

```
In [1]: ## How to reduce dimentionality on Sparse Matrix in Python
def Snippet_122():
    print()
    print(format('How to reduce dimentionality on Sparse Matrix in Python','^82'))
    import warnings
    warnings.filterwarnings("ignore")
    # load libraries
    from sklearn.preprocessing import StandardScaler
    from sklearn.decomposition import TruncatedSVD
    from scipy.sparse import csr_matrix
    from sklearn import datasets
    # Load Digits Data And Make Sparse
    digits = datasets.load_digits()
    # Standardize the feature matrix
    X = StandardScaler().fit_transform(digits.data)
    print(); print(X)
    # Make sparse matrix
    X_sparse = csr_matrix(X)
    print(); print(X_sparse)

    # Create Truncated Singular Value Decomposition
    tsvd = TruncatedSVD(n_components=10)
    # Run Truncated Singular Value Decomposition
    X_sparse_tsvd = tsvd.fit(X_sparse).transform(X_sparse)
    print(); print(X_sparse_tsvd)
    # Show results
    print()
    print('Original number of features:', X_sparse.shape[1])
    print('Reduced number of features:', X_sparse_tsvd.shape[1])
    # Sum of first seven components' explained variance ratios
    print(); print(tsvd.explained_variance_ratio_[0:6].sum())
Snippet_122()
```

\*\*\*\*\*How to reduce dimentionality on Sparse Matrix in Python\*\*\*\*\*

```
[ [ 0.          -0.33501649 -0.04308102 ... -1.14664746 -0.5056698
  -0.19600752]
 [ 0.          -0.33501649 -1.09493684 ...  0.54856067 -0.5056698
  -0.19600752]
 [ 0.          -0.33501649 -1.09493684 ...  1.56568555  1.6951369
  -0.19600752]
 ...
 [ 0.          -0.33501649 -0.88456568 ... -0.12952258 -0.5056698
  -0.19600752]
 [ 0.          -0.33501649 -0.67419451 ...  0.8876023  -0.5056698
  -0.19600752]
 [ 0.          -0.33501649  1.00877481 ...  0.8876023  -0.26113572
  -0.19600752]]
```

```
(0, 1)      -0.3350164872543856
(0, 2)      -0.04308101770538793
(0, 3)      0.2740715207154218
(0, 4)      -0.6644775126361527
(0, 5)      -0.8441293865949171
(0, 6)      -0.40972392088346243
(0, 7)      -0.1250229232970408
(0, 8)      -0.05907755711884675
(0, 9)      -0.6240092623290964
(0, 10)     0.4829744992519545
(0, 11)     0.7596224512649244
(0, 12)     -0.05842586308220443
(0, 13)     1.1277211297338117
(0, 14)     0.8795830595483867
(0, 15)     -0.13043338063115095
(0, 16)     -0.04462507326885248
(0, 17)     0.11144272449970435
(0, 18)     0.8958804382797294
(0, 19)     -0.8606663175537699
(0, 20)     -1.1496484601880896
(0, 21)     0.5154718747277965
(0, 22)     1.905963466976408
(0, 23)     -0.11422184388584329
(0, 24)     -0.03337972630405602
(0, 25)     0.48648927722411006
:
:
(1796, 38)  -0.8226945146290309
(1796, 40)  -0.061343668908253476
(1796, 41)  0.8105536026095989
(1796, 42)  1.3950951873625397
(1796, 43)  -0.19072005925701047
(1796, 44)  -0.5868275383619802
(1796, 45)  1.3634658076459107
(1796, 46)  0.5874903313016945
(1796, 47)  -0.08874161717060432
(1796, 48)  -0.035433262605025426
(1796, 49)  4.179200682513991
(1796, 50)  1.505078217025183
(1796, 51)  0.0881769306516768
(1796, 52)  -0.26718796251356636
(1796, 53)  1.2010187221077009
(1796, 54)  0.8692294429227895
(1796, 55)  -0.2097851269640334
(1796, 56)  -0.023596458909150665
(1796, 57)  0.7715345500122912
(1796, 58)  0.47875261517372414
(1796, 59)  -0.020358468129093202
(1796, 60)  0.4441643511677691
(1796, 61)  0.8876022965425754
(1796, 62)  -0.26113572420685327
(1796, 63)  -0.1960075186604789
```

```
[ [ 1.91421236 -0.95450041 -3.94603576 ...  1.49033451  0.10852532
  -0.81761289]
 [ 0.58898028  0.92462845  3.92476683 ...  0.55371152  1.08076518
  0.06960561]
 [ 1.30204279 -0.3171827  3.02331181 ...  1.15092477  0.77118371
  -1.11099251]
 ...
 [ 1.02259317 -0.14790945  2.46997777 ...  0.52634503  2.02839698
  -2.07554809]
 [ 1.07605019 -0.38090842 -2.45546952 ...  0.77399446  1.09180364
  -0.37528466]
 [-1.25770058 -2.22758648  0.28361382 ... -1.20091066  0.7980868
  -1.83871106]]
```

Original number of features: 64  
 Reduced number of features: 10

0.45611983736807116