

```
In [45]: 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.preprocessing import LabelEncoder
from sklearn import tree
import matplotlib.pyplot as plt
```

```
In [46]: # Data Preprocessing steps
df2 = pd.read_csv("C:/Users/sujit/AppData/Local/Temp/bf494293-ca08-4a7d-a824-017f7f13a6bd_bank.zip.6bd/bank-full.csv", delimiter=';')
```

```
In [47]: df2
```

```
Out[47]:
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no
...
45206	51	technician	married	tertiary	no	825	no	no	cellular	17	nov	977	3	-1	0	unknown	yes
45207	71	retired	divorced	primary	no	1729	no	no	cellular	17	nov	456	2	-1	0	unknown	yes
45208	72	retired	married	secondary	no	5715	no	no	cellular	17	nov	1127	5	184	3	success	yes
45209	57	blue-collar	married	secondary	no	668	no	no	telephone	17	nov	508	4	-1	0	unknown	no
45210	37	entrepreneur	married	secondary	no	2971	no	no	cellular	17	nov	361	2	188	11	other	no

45211 rows × 17 columns

In [48]: df2.head(10)

Out[48]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no
5	35	management	married	tertiary	no	231	yes	no	unknown	5	may	139	1	-1	0	unknown	no
6	28	management	single	tertiary	no	447	yes	yes	unknown	5	may	217	1	-1	0	unknown	no
7	42	entrepreneur	divorced	tertiary	yes	2	yes	no	unknown	5	may	380	1	-1	0	unknown	no
8	58	retired	married	primary	no	121	yes	no	unknown	5	may	50	1	-1	0	unknown	no
9	43	technician	single	secondary	no	593	yes	no	unknown	5	may	55	1	-1	0	unknown	no

```
In [55]: import altair as alt
df2 = df2.head(5000)
chart = alt.Chart(df2).mark_bar().encode(
    x='age:Q',
    y=alt.Y('count()', title='Loan'), # Set the y-axis title here
    color='job:N',
).properties(height=400, width=800)
chart

# Assuming 'y' is the target variable
X = df2.drop(columns=['y'])
y = df2['y']

label_encoders = {}
for column in X.select_dtypes(include=['object']).columns:
    label_encoders[column] = LabelEncoder()
    X[column] = label_encoders[column].fit_transform(X[column])
```

```
In [56]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [43]: y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print("\nClassification Report:\n", classification_report(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

Accuracy: 0.958

Classification Report:

	precision	recall	f1-score	support
no	0.98	0.98	0.98	965
yes	0.39	0.34	0.36	35
accuracy			0.96	1000
macro avg	0.68	0.66	0.67	1000
weighted avg	0.96	0.96	0.96	1000

Confusion Matrix:

```
[[946  19]
 [ 23  12]]
```

```
In [59]: clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train, y_train)
```

```
Out[59]: ▼      DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

```
plt.figure(figsize=(20,10))
tree.plot_tree(cif, feature_names_X.columns.tolist(), class_names=self.classes_.tolist(), filled=True)
plt.show()
```

