```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data = pd.read_csv("luna_pets.csv")
```

```
data.head()
```

	X	Y	Z	R	G	B	Intensity
0	731000.31	9246012.06	3317.59	126	119	163	5911.0
1	731002.53	9246010.16	3316.46	118	110	160	6939.0
2	731000.50	9246012.30	3316.79	127	121	159	3855.0
3	731000.25	9246012.73	3317.08	126	120	157	5654.0
4	731001.47	9246010.60	3317.28	121	114	159	4369.0

Next steps: Generate code with `data` View recommended plots

```
X = data.drop(["Y","Z"], axis=1)
y = data["Y"]
```

```
y = pd.cut(y, bins=2, labels=['baixo venda', 'alta venda'])
```

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
X_train, X_val, y_train, y_val = train_test_split(X_scaled, y, test_size=0.3, random_state=42)
```

```
knn_cls = KNeighborsClassifier(n_neighbors=5)
```

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

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

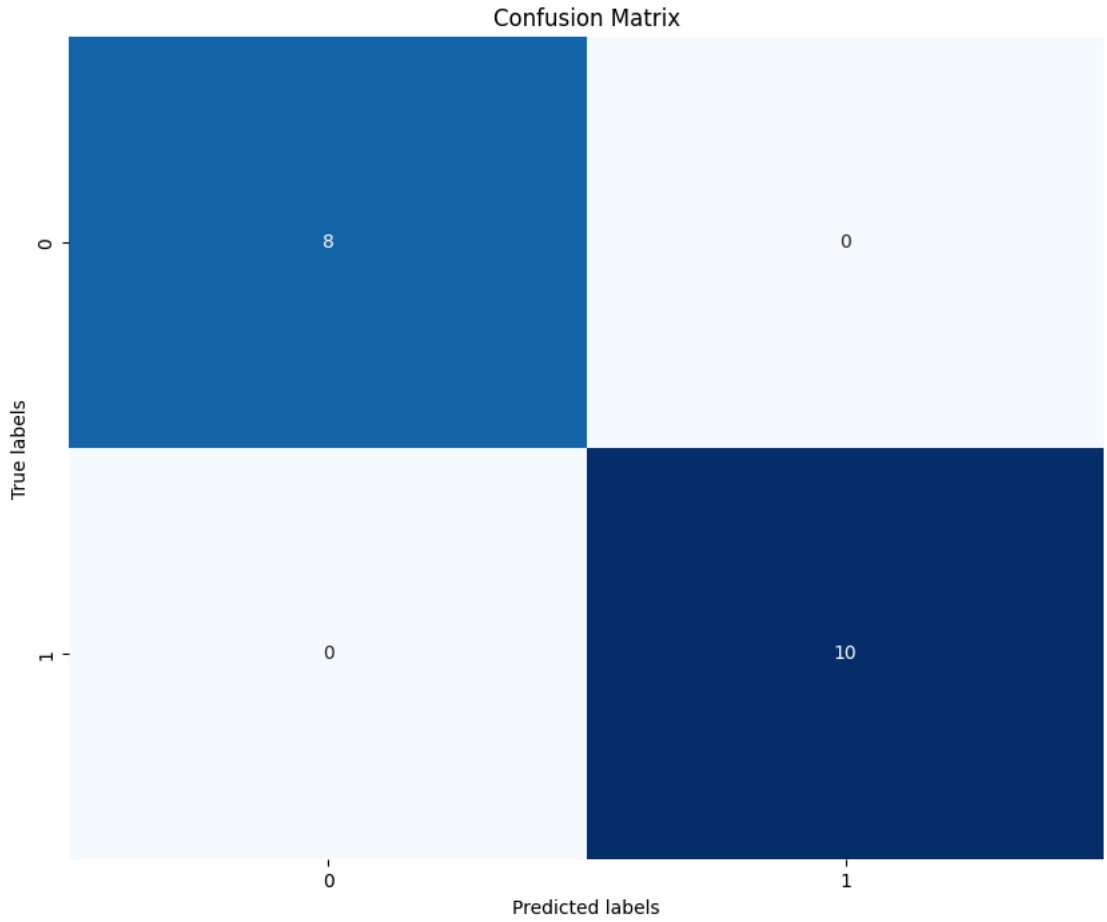
```
y_pred = knn_cls.predict(X_val)
```

```
accuracy = accuracy_score(y_val, y_pred)
precision = precision_score(y_val, y_pred, average='weighted', zero_division='warn')
recall = recall_score(y_val, y_pred, average='weighted', zero_division='warn')
f1 = f1_score(y_val, y_pred, average='weighted', zero_division='warn')
```

```
print("Acurácia:", accuracy)
print("Precisão:", precision)
print("Recall:", recall)
print("F1-score:", f1)
```

```
Acurácia: 1.0
Precisão: 1.0
Recall: 1.0
F1-score: 1.0
```

```
cm = confusion_matrix(y_val, y_pred)
plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False)
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix')
plt.show()
```


Confusion Matrix
8
0
0
10
True labels
Predicted labels
0
1
0
1