# EigenSample: Python package for generating synthetic samples in eigenspace to minimize distortion.

Python implementation of the EigenSample algorithm by Jayadeva et al., 2017, designed to generate synthetic samples in the eigenspace while minimizing distortion. Please note that this implementation is intended solely for learning purposes and does not claim any original work or contributions. Feel free to explore, learn from, and contribute to this repository!

## 1. Generating Synthetic Samples for Classification Problem

```
In [1]: # Importing module
        from sampler import EigenSample
```

```
In [2]: # Importing data
        from sklearn.datasets import load_breast_cancer
```

```
In [3]: # Feature matrix
        data = load_breast_cancer().data

        # Target labels
        target = load_breast_cancer().target
```

```
In [4]: # Importing classification model
        from sklearn.linear_model import LogisticRegression

        # Model
        model = LogisticRegression()

        # Generating synthetic samples
        sampler = EigenSample(data, target, model)
        new_samples = sampler.add_samples()
```

```
In [5]: # New feature matrix
        new_data = new_samples["new_data"]

        # New target labels
        new_target = new_samples["new_target"]
```

```
In [6]: # Print first 5 rows of feature matrix and target labels
        print(f'New Data:\n{new_data[:5]}')
        print(f'New Labels:\n{new_target[:5]}')
```

```
New Data:
[[1.96839061e+01 2.16893609e+01 1.30289275e+02 1.21951246e+03
  1.00989066e-01 1.48614895e-01 1.78317200e-01 1.01118761e-01
  1.88894494e-01 5.99336122e-02 7.47929941e-01 1.14573463e+00
  5.30922095e+00 1.01217978e+02 6.16082553e-03 3.15077430e-02
  4.15850592e-02 1.53785479e-02 1.91865092e-02 3.70154795e-03
  2.40856028e+01 2.90282641e+01 1.61292775e+02 1.81144662e+03
  1.39382385e-01 3.64907132e-01 4.56739881e-01 1.95084766e-01
  3.09629489e-01 8.57086034e-02]
 [1.99610546e+01 2.18090519e+01 1.32200582e+02 1.24767430e+03
  1.01219936e-01 1.50823154e-01 1.82782102e-01 1.03722333e-01
  1.89280176e-01 5.97907640e-02 7.65025749e-01 1.14218742e+00
  5.43107905e+00 1.04254548e+02 6.11692593e-03 3.18084829e-02
  4.20684364e-02 1.55572286e-02 1.91188862e-02 3.69689162e-03
  2.44754637e+01 2.91954046e+01 1.63987718e+02 1.85787550e+03
  1.39732214e-01 3.70425651e-01 4.65944791e-01 1.99098812e-01
  3.10604784e-01 8.57965263e-02]
 [1.92659484e+01 2.15088592e+01 1.27406900e+02 1.17704258e+03
  1.00640897e-01 1.45284697e-01 1.71583838e-01 9.71924074e-02
  1.88312861e-01 6.01490366e-02 7.22148357e-01 1.15108405e+00
  5.12545106e+00 9.66386347e+01 6.22702896e-03 3.10542079e-02
  4.08560952e-02 1.51090859e-02 1.92884890e-02 3.70857000e-03
  2.34976673e+01 2.87762053e+01 1.57228625e+02 1.74142887e+03
  1.38854821e-01 3.56584848e-01 4.42858282e-01 1.89031326e-01
  3.08158681e-01 8.55760100e-02]
 [1.23122927e+01 1.85058123e+01 7.94522026e+01 4.70461898e+02
  9.48483463e-02 8.98794613e-02 5.95594111e-02 3.18687899e-02
  1.78636095e-01 6.37330991e-02 2.93214451e-01 1.24008350e+00
  2.06803080e+00 2.04510897e+01 7.32847008e-03 2.35086435e-02
  2.87281596e-02 1.06259871e-02 2.09851506e-02 3.82539742e-03
  1.37160556e+01 2.45826473e+01 8.96124742e+01 5.76528132e+02
  1.30077626e-01 2.18125180e-01 2.11907062e-01 8.83189068e-02
  2.83688527e-01 8.33700250e-02]
 [1.90994945e+01 2.14369734e+01 1.26258980e+02 1.16012874e+03
  1.00502238e-01 1.43958428e-01 1.68902242e-01 9.56287168e-02
  1.88081222e-01 6.02348304e-02 7.11880708e-01 1.15321448e+00
  5.05226375e+00 9.48148877e+01 6.25339482e-03 3.08735852e-02
  4.05657815e-02 1.50017713e-02 1.93291030e-02 3.71136657e-03
  2.32635190e+01 2.86758215e+01 1.55610057e+02 1.71354394e+03
  1.38644716e-01 3.53270456e-01 4.37329864e-01 1.86620513e-01
  3.07572925e-01 8.55232041e-02]]
New Labels:
[0 0 0 1 0]
```

# 2. Generating Synthetic Samples for Regression Problem

In [7]:

```
# Importing module
from sampler import EigenSample
```

In [8]:

```
# Importing data
from sklearn.datasets import load_diabetes
```

In [9]:

```
# Feature matrix
data = load_diabetes().data

# Target labels
target = load_diabetes().target
```

In [10…

```
# Importing regression model
from sklearn.linear_model import LinearRegression

# Model
model = LinearRegression()

# Generating synthetic samples
sampler = EigenSample(data, target, model)
new_samples = sampler.add_samples()
```

In [11…

```
# New feature matrix
new_data = new_samples["new_data"]

# New target labels
new_target = new_samples["new_target"]
```

In [12…

```
# Print first 5 rows of feature matrix and target labels
print(f'New Data:\n{new_data[:5]}')
print(f'New Labels:\n{new_target[:5]}')
```

```
New Data:
[[ 0.01156088  0.03878495  0.02596051  0.03040261 -0.04557526 -0.03849367
  -0.04684823  0.01008673  0.01377309  0.02202532]
 [-0.02285205 -0.04976591 -0.04833536 -0.04191511 -0.0059919  -0.01555255
   0.07083137 -0.06044246 -0.0496944  -0.04564556]
 [ 0.01655455  0.02853757  0.02178682  0.03202747 -0.04223929 -0.04008021
  -0.0292977  -0.00247168  0.00909352  0.02032338]
 [-0.04588207  0.00297977 -0.019083   -0.05081083  0.01467805  0.03139807
  -0.02662638  0.03285329 -0.00754323 -0.02812064]
 [-0.00908358 -0.02152212 -0.01860066 -0.01827531  0.00860382  0.00474554
   0.02852888 -0.01766514 -0.01614398 -0.0171414 ]]
New Labels:
[192.7154571   78.93939023 186.00030912 124.37473532 123.70172698]
```