MNIST

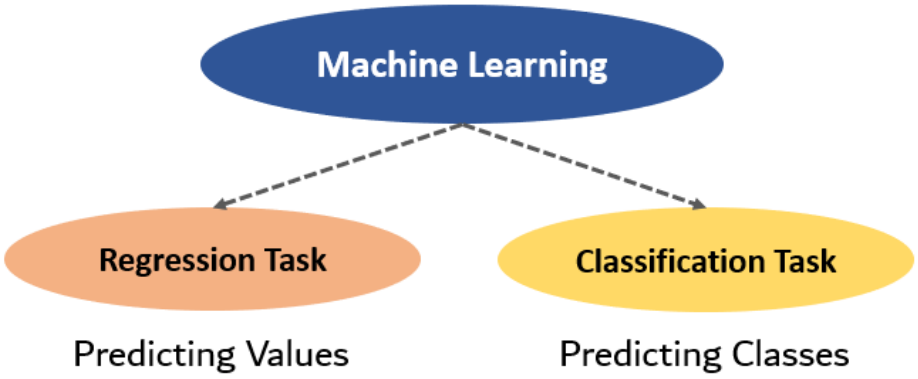
Tuesday, December 3, 2019 4:37 PM

Note Written by: Zulfadli Zainal

The most common supervised learning tasks are regression (predicting values) and classification (predicting classes).

We have done exploring regression task: **1) Linear Regression 2) Decision Trees 3) Random Forests**

Now will focus on classification task.



What is MNIST?

It's a very famous data set to study classification.

It is a set of 70,000 small images of digits handwritten by high school students and employees of the US Census Bureau. Each image is labele

Scikit-Learn provides many helper functions to download popular datasets. MNIST is one of them. The following code fetches the MNIST dataset:

```
from sklearn.datasets import fetch_openml
#Load data
mnist = fetch_openml('mnist_784')
```

```
from sklearn.datasets import fetch_openml
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

# Load data
mnist = fetch_openml('mnist_784')
```

Understand the data: MNIST data structure.

Key	Type	Size	Value
DESCR	str	1	**Author**: Yann LeCun, Corinna Cortes, Christopher J.C. Burges **So ...
categories	dict	0	{}
data	float64	(70000, 784)	[[0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.]
details	dict	14	{'id':'554', 'name':'mnist_784', 'version':'1', 'format':'ARFF', 'uplo ...
feature_names	list	784	['pixel1', 'pixel2', 'pixel3', 'pixel4', 'pixel5', 'pixel6', 'pixel7', ...
frame	NoneType	1	NoneType object of builtins module
target	object	(70000,)	ndarray object of numpy module
target_names	list	1	['class']
url	str	1	https://www.openml.org/d/554

X (Features)

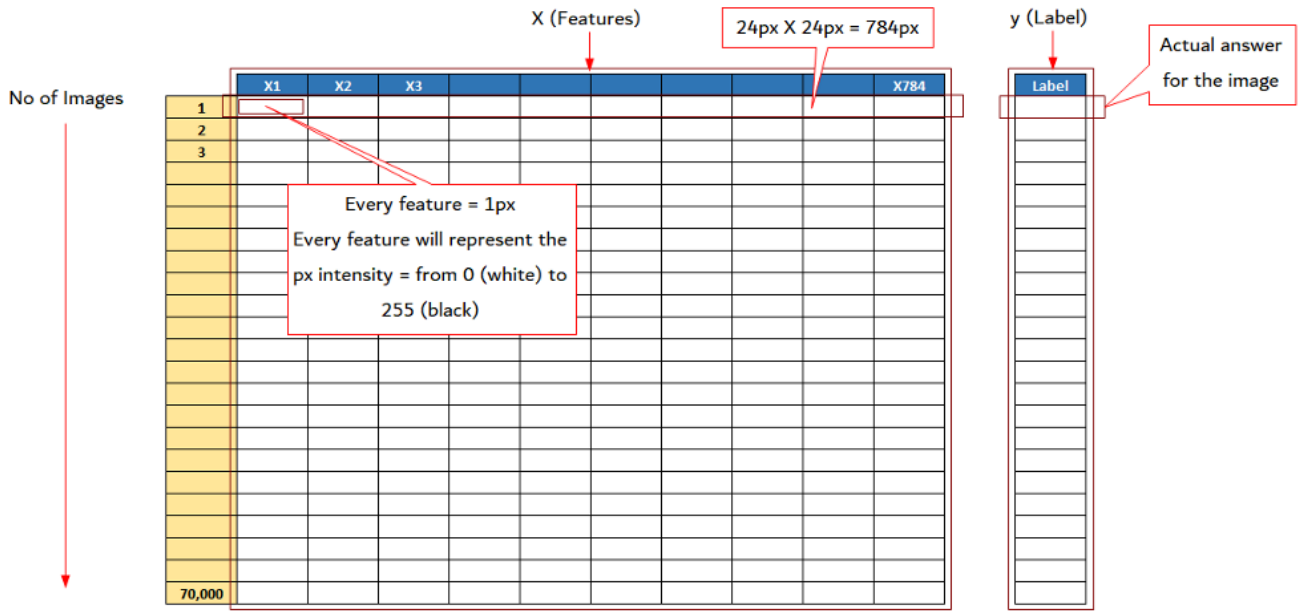
y (Label)

Data:

70,000 Rows in X means number of Images. (Rows)
784 in X means number of features. (Columns)
Every feature (X1, X2,…X784) coming from every pixel block. Picture size is 28x28 pixel = 784.
Content of every feature is the intensity of the pixel, from 0 (white) to 255 (black).

Target:

70,000 Rows in y means the label (answer) for the image - what is actually the image represent.



Look at the data:

```
#Looks Data Array - Understand the data
X, y = mnist['data'], mnist['target']
print(X.shape)
print(y.shape)
```

```
from sklearn.datasets import fetch_openml
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

# Looks Data Array - Understand the data
X, y = mnist['data'], mnist['target']
print(X.shape)
print(y.shape)
```

Result:

X = (70000, 784)
y = (70000,)

Print one image data as example:

```
#Print Any Image
some_digit = X[36000]
some_digit_image = some_digit.reshape(28,28)

plt.imshow(some_digit_image, cmap=matplotlib.cm.binary, interpolation='nearest')
plt.axis('off')
plt.show()

#Check the labeled information for that picture
print(y[36000])
```

```
from sklearn.datasets import fetch_openml
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

# Print Any Image
some_digit = X[36000]
some_digit_image = some_digit.reshape(28, 28)

plt.imshow(some_digit_image, cmap=matplotlib.cm.binary,
        interpolation='nearest')
plt.axis('off')
plt.show()

# Check the labeled information for that picture
print(y[36000])
```

Result:

From our eyes looks like the image wrote 9. So we print the labelled data to make sure.

Result:
>>> y[36000]
9.0

Few more samples from MNIST dataset:

So, to start to train this sample, we can separate test samples with training sample.

```
# Plot train data and test data
# 60000 Images for Training
# 10000 Images for Test
X_train, X_test, y_train, y_test = X[:60000], X[60000:], y[:60000], y[60000:]

# Shuffle Training data
# Permutation based on number of Images
shuffle_index = np.random.permutation(60000)

X_train, y_train = X_train[shuffle_index], y_train[shuffle_index]
```

from sklearn.datasets import fetch_openml
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

 # Plot train data and test data
 # 60000 Images for Training
 # 10000 Images for Test
X_train, X_test, y_train, y_test = X[:60000], X[60000:], y[:60000], y[60000:]

 # Shuffle Training data
 # Permutation based on number of Images
shuffle_index = np.random.permutation(60000)

X_train, y_train = X_train[shuffle_index], y_train[shuffle_index]