```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
from google.colab import drive

file_path = '/content/2_letter_recognition.data'
```

```
dataset = pd.read_csv(file_path, sep = ",")
```

```
X = dataset.iloc[:, 1 : 17]
Y = dataset.select_dtypes(include = [object])
```

```
X_train, X_validation, Y_train, Y_validation = train_test_split(X, Y, test_size = 0.20, random_state = 10)
```

```
scaler = StandardScaler()
scaler.fit(X_train)
```

```
▾ StandardScaler
StandardScaler()
```

```
X_train = scaler.transform(X_train)
X_validation = scaler.transform(X_validation)
```

```
mlp = MLPClassifier(hidden_layer_sizes = (250, 300), max_iter = 1000000, activation = 'logistic')
```

```
from yellowbrick.classifier import confusion_matrix
cm = confusion_matrix(mlp,X_train,Y_train, X_validation, Y_validation, classes="A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,
/usr/local/lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:1098: DataConversionWarni
```

```
# !pip install yellowbrick
```

```
  y = column_or_1d(y, warn=True)
```

```
cm.fit(X_train, Y_train.values.ravel())
```

```
▸      ConfusionMatrix
▸ estimator: MLPClassifier
     ▸ MLPClassifier
```

```
cm.score(X_validation, Y_validation)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_label.py:116: DataConversionWarning: A column-vecto
  y = column_or_1d(y, warn=True)
0.97425
```

```
predictions = cm.predict(X_validation)
predictions
```

```
array(['C', 'I', 'H', ..., 'R', 'B', 'W'], dtype='<U1')
```

```
print("Accuracy: ", accuracy_score(Y_validation, predictions))
```

```
Accuracy:  0.97425
```

Start coding or generate with AI.