

```
import numpy as np
import pandas as pd
```

```
df=pd.read_csv('/content/insurance - insurance.csv')
df.head(2)
```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.90	0	yes	southwest	16884.9240
1	18	male	33.77	1	no	southeast	1725.5523

```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
```

```
lb=LabelEncoder()
```

```
df['sex']=lb.fit_transform(df['sex'])
df['smoker']=lb.fit_transform(df['smoker'])
df['region']=lb.fit_transform(df['region'])

df.head(2)
```

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.90	0	1	3	16884.9240
1	18	1	33.77	1	0	2	1725.5523

```
x=df.drop('charges',axis=1)
y=df['charges']
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_stat
```

```
from sklearn.linear_model import LinearRegression
```

```
lr=LinearRegression()
```

```
lr.fit(x_train,y_train)
```

▼ LinearRegression ⓘ ?
LinearRegression()

```
y_pred=lr.predict(x_test)
```

```
from sklearn.metrics import r2_score
```

```
r2_score(y_test,y_pred)
```

```
0.7998747145449959
```

```
from sklearn.linear_model import Lasso,Ridge
```

```
# ridge regression model  
ridge=Ridge(alpha=10)  
ridge.fit(x_train,y_train)  
  
# train and test scorefor ridge regression  
  
train_score_ridge=ridge.score(x_train,y_train)  
test_score_ridge=ridge.score(x_test,y_test)
```

```
# lasso
```

```
# lasso regression model  
  
lasso=Lasso(alpha=10)  
lasso.fit(x_train,y_train)  
  
# train and test scorefor lasso regression  
  
train_score_ls=lasso.score(x_train,y_train)  
test_score_ls=lasso.score(x_test,y_test)  
  
print('lasso regression train scope' ,train_score_ls)  
print('lasso regression test scope' ,test_score_ls)
```

```
lasso regression train scope 0.7368242847139901  
lasso regression test scope 0.7996147889122028
```

```
# polynomial regression
```

```
from sklearn.preprocessing import PolynomialFeatures
```

```
# mymodel=numpy.poly1d(numpy.polyfit(x,y,3))
```

```
# r2_score(y_test,mymodel(x_test))
```

```
0.9398627467762428
```

```
poly=PolynomialFeatures(degree=4)  
x_poly=poly.fit_transform(x_train)
```

```
poly.fit(x_poly,y_train)
```

▼ PolynomialFeatures ⓘ ?
PolynomialFeatures(degree=4)

```
lr=LinearRegression()  
lr.fit(x_poly,y_train)
```

▼ LinearRegression ⓘ ?
LinearRegression()

```
y_pred=lr.predict(poly.fit_transform(x_test))
```

```
r2_score(y_test,y_pred)
```

```
0.861317444847725
```