pca

April 4, 2024

```
[1]: import numpy as np
     import pandas as pd
```

```
[2]: df = pd.read_csv("D:\MIT ADT\Third Year - Sem 2\ML LAB\Assign 9 -␣
       ↪PCA\MalwareMemoryDump.csv")
```

```
[3]: df.head()
```

```
[3]:   Raw_Type  pslist_nproc  pslist_nppid  pslist_avg_threads  \
     0   Benign            45            17           10.555556
     1   Benign            47            19           11.531915
     2   Benign            40            14           14.725000
     3   Benign            32            13           13.500000
     4   Benign            42            16           11.452381

        pslist_nprocs64bit  pslist_avg_handlers  dlllist_ndlls  \
     0                   0           202.844444           1694
     1                   0           242.234043           2074
     2                   0           288.225000           1932
     3                   0           264.281250           1445
     4                   0           281.333333           2067

        dlllist_avg_dlls_per_proc  handles_nhandles  handles_avg_handles_per_proc  \
     0                  38.500000              9129                    212.302326
     1                  44.127660             11385                    242.234043
     2                  48.300000             11529                    288.225000
     3                  45.156250              8457                    264.281250
     4                  49.214286             11816                    281.333333

        …  svcscan_fs_drivers  svcscan_process_services  \
     0  …                  26                        24
     1  …                  26                        24
     2  …                  26                        27
     3  …                  26                        27
     4  …                  26                        24

        svcscan_shared_process_services  svcscan_interactive_process_services  \
     0                             116                                        0
     1                             118                                        0
     2                             118                                        0
     3                             118                                        0
     4                             118                                        0

        svcscan_nactive  callbacks_ncallbacks  callbacks_nanonymous  SubType  \
     0              121                    87                     0   Benign
     1              122                    87                     0   Benign
     2              120                    88                     0   Benign
     3              120                    88                     0   Benign
     4              124                    87                     0   Benign

        callbacks_ngeneric   Label
     0                   8  Benign
     1                   8  Benign
     2                   8  Benign
     3                   8  Benign
     4                   8  Benign

     [5 rows x 58 columns]
```

[4]:
```
#df['Raw_Type'].unique().sum()
```

[5]:
```
df = df.drop(["Raw_Type"], axis=1)
```

[6]:
```
df.head()
```

[6]:
```
   pslist_nproc  pslist_nppid  pslist_avg_threads  pslist_nprocs64bit  \
0            45            17           10.555556                   0
1            47            19           11.531915                   0
2            40            14           14.725000                   0
3            32            13           13.500000                   0
4            42            16           11.452381                   0

   pslist_avg_handlers  dlllist_ndlls  dlllist_avg_dlls_per_proc  \
0           202.844444           1694                  38.500000
1           242.234043           2074                  44.127660
2           288.225000           1932                  48.300000
3           264.281250           1445                  45.156250
4           281.333333           2067                  49.214286

   handles_nhandles  handles_avg_handles_per_proc  handles_nport  …  \
0              9129                    212.302326              0  …
1             11385                    242.234043              0  …
2             11529                    288.225000              0  …
3              8457                    264.281250              0  …
```

```
4            11816                     281.333333            0  …

   svcscan_fs_drivers  svcscan_process_services  \
0                  26                        24
1                  26                        24
2                  26                        27
3                  26                        27
4                  26                        24

   svcscan_shared_process_services  svcscan_interactive_process_services  \
0                              116                                     0
1                              118                                     0
2                              118                                     0
3                              118                                     0
4                              118                                     0

   svcscan_nactive  callbacks_ncallbacks  callbacks_nanonymous  SubType  \
0              121                    87                     0   Benign
1              122                    87                     0   Benign
2              120                    88                     0   Benign
3              120                    88                     0   Benign
4              124                    87                     0   Benign

   callbacks_ngeneric   Label
0                   8  Benign
1                   8  Benign
2                   8  Benign
3                   8  Benign
4                   8  Benign

[5 rows x 57 columns]
```

[7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58596 entries, 0 to 58595
Data columns (total 57 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   pslist_nproc                            58596 non-null  int64
 1   pslist_nppid                            58596 non-null  int64
 2   pslist_avg_threads                      58596 non-null  float64
 3   pslist_nprocs64bit                      58596 non-null  int64
 4   pslist_avg_handlers                     58596 non-null  float64
 5   dlllist_ndlls                           58596 non-null  int64
 6   dlllist_avg_dlls_per_proc               58596 non-null  float64
 7   handles_nhandles                        58596 non-null  int64
```

```
8   handles_avg_handles_per_proc           58596 non-null  float64
9   handles_nport                          58596 non-null  int64
10  handles_nfile                          58596 non-null  int64
11  handles_nevent                         58596 non-null  int64
12  handles_ndesktop                       58596 non-null  int64
13  handles_nkey                           58596 non-null  int64
14  handles_nthread                        58596 non-null  int64
15  handles_ndirectory                     58596 non-null  int64
16  handles_nsemaphore                     58596 non-null  int64
17  handles_ntimer                         58596 non-null  int64
18  handles_nsection                       58596 non-null  int64
19  handles_nmutant                        58596 non-null  int64
20  ldrmodules_not_in_load                 58596 non-null  int64
21  ldrmodules_not_in_init                 58596 non-null  int64
22  ldrmodules_not_in_mem                  58596 non-null  int64
23  ldrmodules_not_in_load_avg             58596 non-null  float64
24  ldrmodules_not_in_init_avg             58596 non-null  float64
25  ldrmodules_not_in_mem_avg              58596 non-null  float64
26  malfind_ninjections                    58596 non-null  int64
27  malfind_commitCharge                   58596 non-null  int64
28  malfind_protection                     58596 non-null  int64
29  malfind_uniqueInjections               58596 non-null  float64
30  psxview_not_in_pslist                  58596 non-null  int64
31  psxview_not_in_eprocess_pool           58596 non-null  int64
32  psxview_not_in_ethread_pool            58596 non-null  int64
33  psxview_not_in_pspcid_list             58596 non-null  int64
34  psxview_not_in_csrss_handles           58596 non-null  int64
35  psxview_not_in_session                 58596 non-null  int64
36  psxview_not_in_deskthrd                58596 non-null  int64
37  psxview_not_in_pslist_false_avg        58596 non-null  float64
38  psxview_not_in_eprocess_pool_false_avg 58596 non-null  float64
39  psxview_not_in_ethread_pool_false_avg  58596 non-null  float64
40  psxview_not_in_pspcid_list_false_avg   58596 non-null  float64
41  psxview_not_in_csrss_handles_false_avg 58596 non-null  float64
42  psxview_not_in_session_false_avg       58596 non-null  float64
43  psxview_not_in_deskthrd_false_avg      58596 non-null  float64
44  modules_nmodules                       58596 non-null  int64
45  svcscan_nservices                      58596 non-null  int64
46  svcscan_kernel_drivers                 58596 non-null  int64
47  svcscan_fs_drivers                     58596 non-null  int64
48  svcscan_process_services               58596 non-null  int64
49  svcscan_shared_process_services        58596 non-null  int64
50  svcscan_interactive_process_services   58596 non-null  int64
51  svcscan_nactive                        58596 non-null  int64
52  callbacks_ncallbacks                   58596 non-null  int64
53  callbacks_nanonymous                   58596 non-null  int64
54  SubType                                58596 non-null  object
55  callbacks_ngeneric                     58596 non-null  int64
```

```
 56  Label                                      58596 non-null  object
dtypes: float64(15), int64(40), object(2)
memory usage: 25.5+ MB
```

[8]:
```python
cat_cols = df.select_dtypes(exclude=["int64",'float64']).columns
```

[9]:
```python
cat_cols
```

[9]:
```
Index(['SubType', 'Label'], dtype='object')
```

[10]:
```python
from sklearn.preprocessing import LabelEncoder

lbl_enc = LabelEncoder()

for i in cat_cols:
    df[i] = lbl_enc.fit_transform(df[i])
```

[11]:
```python
X = df.drop(['Label'], axis=1)

y = df["Label"]
```

[12]:
```python
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

[13]:
```python
# from sklearn.decomposition import PCA

# pca = PCA(n_components=1)
# X_pca1 = pca.fit_transform(X_scaled)
# pca = PCA(n_components=2)
# X_pca2 = pca.fit_transform(X_scaled)
# pca = PCA(n_components=3)
# X_pca3 = pca.fit_transform(X_scaled)
# pca = PCA(n_components=4)
# X_pca4 = pca.fit_transform(X_scaled)
# pca = PCA(n_components=5)
# X_pca5 = pca.fit_transform(X_scaled)


# eigenvalues = pca.explained_variance_
# eigenvectors = pca.components_

# print("Eigenvalues:")
# print(eigenvalues)
# print("\nEigenvectors:")
# print(eigenvectors)
```

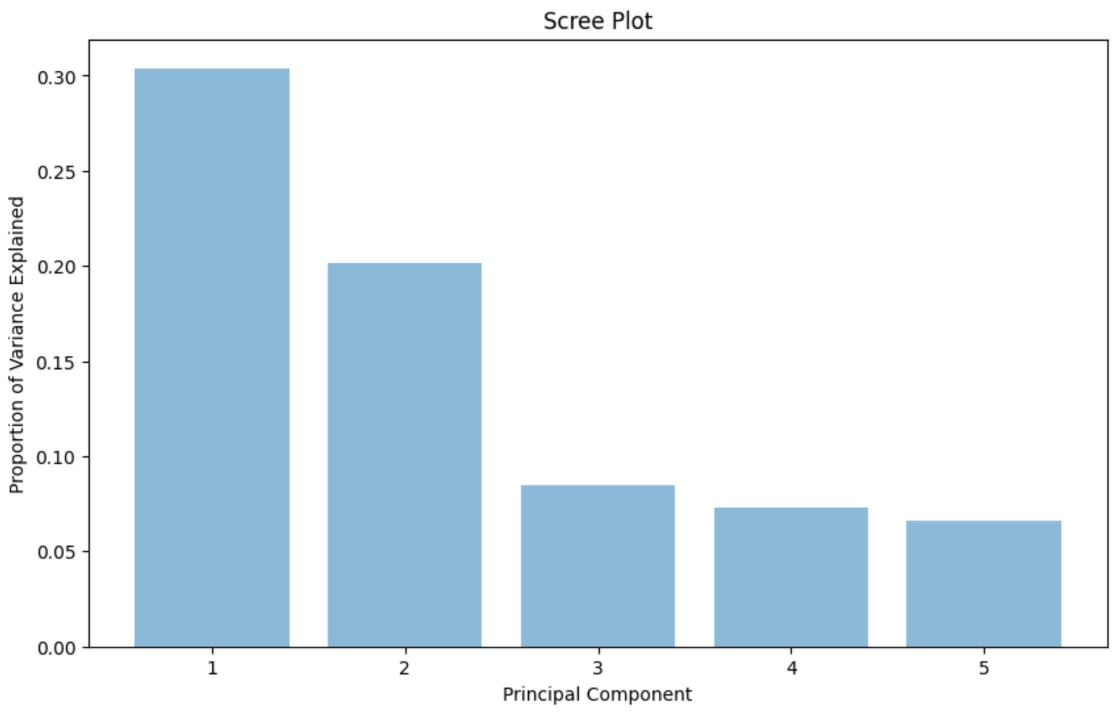
[14]:
```python
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,␣
 ↪random_state=42)
svm_original = SVC(kernel='linear')
svm_original.fit(X_train, y_train)
y_pred_original = svm_original.predict(X_test)
accuracy_original = accuracy_score(y_test, y_pred_original)
print("Accuracy Orignal: ", accuracy_original)
```

```
Accuracy Orignal:  1.0
```

[15]:
```python
from sklearn.decomposition import PCA
for i in range(1,6):
    print("Components: ", i)
    pca = PCA(n_components=i)

    X_pca = pca.fit_transform(X_scaled)
    X_train_pca, X_test_pca, _, _ = train_test_split(X_pca, y, test_size=0.2,␣
 ↪random_state=42)

    svm_pca = SVC(kernel='linear')
    svm_pca.fit(X_train_pca, y_train)
    y_pred_pca = svm_pca.predict(X_test_pca)

    accuracy_pca = accuracy_score(y_test, y_pred_pca)
    print("Accuracy PCA: ", i, "-->", accuracy_pca)
```

```
Components:  1
Accuracy PCA:  1 --> 0.9652730375426621
Components:  2
Accuracy PCA:  2 --> 0.9841296928327645
Components:  3
Accuracy PCA:  3 --> 0.996160409556314
Components:  4
Accuracy PCA:  4 --> 0.9963310580204778
Components:  5
Accuracy PCA:  5 --> 0.997098976109215
```

[16]:
```python
import matplotlib.pyplot as plt

# Get explained variance ratio
explained_variance_ratio = pca.explained_variance_ratio_

# Plot scree plot
plt.figure(figsize=(10, 6))
plt.bar(range(1, len(explained_variance_ratio) + 1), explained_variance_ratio,␣
 ↪alpha=0.5, align='center')
plt.xlabel('Principal Component')
plt.ylabel('Proportion of Variance Explained')
plt.title('Scree Plot')
plt.show()
```


Scree Plot
Proportion of Variance Explained
Principal Component