FEB. 8, 2020 • MACHINE LEARNING

Data Exploration

FEB. 8, 2020 • MACHINE LEARNING

FEB. 8, 2020 • MACHINE LEARNING

GROUP

Imam Nurul Ihsan	1301174688
Khaerani Nur A	1301173712
Hilmy Faruq	1301172745
Vina Putri Damartya	1301173705

FEB. 8, 2020 • MACHINE LEARNING

Imam Nurul Ihsan	3,7,10
Khaerani Nur A	1,6
Hilmy Faruq	4,8
Vina Putri Damartya	2,5,9

Data Tranning

Att1	Att2	Att3	Att4	Att5	Att6	Class
2.5	Besar	Hijau	48	77	3	A
	Besar	Hijau	36	62	3	B
1.2	Besar	Merah	36	63	3	A
8.2	Sedang	Biru	12	32	2	C
4.4	Sedang	Merah	36	74	3	A
5.7	Kecil	Hijau	24	42	3	B
7.7	Kecil	Merah	12	20	2	C
8.5	Sedang	Merah	24		3	C
10.9	Besar	Biru	12	16	2	C
5.8	Besar	Biru	24	52	3	B
3.9		Hijau	48	81	3	A

2.2	Sedang	Merah	24	91	3
4.5	Kecil	Biru	24	33	3
6.3	Sedang	Merah	24	22	2
2.9	Besar	Biru	36	76	3
5.6	Sedang	Merah	36	74	3
7.5	Kecil	Hijau	24	32	3

Data Testing

DETECT MISSING VALUES

```
print (train['att1'])
print (train['att1'].isnull())
print (train['att2'])
print (train['att2'].isnull())
print (train['att3'])
print (train['att3'].isnull())
print (train['att4'])
print (train['att4'].isnull())
print (train['att5'])
print (train['att5'].isnull())
print (train['att6'])
print (train['att6'].isnull())
print (train.isnull().sum())
```

```
0       2.5
1       NaN
2       1.2
3       8.2
4       4.4
5       5.7
6       7.7
7       8.5
8      10.9
9       5.8
10      3.9
Name: att1, dtype: float64
0     False
1      True
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
Name: att1, dtype: bool
```

DETECT MISSING VALUES

```
0       besar
1       besar
2       besar
3      sedang
4      sedang
5       kecil
6       kecil
7      sedang
8       besar
9       besar
10        NaN
Name: att2, dtype: object
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10     True
Name: att2, dtype: bool
```

```
0     hijau
1     hijau
2     merah
3      biru
4     merah
5     hijau
6     merah
7     merah
8      biru
9      biru
10    hijau
Name: att3, dtype: object
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
Name: att3, dtype: bool
```

DETECT MISSING VALUES

```
0      77.0
1      62.0
2      63.0
3      32.0
4      74.0
5      42.0
6      20.0
7       NaN
8      16.0
9      52.0
10     81.0
Name: att5, dtype: float64
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7      True
8     False
9     False
10    False
Name: att5, dtype: bool
```

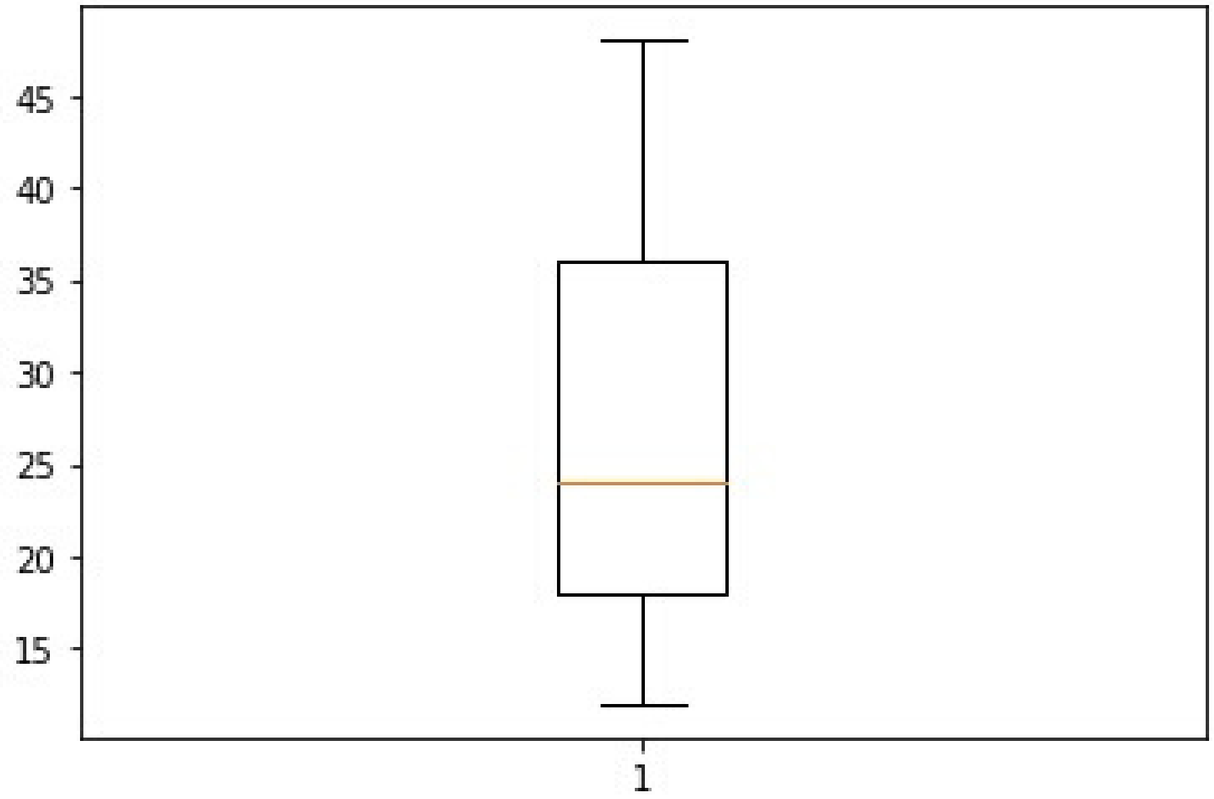
```
0     3
1     3
2     3
3     2
4     3
5     3
6     2
7     3
8     2
9     3
10    3
Name: att6, dtype: int64
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
Name: att6, dtype: bool
```

```
att1      1
att2      1
att3      0
att4      0
att5      1
att6      0
class     0
dtype: int64
```

Jika terdapat nilai kosong akan diisi dengan NaN dan True, tetapi jika terdapat nilai akan terisi dengan nilai dan False

DETECT OUTLIERS USING BOXPLOT

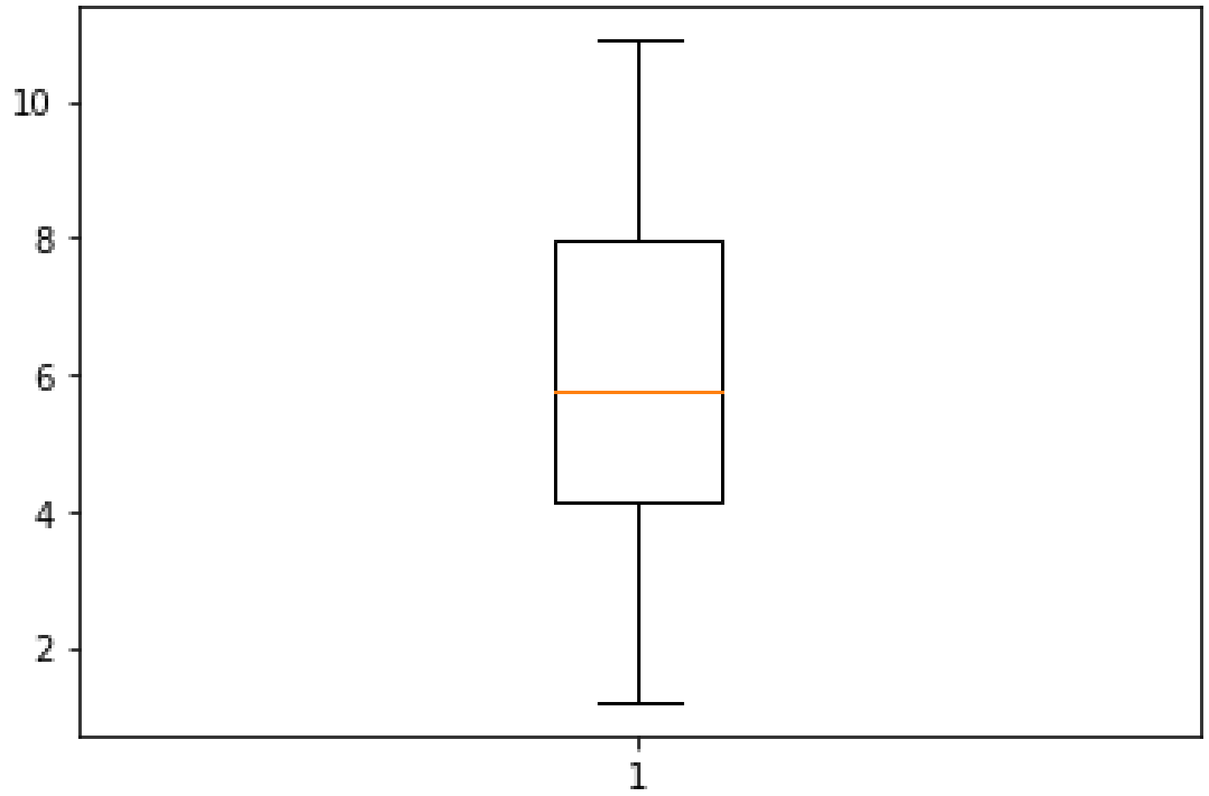
```
import matplotlib.pyplot as plt
plt.boxplot(train['att4'])
plt.show()
```



Outlier adalah data observasi yang muncul dengan nilai-nilai ekstrim, baik secara univariat ataupun multivariat.

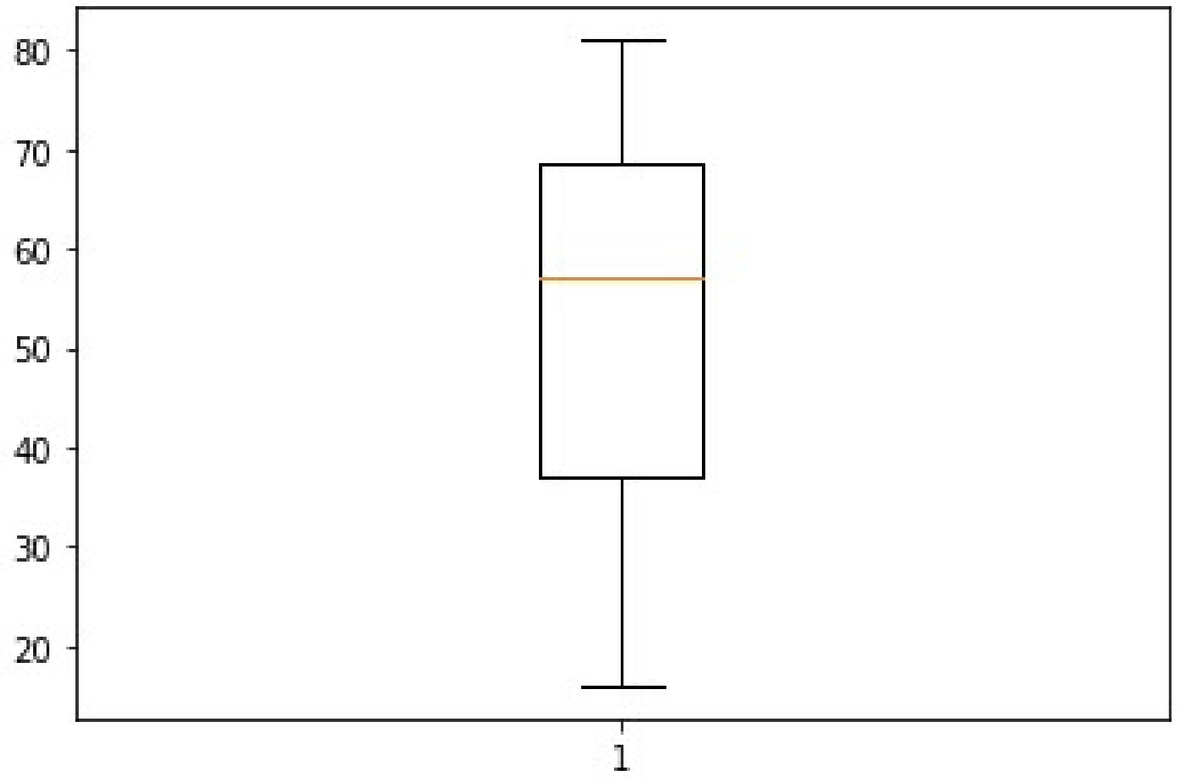
DETECT OUTLIERS USING BOXPLOT

```
plt.boxplot(train['att1'])
plt.show()
```



DETECT OUTLIERS USING BOXPLOT

```
plt.boxplot(train['att5'])
plt.show()
```



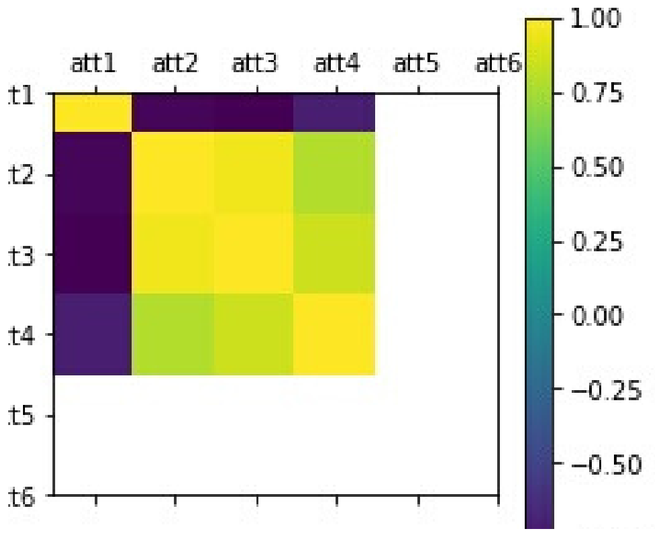
Plot Correlation to do Feature Selection

Correlation gunakan dalam statistik untuk mengukur korelasi antara dua kumpulan data

```
ain.corr()
```

	att1	att4	att5	att6
t1	1.000000	-0.835726	-0.861561	-0.705669
t4	-0.835726	1.000000	0.962330	0.781929
t5	-0.861561	0.962330	1.000000	0.856644
t6	-0.705669	0.781929	0.856644	1.000000

```
t.matshow(train.corr())
t.xticks(range(len(train.columns)-1), train.columns)
t.yticks(range(len(train.columns)-1), train.columns)
t.colorbar()
t.show()
```



ENCODING CATEGORICAL VARIABLE

```
train['att2'] = train['att2'].replace({'besar': 1, 'sedang': 2, 'kecil': 3})
train['att3'] = train['att3'].replace({'hijau': 1, 'merah': 2, 'biru': 3})
train['class'] = train['class'].replace({'A': 1, 'B': 2, 'C': 3})
```

```
train
```

	att1	att2	att3	att4	att5	att6	class
0	2.5	1.0	1	48	77.0	3	1
1	NaN	1.0	1	36	62.0	3	2
2	1.2	1.0	2	36	63.0	3	1
3	8.2	2.0	3	12	32.0	2	3
4	4.4	2.0	2	36	74.0	3	1
5	5.7	3.0	1	24	42.0	3	2
6	7.7	3.0	2	12	20.0	2	3
7	8.5	2.0	2	24	NaN	3	3
8	10.9	1.0	3	12	16.0	2	3
9	5.8	1.0	3	24	52.0	3	2
10	3.9	NaN	1	48	81.0	3	1

Encoding adalah proses konversi informasi dari suatu sumber (objek) menjadi data, yang selanjutnya dikirimkan ke penerima atau pengamat, seperti pada sistem pemrosesan data. Mengubah data set menjadi numerik data karena data numerik yang dapat dibaca oleh mesin

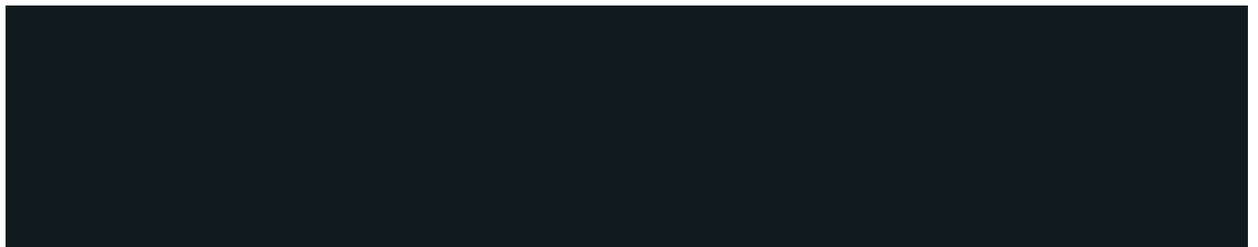
SCALING

SCALING ADALAH SUATU CARA UNTUK MEMBUAT NUMERICAL DATA PADA DATASET MEMILIKI RENTANG NILAI (SCALE) YANG SAMA. TIDAK ADA LAGI SATU VARIABEL DATA YANG MENDOMINASI VARIABEL DATA LAINNYA.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)

X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
print(X_train)
print(X_test)
```

```
[[ 0.41346563  1.97565832 -0.37796447 -1.15470054 -1.33957513 -1.29099445]
 [-0.77948438  0.53881591 -0.37796447  1.15470054  1.33957513  0.77459667]
 [-0.29145938 -0.89802651 -1.88982237  1.15470054  0.74420841  0.77459667]
 [ 0.59421563  0.53881591  1.13389342 -1.15470054 -0.74420841 -1.29099445]
 [ 1.57026564 -0.89802651  1.13389342 -1.15470054 -1.53803071 -1.29099445]
 [ 0.70266563  0.53881591 -0.37796447  0.          0.49613894  0.77459667]
 [-0.27338438 -0.89802651  1.13389342  0.          0.24806947  0.77459667]
 [-1.9362844  -0.89802651 -0.37796447  1.15470054  0.7938223   0.77459667]]
[[-0.30953438  1.97565832 -1.88982237  0.         -0.24806947  0.77459667]
 [-1.46633439 -0.89802651 -1.88982237  2.30940108  1.48841682  0.77459667]
 [-0.96023439 -0.1796053  -1.88982237  2.30940108  1.68687239  0.77459667]]
```



DATA SPLITING

```
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
X = train.iloc[:, :-1].values
y = train.iloc[:, 6].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
X_train
```

```
array([[ 7.7 ,  3.  ,  2.  , 12.  , 20.  ,  2.  ],
       [ 4.4 ,  2.  ,  2.  , 36.  , 74.  ,  3.  ],
       [ 5.75,  1.  ,  1.  , 36.  , 62.  ,  3.  ],
       [ 8.2 ,  2.  ,  3.  , 12.  , 32.  ,  2.  ],
       [10.9 ,  1.  ,  3.  , 12.  , 16.  ,  2.  ],
       [ 8.5 ,  2.  ,  2.  , 24.  , 57.  ,  3.  ],
       [ 5.8 ,  1.  ,  3.  , 24.  , 52.  ,  3.  ],
       [ 1.2 ,  1.  ,  2.  , 36.  , 63.  ,  3.  ]])
```

```
X_train.shape
```

```
(8, 6)
```

```
X_test
```

```
array([[ 5.7,  3. ,  1. , 24. , 42. ,  3. ],
       [ 2.5,  1. ,  1. , 48. , 77. ,  3. ],
       [ 3.9,  1.5,  1. , 48. , 81. ,  3. ]])
```

Data splitting (Pemisahan data) adalah pendekatan untuk melindungi data sensitif dari akses yang tidak sah dengan mengenkripsi data dan menyimpan bagian file yang berbeda di server yang berbeda.

DATA SPLITING

```
X_test.shape
```

```
(3, 6)
```

```
y_train
```

```
array([3, 1, 2, 3, 3, 3, 2, 1], dtype=int64)
```

```
y_train.shape
```

```
(8,)
```

```
y_test
```

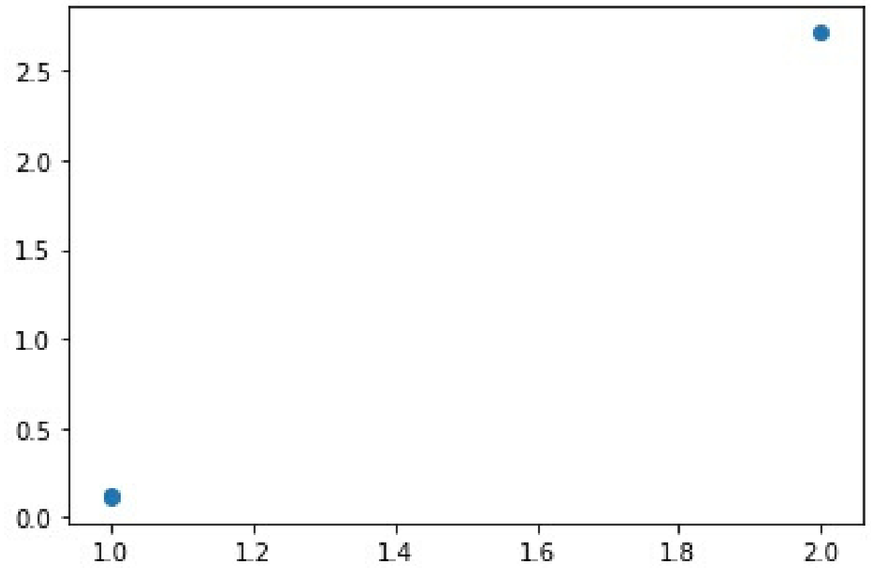
```
array([2, 1, 1], dtype=int64)
```

```
y_test.shape
```

```
(3,)
```

```
from sklearn.linear_model import LinearRegression as lm
model=lm().fit(X_train,y_train)
predictions=model.predict(X_test)
import matplotlib.pyplot as plt
plt.scatter(y_test,predictions)
```

```
<matplotlib.collections.PathCollection at 0x1843108a288>
```



IMPLEMENT K-NN

SELURUH DATASET PELATIHAN DISIMPAN. KETIKA PREDIKSI DIPERLUKAN, CATATAN K-PALING MIRIP DENGAN CATATAN BARU DARI DATASET PELATIHAN KEMUDIAN DITEMUKAN.

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=3)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
y_pred
```

```
array([3, 1, 1], dtype=int64)
```

COUNT PERFORMANCE FOR VALIDATION

```
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn import metrics
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
[[2 0 0]
 [0 0 1]
 [0 0 0]]
              precision    recall  f1-score   support

           1       1.00      1.00      1.00         2
           2       0.00      0.00      0.00         1
           3       0.00      0.00      0.00         0

    accuracy                           0.67         3
   macro avg       0.33      0.33      0.33         3
weighted avg       0.67      0.67      0.67         3

Accuracy: 0.6666666666666666
```

MENGGUNAKAN DATA DARI UJI VALIDASI UNTUK MENGUBAH MODEL, SEHINGGA MEMPENGARUHI MODEL YANG DILATIH SECARA TIDAK LANGSUNG

PREDICT TEST SET RESULTS

```
test['att2'] = test['att2'].replace({'besar': 1, 'sedang': 2, 'kecil': 3})
test['att3'] = test['att3'].replace({'hijau': 1, 'merah': 2, 'biru': 3})
test
```

	att1	att2	att3	att4	att5	att6	class
0	2.2	2	2	24	91	3	NaN
1	4.5	3	3	24	33	3	NaN
2	6.3	2	2	24	22	2	NaN
3	2.9	1	3	36	76	3	NaN
4	5.6	3	2	24	68	3	NaN
5	7.5	3	1	24	32	3	NaN

```
test
```

	att1	att2	att3	att4	att5	att6	class
0	2.2	2	2	24	91	3	NaN
1	4.5	3	3	24	33	3	NaN
2	6.3	2	2	24	22	2	NaN
3	2.9	1	3	36	76	3	NaN
4	5.6	3	2	24	68	3	NaN
5	7.5	3	1	24	32	3	NaN

PREDICT TEST SET RESULTS

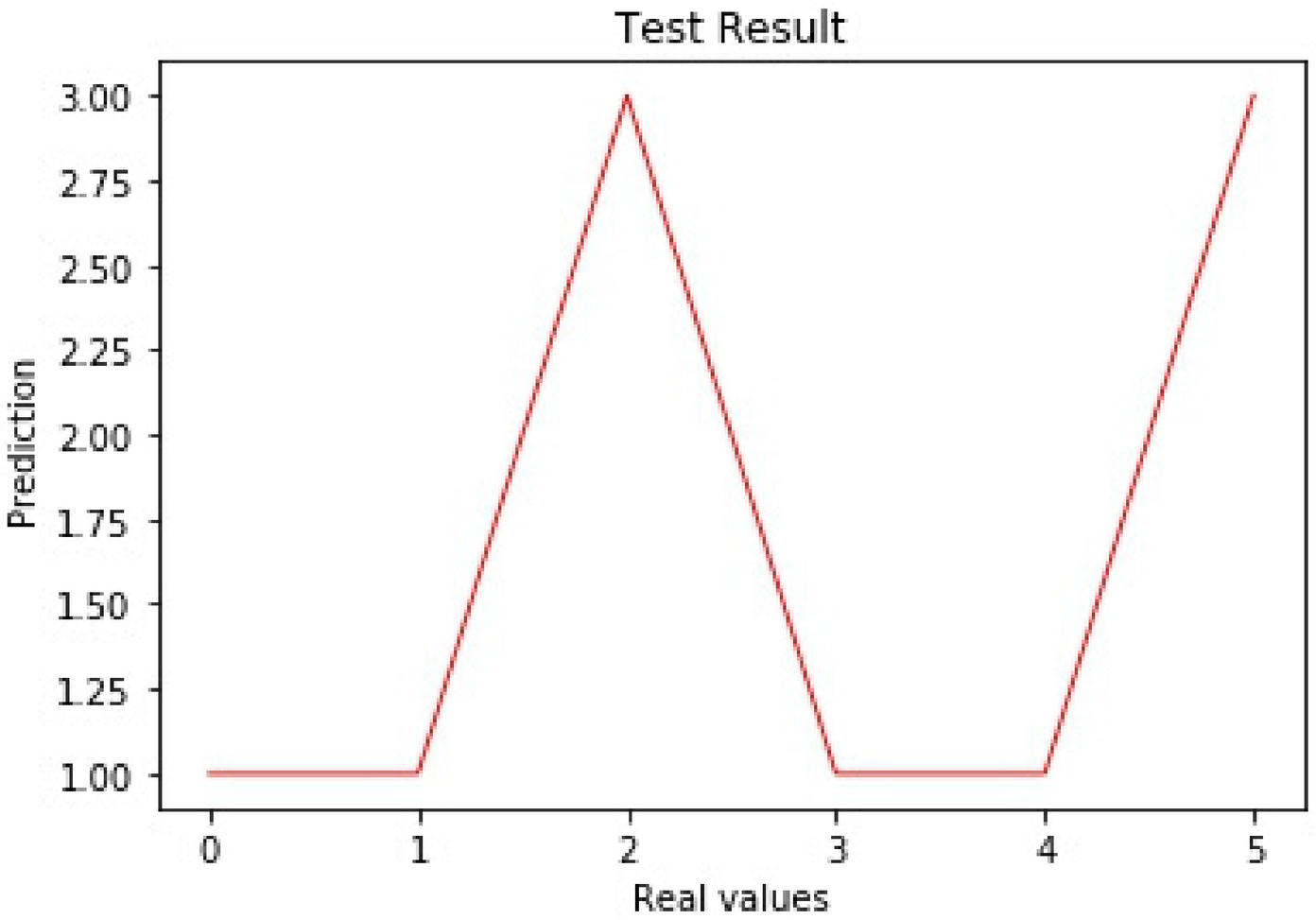
```
X_train = train.iloc[:, :-1].values
X_test = test.iloc[:, :-1].values
y_train = train.iloc[:, 6].values
y_test = test.iloc[:, 6].values

X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

classifier = KNeighborsClassifier(n_neighbors=3)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
y_pred
```

```
array([1, 3, 3, 1, 1, 3], dtype=int64)
```

VISUALIZE TEST SET RESULT



THANKYOU