

```
In [3]: import pandas as pd
data=pd.read_csv("insurance_pre.csv")
data
```

Out[3]:

	age	sex	bmi	children	smoker	charges
0	19	female	27.900	0	yes	16884.92400
1	18	male	33.770	1	no	1725.55230
2	28	male	33.000	3	no	4449.46200
3	33	male	22.705	0	no	21984.47061
4	32	male	28.880	0	no	3866.85520
...	...	...	...	...	...	...
1333	50	male	30.970	3	no	10600.54830
1334	18	female	31.920	0	no	2205.98080
1335	18	female	36.850	0	no	1629.83350
1336	21	female	25.800	0	no	2007.94500
1337	61	female	29.070	0	yes	29141.36030

1338 rows x 6 columns

```
In [4]: data=pd.get_dummies(data.drop_first=True)
```



```
In [3]: import pandas as pd
data=pd.read_csv("insurance_pre.csv")
data
```

Out[3]:

	age	sex	bmi	children	smoker	charges
0	19	female	27.900	0	yes	16884.92400
1	18	male	33.770	1	no	1725.55230
2	28	male	33.000	3	no	4449.46200
3	33	male	22.705	0	no	21984.47061
4	32	male	28.880	0	no	3866.85520
...	...	...	...	...	...	...
1333	50	male	30.970	3	no	10600.54830
1334	18	female	31.920	0	no	2205.98080
1335	18	female	36.850	0	no	1629.83350
1336	21	female	25.800	0	no	2007.94500
1337	61	female	29.070	0	yes	29141.36030

1338 rows × 6 columns

```
In [4]: data=pd.get_dummies(data.drop first=True)
```

1338 rows × 6 columns

```
In [4]: data=pd.get_dummies(data,drop_first=True)
data
```

Out[4]:

	age	bmi	children	charges	sex_male	smoker_yes
0	19	27.900	0	16884.92400	False	True
1	18	33.770	1	1725.55230	True	False
2	28	33.000	3	4449.46200	True	False
3	33	22.705	0	21984.47061	True	False
4	32	28.880	0	3066.85520	True	False
...	...	...	...	...	...	...
1333	50	30.970	3	10600.54830	True	False
1334	18	31.920	0	2205.98080	False	False
1335	18	36.850	0	1629.83350	False	False
1336	21	25.800	0	2007.94500	False	False
1337	61	29.070	0	29141.36030	False	True

1338 rows × 6 columns

```
In [5]: import numpy as np
```

1338 rows x 5 columns

```
In [7]: dependent = data[["charges"]]
dependent
```

Out[7]:

	charges
0	16884.92400
1	1725.55230
2	4449.46200
3	21984.47061
4	3866.85520
...	...
1333	10600.54830
1334	2205.98080
1335	1629.83350
1336	2007.94500
1337	29141.36030

1338 rows x 1 columns

```
In [8]: from sklearn.model_selection import train_test_split
```



1338 rows x 1 columns

```
In [8]: from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(independent, dependent, test_size=0.2, random_state=42)
```

```
In [9]: from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
```

```
x_train=scaler.fit_transform(x_train)
```

```
x_test=scaler.transform(x_test)
```

```
x_train
```

```
Out[9]: array([[ 0.47222651, -1.75652513,  0.73433626, -1.0246016 , -0.50874702],
 [ 0.54331294, -1.03308239, -0.91119211, -1.0246016 , -0.50874702],
 [ 0.8987451 , -0.94368672, -0.91119211, -1.0246016 , -0.50874702],
 ...,
 [ 1.3252637 , -0.89153925, -0.91119211,  0.97598911, -0.50874702],
 [-0.16755139,  2.82086429,  0.73433626, -1.0246016 ,  1.96561348],
 [ 1.1120044 , -0.10932713, -0.91119211,  0.97598911, -0.50874702]])
```

```
In [10]: from sklearn.ensemble import RandomForestRegressor
```

```
from sklearn.datasets import make_regression
```

```
regressor=RandomForestRegressor(criterion="squared_error",max_features="sqrt",random_state=0)
```

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

C:\Users\CSLAB\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

```
return fit_method(estimator, *args, **kwargs)
```



C:\Users\CSLAB\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().  
return fit\_method(estimator, \*args, \*\*kwargs)

Out[10]:

RandomForestRegressor  
RandomForestRegressor(max\_features='sqrt', random\_state=0)

```
In [*]: weight = regressor.feature_importances_  
print("weight of the model-{}", format (weight))  
bais = regressor.feature_importances_  
print("bais of the model -{}", format (bais))  
  
y_pred = regressor.predict(x_test)  
from sklearn.metrics import r2_score  
r_score = r2_score(y_test, y_pred)  
print("r_score=", r_score)  
import numpy as np  
age=int(input("enter the value:"))  
bmi=int(input("enter ther value:"))  
children=int(input("enter the value:"))  
sex=int(input("enter the gender:"))  
smoker=int(input("enter the value:"))
```

1338 rows x 6 columns

```
In [5]: import numpy as np
import matplotlib.pyplot as plt
independent=data[["age", "bmi", "children", "sex_male", "smoker_yes"]]
dependent=data[["charges"]]
```

```
In [6]: independent=data[["age", "bmi", "children", "sex_male", "smoker_yes"]]
independent
```

Out[6]:

	age	bmi	children	sex_male	smoker_yes
0	19	27.900	0	False	True
1	18	33.770	1	True	False
2	28	33.000	3	True	False
3	33	22.705	0	True	False
4	32	28.880	0	True	False
...	...	...	...	...	...
1333	50	30.970	3	True	False
1334	18	31.920	0	False	False
1335	18	36.850	0	False	False
1336	21	25.800	0	False	False
1337	24	29.870	0	False	True

```
In [*]: weight = regressor.feature_importances_
print("weight of the model-{}", format (weight))
bais = regressor.feature_importances_
print("bais of the model -{}", format (bais))

y_pred = regressor.predict(x_test)
from sklearn.metrics import r2_score
r_score = r2_score(y_test, y_pred)
print("r_score=", r_score)
import numpy as np
age=int(input("enter the value:"))
bmi=int(input("enter ther value:"))
children=int(input("enter the value:"))
sex=int(input("enter the gender:"))
smoker=int(input("enter the value:"))

future_prediction = regressor.predict([[age,bmi,children,sex,smoker]])
print("future_prediction = {}", format(future_prediction))

weight of the model={} [0.15316259 0.18834183 0.02677339 0.00922418 0.62249801]
bais of the model -{} [0.15316259 0.18834183 0.02677339 0.00922418 0.62249801]
r_score= 0.8635372570210067
enter the value:0
```