In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
D:\Jupyter Notebook\lib\site-packages\scipy\__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.23.1
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [2]:
```python
data=pd.read_csv(r"D:\PWDS-Pooja Sharma\All G-drive files\Day15\IBM Attrition Data.csv")
```

In [3]:
```python
data.head()
```

Out[3]:

	Age	Attrition	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	JobSatisfaction	MaritalStatus	MonthlyIncome	NumCompaniesWorked	WorkLifeBalance	YearsAtCompany
0	41	Yes	Sales	1	2	Life Sciences	2	4	Single	5993	8	1	6
1	49	No	Research & Development	8	1	Life Sciences	3	2	Married	5130	1	3	10
2	37	Yes	Research & Development	2	2	Other	4	3	Single	2090	6	3	0
3	33	No	Research & Development	3	4	Life Sciences	4	3	Married	2909	1	3	8
4	27	No	Research & Development	2	1	Medical	1	2	Married	3468	9	3	2

In [4]:
```python
data.shape
```

Out[4]:
```
(1470, 13)
```

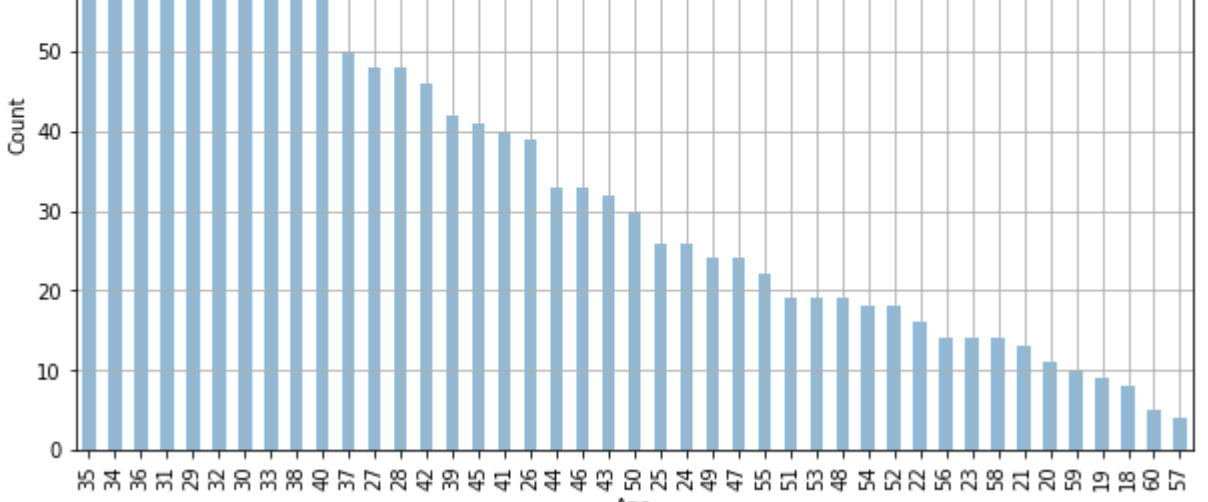
In [5]:
```python
data.info() #check for null values
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   Age                      1470 non-null   int64 
 1   Attrition                1470 non-null   object
 2   Department               1470 non-null   object
 3   DistanceFromHome         1470 non-null   int64 
 4   Education                1470 non-null   int64 
 5   EducationField           1470 non-null   object
 6   EnvironmentSatisfaction  1470 non-null   int64 
 7   JobSatisfaction          1470 non-null   int64 
 8   MaritalStatus            1470 non-null   object
 9   MonthlyIncome            1470 non-null   int64 
 10  NumCompaniesWorked       1470 non-null   int64 
 11  WorkLifeBalance          1470 non-null   int64 
 12  YearsAtCompany           1470 non-null   int64 
dtypes: int64(9), object(4)
memory usage: 149.4+ KB
```

In [6]:
```python
data.isnull().sum() #sum of all null values must be zero
```

Out[6]:
```
Age                        0
Attrition                  0
Department                 0
DistanceFromHome           0
Education                  0
EducationField             0
EnvironmentSatisfaction    0
JobSatisfaction            0
MaritalStatus              0
MonthlyIncome              0
NumCompaniesWorked         0
WorkLifeBalance            0
YearsAtCompany             0
dtype: int64
```

In [7]:
```python
data['Age'].value_counts().plot.bar(figsize=(10,6),xlabel="Age",ylabel="Count",title="Age distribution",alpha=0.5)
plt.grid()
plt.show()
```



In [8]:
```python
data['Attrition'].value_counts()
```

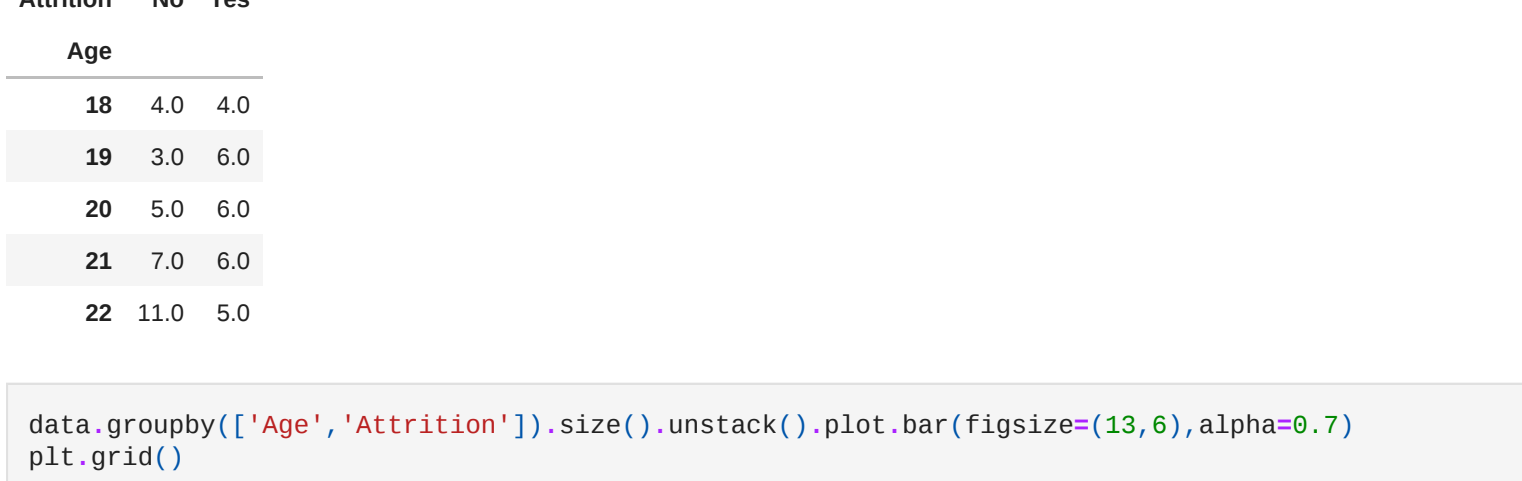
Out[8]:
```
No     1233
Yes     237
Name: Attrition, dtype: int64
```

In [9]:
```python
data.groupby(['Age','Attrition']).size().unstack().head()
```

Out[9]:

Attrition / Age	No	Yes
18	4.0	4.0
19	3.0	6.0
20	5.0	6.0
21	7.0	6.0
22	11.0	5.0

In [10]:
```python
data.groupby(['Age','Attrition']).size().unstack().plot.bar(figsize=(13,6),alpha=0.7)
plt.grid()
```



In [11]:
```python
x=data.groupby(['Age','Attrition']).size().unstack().reset_index()
for i in x['Age']:
    x['% Attrition']=((x['No']*100)/(x['No']+x['Yes']))
```

In [12]:
```python
x.head()
```

Out[12]:

Attrition	Age	No	Yes	% Attrition
0	18	4.0	4.0	50.000000
1	19	3.0	6.0	33.333333
2	20	5.0	6.0	45.454545
3	21	7.0	6.0	53.846154
4	22	11.0	5.0	68.750000

In [13]:
```python
print("Most % of employees attritioned belong to age :",x['Age'][x['% Attrition'].idxmax()]," Percantage of leaving :",x['% Attrition'].max())
print("Least % of employees attritioned belong to age :",x['Age'][x['% Attrition'].idxmin()]," Percantage of leaving :",x['% Attrition'].min())
```

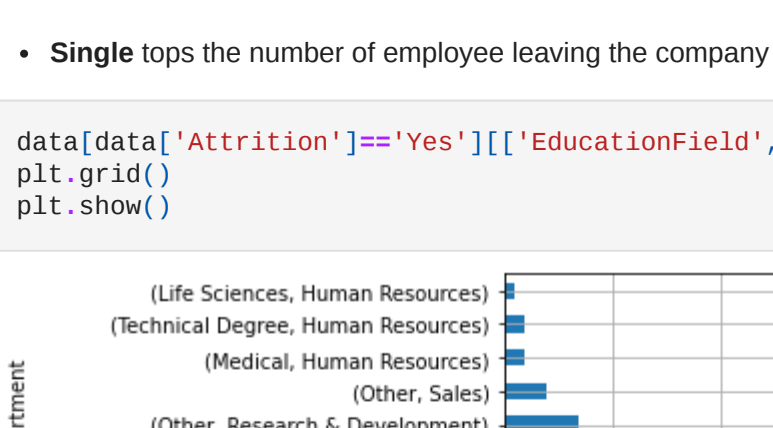
```
Most % of employees attritioned belong to age : 38  Percantage of leaving : 96.55172413793103
Least % of employees attritioned belong to age : 19  Percantage of leaving : 33.333333333333336
```

In [14]:
```python
Left=data[data['Attrition']=='Yes']
working=data[data['Attrition']=='No']
```

In [15]:
```python
data.head()
data.columns
```

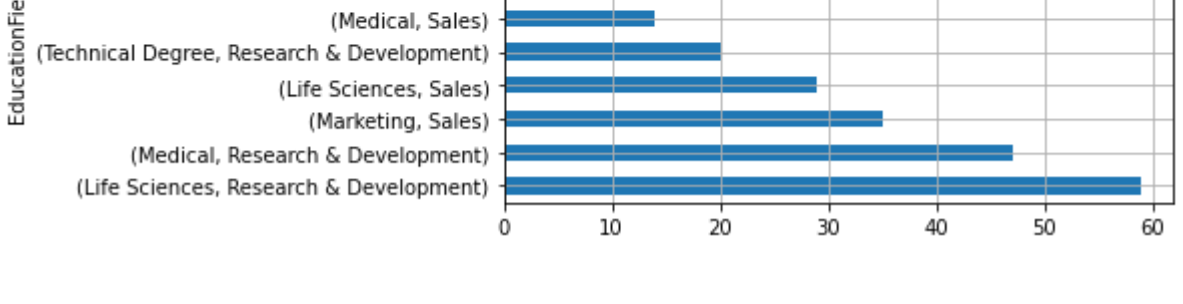
Out[15]:
```
Index(['Age', 'Attrition', 'Department', 'DistanceFromHome', 'Education',
       'EducationField', 'EnvironmentSatisfaction', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'NumCompaniesWorked',
       'WorkLifeBalance', 'YearsAtCompany'],
      dtype='object')
```

In [16]:
```python
data[data['Attrition']=='Yes']['MaritalStatus'].value_counts().plot.bar(xlabel="Marital Status",ylabel='Count',title="Attrition vs Marital Status")
plt.grid()
plt.show()
```



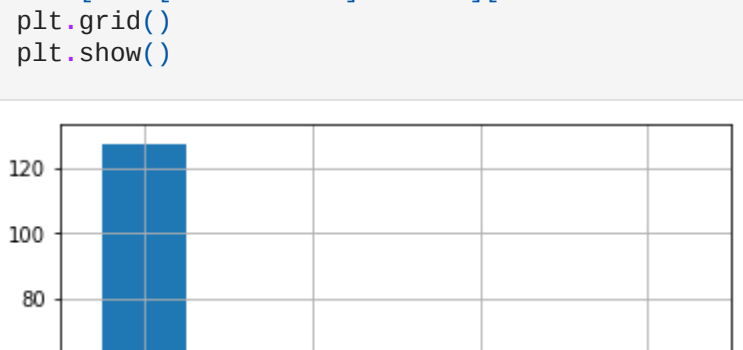
- **Single** tops the number of employee leaving the company

In [17]:
```python
data[data['Attrition']=='Yes'][['EducationField','Department']].value_counts().plot.barh()
plt.grid()
plt.show()
```



- Employees belonging to **Life sciences Education** field tops the attrition

In [18]:
```python
data[data['Attrition']=='Yes']['WorkLifeBalance'].value_counts().plot.bar()
plt.grid()
plt.show()
```



- Employees with **'Best'** and **'Worst'** worklife balance tends to leave more

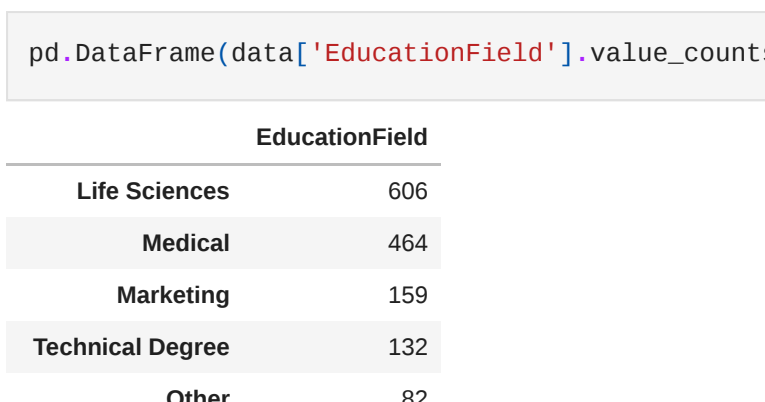
Distribution of employees by Education Field

In [19]:
```python
pd.DataFrame(data['EducationField'].value_counts())
```

Out[19]:

	EducationField
Life Sciences	606
Medical	464
Marketing	159
Technical Degree	132
Other	82
Human Resources	27

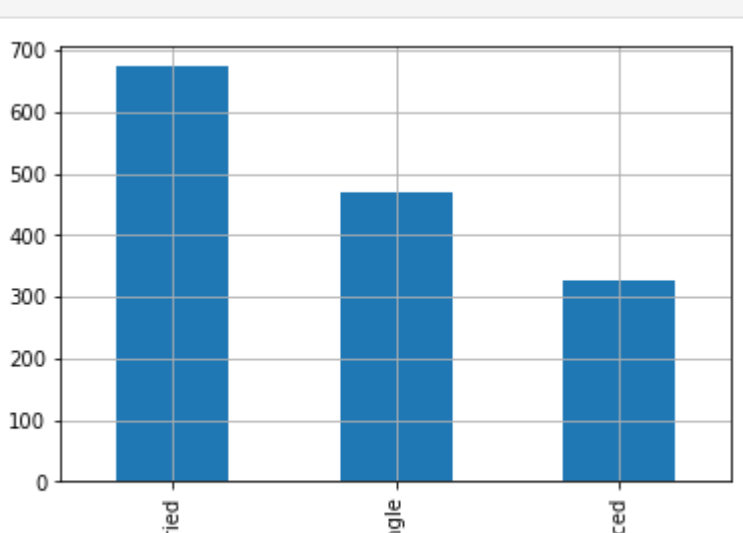
In [44]:
```python
data['EducationField'].value_counts().plot.bar()
plt.grid()
plt.show()
```



In [21]:
```python
print("Max Number of employees belong to :",data['EducationField'].value_counts().idxmax()," :",data['EducationField'].value_counts().max())
```

```
Max Number of employees belong to : Life Sciences  : 606
```

In [22]:
```python
data['MaritalStatus'].value_counts().plot.bar()
plt.grid()
plt.show()
```



- Most of the employees are : **Married**

Q. Build up a logistic regression model to predict which employees are likely to attrite.

In [23]:
```python
data.head()
```

Out[23]:

	Age	Attrition	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	JobSatisfaction	MaritalStatus	MonthlyIncome	NumCompaniesWorked	WorkLifeBalance	YearsAtCompany
0	41	Yes	Sales	1	2	Life Sciences	2	4	Single	5993	8	1	6
1	49	No	Research & Development	8	1	Life Sciences	3	2	Married	5130	1	3	10
2	37	Yes	Research & Development	2	2	Other	4	3	Single	2090	6	3	0
3	33	No	Research & Development	3	4	Life Sciences	4	3	Married	2909	1	3	8
4	27	No	Research & Development	2	1	Medical	1	2	Married	3468	9	3	2

In [24]:
```python
data.dtypes
```

Out[24]:
```
Age                         int64
Attrition                  object
Department                 object
DistanceFromHome            int64
Education                   int64
EducationField             object
EnvironmentSatisfaction     int64
JobSatisfaction             int64
MaritalStatus              object
MonthlyIncome               int64
NumCompaniesWorked          int64
WorkLifeBalance             int64
YearsAtCompany              int64
dtype: object
```

In [25]:
```python
x=data.drop(['Attrition'],axis=1)
x
```

Out[25]:

	Age	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	JobSatisfaction	MaritalStatus	MonthlyIncome	NumCompaniesWorked	WorkLifeBalance	YearsAtCompany
0	41	Sales	1	2	Life Sciences	2	4	Single	5993	8	1	6
1	49	Research & Development	8	1	Life Sciences	3	2	Married	5130	1	3	10
2	37	Research & Development	2	2	Other	4	3	Single	2090	6	3	0
3	33	Research & Development	3	4	Life Sciences	4	3	Married	2909	1	3	8
4	27	Research & Development	2	1	Medical	1	2	Married	3468	9	3	2
...	...	...	...	...	...	...	...	...	...	...	...	...
1465	36	Research & Development	23	2	Medical	3	4	Married	2571	4	3	5
1466	39	Research & Development	6	1	Medical	4	1	Married	9991	4	3	7
1467	27	Research & Development	4	3	Life Sciences	2	2	Married	6142	1	3	6
1468	49	Sales	2	3	Medical	4	2	Married	5390	2	2	9
1469	34	Research & Development	8	3	Medical	2	3	Married	4404	2	4	4

1470 rows × 12 columns

In [26]:
```python
x=pd.get_dummies(x)
print(x)
y=data['Attrition']
```

```
      Age  DistanceFromHome  Education  EnvironmentSatisfaction  \
0      41                 1          2                        2   
1      49                 8          1                        3   
2      37                 2          2                        4   
3      33                 3          4                        4   
4      27                 2          1                        1   
...   ...               ...        ...                      ...   
1465   36                23          2                        3   
1466   39                 6          1                        4   
1467   27                 4          3                        2   
1468   49                 2          3                        4   
1469   34                 8          3                        2   

      JobSatisfaction  MonthlyIncome  NumCompaniesWorked  WorkLifeBalance  \
0                   4           5993                   8                1   
1                   2           5130                   1                3   
2                   3           2090                   6                3   
3                   3           2909                   1                3   
4                   2           3468                   9                3   
...               ...            ...                 ...              ...   
1465                4           2571                   4                3   
1466                1           9991                   4                3   
1467                2           6142                   1                3   
1468                2           5390                   2                2   
1469                3           4404                   2                4   

      YearsAtCompany  Department_Human Resources  ...  Department_Sales  \
0                  6                           0  ...                 1   
1                 10                           0  ...                 0   
2                  0                           0  ...                 0   
3                  8                           0  ...                 0   
4                  2                           0  ...                 0   
...              ...                         ...  ...               ...   
1465               5                           0  ...                 0   
1466               7                           0  ...                 0   
1467               6                           0  ...                 0   
1468               9                           0  ...                 1   
1469               4                           0  ...                 0   

      EducationField_Human Resources  EducationField_Life Sciences  \
0                                  0                             1   
1                                  0                             1   
2                                  0                             0   
3                                  0                             1   
4                                  0                             0   
...                              ...                           ...   
1465                               0                             0   
1466                               0                             0   
1467                               0                             1   
1468                               0                             0   
1469                               0                             0   

      EducationField_Marketing  EducationField_Medical  EducationField_Other  \
0                            0                       0                     0   
1                            0                       0                     0   
2                            0                       0                     1   
3                            0                       0                     0   
4                            0                       1                     0   
...                        ...                     ...                   ...   
1465                         0                       1                     0   
1466                         0                       1                     0   
1467                         0                       0                     0   
1468                         0                       1                     0   
1469                         0                       1                     0   

      EducationField_Technical Degree  MaritalStatus_Divorced  \
0                                   0                       0   
1                                   0                       0   
2                                   0                       0   
3                                   0                       0   
4                                   0                       0   
...                               ...                     ...   
1465                                0                       0   
1466                                0                       0   
1467                                0                       0   
1468                                0                       0   
1469                                0                       0   

      MaritalStatus_Married  MaritalStatus_Single  
0                         0                     1  
1                         1                     0  
2                         0                     1  
3                         1                     0  
4                         1                     0  
...                     ...                   ...  
1465                      1                     0  
1466                      1                     0  
1467                      1                     0  
1468                      1                     0  
1469                      1                     0  

[1470 rows x 21 columns]
```

In [27]:
```python
from sklearn.model_selection import train_test_split
```

In [28]:
```python
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=21)
```

In [29]:
```python
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(1176, 21)
(294, 21)
(1176,)
(294,)
```

In [30]:
```python
from sklearn.preprocessing import StandardScaler
```

In [31]:
```python
sc=StandardScaler()
```

In [32]:
```python
x_train=sc.fit_transform(x_train)
x_test=sc.transform(x_test)
x_train[:5]
```

Out[32]:
```
array([[-0.63234372, -0.14449779,  1.05683329,  0.23737148,  1.14333888,
        -0.44458181, -0.68144876,  0.34846951,  0.65560534, -0.22150245,
         0.73434802, -0.66022529, -0.14123725,  1.20415946, -0.34641016,
        -0.67887375, -0.23990406, -0.32444284, -0.54459355, -0.92618568,
         1.49357599],
       [-0.63234372, -0.26808282, -0.87004415,  0.23737148,  0.24094524,
         0.50684829, -0.68144876,  0.34846951,  0.49267383, -0.22150245,
         0.73434802, -0.66022529, -0.14123725,  1.20415946, -0.34641016,
        -0.67887375, -0.23990406, -0.32444284, -0.54459355,  1.07969711,
        -0.66953406],
       [-0.8538618 ,  0.47342732,  0.09339457, -1.59311285, -1.56384204,
        -0.88439979,  0.52062686,  0.34846951, -0.8107782 ,  4.51462271,
        -1.36175216, -0.66022529,  7.08028494, -0.8304548 , -0.34641016,
        -0.67887375, -0.23990406, -0.32444284,  1.8362318 , -0.92618568,
        -0.66953406],
       [ 0.91828288, -0.39166784,  0.09339457,  1.15261365,  1.14333888,
         0.49042617,  1.3220106 ,  0.34846951,  1.95905737, -0.22150245,
        -1.36175216,  1.51463449, -0.14123725, -0.8304548 , -0.34641016,
         1.47302794, -0.23990406, -0.32444284, -0.54459355, -0.92618568,
         1.49357599],
       [-0.74310276, -0.26808282,  1.05683329,  0.23737148,  0.24094524,
        -0.6410156 , -0.68144876, -1.0597784 ,  0.49267383, -0.22150245,
         0.73434802, -0.66022529, -0.14123725,  1.20415946, -0.34641016,
        -0.67887375, -0.23990406, -0.32444284,  1.8362318 , -0.92618568,
        -0.66953406]])
```

In [33]:
```python
from sklearn.linear_model import LogisticRegression
```

In [34]:
```python
classifier = LogisticRegression()
classifier.fit(x_train,y_train)
```

Out[34]:
```
LogisticRegression()
```

In [35]:
```python
y_pred=classifier.predict(x_test)
```

In [36]:
```python
y_pred
```

Out[36]:
```
array(['No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No'], dtype=object)
```

In [37]:
```python
from sklearn.metrics import confusion_matrix
```

In [38]:
```python
cm=confusion_matrix(y_test,y_pred)
```

In [39]:
```python
cm
```

Out[39]:
```
array([[245,   2],
       [ 40,   7]], dtype=int64)
```

In [40]:
```python
from sklearn.metrics import accuracy_score
print("Accuracy :",accuracy_score(y_test,y_pred))
```

```
Accuracy : 0.8571428571428571
```

In [41]:
```python
x_test[0:1]
```

Out[41]:
```
array([[ 2.57966852,  2.32720265,  0.09339457,  0.23737148, -1.56384204,
         0.78539266,  0.52062686,  1.75671742,  1.79612586, -0.22150245,
        -1.36175216,  1.51463449, -0.14123725, -0.8304548 ,  2.88675135,
        -0.67887375, -0.23990406, -0.32444284, -0.54459355,  1.07969711,
        -0.66953406]])
```

In [42]:
```python
predicted_by_lr=classifier.predict(x_test[0:1])
```

In [43]:
```python
predicted_by_lr
```

Out[43]:
```
array(['No'], dtype=object)
```