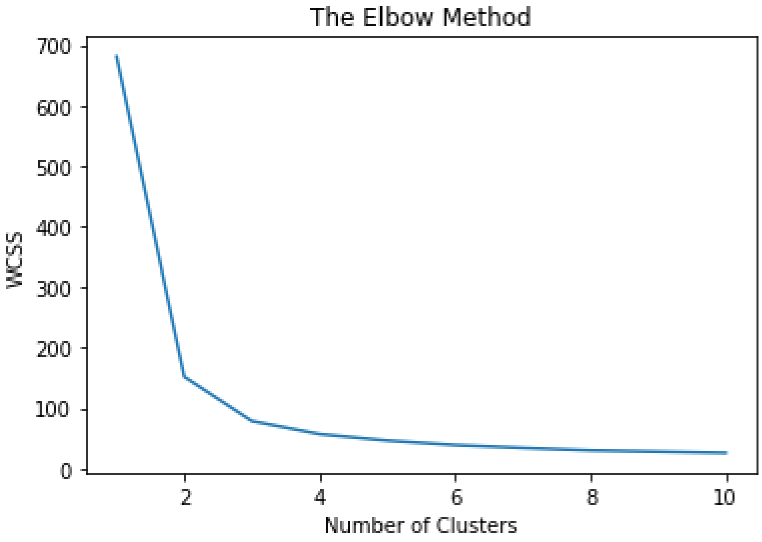
```
#Caleb Hull
#9/25/2022
#Homework 5 CS4410
#15.18 on page 661
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import datasets
from sklearn.cluster import KMeans


iris=datasets.load_iris()
iris_df=pd.DataFrame(iris.data,columns=iris.feature_names)


x=iris_df.iloc[:,[0,1,2,3]]

SSE=[]

Krange=np.arange(1,11)

for k in Krange:
    KM=KMeans(n_clusters=k)
    KM.fit(x)
    SSE.append(KM.inertia_)


plt.plot(Krange,SSE)
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.title('The Elbow Method')
```

```
Text(0.5, 1.0, 'The Elbow Method')
```



```
#15.21
```

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

Double-click (or enter) to edit

```
#split data into training and testing
from sklearn.model_selection import train_test_split

train_img, test_img, train_lbl, test_lbl = train_test_split( mnist.data, mnist.target, test_s
```

```
#apply StandardScaler
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

scaler.fit(train_img)

train_img = scaler.transform(train_img)
test_img = scaler.transform(test_img)
```

```
#apply PCA
from sklearn.decomposition import PCA

pca = PCA(.95)
```

```
pca.fit(train_img)

     PCA(n_components=0.95)
```

```
train_img = pca.transform(train_img)
test_img = pca.transform(test_img)
```

```
from sklearn.linear_model import LogisticRegression
```

```
#apply logistic regression
logisticRegr = LogisticRegression(solver = 'lbfgs')
```

```
logisticRegr.fit(train_img, train_lbl)

     /usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: Convergenc
     STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

     Increase the number of iterations (max_iter) or scale the data as shown in:
         https://scikit-learn.org/stable/modules/preprocessing.html
     Please also refer to the documentation for alternative solver options:
```

```
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
LogisticRegression()
```

```
logisticRegr.predict(test_img[0].reshape(1,-1))
```

```
array(['0'], dtype=object)
```

```
logisticRegr.predict(test_img[0:10])
```

```
array(['0', '4', '1', '2', '4', '7', '7', '1', '1', '7'], dtype=object)
```

```
logisticRegr.score(test_img, test_lbl)
```

```
0.9201
```

✓ 0s completed at 10:52 PM ● ✕