**Project 5**

**Kishan Sarpangala**

Code:

%matplotlib inline

import pandas as pd

import numpy as np

from sklearn import cross\_validation

from sklearn.ensemble import RandomForestClassifier

from sklearn import metrics

import matplotlib.pyplot as plt

from matplotlib.pyplot import \*

from sklearn.datasets import fetch\_mldata

from sklearn.utils import shuffle

import time

X, y = mnist["data"], mnist["target"]

#X\_train, X\_test, y\_train, y\_test = sklearn.cross\_validation.train\_test\_split(mnist.data, mnist.target, test\_size=0.1, random\_state=0)

X\_train, X\_test, y\_train, y\_test = X[:60000], X[60000:], y[:60000], y[60000:]

shuffle\_index = np.random.permutation(60000)

X\_train, y\_train = X\_train[shuffle\_index], y\_train[shuffle\_index]

clf = RandomForestClassifier(n\_estimators=10,n\_jobs=2)

clf.fit(X\_train, y\_train)

clf.score(X\_test, y\_test)

rnd\_clf = RandomForestClassifier(random\_state=42)

rnd\_clf.fit(mnist["data"], mnist["target"])

def plot\_digit(data):

image = data.reshape(28, 28)

plt.imshow(image, cmap = matplotlib.cm.hot,

interpolation="nearest")

plt.axis("off")

plot\_digit(rnd\_clf.feature\_importances\_)

cbar = plt.colorbar(ticks=[rnd\_clf.feature\_importances\_.min(), rnd\_clf.feature\_importances\_.max()])

cbar.ax.set\_yticklabels(['Not important', 'Very important'])

plt.show()