Aneeka Latif
Lab 9
IST 718

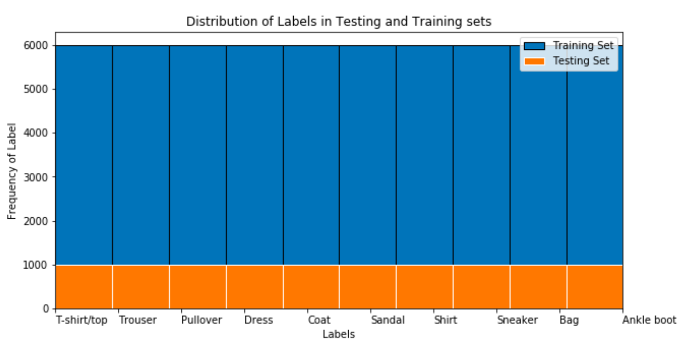
Introduction

Image recognition is used in many different industries: in online banking for mobile deposits, facial recognition as passwords on electronic devices, and vehicle identification as part of enforcing driving regulations by law enforcement. In this assignment, we will compare several classification techniques on accuracy and speed.

About the data

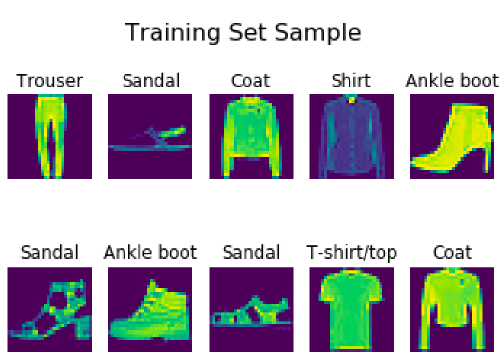
The data is an MNIST Fashion dataset from Zalando consisting of 28x28 pixel sized images of clothing items such as t-shirts, boots, and coats. The data is already split into a training set of 60,000 observations, and a testing set of 10,000 observations.

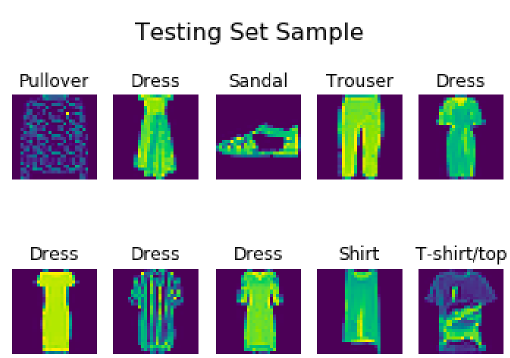
The distribution of classes (clothing items) between the training and testing sets are even, and the sets also have an equal distribution of class observations within themselves.



Distribution image included for *posterity* despite doing little to add to the exploratory analysis

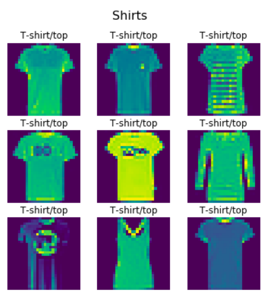
Here's a random sample of 10 clothing items from both the training and testing sets:

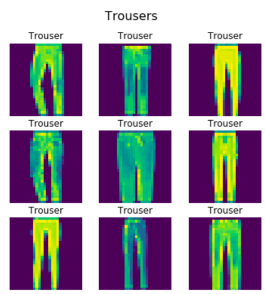


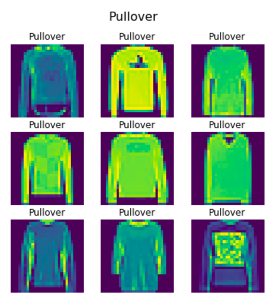


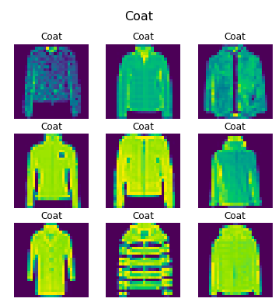
Looking at each of the clothing items, classes that have more variability in shapes will likely cause issues for our classifiers. It appears that male and unisex items generally have a fairly uniform shape for each class. The female clothing items like dresses, sandals (heels), and accessories (bags) have much more variety in shape and style. We should expect to have more difficulty in classifying dresses, sandals, and bags. I think we will also have some difficultly differentiating between Shirt vs T-shirt/Shirts and Coat vs Pullover because those categories have much overlap.

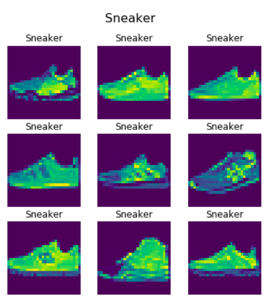
Uniform shape and style:

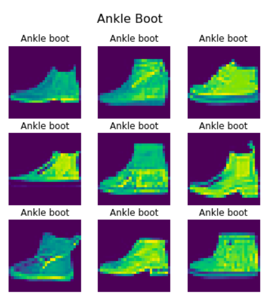


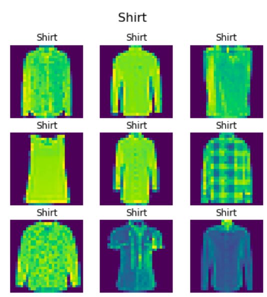






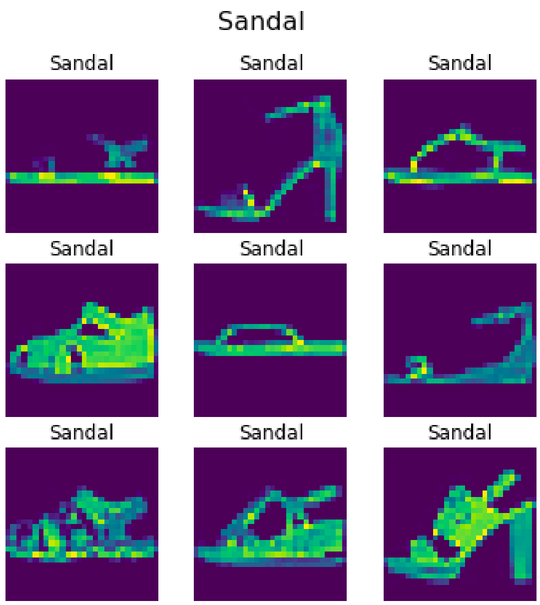


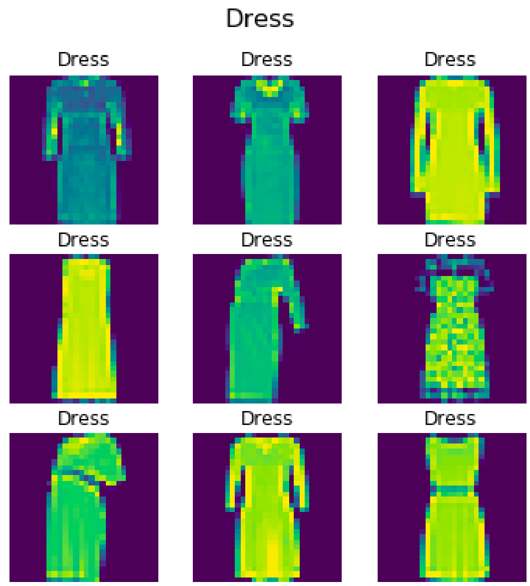


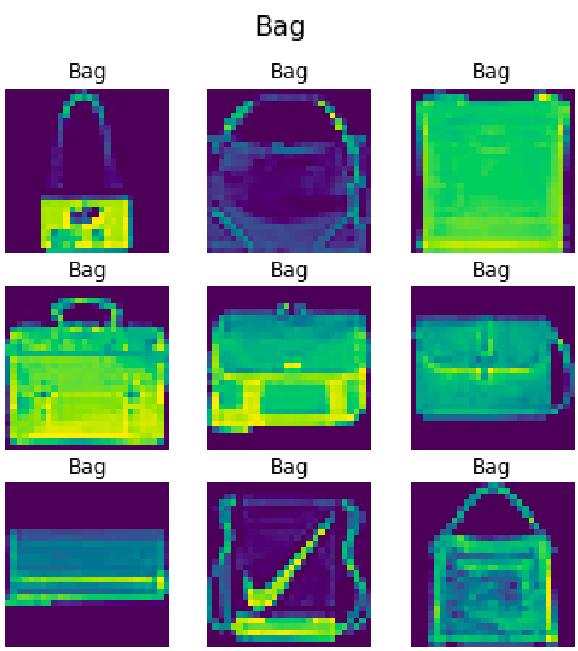


Non Uniform shape and style:

Aneeka Latif
Lab 9
IST 718

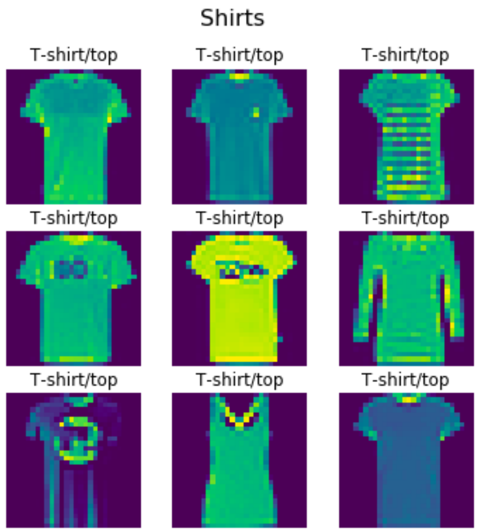




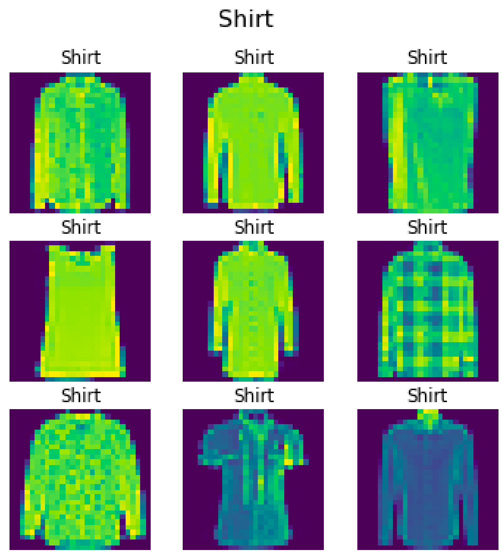


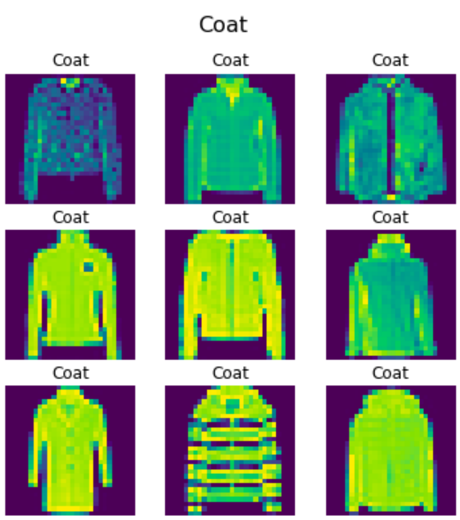
Similar categories:

Aneeka Latif
Lab 9
IST 718

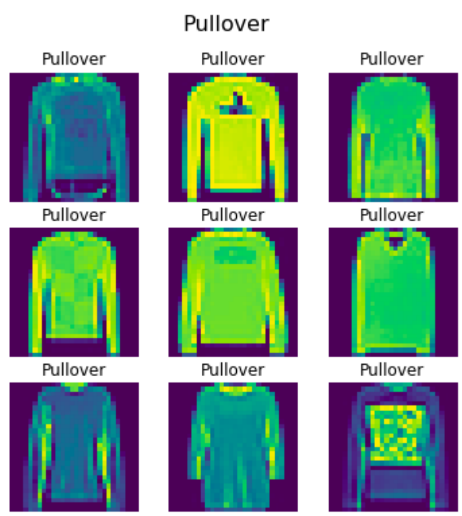


VS





VS



Preprocessing:

The MNIST data is already cropped to appropriate zoom levels and is centered on each image. Minimal preprocessing steps were taken – the images were normalized by dividing each array by 255. In some cases, the image datasets were transformed into smaller dimensional arrays required for certain classifiers. To calculate the processing time for each model, I used python's /time/ method by calculating the delta time prior to model fit and after model fit.

Model 1 / Keras Sequential Model

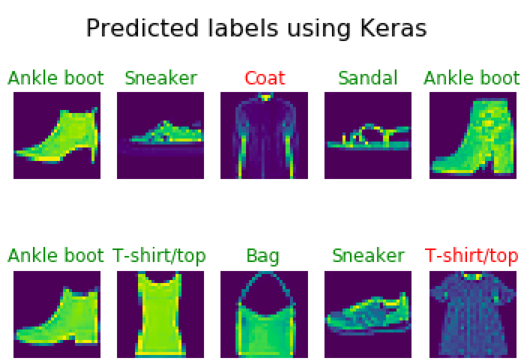
Aneeka Latif
Lab 9
IST 718

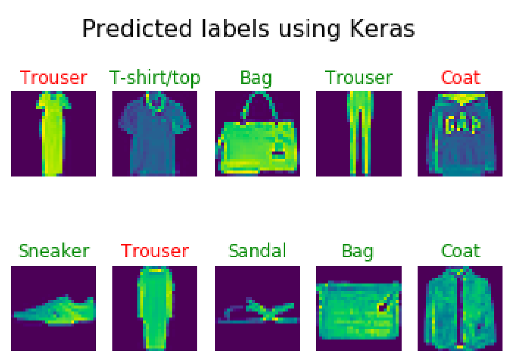
For this model, I compared the accuracy of the unnormalized images with the normalized image dataset. Model 1 is a simple keras model with a few rectified linear unit layers. For better accuracy, I simply increased the number of epoches (iterations over the dataset) which also resulted in increased training time. The loss function used to calculate accuracy is the Sparse Categorical Entropy method because our target classes are integers and we are not usine one-hot encodings.

	Number of Epoches	Accuracy	Loss	Time
Normalized Images	15	88.76%	0.7233	105.31 seconds
Normalized Images	35	89.1%	0.573	227.41 seconds
Non Normalized Images	15	85.81%	0.424	90.32 seconds
Non Normalized Images	35	85.05%	0.5389	210.01 seconds

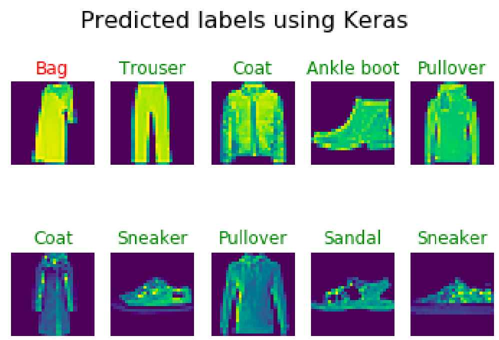
The predictions are calculated using Keras's softmax method which returns arrays of probabilities on how likely the observation is each class: each observation has an array of 10 probabilities – the model's prediction of the observation is the index of the highest probability. To get this number, I used argmax.

Plotting random samples with predicted labels – green for correct and red for incorrect labels – we can see that we do have some issues classifying dresses and bags.

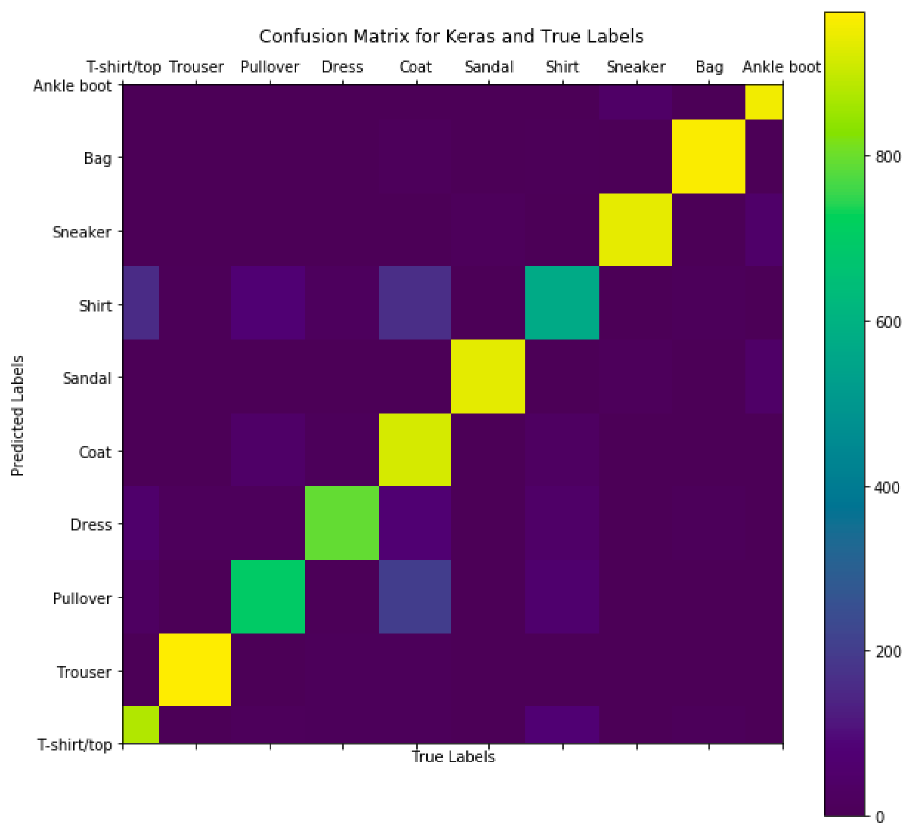




Aneeka Latif
Lab 9
IST 718



Looking at a confusion matrix of our real labels and predicted labels: we can see that we understandably confuse shirts with t-shirt/tops and coats with pullovers. Those garments are very similar and oftentimes, those names are used interchangeably for both. Otherwise, the model performed well and has generally classified clothing items correct with very limited and expected confusion.



Despite the troubles, the keras model was very simple and straightforward, didn't need any preprocessing, and was very fast. The model only struggled with categories that are expected to be difficult to differentiate. It was also reasonably accurate with only default parameters.

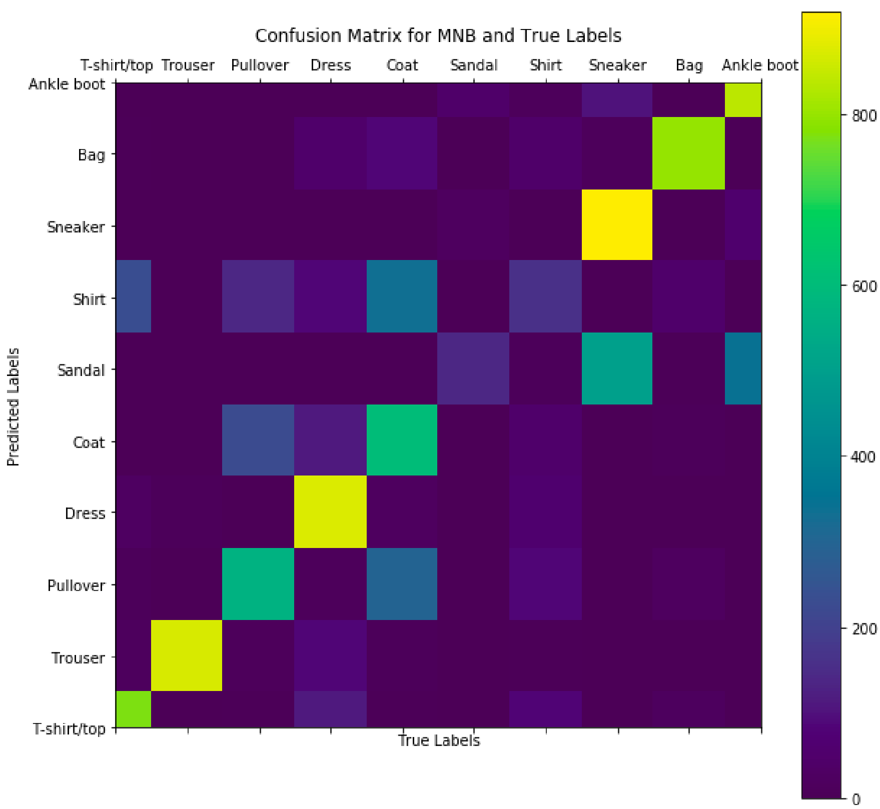
Aneeka Latif
Lab 9
IST 718

Model 2 / Multinomial Naïve Bayes

The Multinomial Naïve Bayes model requires the input data to be a maximum shape of two dimensions, since our input data is in three dimensions, we need to transform it to an appropriate shape. Accuracy is calculated using sklearn's accuracy method.

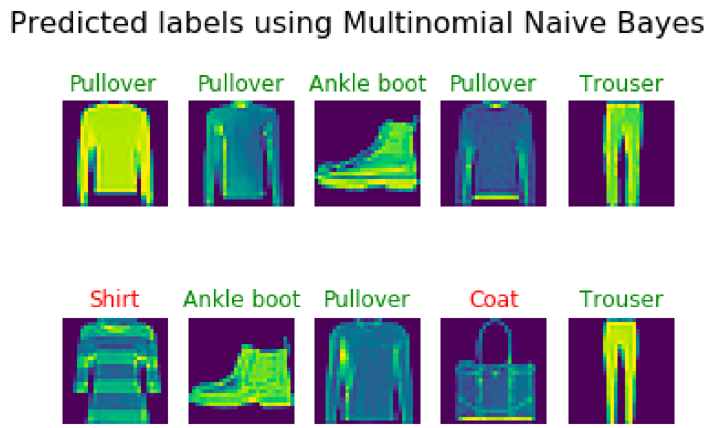
MNB Parameters	Accuracy	Time
alpha=1.0, class_prior=None, fit_prior=True	65.52%	0.25 seconds
alpha=0.8, class_prior=None, fit_prior=False	65.52%	0.23 seconds
alpha=0.2, class_prior=None, fit_prior=True	65.52%	0.246 seconds

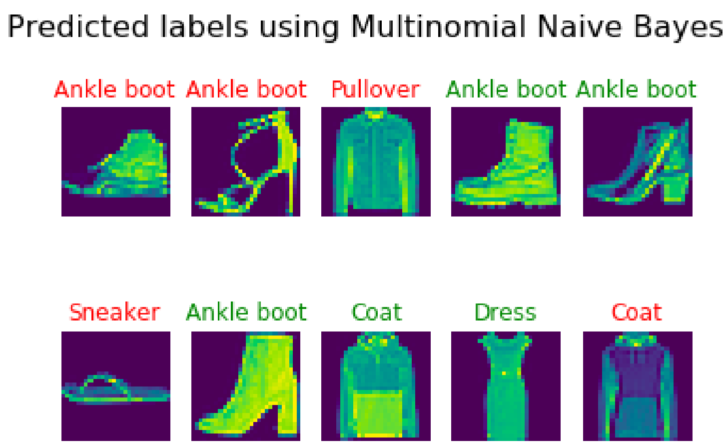
Plotting the confusion matrix – we can see that the model only did well in classifying sneakers, dresses, and trousers. Shirts and sandals were especially poor. Since sneakers and sandals are both footwear, we can expect mild confusion. Shirts, coats, and pullovers are all very similar shapes, so that can also be justified confusion. Shirts, however, should never be misclassified as bags; they are wildly different shapes. We don't see a straight yellow/green diagonal line through this matrix so it's clear even without the accuracy metrics that this model performed poorly.



Aneeka Latif
Lab 9
IST 718

If we look at random samples of predicted data, we can see that sandals are getting confused for ankle boots and tops/outerwear are being confused for similarly shaped clothing items. The most egregious mistake is the bag being mistaken for a coat.





The model performed notably worse than the keras model; however, it was much faster.

Model 3 / Decision Tree

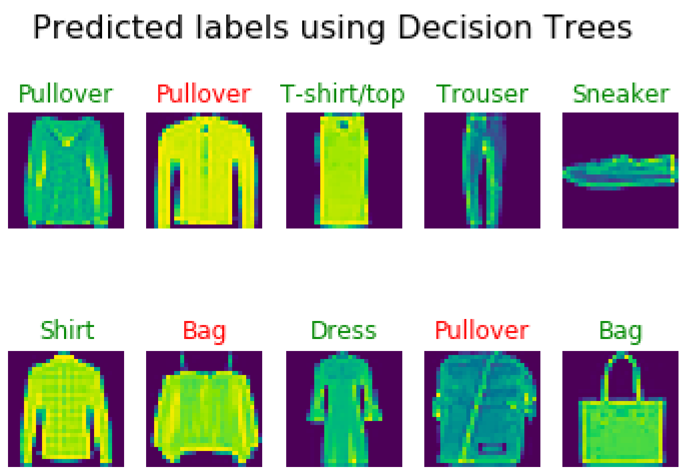
Again, the decision tree takes input that is a maximum shape of two dimensions, so we used our previously transformed image set for the MNB model. The accuracy was calculated using sklearn metric's accuracy method. Parameters we can change are the criterion which changes the evaluatio method of the splitting method, and the splitting method itself. Higher max depth prevents overfitting while lower max depth prevents underfitting the model.

Decision Tree parameters	Accuracy	Time
criterion='gini', splitter='best'	79.14%	46.88 seconds

Aneeka Latif
Lab 9
IST 718

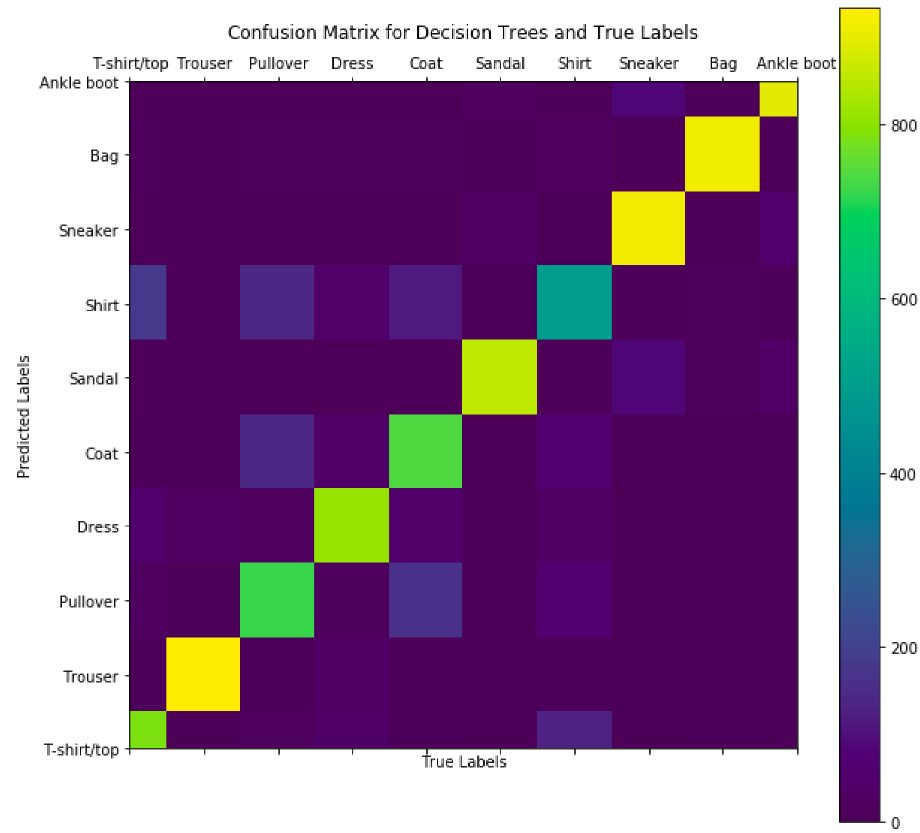
criterion='entropy', splitter='best'	80.32%	47.13 seconds
criterion='entropy', splitter='random'	79.30%	7.13 seconds
criterion='entropy', splitter='random', max_depth=3	53.66%	2.36 seconds
criterion='entropy', splitter='random', max_depth=10	80.50%	5.17 seconds
criterion='entropy', splitter='random', max_depth=13	80.71%	6.00 seconds

Looking at a random set: we seem to have the same expected issues with footwear, bags, and dresses.



Our confusion matrix indicates that this is a more accurate model than the MNB model but is still not as accurate as the Keras model. We can see increased misclassification in the center and towards the lower lefthand quadrant of the matrix.

Aneeka Latif
Lab 9
IST 718



Conclusion

There are many different ways to classify images, some require more pre-processing and data manipulation than others: in example, the keras model required virtually no preprocessing, but the MNB model and the Decision tree models required the data to fit dimensionality equal to or lower than 2. The keras model offers more options in the form of different layers and ultimately infinite number of combinations/additional layers for model tuning. For the MNB model, we could really only adjust the alpha parameter, and the flag for prior_fit. Adjusting the parameters for the MNB had no effect on the accuracy. Adjusting the keras parameters had a very clear increase in accuracy. The decision tree model parameter tuning resulted in the widest range of accuracy values.

Best Models	Accuracy	Time (seconds)	Advantage	Disadvantage
Normalized Images / Keras	89.1%	227.41	Most accurate	Slowest
alpha=0.8, class_prior=None, fit_prior=False / MNB	65.52%	0.23	Fastest	Least accurate
criterion='entropy', splitter='random', max_depth=13 /DT	80.71%	6.00	Reasonably accurate Reasonably fast	Prone to over/underfitting

Aneeka Latif
Lab 9
IST 718

In terms of accuracy, the best model was the Keras model, it was also the most straightfoward and simplest to execute. The only con is that it did significantly longer time to process than the other two models. A close second is the decision tree, it was ~10% less accurate but 35 times faster than the Keras sequential model. The multinomial naïve bayes model, was the fastest but had the poorest accuracy. The decision tree model seems like the best compromise, it was reasonably quick and reasonably accurate. If you had the time to spare, I would recommend using the keras sequential model; if you couldn't afford the processing time, the decision tree model would be the next best choice.

Citations

Zalandoresearch. (2019, August 10). zalandoresearch/fashion-mnist. Retrieved from https://github.com/zalandoresearch/fashion-mnist