```
In [164]: import pandas as pd
          import numpy as np
          from sklearn.linear_model import Ridge
          from sklearn.utils import resample
```

```
In [165]: df = np.loadtxt('hw2_lssvm_all.dat')
```

```
In [166]: dftrain = df[0:400]
          dftest = df[400:]
```

```
In [167]: Lambda = [0.05, 0.5, 5, 50, 500]
```

```
In [ ]:
```

```
In [168]: Ein = []
          Eout = []
          for l in Lambda:
              clf = Ridge(alpha=l)
              clf.fit(dftrain[:,0:-1], dftrain[:,-1])
              y_pred_in = (clf.predict(dftrain[:,0:-1])>0)*2-1
              y_pred_out = (clf.predict(dftest[:,0:-1])>0)*2-1
              Ein = Ein + [sum(y_pred_in != dftrain[:,-1])/len(y_pred_in)]
              Eout = Eout + [sum(y_pred_out != dftest[:,-1])/len(y_pred_out)]
```

```
In [169]: Ein

Out[169]: [0.3175, 0.3175, 0.3175, 0.32, 0.3225]
```

```
In [170]: Eout

Out[170]: [0.36, 0.36, 0.36, 0.37, 0.37]
```

With λ increases, Ein and Eout increase.

```
In [171]: Ein = []
          Eout = []
          for l in Lambda:
              y_bag_out = np.zeros(100)
              y_bag_in = np.zeros(400)
              for t in range(250):
                  df_bs = resample(dftrain,n_samples=400,random_state=t)
                  clf = Ridge(alpha=l)
                  clf.fit(df_bs[:,0:-1], df_bs[:,-1])
                  y_bag_out = y_bag_out + (( clf.predict(dftest[:,0:-1]) > 0 )*2-1)
                  y_bag_in = y_bag_in +   (( clf.predict(dftrain[:,0:-1]) > 0 )*2-1)



              Ein = Ein + [sum( ((y_bag_in>0)*2-1) != dftrain[:,-1] ) / len(y_bag_in)]
              Eout = Eout + [sum( ((y_bag_out>0)*2-1) != dftest[:,-1] )/ len(y_bag_out)]
```

```
In [172]: Ein

Out[172]: [0.3175, 0.3175, 0.3175, 0.3175, 0.3225]
```

```
In [173]: Eout
```

```
Out[173]: [0.36, 0.36, 0.36, 0.36, 0.37]
```

With λ increases, Ein and Eout increase. And there is no improvement after using bootstrpping.

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```