proposed_model

January 30, 2019

```
In [1]: import numpy
        from sklearn.utils import shuffle
        from sklearn.model_selection import train_test_split
        X = numpy.loadtxt("./data/Train/X_train.txt")
        y = numpy.loadtxt("./data/Train/y_train.txt")
        X_test = numpy.loadtxt("./data/Test/X_test.txt")
        y_test = numpy.loadtxt("./data/Test/y_test.txt")
```

```
In [3]: X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.20)
```

```
In [3]: from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
        lda = LDA(n_components=10)
        X_train_new = lda.fit_transform(X_train, y_train)
```

```
/home/mahdi/.local/share/virtualenvs/machine-learning-final-project-uk2p9d2v/lib/python3.6/site-
  warnings.warn("Variables are collinear.")
```

```
In [5]: from sklearn.model_selection import cross_val_score
        from sklearn.neural_network import MLPClassifier
        clf = MLPClassifier(hidden_layer_sizes=(54), activation='tanh')
        clf.fit(X_train_new, y_train)
        validation_Score = clf.score(lda.transform(X_valid), y_valid)
```

```
/home/mahdi/.local/share/virtualenvs/machine-learning-final-project-uk2p9d2v/lib/python3.6/site-
  % self.max_iter, ConvergenceWarning)
```

```
In [7]: print("validation score: {}".format(validation_Score))
```

```
validation score: 0.9652509652509652
```

```
In [8]: X_train = X
        y_train = y
        X_train_new = lda.fit_transform(X_train, y_train)
        print("shape of X_train is :", X_train_new.shape)
        clf.fit(X_train_new, y_train)
```

```
/home/mahdi/.local/share/virtualenvs/machine-learning-final-project-uk2p9d2v/lib/python3.6/site-
  warnings.warn("Variables are collinear.")
```

```
shape of X_train is : (7767, 10)
```

```
/home/mahdi/.local/share/virtualenvs/machine-learning-final-project-uk2p9d2v/lib/python3.6/site-
  % self.max_iter, ConvergenceWarning)
```

```
Out[8]: MLPClassifier(activation='tanh', alpha=0.0001, batch_size='auto', beta_1=0.9,
               beta_2=0.999, early_stopping=False, epsilon=1e-08,
               hidden_layer_sizes=54, learning_rate='constant',
               learning_rate_init=0.001, max_iter=200, momentum=0.9,
               n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
               random_state=None, shuffle=True, solver='adam', tol=0.0001,
               validation_fraction=0.1, verbose=False, warm_start=False)
```

```
In [9]: ## this is just for evaluate model performance on test data, and will be used in submitti
        from sklearn.metrics import accuracy_score
        X_test_new = lda.transform(X_test)
        y_pred = clf.predict(X_test_new)
        score = accuracy_score(y_test, y_pred)
        print("test score is : ", score)
```

```
test score is :  0.9869962662546672
```