

```
# Step 1: Upload the CSV in Colab
from google.colab import files
uploaded = files.upload()
```

Enable browser notifications to get alerts when executions complete

OK

No th

Choose Files healthcare_...ntments.csv

- healthcare_noshows_appointments.csv(text/csv) - 11637103 bytes, last modified: 5/10/2025 - 100% done

```
import pandas as pd
import numpy as np

df = pd.read_csv('healthcare_noshows_appointments.csv')

df.drop(['PatientId', 'AppointmentID'], axis=1, inplace=True)

df['ScheduledDay'] = pd.to_datetime(df['ScheduledDay'])
df['AppointmentDay'] = pd.to_datetime(df['AppointmentDay'])

df['AppointmentWeekday'] = df['AppointmentDay'].dt.day_name()

df['Gender'] = df['Gender'].map({'M': 1, 'F': 0})

df['Showed_up'] = df['Showed_up'].map({True: 1, False: 0})

print(df.info())
print(df.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 106987 entries, 0 to 106986
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Gender                106987 non-null  int64
1   ScheduledDay          106987 non-null  datetime64[ns]
2   AppointmentDay        106987 non-null  datetime64[ns]
3   Age                  106987 non-null  int64
4   Neighbourhood         106987 non-null  object
5   Scholarship           106987 non-null  bool
6   Hipertension          106987 non-null  bool
7   Diabetes              106987 non-null  bool
8   Alcoholism            106987 non-null  bool
9   Handcap               106987 non-null  bool
10  SMS_received          106987 non-null  bool
11  Showed_up             106987 non-null  int64
12  Date.diff              106987 non-null  int64
13  AppointmentWeekday    106987 non-null  object
dtypes: bool(6), datetime64[ns](2), int64(4), object(2)
memory usage: 7.1+ MB
None
```

	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	\
0	0	2016-04-29	2016-04-29	62	JARDIM DA PENHA	False	
1	1	2016-04-29	2016-04-29	56	JARDIM DA PENHA	False	
2	0	2016-04-29	2016-04-29	62	MATA DA PRAIA	False	
3	0	2016-04-29	2016-04-29	8	PONTAL DE CAMBURI	False	
4	0	2016-04-29	2016-04-29	56	JARDIM DA PENHA	False	

	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received	Showed_up	\
0	True	False	False	False	False	1	
1	False	False	False	False	False	1	
2	False	False	False	False	False	1	
3	False	False	False	False	False	1	
4	True	True	False	False	False	1	

	Date.diff	AppointmentWeekday
0	0	Friday
1	0	Friday
2	0	Friday
3	0	Friday
4	0	Friday

```
print(df.dtypes)
```

```
Gender                int64
ScheduledDay          datetime64[ns]
AppointmentDay        datetime64[ns]
Age                  int64
Neighbourhood         object
Scholarship           bool
Hipertension          bool
Diabetes              bool
```

```

Alcoholism          bool
Handcap             bool
SMS_received        bool
Showed_up           int64
Date.diff           int64
AppointmentWeekday  object
dtype: object

```

Enable browser notifications to get alerts when executions complete

```

# Import required libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report

# Step 2: Load and clean the data
df = pd.read_csv('healthcare_noshows_appointments.csv')
df.drop(['PatientId', 'AppointmentID'], axis=1, inplace=True)

df['ScheduledDay'] = pd.to_datetime(df['ScheduledDay'])
df['AppointmentDay'] = pd.to_datetime(df['AppointmentDay'])

df['Gender'] = df['Gender'].map({'M': 1, 'F': 0})
df['Showed_up'] = df['Showed_up'].map({True: 1, False: 0})

# Step 3: Create weekday column
if 'AppointmentWeekday' not in df.columns:
    df['AppointmentWeekday'] = df['AppointmentDay'].dt.day_name()

# Step 4: Create dummy variables for weekdays
df = pd.get_dummies(df, columns=['AppointmentWeekday'], drop_first=True)

# Convert Neighbourhood to dummy variables
df = pd.get_dummies(df, columns=['Neighbourhood'], drop_first=True)

# Step 5: Define features and label
df.drop(['ScheduledDay', 'AppointmentDay'], axis=1, inplace=True)

feature_columns = [col for col in df.columns if col != 'Showed_up']
X = df[feature_columns]
y = df['Showed_up']

# Step 6: Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 7: Train model
model = DecisionTreeClassifier(random_state=42)
model.fit(X_train, y_train)

# Step 8: Predictions and evaluation
y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

```

➦ Accuracy: 0.7274044303205907

```

Classification Report:
              precision    recall  f1-score   support

     0       0.33         0.33         0.33         4325
     1       0.83         0.83         0.83        17073

 accuracy          0.73
 macro avg         0.58
weighted avg         0.73

```

Enable browser notifica
Settings to get alerts wl
executions complete