```
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("lunap.csv")
```

```
data.head()
```

	date	sales	customer	cost_of_operation	value_of_sales	profit_from_sales
0	1.1.22	150	120	5000	35000	15000
1	2.1.22	200	140	5000	35000	18000
2	3.1.22	250	160	5000	35000	20000
3	4.1.22	300	180	5000	35000	22000
4	5.1.22	350	200	5000	35000	24500

Next steps: View recommended plots

```
X = data.drop(["profit_from_sales","date"], axis=1)
y = data["profit_from_sales"]
```

```
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()
```

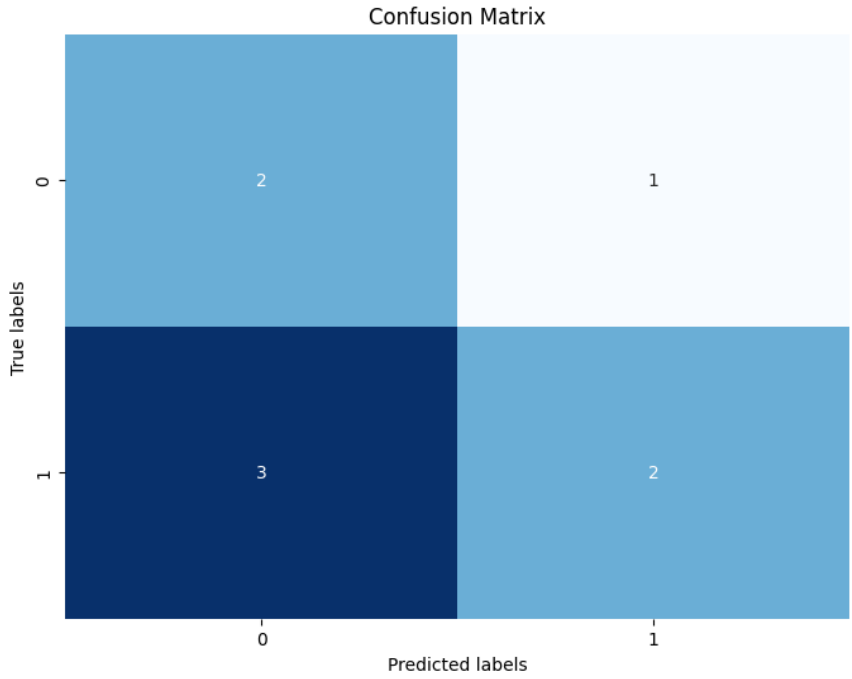
```
y_pred = knn_cls.predict(X_val)
```

```
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: 0.5
Precisão: 0.5666666666666667
Recall: 0.5
F1-score: 0.5
```

```
cm = confusion_matrix(y_val, y_pred)
plt.figure(figsize=(8, 6))
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
2
1
3
2
0
1
True labels
0
1
Predicted labels