

```
In [11]: ## SVC
```

```
In [12]: % matplotlib inline

from IPython.core.display import display, HTML
display(HTML("<style>.container { width:90% !important; }</style>"))
import numpy as np
import pandas as pd
pd.set_option('display.max_columns', 100)

df = pd.read_csv('User_Knowledge.csv')

df.loc[df.UNS == 'very_low','grade'] = 0
df.loc[df.UNS == 'Low','grade'] = 1
df.loc[df.UNS == 'Middle','grade'] = 2
df.loc[df.UNS == 'High','grade'] = 3

df.sample(5)
```

Out[12]:

	STG	SCG	STR	LPR	PEG	UNS	grade
15	0.12	0.12	0.75	0.35	0.80	High	3.0
25	0.09	0.30	0.68	0.18	0.85	High	3.0
45	0.17	0.36	0.80	0.14	0.66	Middle	2.0
9	0.00	0.00	0.50	0.20	0.85	High	3.0
147	0.33	0.27	0.20	0.33	0.10	very_low	0.0

Attribute Information

- STG (The degree of study time for goal object materials), (input value)
- SCG (The degree of repetition number of user for goal object materials) (input value)
- STR (The degree of study time of user for related objects with goal object) (input value)
- LPR (The exam performance of user for related objects with goal object) (input value)
- PEG (The exam performance of user for goal objects) (input value)
- UNS (The knowledge level of user) (target value)
 - Very Low: 50
 - Low:129
 - Middle: 122
 - High 130

```
In [13]: y = list(df['UNS'])
y_grade = df.grade

# feature selection, dropping SCG
X = df.drop(columns=['SCG']).iloc[:,0:4]

X.sample()
```

Out[13]:

	STG	STR	LPR	PEG
92	0.251	0.57	0.6	0.09

```
In [14]: from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import StandardScaler

        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_
        state=0)
        sc = StandardScaler()
        X_train=sc.fit_transform(X_train)
        X_test=sc.transform(X_test)
```

SVC approach

```
In [15]: from sklearn.svm import SVC

        svc = SVC()
        svc_pca = SVC()

        svc.fit(X_train, y_train)
        print("PCA accuracy: ", svc.score(X_test, y_test))

        PCA accuracy:  0.906976744186
```

PCA + SVC approach

```
In [16]: from sklearn.decomposition import PCA

        pca = PCA(n_components=2)
        X_train_pca = pca.fit_transform(X_train)
        X_test_pca = pca.transform(X_test)

        svc_pca.fit(X_train_pca, y_train)
        print("PCA + SVC accuracy: ", svc_pca.score(X_test_pca, y_test))

        PCA + SVC accuracy:  0.558139534884
```

```
In [17]: p=svc.predict(X_test)
```

```
In [18]: from sklearn import metrics
        from sklearn.metrics import confusion_matrix
        print("SVM classification accuracy :",metrics.accuracy_score(y_test, p))

        SVM classification accuracy : 0.906976744186
```