

# DIGIT RECOGNITION using RANDOM FOREST

## Importing Libraries

In [1]:

```
import pandas as pd
import numpy as np
```

## Load Dataset

In [3]:

```
dataset = pd.read_csv("D:/Dataset.csv")
```

## Summarize Dataset

In [5]:

```
print(dataset.shape)
print(dataset.head(5))
```

```
(42000, 785)
  label  pixel0  pixel1  pixel2  pixel3  pixel4  pixel5  pixel6  pixel7  \
0      1      0      0      0      0      0      0      0      0
1      0      0      0      0      0      0      0      0      0
2      1      0      0      0      0      0      0      0      0
3      4      0      0      0      0      0      0      0      0
4      0      0      0      0      0      0      0      0      0

  pixel8  ...  pixel774  pixel775  pixel776  pixel777  pixel778  pixel779  \
0      0  ...      0      0      0      0      0      0
1      0  ...      0      0      0      0      0      0
2      0  ...      0      0      0      0      0      0
3      0  ...      0      0      0      0      0      0
4      0  ...      0      0      0      0      0      0

  pixel780  pixel781  pixel782  pixel783
0      0      0      0      0
1      0      0      0      0
2      0      0      0      0
3      0      0      0      0
4      0      0      0      0
```

[5 rows x 785 columns]

## Segregate Dataset into X(Input/ Independent Variable) & Y(Output/ Dependent Variable)

In [6]:

```
X = dataset.iloc[:,1:]
print(X)
print(X.shape)
```

```
  pixel0  pixel1  pixel2  pixel3  pixel4  pixel5  pixel6  pixel7  pixel8  \
0      0      0      0      0      0      0      0      0      0
1      0      0      0      0      0      0      0      0      0
2      0      0      0      0      0      0      0      0      0
3      0      0      0      0      0      0      0      0      0
4      0      0      0      0      0      0      0      0      0
...      ...      ...      ...      ...      ...      ...      ...      ...
41995    0      0      0      0      0      0      0      0      0
41996    0      0      0      0      0      0      0      0      0
```

41997	0	0	0	0	0	0	0	0	0
41998	0	0	0	0	0	0	0	0	0
41999	0	0	0	0	0	0	0	0	0

	pixel9	...	pixel774	pixel775	pixel776	pixel777	pixel778	\
0	0	...	0	0	0	0	0	
1	0	...	0	0	0	0	0	
2	0	...	0	0	0	0	0	
3	0	...	0	0	0	0	0	
4	0	...	0	0	0	0	0	
...	...	...	...	...	...	...	...	
41995	0	...	0	0	0	0	0	
41996	0	...	0	0	0	0	0	
41997	0	...	0	0	0	0	0	
41998	0	...	0	0	0	0	0	
41999	0	...	0	0	0	0	0	

	pixel779	pixel780	pixel781	pixel782	pixel783
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
...	...	...	...	...	...
41995	0	0	0	0	0
41996	0	0	0	0	0
41997	0	0	0	0	0
41998	0	0	0	0	0
41999	0	0	0	0	0

[42000 rows x 784 columns]  
(42000, 784)

In [7]:

```
Y = dataset.iloc[:,0]
print(Y)
print(Y.shape)
```

```
0      1
1      0
2      1
3      4
4      0
..
41995   0
41996   1
41997   7
41998   6
41999   9
Name: label, Length: 42000, dtype: int64
(42000,)
```

### Splitting Dataset into Test & Train

In [17]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.25, random_state = 0)
```

### Training

In [18]:

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

Out[10]:

```
RandomForestClassifier()
```

In [19]:

```
y_pred = model.predict(X_test)
```

### ***Model Accuracy***

In [21]:

```
from sklearn.metrics import accuracy_score
print("Accuracy of the Model: {0}%".format(accuracy_score(y_test, y_pred)*100))
```

Accuracy of the Model: 96.43809523809523%

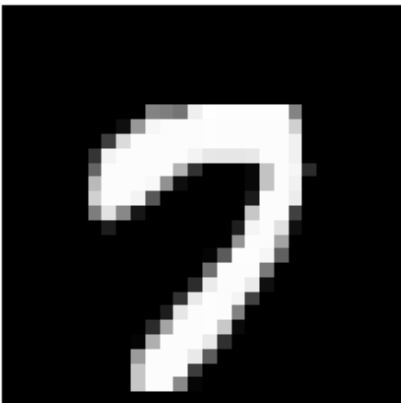
In [28]:

```
import matplotlib.pyplot as plt
index = 10
print("Predicted " + str(model.predict(X_test)[index]))
plt.axis('off')
plt.imshow(X_test.iloc[index].values.reshape((28, 28)), cmap = 'gray')
```

Predicted 7

Out[28]:

<matplotlib.image.AxesImage at 0x18f206f0670>



In [29]:

```
import matplotlib.pyplot as plt
index = 0
print("Predicted " + str(model.predict(X_test)[index]))
plt.axis('off')
plt.imshow(X_test.iloc[index].values.reshape((28, 28)), cmap = 'gray')
```

Predicted 3

Out[29]:

<matplotlib.image.AxesImage at 0x18f20740cd0>



