Day 18

9:42 pm

Go over Handwritten Digit recognizer python code.

Day 14 9:42pm Mon Dec

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn import metrics
import joblib
from sklearn.decomposition import PCA
import numpy as np
from sklearn.utils import shuffle
```

useful libraries

need to install

```
dataframe = pd.read_csv('csv/dataset5labels.csv')
dataframe = dataframe.sample(frac=1).reset_index(drop=True)
```

reading data to use

shuffle the data

```
print dataframe
```

Assigning features and labels

```
X = dataframe.drop(['label'], axis=1)
Y = dataframe['label']
```

eg: if data is

label
1
1
1
...
2
2
...

becomes

label
2
1
3
1
...

```
# X_train, Y_train =  X[0:198], Y[0:198]
# X_test,Y_test = X[198:],Y[198:]
X_train, Y_train =  X, Y
X_test,Y_test = X,Y
```

Training dataset

Test dataset

```
grid_data = X_train.values[40].reshape(28,28)
plt.imshow(grid_data,interpolation=None,cmap="gray")
plt.title(Y_train.values[40])
plt.show()
```

Why is the same dataset used for testing and training?

Isn't it cheating. Don't do this and let's see the reason

The value should be the same. Trying to verify if the labels and figures unzipped

show value of X_train at the index 40

```
model = svm.SVC(kernel="linear",C=2)
print "Fitting this might take some time ....."

model.fit(X_train,Y_train)

joblib.dump(model, "model/svm_0to5label_linear_2")
#model = joblib.load("svm_class_1")
print "predicting ....."
predictions = model.predict(X_test)

print "Getting Accuracy ....."
print "Score", metrics.accuracy_score(Y_test, predictions)
```

Training model

save the model for future use

Accuracy of predictions

Most of the libraries used were from sklearn.