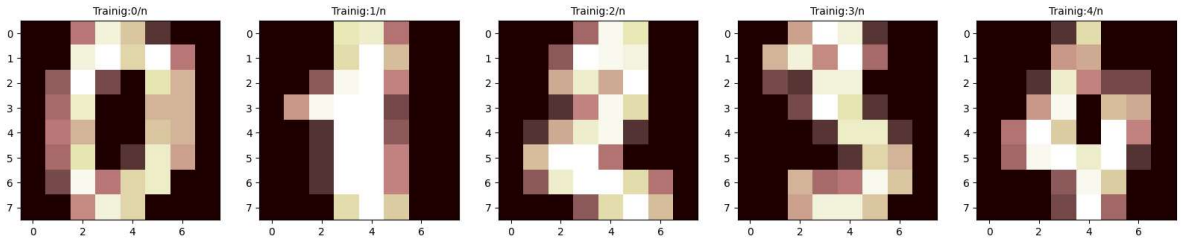
In [ ]:
```
```

In [5]:
```
import re
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import metrics
%matplotlib inline
digits=load_digits()
```

In [6]:
```
print("Image Data Shape",digits.data.shape)
print("Label Data Shape",digits.target.shape)
```

```
Image Data Shape (1797, 64)
Label Data Shape (1797,)
```

In [15]:
```
plt.figure(figsize=(20,4))
for index,(image,label) in enumerate(zip(digits.data[0:5],digits.target[0:5])
    plt.subplot(1,5,index+1)
    plt.imshow(np.reshape(image,(8,8)),cmap=plt.cm.pink)
    plt.title('Trainig:%i/n'%label,fontsize=10)
```



In [18]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(digits.data,digits.target,test_
```

In [19]:
```
print(x_train.shape)
```

```
(1257, 64)
```

In [20]:
```
print(y_train.shape)
```

```
(1257,)
```

In [21]:
```
print(x_test.shape)
```

```
(540, 64)
```

In [22]:
```
print(y_test.shape)
```

```
(540,)
```

In [23]:
```
from sklearn.linear_model import LogisticRegression
```

In [25]:
```
LogisticRegr=LogisticRegression(max_iter=10000)
LogisticRegr.fit(x_train,y_train)
```

Out[25]:
```
LogisticRegression(max_iter=10000)
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [28]:
```
print(LogisticRegr.predict(x_test))
```

```
[4 0 9 1 8 7 1 5 1 6 6 7 6 1 5 5 8 6 2 7 4 6 4 1 5 2 9 5 4 6 5 6 3 4 0 9 9
 8 4 6 8 8 5 7 9 8 9 6 1 7 0 1 9 7 3 3 1 8 8 8 9 8 5 8 4 9 3 5 8 4 3 1 3 8
 7 3 3 0 8 7 2 8 5 3 8 7 6 4 6 2 2 0 1 1 5 3 5 7 1 8 2 2 6 4 6 7 3 7 3 9 4
 7 0 3 5 1 5 0 3 9 2 7 3 2 0 8 1 9 2 1 5 1 0 3 4 3 0 8 3 2 2 7 3 1 6 7 2 8
 3 1 1 6 4 8 2 1 8 4 1 3 1 1 9 5 4 8 7 4 8 9 5 7 6 9 4 0 4 0 0 9 0 6 5 8 8
 3 7 9 2 0 8 2 7 3 0 2 1 9 2 7 0 6 9 3 1 1 3 5 2 5 5 2 1 2 9 4 6 5 5 5 9 7
 1 5 9 6 3 7 1 7 5 1 7 2 7 5 5 4 8 6 6 2 8 7 3 7 8 0 9 5 7 4 3 4 1 0 3 3 5
 4 1 3 1 2 5 1 4 0 3 1 5 5 7 4 0 1 0 9 5 5 5 4 0 1 8 6 2 1 1 1 7 9 6 7 9 7
 0 4 9 6 9 2 7 2 1 0 8 2 8 6 5 7 8 4 5 7 8 6 4 2 6 9 3 0 0 8 0 6 6 7 1 4 5
 6 9 7 2 8 5 1 2 4 1 8 8 7 6 0 8 0 6 1 5 7 8 0 4 1 4 5 9 2 2 3 9 1 3 9 3 2
 8 0 6 5 6 2 5 2 3 2 6 1 0 7 6 0 6 2 7 0 3 2 4 2 3 6 9 7 7 0 3 5 4 1 2 2 1
 2 7 7 0 4 9 8 5 6 1 6 5 2 0 8 2 4 3 3 2 9 3 8 9 9 5 9 0 3 4 7 9 8 5 7 5 0
 5 3 5 0 2 7 3 0 4 3 6 6 1 9 6 3 4 6 4 6 7 2 7 6 3 0 3 0 1 3 6 1 0 4 3 8 4
 3 3 4 8 6 9 6 3 3 0 5 7 8 9 1 5 3 2 5 1 7 6 0 6 9 5 2 4 4 7 2 0 5 6 2 0 8
 4 4 4 7 1 0 4 1 9 2 1 3 0 5 3 9 8 2 6 0 0 4]
```

In [27]:
```
score=LogisticRegr.score(x_test,y_test)
print(score)
```

```
0.9537037037037037
```

In [2]:
```
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
```

In [6]:
```
df=pd.read_csv(r"C:\Users\91756\Documents\python\gender_submission.csv")
df
```

Out[6]:

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1
5	897	0
6	898	1
7	899	0
8	900	1
9	901	0
10	902	0
11	903	0

In [7]:
```
pd.set_option('display.max_rows',100000000000)
pd.set_option('display.max_columns',100000000000)
pd.set_option('display.width',95)
```

In [8]:
```
print('This DataFrame has %d Rows and %d columns'%(df.shape))
```

```
This DataFrame has 418 Rows and 2 columns
```

In [9]:
```
df.head()
```

Out[9]:

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1

In [19]:
```
features_matrix=df.iloc[:,0:34]
```

In [20]:
```
target_vector=df.iloc[:,-1]
```

In [21]:
```
print('The features Matrix Has %d Rows and %d Columns(s)'%(features_matrix.sha
print('The Target Matrix has %d Rows and %d Columns(s)'%(np.array(target_vecto
```

```
The features Matrix Has 418 Rows and 2 Columns(s)
The Target Matrix has 418 Rows and 1 Columns(s)
```

In [23]:
```
features_matrix_standardized=StandardScaler().fit_transform(features_matrix)
```

In [29]:
```
algorithm=LogisticRegression(penalty='l2',dual=False,tol=1e-4,C=1.0,fit_interc
```

In [30]:
```
Logistic_Regression_Model=algorithm.fit(features_matrix_standardized,target_ve
```

In [35]:
```
observation=[[0.4,1]]
```

In [36]:
```
predictions = Logistic_Regression_Model.predict(observation)
print('The Model predicted The observation To Belong To class %s'%(predictions
```

```
The Model predicted The observation To Belong To class [1]
```

In [37]:
```
print('The Algorithm Was Trained To predict one of the two classes:%s'%(algori
```

```
The Algorithm Was Trained To predict one of the two classes:[0 1]
```

In [45]:
```
print("""The Model says The probability of the observation We passed Belonging
print()
print("""The Model says The probability of the observation We passed Belonging
```

```
The Model says The probability of the observation We passed Belonging To cla
ss['0'] Is 0.0549611529834666

The Model says The probability of the observation We passed Belonging To cla
ss['1'] Is 0.0549611529834666
```

In [ ]:

In [ ]: