```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
df=pd.read_csv('Classified Data',index_col=0)
```

```
df.head()
```

	WTT	PTI	EQW	SBI	LQE	QWG	FDJ	PJF	HQE
0	0.913917	1.162073	0.567946	0.755464	0.780862	0.352608	0.759697	0.643798	0.879422
1	0.635632	1.003722	0.535342	0.825645	0.924109	0.648450	0.675334	1.013546	0.621552
2	0.721360	1.201493	0.921990	0.855595	1.526629	0.720781	1.626351	1.154483	0.957877
3	1.234204	1.386726	0.653046	0.825624	1.142504	0.875128	1.409708	1.380003	1.522692

```
from sklearn.preprocessing import StandardScaler
```

```
scaler=StandardScaler()
```

```
scaler.fit(df.drop('TARGET CLASS',axis=1))
```

```
▾ StandardScaler
StandardScaler()
```

```
scaled_features=scaler.transform(df.drop('TARGET CLASS',axis=1))
```

```
fd=pd.DataFrame(scaled_features,columns=df.columns[:-1])
```

```
df.columns[:-1]
```

```
Index(['WTT', 'PTI', 'EQW', 'SBI', 'LQE', 'QWG', 'FDJ', 'PJF', 'HQE', 'NXJ'], dtype='object')
```

```
from sklearn.model_selection import train_test_split as tts
```

```
X=fd
y=df['TARGET CLASS']
```

```
X_train,X_test,y_train,y_test=tts(X,y,test_size=0.3,random_state=20)
```

```
from sklearn.neighbors import KNeighborsClassifier as knc
```

```
knn=knc(n_neighbors=1)
```

```
knn.fit(X_train,y_train)
```

```
▾       KNeighborsClassifier
KNeighborsClassifier(n_neighbors=1)
```

```
pred=knn.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
```

```
print(confusion_matrix(y_test,pred))
```

```
[[127  18]
 [ 13 142]]
```

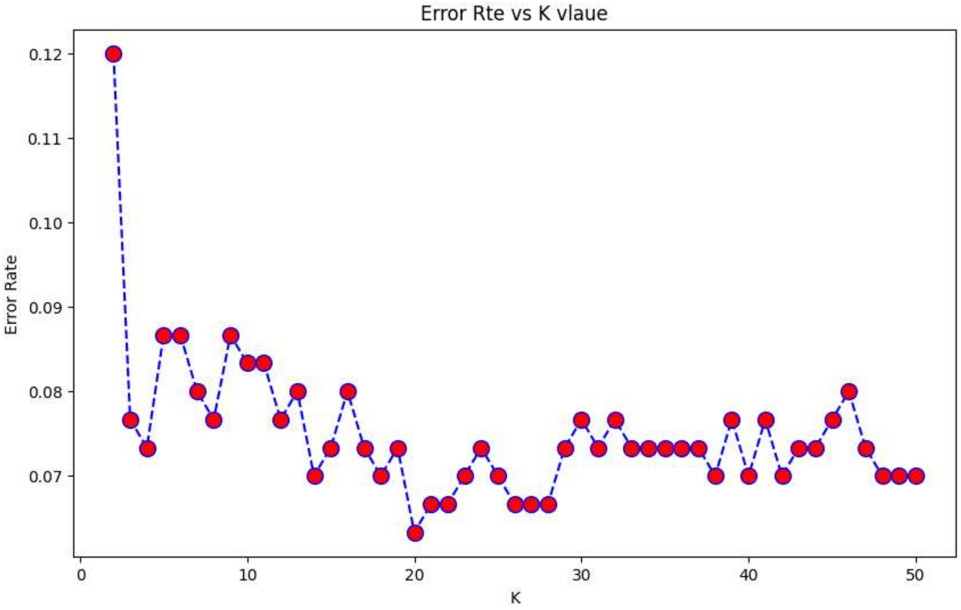
```
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

           0       0.91      0.88      0.89       145
           1       0.89      0.92      0.90       155

    accuracy                           0.90       300
   macro avg       0.90      0.90      0.90       300
weighted avg       0.90      0.90      0.90       300
```

```
error_rate=[]
for i in range(2,51):
  knn=knc(n_neighbors=i)
  knn.fit(X_train,y_train)
  pred_i=knn.predict(X_test)
  error_rate.append(np.mean(pred_i != y_test))
```

```
plt.figure(figsize=(10,6))
plt.plot(range(2,51),error_rate,color='blue',linestyle='dashed',marker='o',markerfacecolor='red',markersize=10)
plt.title('Error Rte vs K vlaue')
plt.xlabel('K')
plt.ylabel('Error Rate')
```

```
Text(0, 0.5, 'Error Rate')
```


Error Rte vs K vlaue
Error Rate
K