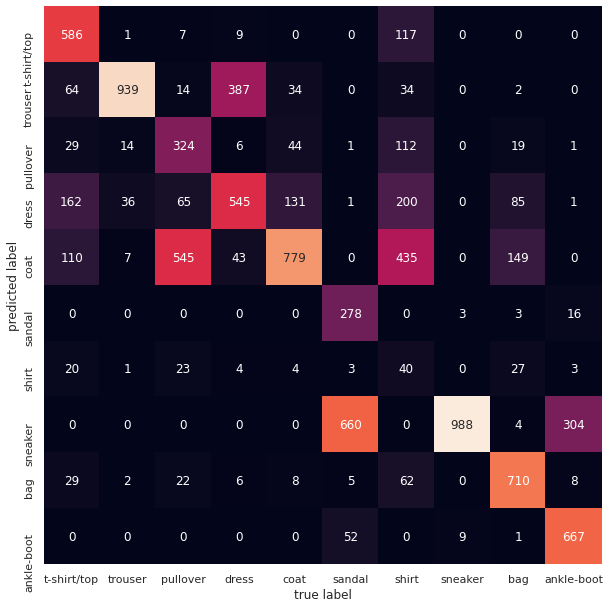
**Modeling**

*Accuracy*

The objective of this classification problem is to maximize the model accuracy. Each algorithm was trained and tested with that objective. Also considered for final model recommendation is the computational demand and subsequent train time needed to build the model.

**Gaussian Naïve Bayes**

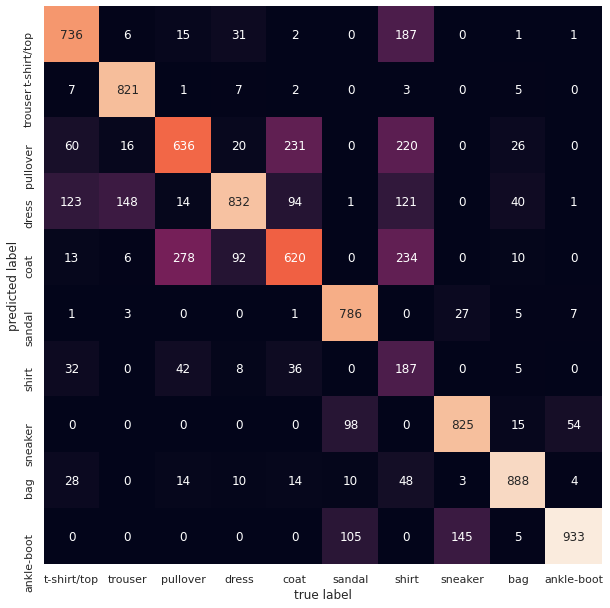
The first model tested was a Gaussian Naïve Bayes model which boasted efficient train time, but poor results with a weighted accuracy of 59%. As displayed in the confusion matrix below, the model had difficulty on alike images (sandal, sneaker, ankle-boot; coat, pullover) more so than completely dissimilar articles of clothing (bag, trouser). This makes conceptual sense amongst the related and dissimilar images. Interestingly, for ‘shirt’ the model had an f1-score of just 0.07 due to extremely low recall at 0.04. On a positive note, the model recall of sneaker was 0.99.

**

*Figure 3.* Gaussian Naïve Bayes confusion matrix.

**CatBoost**

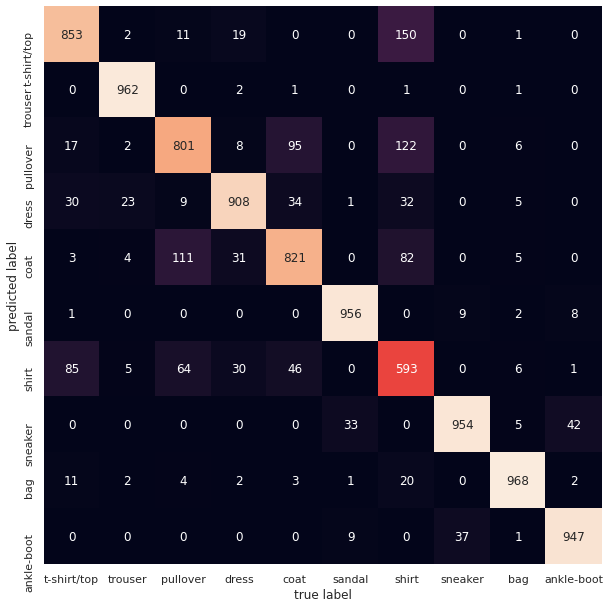
The next model tested was the CatBoost classifier with a moderate train time of 323 seconds and an accuracy score of 73%. Similar to the Gaussian Naïve Bayes model, classifying shirt proved to be a challenge with an f1-score of 0.29. Significantly better results were achieved when classifying the various shoes which helped improve the overall model accuracy.



*Figure 4.* CatBoost confusion matrix.

**Random Forest**

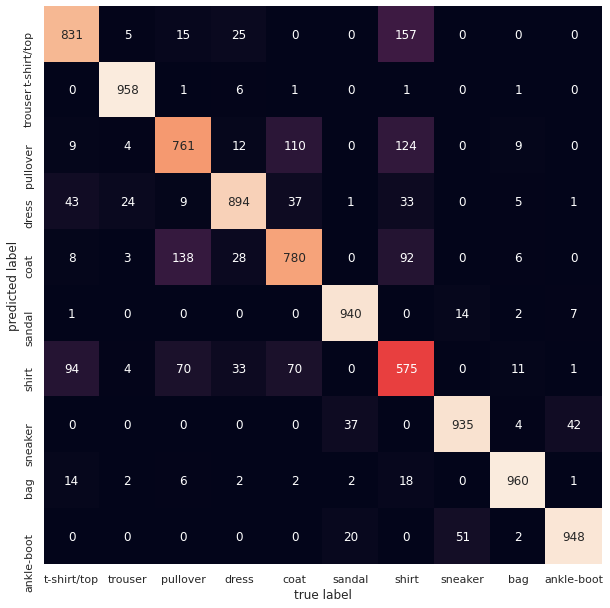
The third model tested was the Random Forest classifier and yielded significant improvements on accuracy (88%) and a shorter train time when compared with the CatBoost model. Major gains were made on some of the less accurate items (sandals and dress) while previously high f1-score items such as (ankle-boot, bag, sneaker, coat, and trouser) rose into the 0.9+ f1-score range. The execution time for this model was about 1/3 that of the CatBoost completing in 101 seconds.

****

*Figure 4.* Random Forest confusion matrix.

**XGBoost**

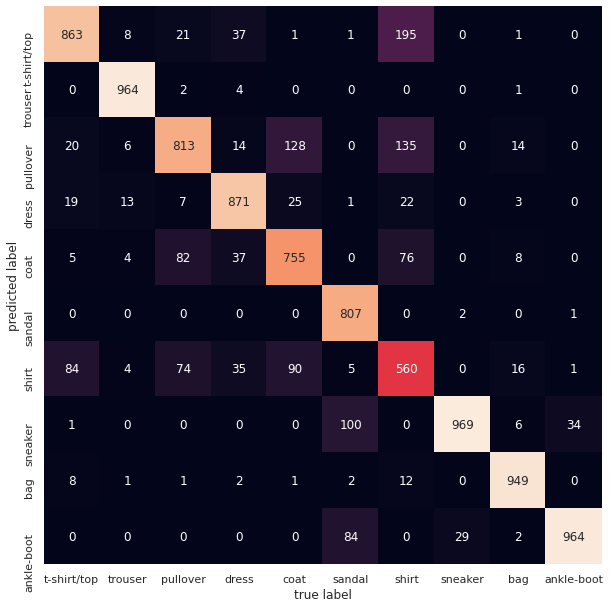
Another model tested was the XGBoost. This model scored similarly to the Random Forest resulting in an accuracy of 0.86. However, this model took significantly longer to run: 1524 seconds. The loss in efficiency and slightly lower accuracy place this model ahead of the CatBoost, but beneath the Random Forest in recommendation rank. The confusion matrix for this model can be found on the next page in Figure 5.



*Figure 5.* XGBoost confusion matrix.

**KNN**

A KNN was also tested and scored similarly to the Random Forest and XGBoost with an accuracy of 0.85. The KNN model took significantly longer to run: 29014 seconds and loss in efficiency and slightly lower accuracy place this model ahead of the CatBoost, but beneath the Random Forest and XGBoost in recommendation rank. The confusion matrix for this model can be found on the next page in Figure 6.

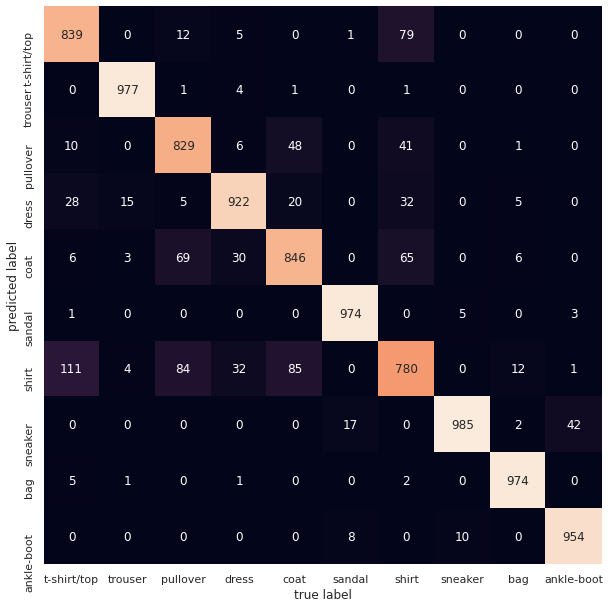


*Figure 6.* KNN confusion matrix.

**Neural Network Models**

Various approaches were taken in training the more complex convolutional neural network models using keras and tensor flow. Some techniques included preprocessing the data through rotation and shifting the images and ZCA whitening. Accuracy gains here were again found raising as high as 92%. Computational and time sacrifices were made with training time lasting 10-50 minutes based upon the preprocessing and number of epochs.

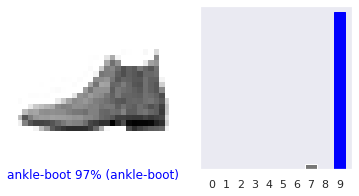
Classifying ‘shirt’ improved from the Gaussian Naïve Bayes f1-score of 0.07 to the best CNN score of 0.73-nearly a 10-fold improvement.

****

*Figure 7.* Convolutional neural network confusion matrix.

An additional model derived from <https://www.tensorflow.org/tutorials/keras/classification> was built and modified achieving a suspiciously high accuracy score of 96%.

Below in figures 8 and 9, the classified image along with the predicted confidence is reported. Displayed in the bar graph, it is evident to see where the misclassifications occurred. For example, there is 69% confidence for the coat in the lower right quadrant largely due to misclassifying it as a pullover. Visually, this would be a hard challenge for a human eye to decipher and clearly presented challenges for the model.

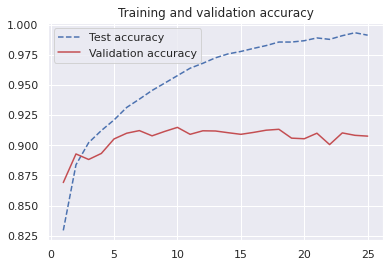
****

*Figure 8.* CNN model classifies this image as an ankle-boot with 97% confidence.

****

*Figure 9.* CNN model classifies images with varying levels of confidence.

It was evident that overfitting occurred during the trained models with validation (test) plateauing at around 91% accuracy as depicted in Figure 10. Beyond epoch 7, very little was improved upon.



*Figure 10.* CNN model test vs validation accuracy.

*Tradeoffs*

While the convolutional neural network models vastly outperformed the boosted trees, KNN and naïve bayes classifiers in accuracy, the gain came with a cost in computational and time constraints when training the model. In order to maximize accuracy, training time was needed and with a larger dataset or less compute power, these limitations could prove to be prohibitive. More complex models can and should be trained with a more powerful computer given the goal of maximizing accuracy for a larger scale computer vision problem. Below, 9 of the different models are displayed with their results and recommendations.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Accuracy** | **Train Time/ Compute Usage** | **Recommendation** |
| Gaussian NB | 59% | Fast/Minimal | 9 |
| CNN | 92% | Slow/High | 1 |
| CNN w/ Augmentation | 91% | Slow/High | 3 |
| CatBoost | 73% | Moderately Fast/Low | 8 |
| RandomForest | 88% | Fast/Low | 2 |
| CNN 1D Array | 91% | Slow/High | 3 |
| KNN | 85% | Slow/High | 7 |
| Pytorch 25 Epoch | 92% | Slow/High | 3 |
| Pytorch 150 Epoch | 91% | Slow/High | 6 |
| Basic AutoML (22) | 70% | Slow/High | 10 |

*Table 2.* A list model results, train time and overall recommendation.

The decision-making logic for the recommendation started with overall accuracy percentage. From there, a consideration was taken for the compute time/usage. It would be recommended to use the Keras CNN or PyTorch to optimize on accuracy despite the lengthy train time. However, if the business problem dictated the need for a quick model with solid accuracy, a random forest could be considered. This lead to the recommendation of the random forest over some of the lengthier but slightly more accurate neural network models. Additionally, AutoML models were trained and tested (total 22). This approach was not beneficial but was a good learning experience with the objective of trying something new and competing against the max models run without writing a loop to do so. Parameters could be adjusted, but would require greater train time and would likely yield less value than some of the CNN models.

**Conclusion & Future Directions**

The model results were solid for a novice data scientist but would not hold up for safety standards in an autonomous Tesla or even for the infamous “Tesla Bot”. This model can and would save countless hours of hand labeling images for a retail store and there were limited examples of gross misclassifications that would be potentially disruptive for the consumer (labeling a shoe a shirt). Still, improvements mentioned in the tradeoffs can and should be considered. Research demonstrated the Google’s AutoML was able to score 3-4pp higher than the CNN recommended in this analysis, but was expected to take 24 hours to run completely. Published research papers classifying this dataset can be found using this link <https://paperswithcode.com/paper/fine-tuning-darts-for-image-classification> (~97% accuracy). Many of the top examples found outside of a research team were in the range of 94% accurate leaving clear room for growth, but within the striking range of the CNN models explored in this paper. More work can be done ot augment that data in order to maximize the amount of information derived from the images available at the expense of compute time. A recommendation for the business partner requesting this algorithm might be to set a threshold of confidence (i.e., 80%) and accept the model predictions as labels above that interval and the remainder may need a more granular or time intensive approach (the pre-model baseline).