

# LOOCV

April 16, 2023

```
[ ]: from numpy import *
      from matplotlib.pyplot import *
      from sklearn.linear_model import LinearRegression
      L = LinearRegression()

[ ]: def do(n=1):
      #Creating data
      X = random.randn(100,5)
      w = linspace(0.2,0.6,5)
      Y = sum(w*X,axis=-1) + random.randn(100)
      O = []
      while len(O) < n:
          r = random.randint(100)
          if r not in O:
              O.append(r)
      for o in O:
          Y[o] = sum(w * X[o]) + 10*random.randn()
      print("Rank, Index, Error")
      E0 = []
      for o in O :
          L.fit(X[r_[0:o,o+1:100]],Y[r_[0:o,o+1:100]])
          E0.extend(abs(L.predict([X[o]]) - Y[o]))
      E0 = array(E0)
      O = array(O)
      s = E0.argsort()
      E0 = E0[s]
      O = O[s]
      print("\n Actual Outliners :")
      for i in range(n-1,-1,-1):
          print(n-i,O[i],E0[i])
      E = zeros(n)
      ind = [0]*n
      for i in range(100):
          L.fit(X[r_[0:i,i+1:100]],Y[r_[0:i,i+1:100]])
          e = abs(L.predict([X[i]]) - Y[i])
          for j in range(n-1,-1,-1):
              if e > E[j] :
```

```

        for k in range(j):
            E[k] = E[k+1]
            ind[k] = ind[k+1]
        E[j] = e
        ind[j] = i
        break
print("\n Predicted :")
for i in range(n-1,-1,-1):
    print(n-i,ind[i],E[i])
print('\n',100*sum(ind==0)/n,"% were predicted correctly")

```

[ ]: do()

Rank, Index, Error

Actual Outliners :  
1 13 7.08595946391043

Predicted :  
1 13 7.08595946391043

100.0 % were predicted correctly

[ ]: do(5)

Rank, Index, Error

Actual Outliners :  
1 3 11.076869512925828  
2 31 9.756089175431805  
3 47 3.176989169162955  
4 91 2.4397070171851567  
5 11 1.6308684362312615

Predicted :  
1 3 11.076869512925828  
2 31 9.756089175431805  
3 47 3.176989169162955  
4 4 2.8991201924666097  
5 48 2.7462432621864727

60.0 % were predicted correctly