Project 3 : Part 2

Luis Garduno

1. About Fashion MNIST Dataset

Dataset: Fashion MNIST

---

2. Data Preparation

2.1 Loading Data & Adjustments

In [1]:

```
import tensorflow as tf
from tensorflow import keras

fashion_mnist = keras.datasets.fashion_mnist
(X_train, y_train), (X_test, y_test) = fashion_mnist.load_data()
```

```
2022-04-12 06:55:34.248407: I tensorflow/stream_executor/platform/default/dso_
loader.cc:49] Successfully opened dynamic library libcudart.so.10.1
```

In [2]:

```
class_names = ["T-shirt/top", "Trouser", "Pullover", "Dress", "Coat", "Sandal",
"Shirt", "Sneaker", "Bag", "Ankle boot"]
```

In [3]:

```
from sklearn.preprocessing import StandardScaler

X_test, X_train = X_train[:5000] / 255.0, X_train[:5000] / 255.0
y_test, y_train = y_train[:5000] / 255.0, y_train[:5000] / 255.0

#X_train = X_train.reshape(X_train.shape[0], 1, 28, 28) / 255.0 - 0.5
#X_test = X_test.reshape(X_test.shape[0], 1, 28, 28) / 255.0 - 0.5

# One-Hot Encoding
y_train_OHE = keras.utils.to_categorical(y_train, 10)
y_test_OHE = keras.utils.to_categorical(y_test, 10)
```

In [4]:

```
from sklearn.linear_model import LogisticRegression, SGDClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.pipeline import make_pipeline
from sklearn import ensemble, tree
```

```
models = {}

#sgd_pipe = make_pipeline(StandardScaler(), SGDClassifier())
#sgd_pipe.fit(X_train)
#sgd_pipe.predict(X_test)

models["DecTree"] = tree.DecisionTreeClassifier()
models["RandForest"] = ensemble.RandomForestClassifier(n_estimators=250)

models["KNeighbors"] = KNeighborsClassifier()

models["SGD"] = SGDClassifier(loss="hinge", penalty="l2")
models["lbfgs_LR"] = LogisticRegression(multi_class="multinomial",
solver="lbfgs", max_iter=300)
models["newton-cg_LR"] = LogisticRegression(multi_class="multinomial",
solver="newton-cg", max_iter=300)
```

In [5]:

```
from sklearn.metrics import accuracy_score, mean_squared_error
from sklearn.linear_model import LogisticRegression as SK_LogisticRegression

ss = StandardScaler()
lr_lr = SK_LogisticRegression()

#lr_lr.fit(X_train, y_train)
#yhat = lr_lr.predict(X_test)
#print('Accuracy of:',accuracy_score(y_test,yhat))
```

In [ ]:

```

```