In [1]:
```
import numpy as np
from scipy.io import loadmat
from matplotlib import pyplot as plt
```

In [7]:
```
#load the dataset
data=loadmat("ex8data1.mat")
X=data['X']
Xval=data['Xval']
Yval=data['yval']
```

In [8]:
```
X.shape
```

Out[8]: (307, 2)

In [9]:
```
Xval.shape
```

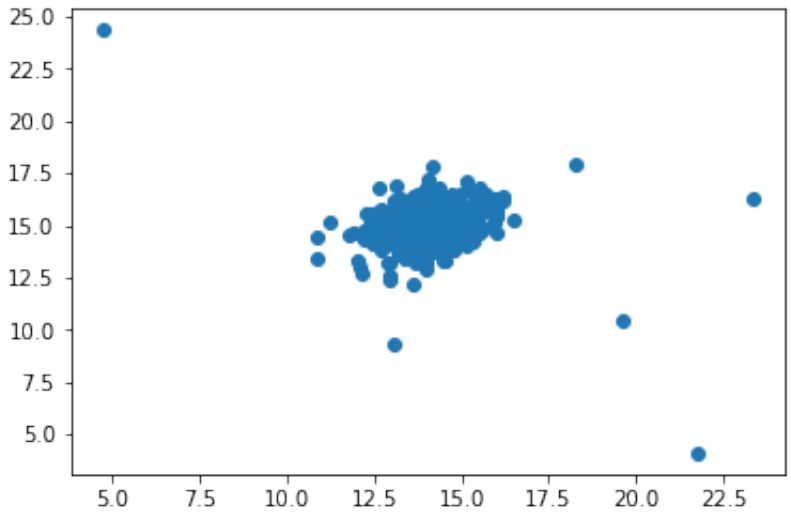
Out[9]: (307, 2)

In [11]:
```
Yval.shape
```

Out[11]: (307, 1)

In [13]:
```
#plot scatter plot
plt.scatter(X[:,0],X[:,1])
```

Out[13]: <matplotlib.collections.PathCollection at 0x10f362cc0>



In [15]:
```
#Gausian distribution -Mean and variance
mu=X.mean(axis=0)
sigma=X.var(axis=0)
mu,sigma
```

Out[15]: (array([14.11222578, 14.99771051]), array([1.83263141, 1.70974533]))

In [16]:
```
X
```

Out[16]:
```
array([[13.04681517, 14.74115241],
       [13.40852019, 13.7632696 ],
       [14.19591481, 15.85318113],
       [14.91470077, 16.17425987],
       [13.57669961, 14.04284944],
       [13.92240251, 13.40646894],
       [12.82213164, 14.22318782],
       [15.67636615, 15.89169137],
       [16.16287532, 16.20299807],
       [12.66645095, 14.89908374],
       [13.98454962, 12.95800822],
       [14.06146043, 14.54908874],
       [13.38988671, 15.56202142],
       [13.39350475, 15.62698794],
       [13.97900926, 13.28061494],
       [14.16791259, 14.46583829],
       [13.96176145, 14.75182421],
       [14.45899735, 15.07018563],
       [14.58476372, 15.82743424],
```

In [20]:
```
from scipy import stats
p=np.zeros((X.shape[0],X.shape[1])) #all 307 values initized to ze
p[:,0]=stats.norm(mu[0],sigma[0]).pdf(X[:,0])
p[:,1]=stats.norm(mu[1],sigma[1]).pdf(X[:,1])

```

Out[20]:
```
(307, 2)
```

In [25]:
```
#let us find for xval
pval=np.zeros((Xval.shape[0],Xval.shape[1]))
pval[:,0]=stats.norm(mu[0],sigma[0]).pdf(Xval[:,0])
pval[:,1]=stats.norm(mu[1],sigma[1]).pdf(Xval[:,1])
pval.shape
```

Out[25]:
```
(307, 2)
```

In [30]:
```
#find the threshold
def select_threshold(pval,yval):
    best_epsilon=0
    best_f1=0
    f1=0
    step=(pval.max()-pval.min())/1000
    for epsilon in np.arange(pval.min(),pval.max(),step):
        preds=pval<epsilon
        tp=np.sum(np.logical_and(preds==1,yval==1)).astype(float)
        fp=np.sum(np.logical_and(preds==1,yval==0)).astype(float)
        fn=np.sum(np.logical_and(preds==0,yval==1)).astype(float)

        precision=tp/(tp+fp)
        recall=tp/(tp+fn)
        f1=(2*precision*recall)/(precision+recall)

        if f1>best_f1:
            best_f1=f1
            best_epsilon=epsilon

    return best_epsilon,best_f1

epsilon,f1=select_threshold(pval,Yval)

epsilon,f1


```

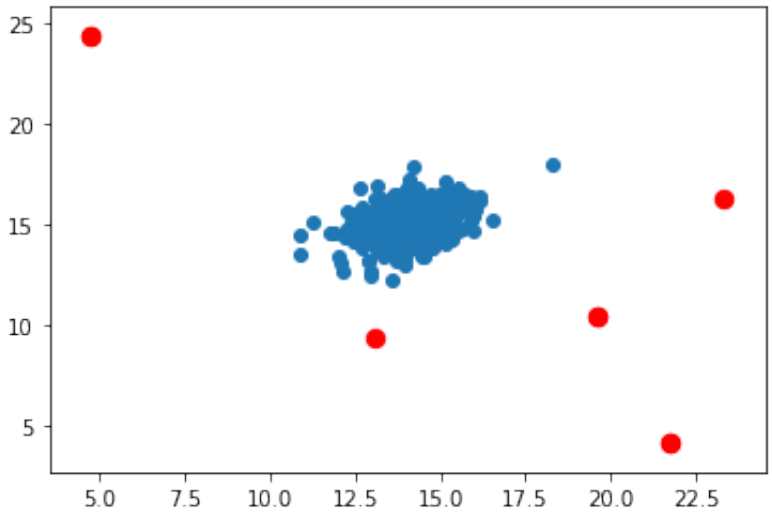
```
/Users/apple/anaconda3/lib/python3.6/site-packages/ipykernel_launche
r.py:13: RuntimeWarning: invalid value encountered in double_scalars
  del sys.path[0]
```

Out[30]: (0.009566706005956842, 0.7142857142857143)

In [38]:
```
#find outlier
outliers=np.where(p<epsilon)
plt.scatter(X[:,0],X[:,1])
plt.scatter(X[outliers[0],0],X[outliers[0],1],color='r',s=70)

```

Out[38]: <matplotlib.collections.PathCollection at 0x116c10358>



End of program 4

In [ ]:
```
1
```