```
#Daksh·Gupta
#GH·Raisoni·Institute·of·Engineering·and·Technology
#2nd·year
#CSE·branch
#MAJOR PROJECT(APPLYING REGRESSOR)
```

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

```
df=pd.read_csv('/content/framingham.csv')
df
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStr
0	1	39	4.0	0	0.0	0.0	
1	0	46	2.0	0	0.0	0.0	
2	1	48	1.0	1	20.0	0.0	
3	0	61	3.0	1	30.0	0.0	
4	0	46	3.0	1	23.0	0.0	
...	...	...	...	...	...	...	
4233	1	50	1.0	1	1.0	0.0	
4234	1	51	3.0	1	43.0	0.0	
4235	0	48	2.0	1	20.0	NaN	
4236	0	44	1.0	1	15.0	0.0	
4237	0	52	2.0	0	0.0	0.0	

4238 rows × 16 columns

```
df.isnull().sum()
```

```
male                0
age                 0
education         105
currentSmoker       0
cigsPerDay         29
BPMeds             53
prevalentStroke     0
prevalentHyp        0
diabetes            0
totChol            50
sysBP               0
diaBP               0
BMI                19
heartRate           1
glucose           388
TenYearCHD          0
dtype: int64
```

```
df=df.dropna()  #REMOVING NaN values
```

```
x=df.iloc[:,0:15].values    #assigning input output
x
```

```
array([[  1.  ,  39.  ,   4.  , ...,  26.97,  80.  ,  77.  ],
       [  0.  ,  46.  ,   2.  , ...,  28.73,  95.  ,  76.  ],
       [  1.  ,  48.  ,   1.  , ...,  25.34,  75.  ,  70.  ],
       ...,
       [  1.  ,  50.  ,   1.  , ...,  25.97,  66.  ,  86.  ],
       [  1.  ,  51.  ,   3.  , ...,  19.71,  65.  ,  68.  ],
       [  0.  ,  52.  ,   2.  , ...,  21.47,  80.  , 107.  ]])
```

```
y=df.iloc[:,15].values
y
```

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

```
x.shape
```

```
(3656, 15)
```

```
y.shape
```

```
(3656,)
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=0)
```

```
from sklearn.preprocessing import MinMaxScaler
```

```
scaler=MinMaxScaler()
```

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

```
from sklearn.linear_model import LogisticRegression
```

```
model=LogisticRegression()
```

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

```
LogisticRegression()
```

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

```
LogisticRegression()
```

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

```
y_pred
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]])
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)*100
```

```
83.47921225382933
```

```
model.predict([[0,23,3,1,10,0,0,0,0,210,110,80,30,95,77]])  #to predict whether person will have chronic heart disease aft
```

```
array([1])
```

```
#Daksh Gupta
#GH Raisoni Institute of Engineering and Technology
#2nd year
#CSE branch
#MAJOR PROJECT 2
#Exploratory data analysis(15+ conclusions)
```

```
df=pd.read_csv('/content/IT Salary Survey EU  2020.csv')
df
```

ty el	Your main technology / programming language	Other technologies/programming languages you use often	...	Annual bonus+stocks one year ago. Only answer if staying in same country	Number of vacation days	Employment status
ior	TypeScript	Kotlin, Javascript / Typescript	...	10000	30	Full-time employee
ior	Ruby	NaN	...	5000	28	Full-time employee
ad	Javascript / Typescript	Javascript / Typescript, Docker	...	100000	30	Self-employed (freelancer)
ior	Javascript	NaN	...	NaN	24	Full-time employee
ior	C# .NET	.NET, SQL, AWS, Docker	...	NaN	29	Full-time employee
...	...	...	...	...	...	...
ior	Java	Python, Javascript / Typescript, Java / Scala,...	...	72000	26	Full-time employee
ior	consumer analysis	NaN	...	2500	unlimited	Full-time employee

```
import seaborn as sns
```

```
df.City.unique()
```

```
array(['Munich', 'Berlin', 'Hamburg', 'Wolfsburg', 'Stuttgart',
       'Schleswig-Holstein', 'London', 'Konstanz area', 'Frankfurt',
       'Cologne', 'Kempten', 'Münster', 'Erlangen', 'Vienna', 'Moldova',
       'Rosenheim', 'Mannheim ', 'Boeblingen', 'Düsseldorf', 'Ingolstadt',
       'Nürnberg', 'Ansbach', 'Leipzig', 'Mannheim', 'Tuttlingen', 'Bonn',
       'Moscow', 'Koblenz', 'Warsaw', 'Heidelberg', 'Karlsruhe', 'Köln',
       'Aachen', 'Karlsruhe ', 'Samara', 'Riga, Latvia', 'Dusseldorf',
       'Zurich', 'Helsinki', 'Würzburg', 'Kiev', 'Den Haag', 'Amsterdam',
       'Cracovia', 'Tallinn', 'Prague', 'Utrecht', 'Stockholm',
       'Braunschweig ', 'Dresden', 'Kyiv', 'Stuttgart ', 'Malta',
       'Lübeck', 'Nuremberg ', 'Bodensee', 'Milan', 'Salzburg', 'Rome',
       'Wroclaw', 'Cupertino', 'Paris', 'Dublin ', 'Paderborn',
       'Konstanz', 'Ulm', 'Düsseldorf ', 'Barcelona', 'Bölingen',
       'Tampere (Finland)', 'Hannover', 'Bucharest', 'Siegen', 'Minsk',
       'Nuremberg', 'Marseille', 'Friedrichshafen', 'Walldorf',
       'Eindhoven', 'France', 'Regensburg', 'Warsaw, Poland', 'Heilbronn',
       'Dortmund', 'Jena', 'City in Russia', 'Cracow', 'Brunswick',
       'Dublin', 'Madrid', 'Lisbon', 'Sofia', 'Luttich', 'Milano',
       'Bielefeld', 'Basel', 'NJ, USA', 'Istanbul', 'warsaw',
       'Hildesheim', 'Heidelberg ', 'Murnau am Staffelsee ', 'Zürich',
       'Innsbruck', 'Brussels ', 'Hildesheim ', 'Prefer not to say',
       'Dusseldurf', 'Fr', 'Darmstadt', 'Duesseldorf', 'Cambridge',
       'Brussels', 'Copenhagen', 'Sevilla', 'Krakow', 'Ingolstadt ',
       'Saint-Petersburg', 'Saarbrücken'], dtype=object)
```

```
df.isnull().sum()
```

```
Timestamp
0
Age
27
Gender
10
City
0
Position
6
Total years of experience
16
Years of experience in Germany
32
Seniority level
12
Your main technology / programming language
127
Other technologies/programming languages you use often
157
Yearly brutto salary (without bonus and stocks) in EUR
0
Yearly bonus + stocks in EUR
424
Annual brutto salary (without bonus and stocks) one year ago. Only answer if staying in the same country
368
Annual bonus+stocks one year ago. Only answer if staying in same country
639
Number of vacation days
68
Employment status
17
Contract duration
29
Main language at work
16
Company size
18
Company type
25
Have you lost your job due to the coronavirus outbreak?
20
Have you been forced to have a shorter working week (Kurzarbeit)? If yes, how many hours per week
880
Have you received additional monetary support from your employer due to Work From Home? If yes, how much in 2020 in
EUR    791
dtype: int64
```

```
df.info
```

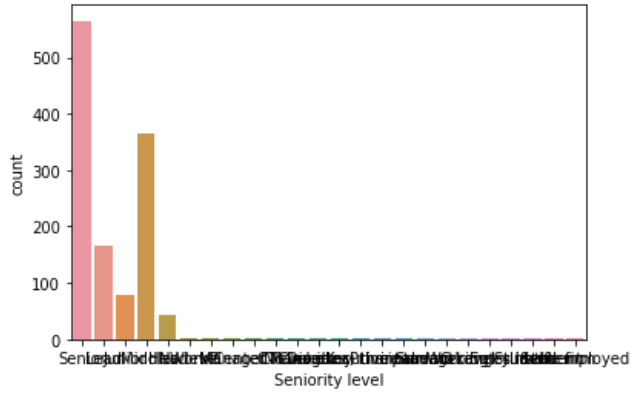
```
3                                                    No
4                                                    No
...                                                 ...
1248                                                Yes
1249                                                 No
1250                                                 No
1251                                                 No
1252                                                 No

     Have you been forced to have a shorter working week (Kurzarbeit)? If yes, how many hours per week  \
0                                                   NaN
1                                                   NaN
2                                                   NaN
3                                                   NaN
4                                                   NaN
...                                                 ...
1248                                                NaN
1249                                                NaN
1250                                                NaN
1251                                                NaN
1252                                               30.0

     Have you received additional monetary support from your employer due to Work From Home? If yes, how much in
2020 in EUR
0                                                   NaN
1                                                   NaN
2                                                   NaN
3                                                   NaN
4                                                   NaN
...                                                 ...
1248                                                NaN
1249                                                  0
1250                                                NaN
1251                                                  0
1252                                                600
```

```
df.groupby('Gender').size()
```

```
Gender
Diverse       2
Female      192
Male       1049
dtype: int64
```

```
sns.countplot(df['Seniority level'])
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: P
  warnings.warn(
<matplotlib.axes._subplots.AxesSubplot at 0x7f80d90f8d00>
```




```
df['Seniority level'].nunique()
```

```
24
```

```
import numpy as np
young = np.sum((df['Age']>=0)&(df['Age']<20)) #0 to 19
adult = np.sum((df['Age']>=20)&(df['Age']<40)) #20 to 39
midage = np.sum((df['Age']>=40)&(df['Age']<60)) #40 to 59
old = np.sum((df['Age']>=60))#60 and above
```

```
young  #number of people aged between 0 and 20
```

```
0
```

```
adult   #number of people aged between 20 and 40
```

```
1103
```

```
midage   #number of people aged between 40 and 60
```

```
120
```

```
old     #number of people aged above 60
```

```
3
```

```
df=df.dropna()
yoe1=np.sum((df['Total years of experience'])=='10')  #number of people with exactly 10 years of experience
yoe1
```

```
15
```

```
java=np.sum((df['Your main technology / programming language']=='Java'))   #number of people with programming language java
java
```

```
24
```

```
fulltime=np.sum((df['Employment status']=='Full-time employee'))    #number of people with employment status full time
fulltime
```

```
141
```

```
unlimited=np.sum((df['Contract duration']=='Unlimited contract'))   #number of people with contract duration unlimited
unlimited
```

```
138
```

```
temp=np.sum((df['Contract duration']=='Temporary contract'))     #number of people with contract duration temporary
temp
```

```
6
```

```
df['Contract duration'].unique()
```

```
array(['Unlimited contract', 'Temporary contract'], dtype=object)
```

```
lost=np.sum((df['Have you lost your job due to the coronavirus outbreak?']=='Yes'))  #number of people that lost their jo
lost
```

```
6
```

✓ 0s completed at 12:50 AM