**Project Proposal**

Image processing for the apparel and fashion industry

**Problem:**

Be able to identify images of clothing items from targeted social media sites.

An apparel/fashion company wanting to start a new marketing campaign targeting users of social media such as Pinterest and Instagram.

People who post a lot of apparel pictures to **Pinterest or Instagram** may be planning a purchase or an event like a vacation. By identifying these images, the apparel/fashion company can develop:

* Models that can identify what makes a *blue blouse,* can use the information to recommend products that are visually similar.
* Understanding a person’s style in clothing can also help target them for related products.
* A following on these social media sites.

Also identify people who are following specific sites or people on Pinterest and Instagram that focus on apparel, can be potential customers

**Data Wrangling Techniques used:**

* Data Overview
* Process
  + Load data – both training and test datasets
    - Split data between Features (X) and Labels (y)
    - Convert to Numpy Array
  + Basic data exploration
  + Slicing and dicing
    - Getting specific subset of the data (3 images) all occurrences
    - Getting specific subset of data for (3 images), only first occurrence
    - Getting first 3 items of data and processing it for plotting
  + Reduce the dimensionality of a data set (reduce noise and improve performance)
* Summary

**Data Overview**

The data used for this project is from [Kaggle](https://www.kaggle.com/zalando-research/fashionmnist). It consists of training data and test data in CSV format. The data consist of 10 different classifications:

0-Tshirt/top 1-Trousers 2-Pullover 3-Dress 4-Coat

5-Sandal 6-Shirt 7-Sneaker 8-Bag 9-Ankle boot

**Process**

**Load Data**

Used the pandas library to read the two data files (csv format), stored as data frames (DF).

Ex: df\_training = pd.read\_csv('C:\\U…’)

Split data between Features (X) and Labels (y).

Ex: df\_X\_train = df\_training.iloc[:, 1:]

df\_y\_train = df\_training.iloc[:, :1]

Converted the DF to numpy arrays

Ex: X\_train = df\_X\_train.as\_matrix()

y\_train = df\_y\_train.as\_matrix()

**Basic data exploration**

Identifying the shape of the numpy array and understanding of the data types.

Ex: X\_train.shape

X\_train.dtype

**Slicing and Dicing**

Getting specific subset of the data (3 images) all occurrences

Ex: test\_subset = df\_test.loc[df\_test['label'].isin([1,3,7])]

Getting specific subset of data for (3 images), only first occurrence

Ex: frst\_img = np.where(y2\_train==1)[0][0]

secnd\_img = np.where(y2\_train==3)[0][0]

thrd\_img = np.where(y2\_ train ==7)[0][0]

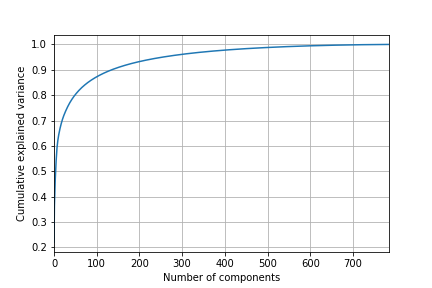
Getting first 3 items of data and processing it for plotting, using enumerate and zip functions

Ex: for index, (image, label) in enumerate(zip(X\_train[0:3], y2\_train[0:3])):

**Reduce the dimensionality of a data set**

To be able to process large datasets used principal component analysis (PCA) to reduce the dimensionality of a data set. This is an Unsupervised clustering algorithm.

**PCA data based on Training Data**



Used sklearn libraries.

Ex: sklearn50\_pca = PCA(n\_components=50)

Xtest\_50 = sklearn50\_pca.fit\_transform(X\_std)

**Summary**

The data was clean, *wrangling* was focused on different methods of slicing and dicing needed for different operations. But the major processing of the data was implementingprincipal component analysis (PCA) to reduce the dimensionality of a data set