

TASK-03

PROBLEM STATEMENT - Build a decision tree classifier to predict whether a customer will purchase a product or service based on their demographic and behavioral data. Use a dataset such as the Bank Marketing dataset from the UCI Machine Learning Repository.

Description:- The data is related with banking institution.

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

```
# Load the dataset
data = pd.read_csv('bank.csv')
data
```

default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	depos
no	2343	yes	no	unknown	5	may	1042	1	-1	0	unknown	y
no	45	no	no	unknown	5	may	1467	1	-1	0	unknown	y
no	1270	yes	no	unknown	5	may	1389	1	-1	0	unknown	y
no	2476	yes	no	unknown	5	may	579	1	-1	0	unknown	y
no	184	no	no	unknown	5	may	673	2	-1	0	unknown	y
...
no	1	yes	no	cellular	20	apr	257	1	-1	0	unknown	
no	733	no	no	unknown	16	jun	83	4	-1	0	unknown	
no	29	no	no	cellular	19	aug	156	2	-1	0	unknown	
no	0	no	yes	cellular	8	may	9	2	172	5	failure	
no	0	no	no	cellular	9	jul	628	1	-1	0	unknown	

Next steps:

[Generate code with data](#)
[View recommended plots](#)

```
# Preprocess the data
X = data.drop('deposit', axis=1)
y = data['deposit']
X = pd.get_dummies(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Create the classifier
clf = DecisionTreeClassifier()
```

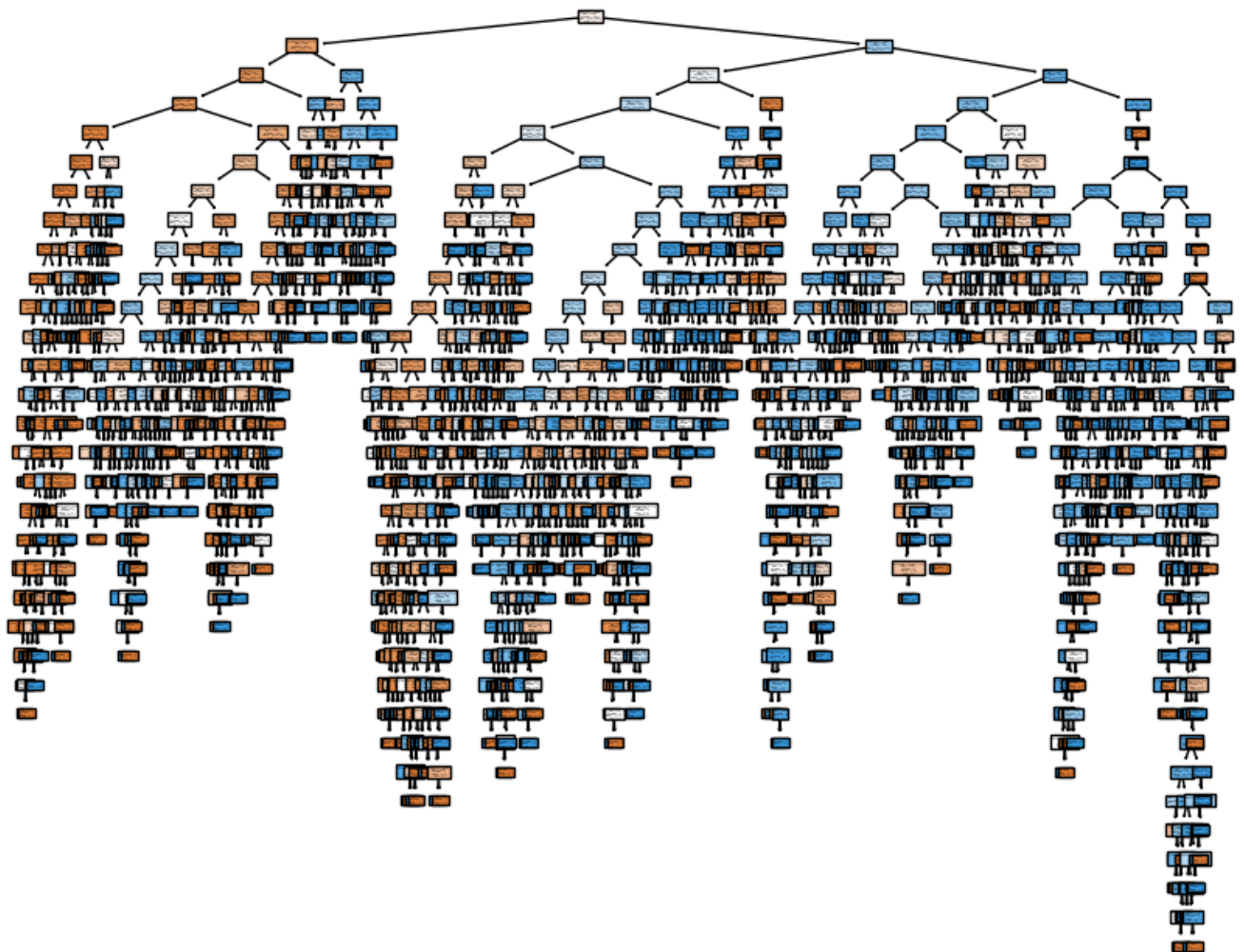
```
# Train the classifier
clf.fit(X_train, y_train)
```

```
# Make predictions
y_pred = clf.predict(X_test)
```

```
# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Accuracy: 0.781012091356919

```
# Visualize the decision tree
plt.figure(figsize=(10, 8))
plot_tree(clf, feature_names=X.columns, class_names=['no', 'yes'], filled=True)
plt.show()
```



```

# Create the classifier with pruning enabled
clf = DecisionTreeClassifier(ccp_alpha=0.01)

# Train the classifier
clf.fit(X_train, y_train)

# Make predictions
y_pred = clf.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)

# Visualize the pruned decision tree
plt.figure(figsize=(10, 8))
plot_tree(clf, feature_names=X.columns, class_names=['no', 'yes'], filled=True)
plt.show()

```

