Basma El Shenawy

900150283

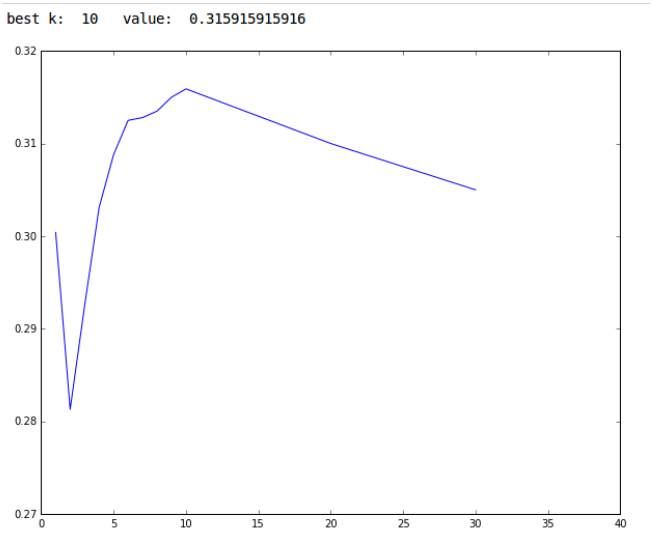
Assignment 1 report

March 3rd, 2018

Part 1

The aim of my training and validation was to find the K that produces the best accuracy and standard deviation. I started by getting the accuracies of a wide set of data to find where this best K might fall. See the results below.

```
for k= 1
Got 968 / 3330 correct => accuracy: 0.290691
Got 1002 / 3330 correct => accuracy: 0.300901
Got 1031 / 3330 correct => accuracy: 0.309610
average accuracy:  0.3004004004
for k= 2
Got 908 / 3330 correct => accuracy: 0.272673
Got 938 / 3330 correct => accuracy: 0.281682
Got 964 / 3330 correct => accuracy: 0.289489
average accuracy:  0.281281281281
for k= 3
Got 928 / 3330 correct => accuracy: 0.278679
Got 987 / 3330 correct => accuracy: 0.296396
Got 1007 / 3330 correct => accuracy: 0.302402
average accuracy:  0.292492492492
for k= 4
Got 962 / 3330 correct => accuracy: 0.288889
Got 1026 / 3330 correct => accuracy: 0.308108
Got 1040 / 3330 correct => accuracy: 0.312312
average accuracy:  0.303103103103
for k= 5
Got 975 / 3330 correct => accuracy: 0.292793
Got 1046 / 3330 correct => accuracy: 0.314114
Got 1064 / 3330 correct => accuracy: 0.319520
average accuracy:  0.308808808809
for k= 6
Got 982 / 3330 correct => accuracy: 0.294895
Got 1055 / 3330 correct => accuracy: 0.316817
Got 1085 / 3330 correct => accuracy: 0.325826
average accuracy:  0.312512512513
for k= 7
Got 989 / 3330 correct => accuracy: 0.296997
Got 1058 / 3330 correct => accuracy: 0.317718
Got 1078 / 3330 correct => accuracy: 0.323724
average accuracy:  0.312812812813
for k= 8
Got 989 / 3330 correct => accuracy: 0.296997
Got 1059 / 3330 correct => accuracy: 0.318018
Got 1084 / 3330 correct => accuracy: 0.325526
average accuracy:  0.313513513514
for k= 9
Got 999 / 3330 correct => accuracy: 0.300000
Got 1065 / 3330 correct => accuracy: 0.319820
Got 1083 / 3330 correct => accuracy: 0.325225
average accuracy:  0.315015015015
for k= 10
Got 1007 / 3330 correct => accuracy: 0.302402
Got 1066 / 3330 correct => accuracy: 0.320120
Got 1083 / 3330 correct => accuracy: 0.325225
average accuracy:  0.315915915916
for k= 20
Got 1005 / 3330 correct => accuracy: 0.301802
Got 1035 / 3330 correct => accuracy: 0.310811
Got 1057 / 3330 correct => accuracy: 0.317417
average accuracy:  0.31001001001
for k= 30
Got 993 / 3330 correct => accuracy: 0.298198
Got 1013 / 3330 correct => accuracy: 0.304204
Got 1041 / 3330 correct => accuracy: 0.312613
average accuracy:  0.305005005005
```



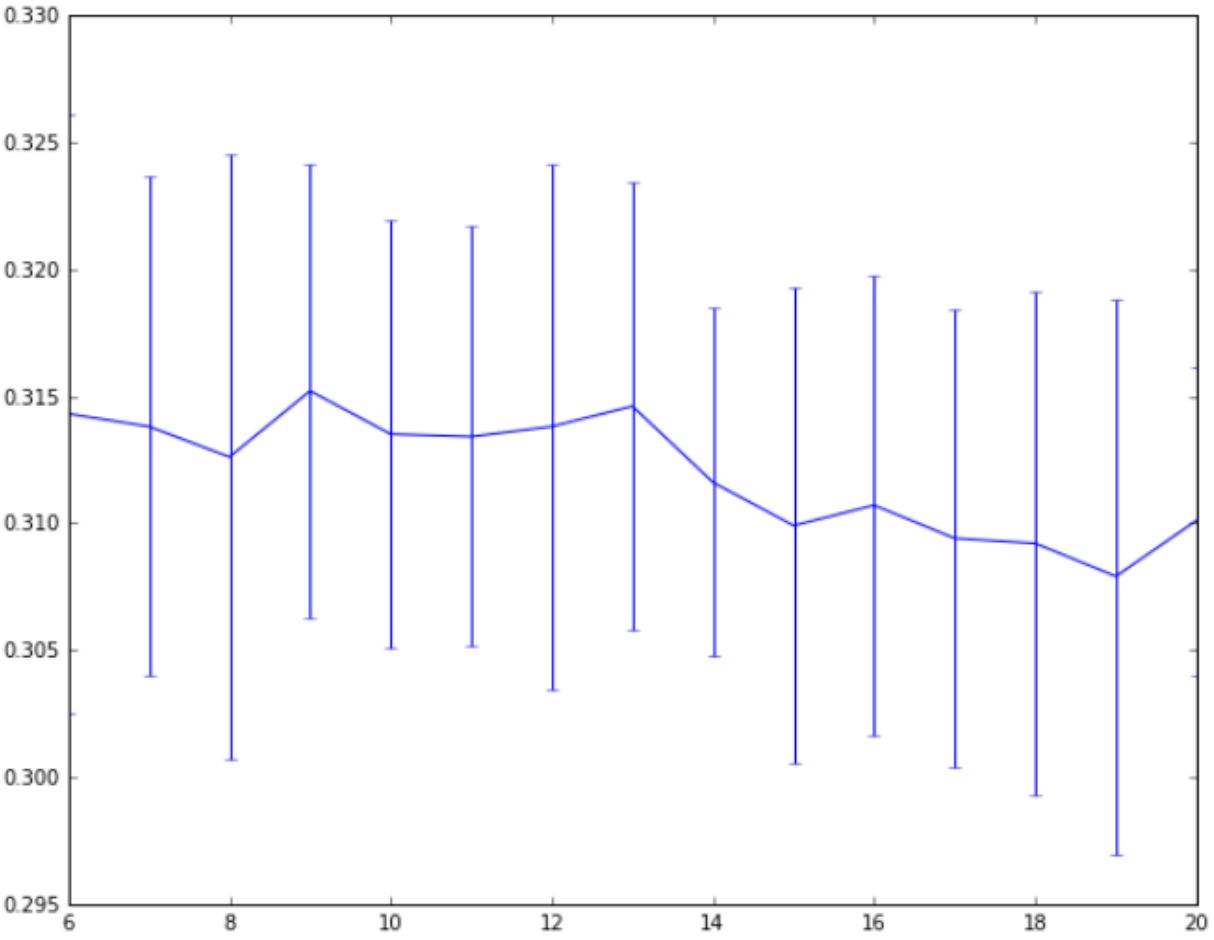
I then tried testing the bigger numbers just to ensure that I got the best K. The average accuracy is less than that of the smaller values as we see in the following results.

```
for k= 50
Got 955 / 3330 correct => accuracy: 0.286787
Got 1005 / 3330 correct => accuracy: 0.301802
Got 1041 / 3330 correct => accuracy: 0.312613
average accuracy:  0.3004004004
for k= 100
Got 928 / 3330 correct => accuracy: 0.278679
Got 962 / 3330 correct => accuracy: 0.288889
Got 979 / 3330 correct => accuracy: 0.293994
average accuracy:  0.287187187187
for k= 150
Got 906 / 3330 correct => accuracy: 0.272072
Got 931 / 3330 correct => accuracy: 0.279580
Got 958 / 3330 correct => accuracy: 0.287688
average accuracy:  0.27977977978
for k= 200
Got 884 / 3330 correct => accuracy: 0.265465
Got 914 / 3330 correct => accuracy: 0.274474
Got 916 / 3330 correct => accuracy: 0.275075
average accuracy:  0.271671671672
```

As we can see from the previous data, the maximums occur somewhere between k=6 and k=20. Therefore, I repeated the results whose average accuracy is between 0.31 and 0.32 (k=6 till k=20) but calculated the standard deviation as well in order to find the best K. see result below.

```
for k= 6
Got 993 / 3330 correct => accuracy: 0.298198
Got 1061 / 3330 correct => accuracy: 0.318619
Got 1086 / 3330 correct => accuracy: 0.326126
average accuracy:  0.314314314314
for k= 7
Got 1000 / 3330 correct => accuracy: 0.300300
Got 1058 / 3330 correct => accuracy: 0.317718
Got 1077 / 3330 correct => accuracy: 0.323423
average accuracy:  0.313813813814
for k= 8
Got 987 / 3330 correct => accuracy: 0.296396
Got 1055 / 3330 correct => accuracy: 0.316817
Got 1081 / 3330 correct => accuracy: 0.324625
average accuracy:  0.312612612613
for k= 9
Got 1008 / 3330 correct => accuracy: 0.302703
Got 1066 / 3330 correct => accuracy: 0.320120
Got 1075 / 3330 correct => accuracy: 0.322823
average accuracy:  0.315215215215
for k= 10
Got 1005 / 3330 correct => accuracy: 0.301802
Got 1057 / 3330 correct => accuracy: 0.317417
Got 1070 / 3330 correct => accuracy: 0.321321
average accuracy:  0.313513513514
for k= 11
Got 1005 / 3330 correct => accuracy: 0.301802
Got 1059 / 3330 correct => accuracy: 0.318018
Got 1067 / 3330 correct => accuracy: 0.320420
average accuracy:  0.313413413413
for k= 12
Got 997 / 3330 correct => accuracy: 0.299399
Got 1062 / 3330 correct => accuracy: 0.318919
Got 1076 / 3330 correct => accuracy: 0.323123
average accuracy:  0.313813813814
for k= 13
Got 1012 / 3330 correct => accuracy: 0.303904
Got 1047 / 3330 correct => accuracy: 0.314414
Got 1084 / 3330 correct => accuracy: 0.325526
average accuracy:  0.314614614615
```

```
for k= 14
Got 1006 / 3330 correct => accuracy: 0.302102
Got 1048 / 3330 correct => accuracy: 0.314715
Got 1059 / 3330 correct => accuracy: 0.318018
average accuracy:  0.311611611612
for k= 15
Got 988 / 3330 correct => accuracy: 0.296697
Got 1055 / 3330 correct => accuracy: 0.316817
Got 1053 / 3330 correct => accuracy: 0.316216
average accuracy:  0.309909909991
for k= 16
Got 992 / 3330 correct => accuracy: 0.297898
Got 1057 / 3330 correct => accuracy: 0.317417
Got 1055 / 3330 correct => accuracy: 0.316817
average accuracy:  0.310710710711
for k= 17
Got 989 / 3330 correct => accuracy: 0.296997
Got 1042 / 3330 correct => accuracy: 0.312913
Got 1060 / 3330 correct => accuracy: 0.318318
average accuracy:  0.309409409409
for k= 18
Got 984 / 3330 correct => accuracy: 0.295495
Got 1044 / 3330 correct => accuracy: 0.313514
Got 1061 / 3330 correct => accuracy: 0.318619
average accuracy:  0.309209209209
for k= 19
Got 976 / 3330 correct => accuracy: 0.293093
Got 1037 / 3330 correct => accuracy: 0.311411
Got 1063 / 3330 correct => accuracy: 0.319219
average accuracy:  0.307907907908
for k= 20
Got 1006 / 3330 correct => accuracy: 0.302102
Got 1037 / 3330 correct => accuracy: 0.311411
Got 1055 / 3330 correct => accuracy: 0.316817
average accuracy:  0.31011011011
```



As we can see from the data, the best K is =9 with validation accuracy of 0.315215215215.. So, I used the entire 10000 dataset as training data and tested using the testing batch. I got the overall accuracy and the accuracy of each class See the results below.

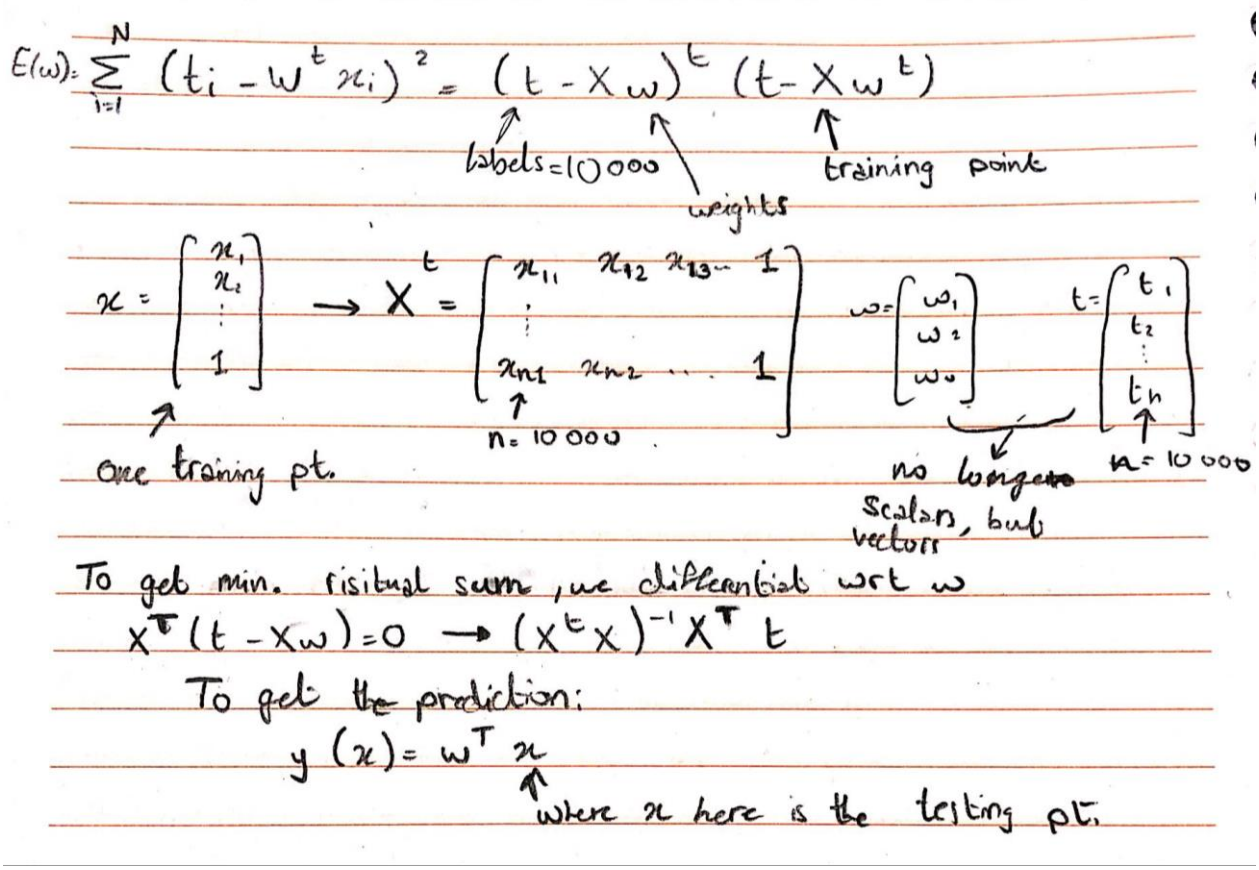
```
Got 3241 / 9990 correct => accuracy: 0.324424
```

The average accuracy of each of the 10 classes:

```
CCRn is:  0.542000  for class: plane
CCRn is:  0.166000  for class: car
CCRn is:  0.443000  for class: bird
CCRn is:  0.144000  for class: cat
CCRn is:  0.422000  for class: deer
CCRn is:  0.191000  for class: dog
CCRn is:  0.271000  for class: frog
CCRn is:  0.197000  for class: horse
CCRn is:  0.657000  for class: ship
CCRn is:  0.208000  for class: truck
```

Part 2

How do I get the w?



Accuracy of the training:

```
0.50944
```

Accuracy of testing:

```
ACCR for testing data is : 0.363700
```

The average accuracy of each of the 10 classes:

```
CCRn is:  0.469000  for class: plane
CCRn is:  0.445000  for class: car
CCRn is:  0.207000  for class: bird
CCRn is:  0.177000  for class: cat
CCRn is:  0.243000  for class: deer
CCRn is:  0.285000  for class: dog
CCRn is:  0.449000  for class: frog
CCRn is:  0.426000  for class: horse
CCRn is:  0.508000  for class: ship
CCRn is:  0.428000  for class: truck
```

I believe that no overfitting occurred because the difference between the accuracy of the testing and the accuracy of the training is very small. (about 1.5)