```
#create a sampleData set
import pandas as pd
import numpy as np

data = {
    'customer_id': np.arange(100),
    'age': np.random.randint(18, 65, 100),
    'tenure': np.random.randint(0, 72, 100),
    'churned': np.random.choice([0, 1], 100, p=[0.8, 0.2])
}
df = pd.DataFrame(data)
```

```
df
```

	customer_id	age	tenure	churned
0	0	37	27	0
1	1	57	24	1
2	2	46	10	1
3	3	35	31	1
4	4	20	19	0
...	...	...	...	...
95	95	27	37	1
96	96	61	26	1
97	97	40	67	0
98	98	56	31	0
99	99	55	36	0

100 rows × 4 columns

Next steps: Generate code with df   View recommended plots   New interactive sheet

```
#calculate averages
churn_stats = df.groupby('churned')[['age', 'tenure']].mean()

print(churn_stats)
```

```
               age     tenure
churned
0        39.525641  39.089744
1        40.863636  31.909091
```

```
#Split the dataset
from sklearn.model_selection import train_test_split

X = df[['age', 'tenure']]
y = df['churned']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
#model training
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

model = LogisticRegression()

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

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

```
Accuracy: 0.9
```