In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
df = pd.DataFrame()
```

In [3]:
```
size = 100
```

In [4]:
```
df['Company'] = np.random.choice(['TCS', 'HCL', 'Infosys', 'Tech Mahindra'], size)
df['Masters'] = np.random.randint(0, 2, size)
```

In [5]:
```
grp = df.groupby('Company')
grp.first()
```

Out[5]:

Company	Masters
HCL	1
Infosys	0
TCS	1
Tech Mahindra	1

logistic regression

In [6]:
```
#since ml doesn't work on object datatype hence we need to convert it into a dummy variables
company = pd.get_dummies(df['Company'], drop_first=True)
```

In [7]:
```
from sklearn.preprocessing import LabelEncoder
```

In [8]:
```
le_company = LabelEncoder()
le_company
```

Out[8]:
```
LabelEncoder()
```

In [9]:
```
df['company_n'] = le_company.fit_transform(df['Company'])
newdf = df.drop('Company', axis=1)
newdf
```

Out[9]:

	Masters	company_n
0	1	3
1	0	3
2	1	0
3	1	0
4	1	2
...	...	...
95	0	0
96	1	2
97	1	3
98	1	3
99	0	3

100 rows × 2 columns

In [10]:
```
import matplotlib.pyplot as plt
```

In [11]:
```
plt.scatter(newdf.compan)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
C:\Users\PRATIK~1\AppData\Local\Temp/ipykernel_16884/3222628025.py in <module>
----> 1 plt.scatter(newdf.compan)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\generic.py in __getattr__(self, name)
   5485         ):
   5486             return self[name]
-> 5487         return object.__getattribute__(self, name)
   5488
   5489     def __setattr__(self, name: str, value) -> None:

AttributeError: 'DataFrame' object has no attribute 'compan'
```

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

```
reg = LogisticRegression()
reg
```

In [ ]:

In [ ]:

predicting whether the bank will give loan to its customers based on their credit score

In [29]:
```
import pandas as pd
from sklearn.linear_model import LogisticRegression
import numpy as np
import matplotlib.pyplot as plt
```

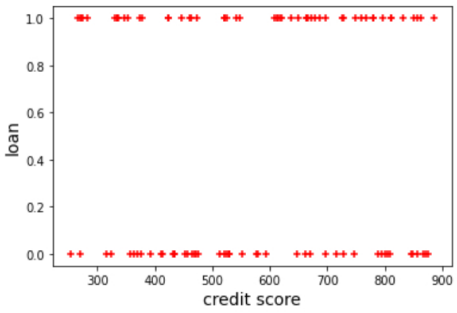
In [30]:
```
#creating a dataframe
df = pd.DataFrame()

size = 100

df['credit_score'] = np.random.randint(250, 900, size)
df['loan'] = np.random.randint(0, 2, size)

df.head()
```

Out[30]:

	credit_score	loan
0	618	1
1	856	1
2	663	1
3	849	0
4	412	0

In [31]:
```
plt.scatter(df.credit_score, df.loan, marker='+', color='red')
plt.xlabel('credit score', fontsize=14)
plt.ylabel('loan', fontsize=14)
```

Out[31]:
```
Text(0, 0.5, 'loan')
```



In [32]:
```
#creating logistic model
reg = LogisticRegression()
reg
```

Out[32]:
```
LogisticRegression()
```

In [33]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(df[['credit_score']], df[['loan']], train_size=0.9)
```

In [34]:
```
reg.fit(x_train, y_train)
```

```
C:\Users\PRATIKBAWANE\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\utils\validation.py:1111: 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().
  y = column_or_1d(y, warn=True)
```

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

In [35]:
```
x_test
```

Out[35]:

	credit_score
65	335
31	392
46	698
61	424
34	868

76	846
1	856
75	728
77	374
91	768

In [36]:
```
reg.predict([[450]])
```

```
C:\Users\PRATIKBAWANE\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:450: UserWarning: X does not ha
ve valid feature names, but LogisticRegression was fitted with feature names
  warnings.warn(
```

Out[36]:
```
array([1])
```

In [37]:
```
predict = reg.predict(x_test)
predict
```

Out[37]:
```
array([0, 0, 1, 1, 1, 1, 1, 1, 0, 1])
```

In [38]:
```
plt.figure()
plt.scatter(x_test, predict, marker='+', color='red')
plt.xlabel('credit score', size=14)
plt.ylabel('loan', size=14)
```

Out[38]:
```
Text(0, 0.5, 'loan')
```