# Capstone Project 1
# Final Report
# Clothing Categorization

Parker Williamson
3/7/2018
Springboard Data Science Career Track

# Goal

- Correctly categorize clothing images

# Benefits/Customers

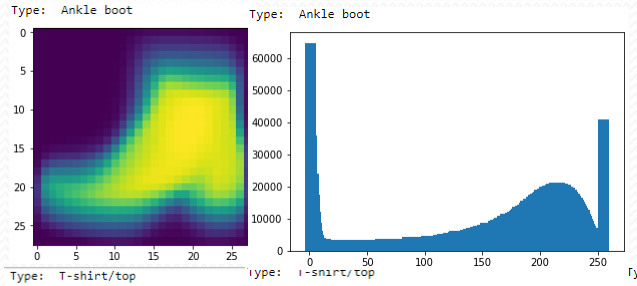
- Online sellers of clothing
- Eventually clothing folding/sorting

# Datasets

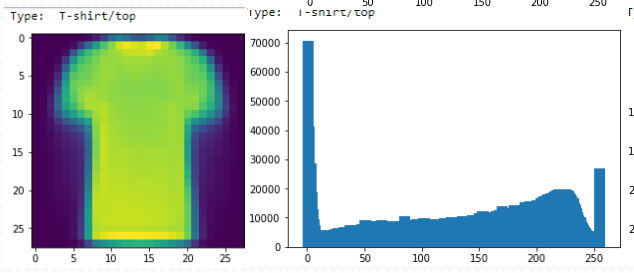
- Fashion.mnist
- A Kaggle clothing dataset was the dataset I used to train and compare the classifiers

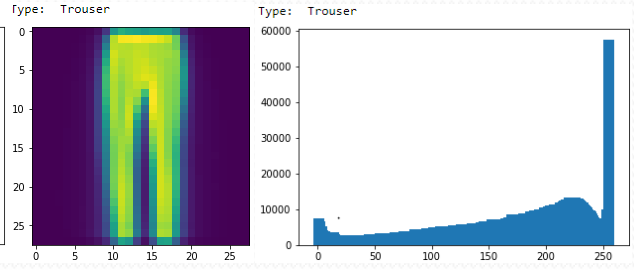
- https://www.kaggle.com/zalando-research/fashionmnist/data

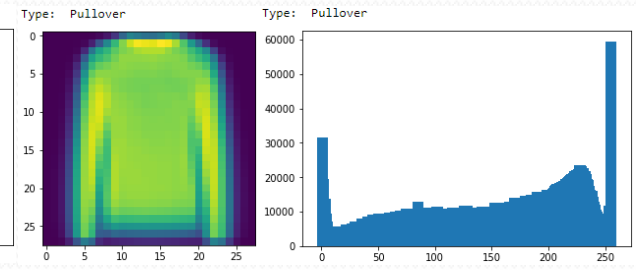
# Data Wrangling

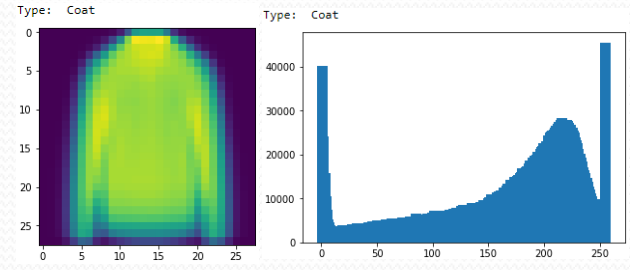


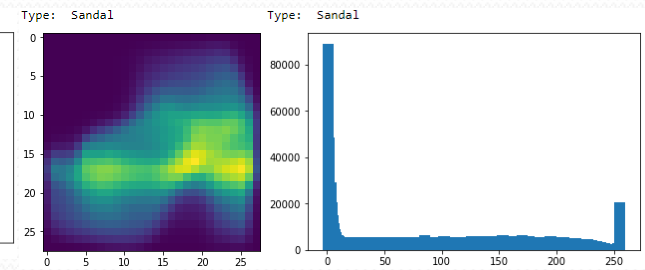

- Average images and histograms of each of the image classes are shown

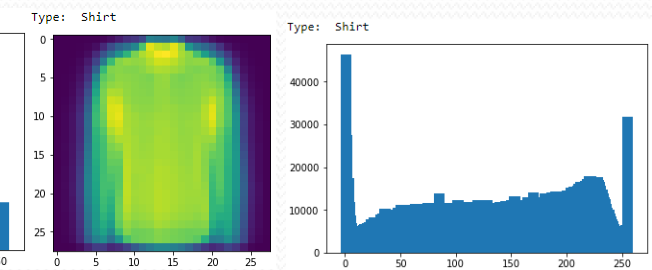


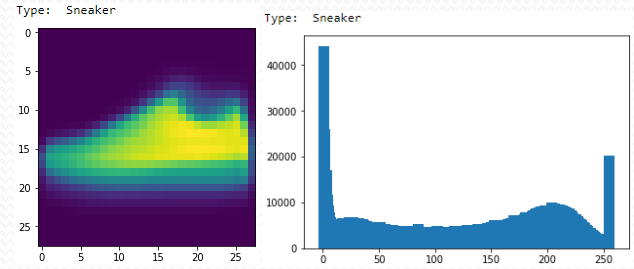



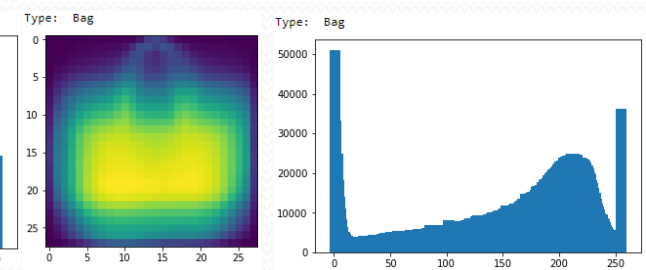



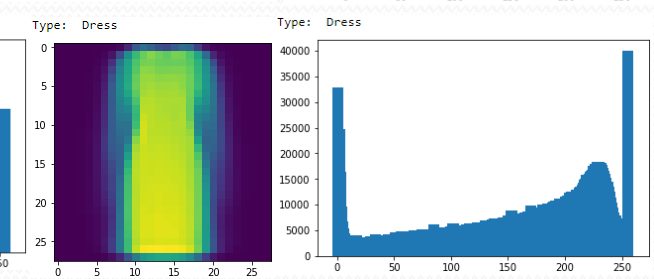



# Statistical Analysis

- Compared the order means to see if any of them were statistically similar using a paired z-test
- If the means could be the same the p-score would be about .05

```
p-score 0-1(Z): 0.0
p-score 1-2(Z): 0.0
p-score 2-3(Z): 0.0
p-score 3-4(Z): 0.0
p-score 4-5(Z): 0.0
p-score 5-6(Z): 0.0001
p-score 6-7(Z): 0.0
p-score 7-8(Z): 0.0
p-score 8-9(Z): 0.0
```

# Preprocessing

- Normalization (with a range of 0-1)
  - Make the range of values fill and be between 0-1
- Standardization
  - Spread out the data more evenly by scaling it using the standard deviation and mean
  - (Data –mean(Data))/standard deviation

# Analysis

- MLP (size of hidden layers 784-100-100):
  - With no preprocessing – 87.65% accuracy
  - Normalized(0-1) – 90.37% accuracy
  - Standardized(0-1) – 90.09% accuracy
- SVC:
  - With no preprocessing – 70.67% accuracy
  - Normalized(0-1) – 85.57% accuracy
  - Standardized(0-1) – 81.87%
- Logistic Regression:
  - With no preprocessing – supposed to be normalized
  - Normalized(0-1) – 84.45% accuracy
  - Standardized(0-1) – 85.19% accuracy
- CNN:
  - Normalized(0-255) – 92.25% accuracy
  - Standardized – **92.32% accuracy**
- Ensemble the above models (Random Forest with 20 trees):
  - 86.47% accuracy
- Use VGG19 as a feature extractor before a neural net:
  - 85.71% accuracy

# Analysis – Confusion Matrices

X – Predicted

Y – Actual (top is 0) Combined

Combined

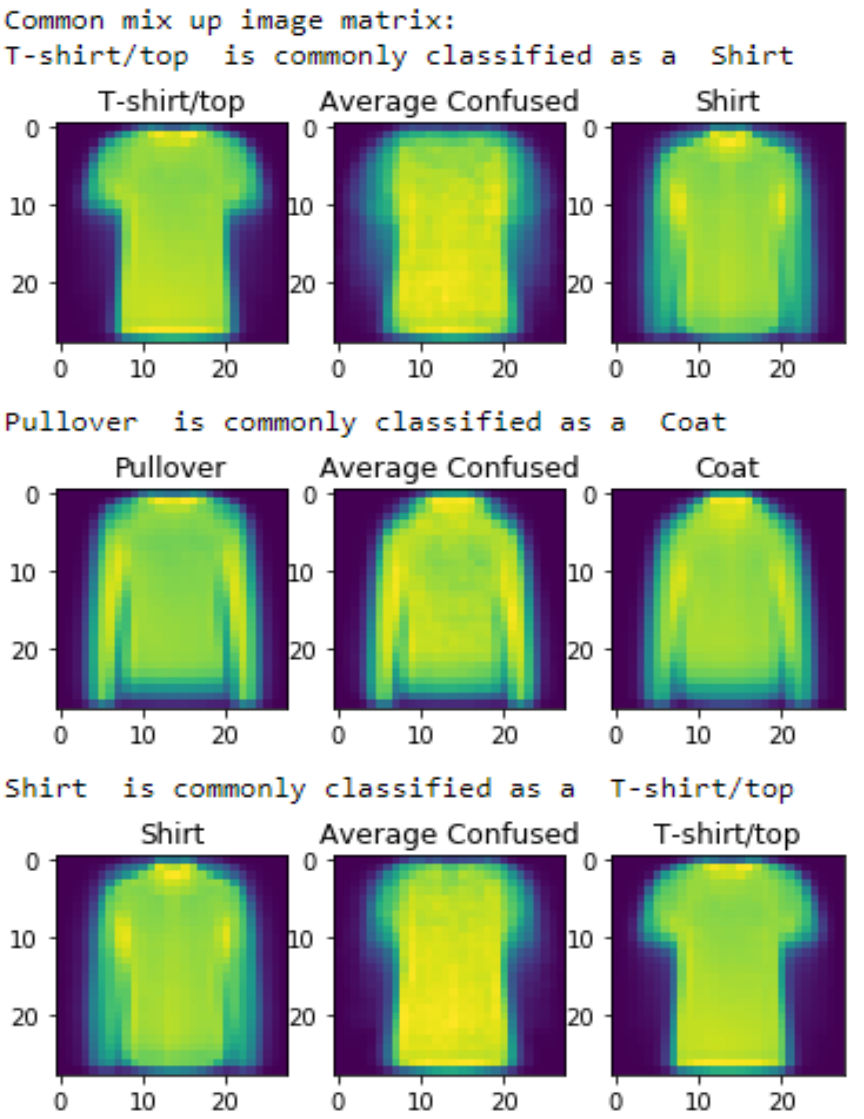
```
[[3378   13   63  126   12    7  349    1   51    0]
 [  14 3910    9   45    2    2   16    2    0    0]
 [  66   13 3173   49  393    0  276    0   30    0]
 [  98   53   39 3606   93    4   91    2   14    0]
 [   7    9  207  124 3388    2  252    0   11    0]
 [   4    6    1    0    0 3738    0  147   22   82]
 [ 562   19  308  136  282    1 2638    0   54    0]
 [   0    0    0    0    0   82    0 3747    2  169]
 [  18    8   25   21   10   23   41   13 3838    3]
 [   1    0    0    0    2   28    0  129    5 3835]]
```

CNN*4

```
[[3456    0   48   52    0    4  412    0   28    0]
 [   8 3944    4   32    4    0    8    0    0    0]
 [  48    0 3396   52  232    0  268    0    4    0]
 [  36   16   12 3764   60    0  112    0    0    0]
 [   0   12   64   92 3580    0  248    0    4    0]
 [   0    0    0    0    0 3908    0   48    8   36]
 [ 296    0  100   68  140    0 3384    0   12    0]
 [   0    0    0    0    0   20    0 3836    0  144]
 [  16    8    8    0    8    4   20    8 3924    4]
 [   0    0    0    0    0    4    0   80    0 3916]]
```

# Visualizations

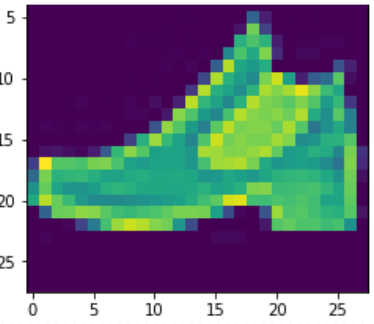
- The commonly confused categories (above 95 percentile) are show with an average of the correct class on the right the average of all the confused images in the middle and the average image of the class it was classified as

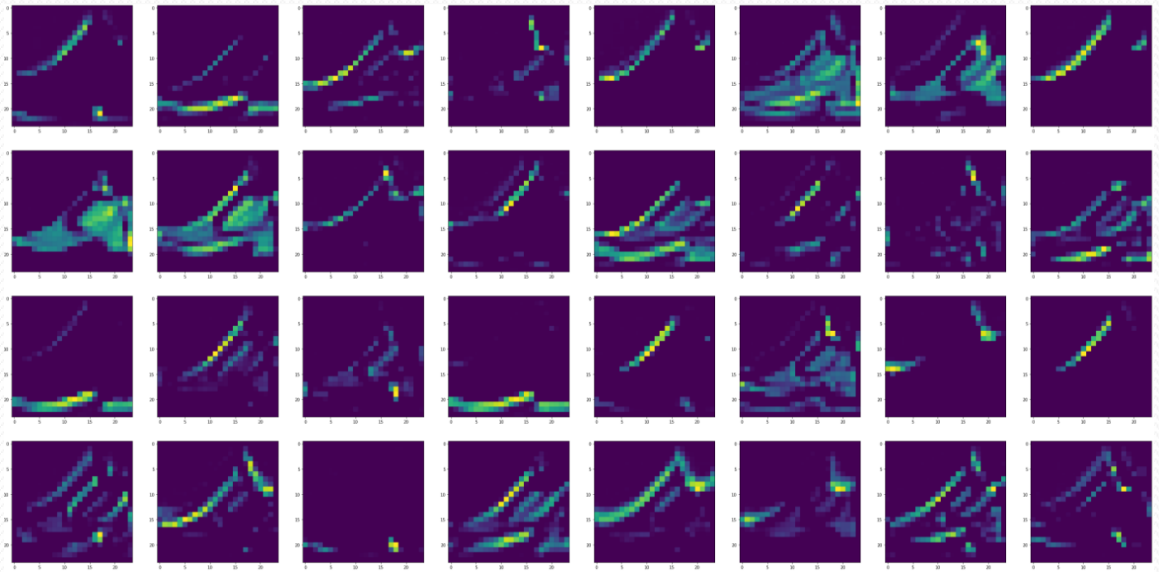


# Visualizations – CNN in progress

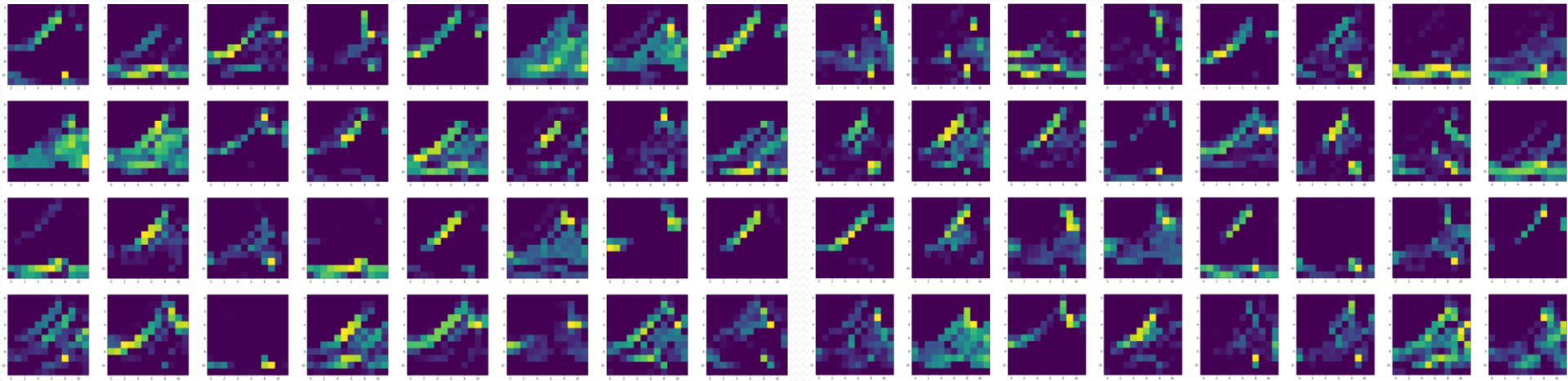
PHASE 1

Original image

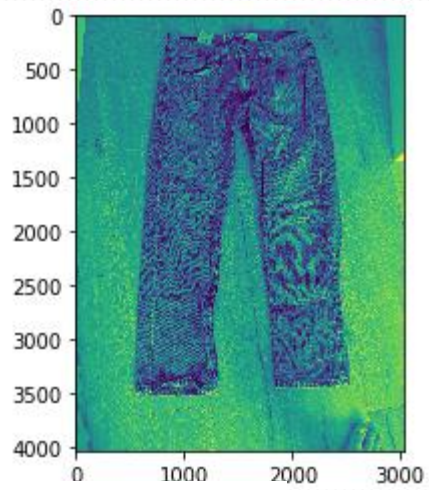


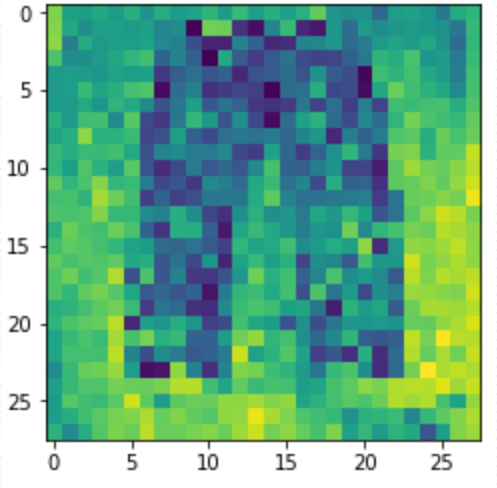



PHASE 2

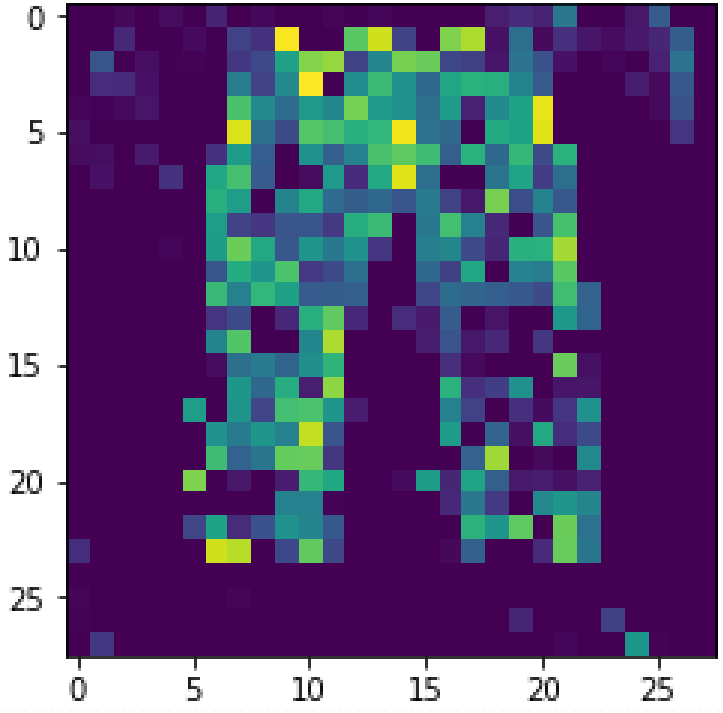


# Testing with my Photos



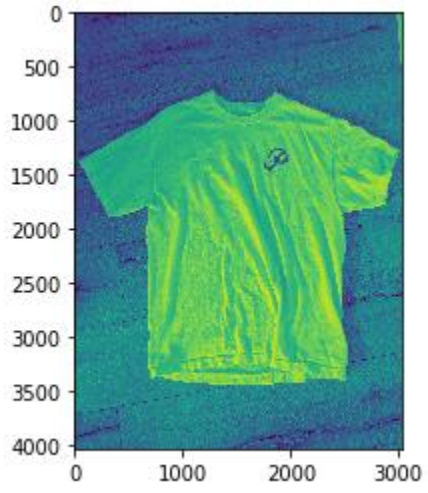



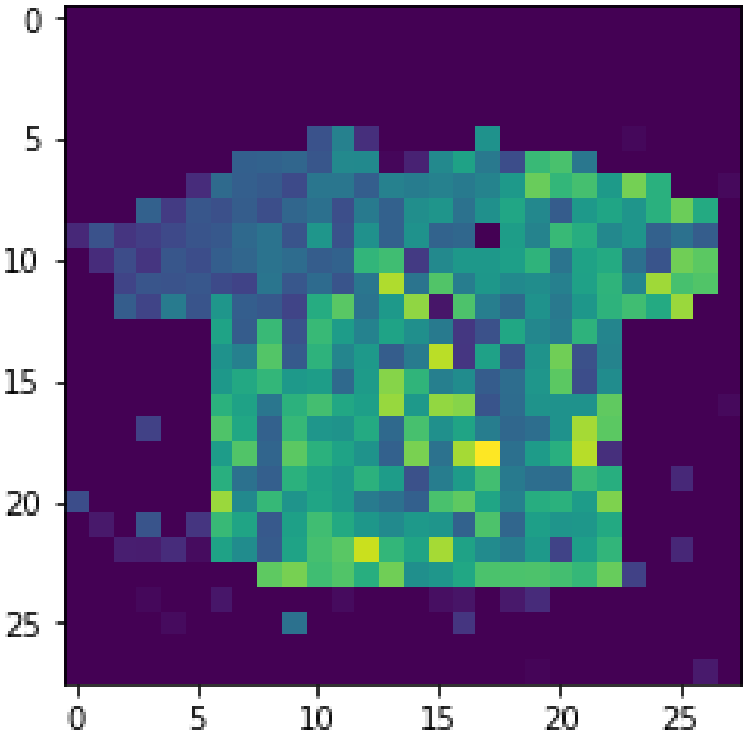

- MLP classified it as a coat, SVM and LogReg both correctly said trousers and CNN said shirt

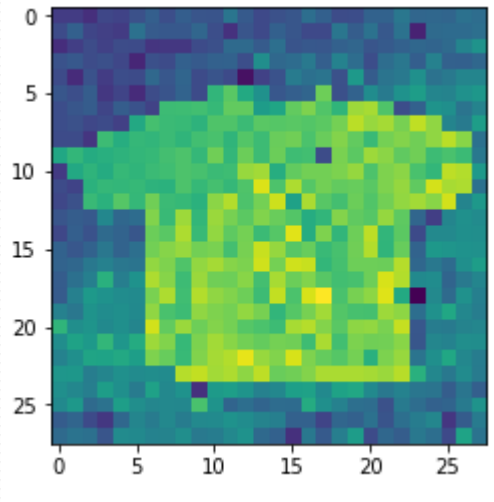


# Testing with my Photos

- MLP classified it as a bag, whereas SVM, LogReg and CNN classified it as a shirt



# Testing with my Photos

- Either make sure to train on images filtered in the same way or look out for overfitting with MLP and CNN (MLP is the main classifier that seems it could be overfitting)
- If training on slightly different images, SVM and LogReg may have more robust results