```
from sklearn.cluster import KMeans
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.datasets import load_iris
import numpy as np
```

```
iris = load_iris()
df_iris = pd.DataFrame(data= np.c_[iris['data'], iris['target']], columns= iris['feature_name
```

```
df_iris.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Species
0	5.1	3.5	1.4	0.2	0.0
1	4.9	3.0	1.4	0.2	0.0
2	4.7	3.2	1.3	0.2	0.0
3	4.6	3.1	1.5	0.2	0.0
4	5.0	3.6	1.4	0.2	0.0

```
sse = []
k_rng = range(1,10)
for k in k_rng:
    km = KMeans(n_clusters=k)
    km.fit(df_iris)
    sse.append(km.inertia_)
```

```
plt.title('Elbow Method')
plt.xlabel('K')
plt.ylabel('Sum of squared error')
plt.plot(k_rng,sse)
```

```
[<matplotlib.lines.Line2D at 0x7fe23b404750>]
```


```
from sklearn.datasets import fetch_openml
mnist = fetch_openml('mnist_784', version=1, return_X_y=True)
```

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
import tensorflow as tf



DATASET_SIZE = 70000
TRAIN_RATIO = 0.7
VALIDATION_RATIO = 0.2
TEST_RATIO = 0.1

(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()


pca = PCA(n_components=3, random_state=11)
pca.fit(x_train)
#I'm having trouble with this portion of the exercise. I can't use the StandardScaler or the
#because of the way I loaded in the dataset, but I couldn't split the data when I loaded it u
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-62-9ef50204462a> in <module>
      1 pca = PCA(n_components=3, random_state=11)
----> 2 pca.fit(x_train)

                                  3 frames

/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py in
check_array(array, accept_sparse, accept_large_sparse, dtype, order, copy,
force_all_finite, ensure_2d, allow_nd, ensure_min_samples, ensure_min_features,
estimator)
    794             raise ValueError(
    795                 "Found array with dim %d. %s expected <= 2."
--> 796                 % (array.ndim, estimator_name)
    797             )
    798

ValueError: Found array with dim 3. Estimator expected <= 2.
```

```
logisticRegr = LogisticRegression(solver = 'lbfgs')
logisticRegr.fit(x_train, y_train)
```


0s completed at 7:29 PM