Scikit-learn comes with preloaded datasets. Load the digits dataset from that collection (http://scikitlearn.org/stable/auto_examples/datasets/plot_digits_last_image.html (http://scikitlearn.org/stable/auto_examples/datasets/plot_digits_last_image.html)). Using Scikit-learn, perform a PCA transformation such that the transformed dataset can explain 95% of the variance in the original dataset. Find out the number of components in the projected subspace.

Objective: Understand and practice Principal Component Analysis using Scikit-learn.

In [3]:
```
import sklearn
from sklearn.datasets import load_digits
```

In [4]:
```
#We will set the data set now to the loaded data and define X and Y as the training and target variables
data = load_digits()
```

In [5]:
```
x =data.data
y =data.target
```

In [6]:
```
# Now we will use the model selection module with SK Learn Library and import Train Test split function.
#Train test split takes the data and splits it into training and testing sets
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=1)
```

In [7]:
```
print(x_train.shape)
```

```
(1437, 64)
```

In [8]:
```
#Transforming the train and test sets such that they explain 95% of variance
from sklearn.decomposition import PCA
sklearn_pca = PCA(n_components =0.95)
sklearn_pca.fit(x_train)
x_train_transformed = sklearn_pca.transform(x_train)

print(x_train_transformed.shape)
print(x_test.shape)
x_test_transformed = sklearn_pca.transform(x_test)
print(x_test_transformed.shape)
```

```
(1437, 28)
(360, 64)
(360, 28)
```

In [ ]:
```

```