```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from matplotlib import pyplot as plt


from google.colab import files
uploaded = files.upload()
```

Choose Files insurance_data.csv

- **insurance_data.csv**(text/csv) - 155 bytes, last modified: 3/24/2025 - 100% done

```
Saving insurance_data.csv to insurance_data.csv
```

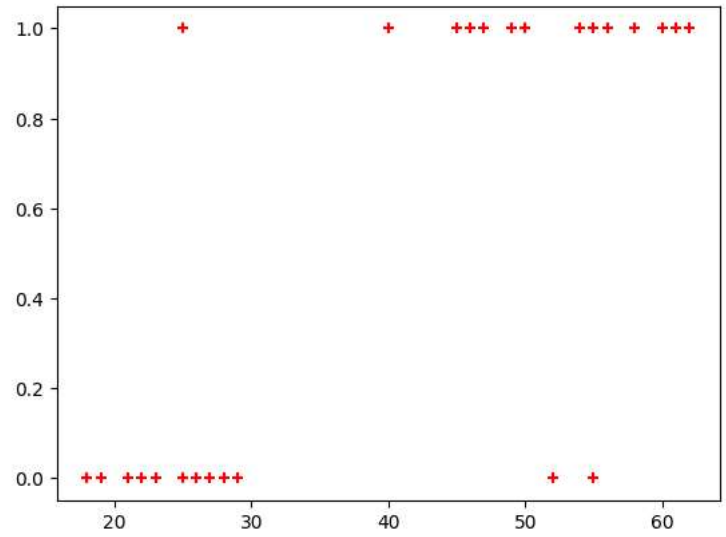
```
df = pd.read_csv('insurance_data.csv')


df
```

	age	bought_insurance
0	22	0
1	25	0
2	47	1
3	52	0
4	46	1
5	56	1
6	55	0
7	60	1
8	62	1
9	61	1
10	18	0
11	28	0
12	27	0
13	29	0
14	49	1
15	55	1
16	25	1
17	58	1
18	19	0
19	18	0
20	21	0
21	26	0
22	40	1
23	45	1
24	50	1
25	54	1
26	23	0

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

```
plt.scatter(df.age,df.bought_insurance,marker='+',color='red')
```

```
<matplotlib.collections.PathCollection at 0x79be5eda76d0>
```




```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df[['age']],df.bought_insurance,train_size=0.9,random_state=10)
X_train.shape
```

```
(24, 1)
```

```
X_test
```

	age
7	60
5	56
18	19

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

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
```

```
model.fit(X_train, y_train)
```

```
X_test
```

```
y_test
```

```
y_predicted = model.predict(X_test)
y_predicted
```

```
array([1, 1, 0])
```

```
model.score(X_test,y_test)
```

```
model.predict_proba(X_test)
```

```
y_predicted = model.predict([[60]])
y_predicted
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but Logistic
  warnings.warn(
array([1])
```

```
model.coef_
```

```
array([[0.1274065]])
```

```
model.intercept_
```

```
array([-4.97339194])
```

```
import math
def sigmoid(x):
  return 1 / (1 + math.exp(-x))

def prediction_function(age):
   z = 0.127 * age - 4.973 # 0.12740563 ~ 0.0127 and -4.97335111 ~ -4.97
   y = sigmoid(z)
   return y

age = 35
prediction_function(age)
```

```
0.3709834769552775
```