姓名:王國倫　　學號:0860077　　Homework 2

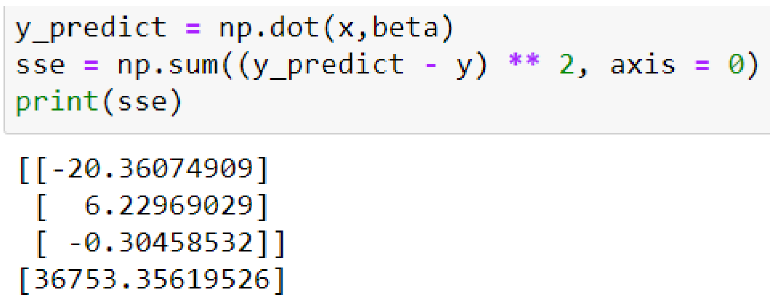
Problem 1:

(a) k=20962, sse=93413.0895. This algorithm is converged.

(b) k=∞, $sse = \infty$. This algorithm is diverged.

(c) According to above results, there are different learning rate, the k and sse are also different. The possible reason is learning rate, which decides parameters update speed, the overhigh value easy cause drastically oscillation, lead to the parameter can't converge. In the contrary, the overlow value converge slowly, and require plenty of time to train. As a result, the properly chose learning rate is key factor.

(d) The sse in problem2 in HW1 is 36753.3562 shown as below picture, and the sse in problem1 in HW2 is 93413.0895. The result shows that gradient descent can continue training, decreasing the sse in HW2, but the data lexpend provided from MEAP93.csv is too smooth, cause gradient is so small, the parameter renew slowly, let the error converge, and stop to train.



```
y_predict = np.dot(x,beta)
sse = np.sum((y_predict - y) ** 2, axis = 0)
print(sse)
```

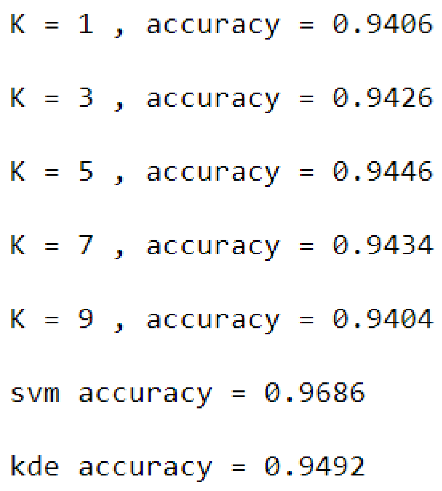
```
[[-20.36074909]
 [  6.22969029]
 [ -0.30458532]]
[36753.35619526]
```

Problem 2:

(a)(b)(c) the detail code in HW2_0860077_王國倫(2).ipynb

(d) The k nearest neighbor, support vector machine, and kernel density estimation accuracy are show as below picture. In order to decrease time complexity, the above algorithm adopt 10000 training data to train, and 5000 testing data to predict. The KNN time complexity is O(ndk), where n is the number of testing data, d is number of training data, k is total pixel of image. There has 28*28 pixel in mnist image. The SVM training time complexity is O(n^3), where n is the number of training data. and testing time complexity is O($n_{sv}d$), where n_{sv} is the number of support vectors, d is the number of testing data. The KDE training time complexity is O(t), where t is the number of types, and testing time complexity is O(td), where t is the number of types, d is the number of testing data.

The k nearest neighbor requires to calculate every image distance between testing and training data, and find the nearest points as predict result. So, this algorithm doesn't need training in advance. The propose of support vector machine algorithm is to find hyperplane to separate different types, to be more specific, to find the appropriate decision boundary, and maximum their margins. The kernel density estimation is to estimate probability density function with finite training data, and classify testing data based on every PDF, in other words, to find the most likely probability to be sample from every digit.



```
K = 1 , accuracy = 0.9406

K = 3 , accuracy = 0.9426

K = 5 , accuracy = 0.9446

K = 7 , accuracy = 0.9434

K = 9 , accuracy = 0.9404

svm accuracy = 0.9686

kde accuracy = 0.9492
```