

In [2]:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.preprocessing import LabelEncoder

%matplotlib inline
```

In [3]:

```
smarthouse = pd.read_csv("smart_houses.csv")
```

In [4]:

```
le = LabelEncoder()
smarthouse['familia'] = le.fit_transform(smarthouse['familia'].values)
#smarthouse['familia']=le.inverse_transform(smarthouse['familia'])
```

In [5]:

```
target = smarthouse['tipo_sh']
```

In [6]:

```
cols_to_use = ['n_moradores', 'idosos', 'def_loc', 'def_b_v', 'def_cog', 'def_aud', 'comodos',
               'r_anual', 'A', 'B', 'C', 'D', 'E', 'instalacao', 'preco', 'crianca', 'familia']
```

In [7]:

```
X = smarthouse[cols_to_use]
y = target
```

In [8]:

```
X_treino, X_teste, y_treino, y_teste = train_test_split(X, y, test_size=0.30)
```

## NAIVE BAYES

In [9]:

```
gnb = GaussianNB()
pred = gnb.fit(X_treino, y_treino).predict(X_teste)

print("Naive-Bayes accuracy: ", accuracy_score(y_teste, pred) )
```

Naive-Bayes accuracy: 0.59375

In [10]:

```
print(confusion_matrix(y_teste, pred))
print(classification_report(y_teste, pred))
```

```
[[1 0 0 0 0 0]
 [0 7 0 1 0 0]
 [0 0 1 5 1 0]
 [0 0 1 2 1 0]
 [0 2 0 1 1 1]
 [0 0 0 0 0 7]]
```

	precision	recall	f1-score	support
SH	1.00	1.00	1.00	1
SHBV	0.78	0.88	0.82	8
SHDA	0.50	0.14	0.22	7
SHDC	0.22	0.50	0.31	4
SHDL	0.33	0.20	0.25	5
SHI	0.88	1.00	0.93	7
accuracy			0.59	32
macro avg	0.62	0.62	0.59	32
weighted avg	0.61	0.59	0.57	32

In [11]:

```
print (pd.crosstab(y_teste, pred, rownames=['Real'], colnames=['Predito'], margins=True))
```

Predito	SH	SHBV	SHDA	SHDC	SHDL	SHI	All
Real							
SH	1	0	0	0	0	0	1
SHBV	0	7	0	1	0	0	8
SHDA	0	0	1	5	1	0	7
SHDC	0	0	1	2	1	0	4
SHDL	0	2	0	1	1	1	5
SHI	0	0	0	0	0	7	7
All	1	9	2	9	3	8	32

In [12]:

```
#smarthouse['familia']=le.inverse_transform(smarthouse['familia'])
```

In [13]:

```
#from yellowbrick.classifier import ClassificationReport
#visualizer = ClassificationReport(gnb, classes=target)
#visualizer.fit(X_treino, y_treino) # Fit the training data to the visualizer
#visualizer.score(X_teste, y_teste) # Evaluate the model on the test data
#g = visualizer.poof() # Draw/show/poof the data
```

...

## LINEAR SVC

In [17]:

```
from sklearn.svm import LinearSVC
svc_model = LinearSVC(max_iter = 2500, random_state=0)
pred = svc_model.fit(X_treino, y_treino).predict(X_teste)
print("LinearSVC accuracy : ",accuracy_score(y_teste, pred, normalize = True))
```

LinearSVC accuracy : 0.5625

C:\Users\gabid\anaconda3\lib\site-packages\sklearn\svm\\_base.py:947: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.

"the number of iterations.", ConvergenceWarning)

In [18]:

```
print (pd.crosstab(y_teste, pred, rownames=['Real'], colnames=['Predito'], margins=True))
```

Predito	SHBV	SHDA	SHDC	SHDL	SHI	All
Real						
SH	0	0	0	1	0	1
SHBV	6	1	1	0	0	8
SHDA	4	0	3	0	0	7
SHDC	3	0	1	0	0	4
SHDL	0	0	1	4	0	5
SHI	0	0	0	0	7	7
All	13	1	6	5	7	32

In [19]:

```
print(confusion_matrix(y_teste, pred))
print(classification_report(y_teste, pred))
```

```
[[0 0 0 0 1 0]
 [0 6 1 1 0 0]
 [0 4 0 3 0 0]
 [0 3 0 1 0 0]
 [0 0 0 1 4 0]
 [0 0 0 0 0 7]]
```

	precision	recall	f1-score	support
SH	0.00	0.00	0.00	1
SHBV	0.46	0.75	0.57	8
SHDA	0.00	0.00	0.00	7
SHDC	0.17	0.25	0.20	4
SHDL	0.80	0.80	0.80	5
SHI	1.00	1.00	1.00	7
accuracy			0.56	32
macro avg	0.40	0.47	0.43	32
weighted avg	0.48	0.56	0.51	32

C:\Users\gabid\anaconda3\lib\site-packages\sklearn\metrics\\_classification.py:1272: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero\_division` parameter to control this behavior.

\_warn\_prf(average, modifier, msg\_start, len(result))

# K- NEIGHBORS

In [20]:

```
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier(n_neighbors=3)
neigh.fit(X_treino, y_treino)
pred = neigh.predict(X_teste)
print ("KNeighbors accuracy score : ",accuracy_score(y_teste, pred))
```

KNeighbors accuracy score : 0.4375

In [21]:

```
print (pd.crosstab(y_teste, pred, rownames=['Real'], colnames=['Predito'], margins=True))
```

Predito	SHBV	SHDA	SHDC	SHDL	SHI	All
Real						
SH	0	0	0	1	0	1
SHBV	5	0	3	0	0	8
SHDA	4	3	0	0	0	7
SHDC	2	1	1	0	0	4
SHDL	0	4	0	1	0	5
SHI	0	0	1	2	4	7
All	11	8	5	4	4	32

In [22]:

```
print(confusion_matrix(y_teste, pred))
print(classification_report(y_teste, pred))
```

```
[[0 0 0 0 1 0]
 [0 5 0 3 0 0]
 [0 4 3 0 0 0]
 [0 2 1 1 0 0]
 [0 0 4 0 1 0]
 [0 0 0 1 2 4]]
```

	precision	recall	f1-score	support
SH	0.00	0.00	0.00	1
SHBV	0.45	0.62	0.53	8
SHDA	0.38	0.43	0.40	7
SHDC	0.20	0.25	0.22	4
SHDL	0.25	0.20	0.22	5
SHI	1.00	0.57	0.73	7
accuracy			0.44	32
macro avg	0.38	0.35	0.35	32
weighted avg	0.48	0.44	0.44	32

C:\Users\gabid\anaconda3\lib\site-packages\sklearn\metrics\\_classification.p  
y:1272: UndefinedMetricWarning: Precision and F-score are ill-defined and be  
ing set to 0.0 in labels with no predicted samples. Use `zero\_division` para  
meter to control this behavior.

```
_warn_prf(average, modifier, msg_start, len(result))
```

In [ ]:

