problem definition

Is the person suffering from heart disease or not

target definition

after we check the age, sex, blood pressure, and other attributes we get the result if that person suffering from heart disease or not

In [10]:

```
from sklearn.metrics import confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
import pandas as pd
import numpy as np
import seaborn as sn
```

```
Matplotlib is building the font cache; this may take a moment.
```

In [3]:

```
# import data_set
data = pd.read_csv("D:/project/heart.csv")
data.dropna(inplace=True)  # for removing null values

# splitting to independent and target values
x = data.iloc[:, :-1].values
y = data.iloc[:, -1].values
```

In [4]:

```
data.describe()
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	
count	301.000000	301.000000	301.000000	301.000000	301.000000	301.000000	301.000000	30
mean	54.382060	0.684385	0.960133	131.584718	246.448505	0.146179	0.531561	14
std	9.065882	0.465534	1.028788	17.579258	51.939980	0.353874	0.525833	2
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	7
25%	48.000000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	13
50%	55.000000	1.000000	1.000000	130.000000	241.000000	0.000000	1.000000	15
75%	61.000000	1.000000	2.000000	140.000000	275.000000	0.000000	1.000000	16
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	20

In [5]:

```
data.info
```

Out[5]:

```
<bound method DataFrame.info of      age  sex  cp  trestbps  chol  fbs  rest
ecg  thalach  exang  oldpeak  \
1     37    1   2       130   250    0        1      187      0      3.5
3     56    1   1       120   236    0        1      178      0      0.8
4     57    0   0       120   354    0        1      163      1      0.6
5     57    1   0       140   192    0        1      148      0      0.4
6     56    0   1       140   294    0        0      153      0      1.3
..   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
298   57    0   0       140   241    0        1      123      1      0.2
299   45    1   3       110   264    0        1      132      0      1.2
300   68    1   0       144   193    1        1      141      0      3.4
301   57    1   0       130   131    0        1      115      1      1.2
302   57    0   1       130   236    0        0      174      0      0.0

     slope  ca  thal  target
1        0   0     2     1.0
3        2   0     2     1.0
4        2   0     2     1.0
5        1   0     1     1.0
6        1   0     2     1.0
..     ...  ..   ...     ...
298      1   0     3     0.0
299      1   0     3     0.0
300      1   2     3     0.0
301      1   1     3     0.0
302      1   1     2     0.0

[301 rows x 14 columns]>
```

In [8]:

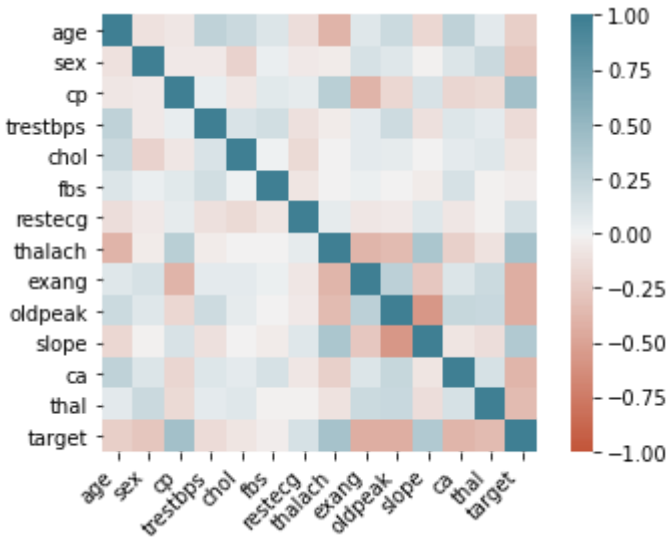
```
corr = data.corr()
corr
```

Out[8]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach
age	1.000000	-0.108759	-0.075558	0.278192	0.211840	0.113449	-0.118960	-0.396454
sex	-0.108759	1.000000	-0.054199	-0.059244	-0.202427	0.038182	-0.061299	-0.039543
cp	-0.075558	-0.054199	1.000000	0.042948	-0.075706	0.080153	0.051628	0.297933
trestbps	0.278192	-0.059244	0.042948	1.000000	0.123850	0.173220	-0.112348	-0.046564
chol	0.211840	-0.202427	-0.075706	0.123850	1.000000	0.014375	-0.155337	-0.007301
fbs	0.113449	0.038182	0.080153	0.173220	0.014375	1.000000	-0.078617	-0.007448
restecg	-0.118960	-0.061299	0.051628	-0.112348	-0.155337	-0.078617	1.000000	0.047677
thalach	-0.396454	-0.039543	0.297933	-0.046564	-0.007301	-0.007448	0.047677	1.000000
exang	0.096238	0.140677	-0.392810	0.069349	0.064722	0.030589	-0.075766	-0.377721
oldpeak	0.209627	0.095786	-0.157728	0.191153	0.055913	-0.002477	-0.054396	-0.346544
slope	-0.159348	-0.021184	0.136720	-0.116732	-0.003384	-0.041368	0.089921	0.388323
ca	0.276967	0.117098	-0.177743	0.103258	0.068162	0.144331	-0.077208	-0.211514
thal	0.073209	0.215034	-0.149808	0.068110	0.096471	-0.015973	-0.021118	-0.095573
target	-0.225627	-0.280538	0.431692	-0.147525	-0.082308	-0.034485	0.144227	0.420558

In [12]:

```
ax = sn.heatmap(
    corr,
    vmin=-1, vmax=1, center=0,
    cmap=sn.diverging_palette(20, 220, n=200),
    square=True
)
ax.set_xticklabels(
    ax.get_xticklabels(),
    rotation=45,
    horizontalalignment='right'
);
```

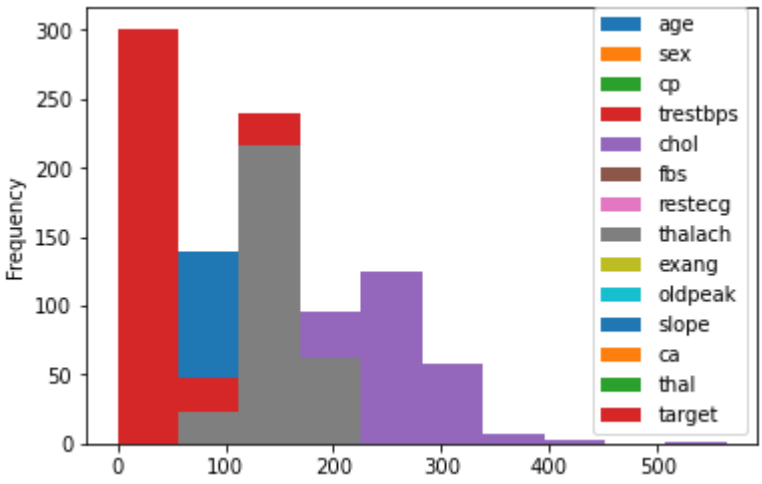


In [13]:

```
data.plot.hist()
```

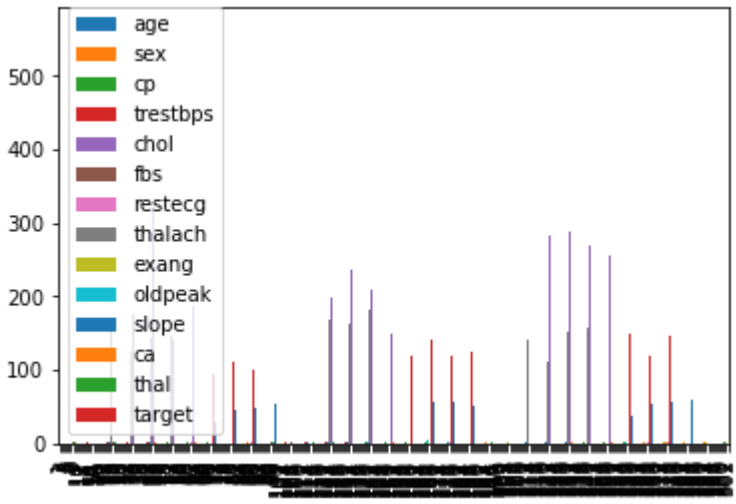
Out[13]:

```
<AxesSubplot:ylabel='Frequency'>
```



In [14]:

```
data.plot.bar();
```



In [17]:

```
# split to training and test set
x_train, x_test, y_train, y_test = train_test_split(
    x, y, test_size=0.25, random_state=0)
# print(x_train, y_train)
```

In [18]:

```
st = StandardScaler()
x_train = st.fit_transform(x_train)
x_test = st.transform(x_test)

# learning (fitting the model)
classifier = DecisionTreeClassifier(criterion='gini', random_state=10)
classifier.fit(x_train, y_train)

# fitting the Naive Bayes model
NBclassifier = GaussianNB()
NBclassifier.fit(x_train, y_train)

# fitting the knn model
KNNclassifier = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2)
KNNclassifier.fit(x_train, y_train)

# prediction on test set
y_pred = classifier.predict(x_test)
# print(y_pred)
```

In [19]:

```
# measuring the accuracy of the model
cm = confusion_matrix(y_test, y_pred)
# print(cm)
```

In [*]:

```
# Take input from user
age = float(input("Enter age: "))
sex = float(input("Enter sex: "))
cp = float(input("Enter cp: "))
trestbps = float(input("Enter trestbps: "))
chol = float(input("Enter chol: "))
fbs = float(input("Enter fbs: "))
restecg = float(input("Enter restecg: "))
thalach = float(input("Enter thalach: "))
exang = float(input("Enter exang: "))
oldpeak = float(input("Enter oldpeak: "))
slope = float(input("Enter slope: "))
ca = float(input("Enter ca: "))
thal = float(input("Enter thal: "))


# input must be 2D array
result = classifier.predict(
    [[age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal]]
print('target ==' + result)
```

```
Enter age: 25

Enter sex:
```

In [ ]:

```
```