COMP 551 – Assignment 2 – Report

Ali Shobeiri - 26066559

All text files submissions can be found in the txt Files folder.

1. Please see: *DS1.csv, DS1_train.csv, DS1_valid.csv, DS1_test.csv*

2. a. Please see: *Assignment2_260665549_2_1a.txt, values:*
   Precision: 0.9443037974683545
   Recall: 0.9325
   Accuracy: 0.93875
   F1 Measure: 0.938364779874214

   b. Please see: *Assignment2_260665549_2_1b.txt, values:*

```
w:  [ 1.40542533e+01 -8.26578614e+00 -5.06632505e+00 -2.69201272e+00
 -9.37280780e+00 -4.38523686e+00  1.58689599e+01 -2.34300606e+01
 -2.78745134e+01  9.09608016e+00 -1.28541098e+01 -1.16922993e+01
  1.48472460e+01  1.22183007e+01 -5.71140687e+00  1.27236593e+01
  2.80432792e+01 -6.58936552e+00  9.58045183e-03 -4.93335203e+00]
wo:  26.41842219565859
```

3. a. The KNN classifier performed worse than the GDA classifier. As we increase our value for K, the performance seems to increase. This happens as the higher we increase our value for K, the less one data point can affect our classifier. In other words, we reduce potential noise and achieve a better F1 measure by increasing our sampling of neighbors. By increasing our neighbors, we are increasing the likelihood that a point can encounter more points of the same class and get correctly classified.
   We find our best number of neighbors with the best F1 performance to be, 199 which was one of the highest K value we tested. See: *Assignment2_260665549_3.txt, values:*

   k: 1
   Precision: 0.507537688442211
   Recall: 0.505
   Accuracy: 0.5075
   F1 Measure: 0.5062656641604011

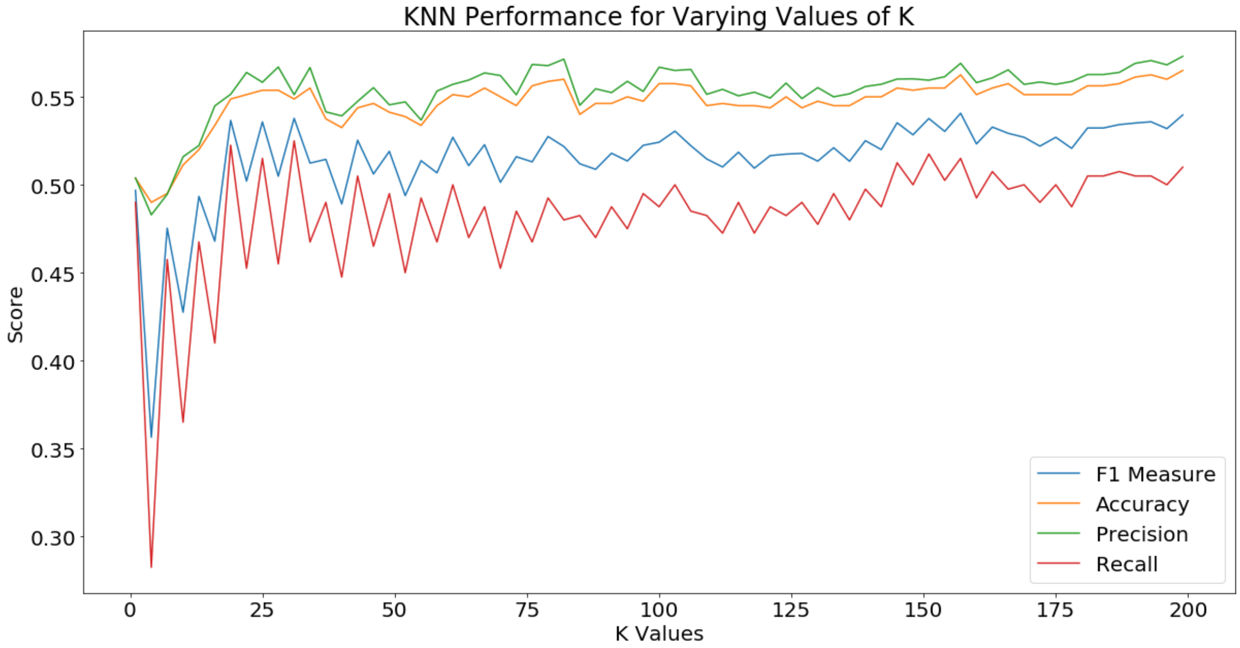
   k: 61
   Precision: 0.5751295336787565
   Recall: 0.555
   Accuracy: 0.5725
   F1 Measure: 0.5648854961832063

   k: 121
   Precision: 0.5743073047858942
   Recall: 0.57
   Accuracy: 0.57375
   F1 Measure: 0.5721455457967376

   k: 181
   Precision: 0.5934343434343434
   Recall: 0.5875
   Accuracy: 0.5925
   F1 Measure: 0.5904522613065327

   Best k we found, with best F1:
   k: 199
   Best Validation F1: 0.5982478097622027

A plot of several accuracy metrics with varying Ks is shown below:



b. Please see *Assignment2_260665549_3_b.txt,* values:

Precision: 0.5550351288056206
Recall: 0.5925
Accuracy: 0.55875
F1 Measure: 0.5731559854897219

4. Please see *DS2.csv, DS2_train.csv, DS2_valid.csv, DS2_test.csv*

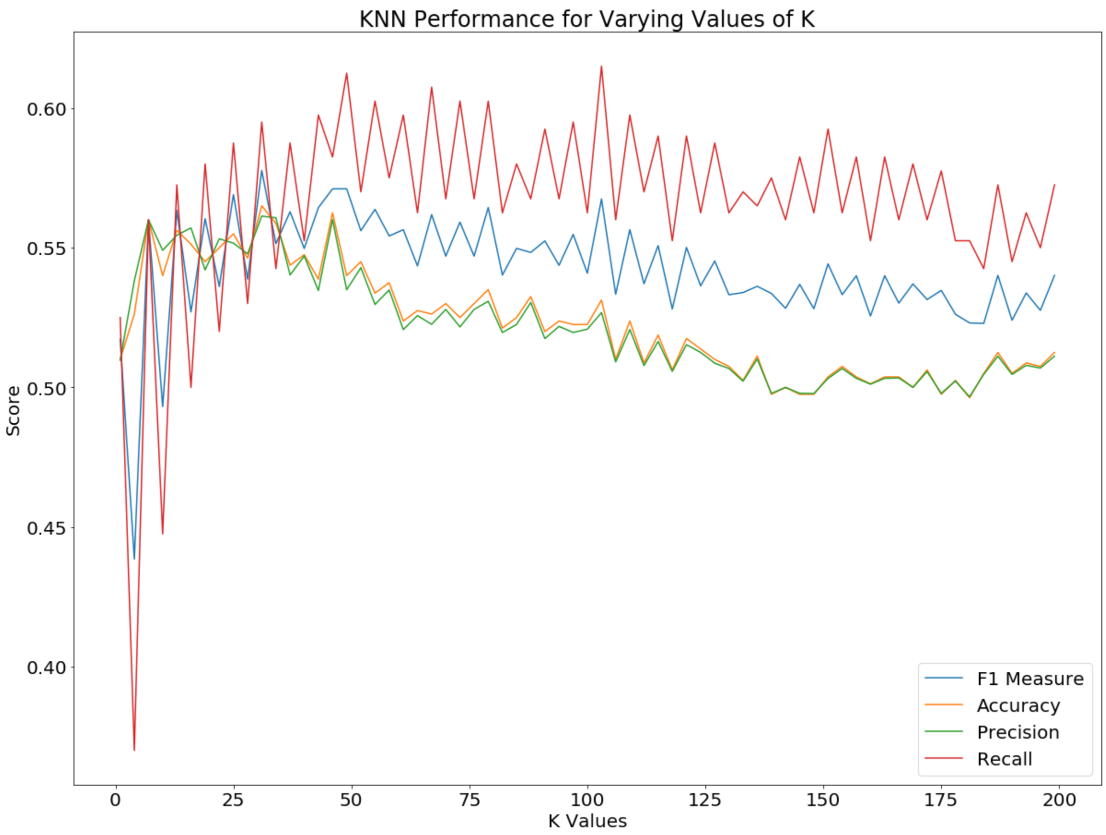
5. 1. a. Please see *Assignment2_260665549_5_1_a.txt*, values:

Precision: 0.5275779376498801
Recall: 0.55
Accuracy: 0.52875
F1 Measure: 0.5385556915544676

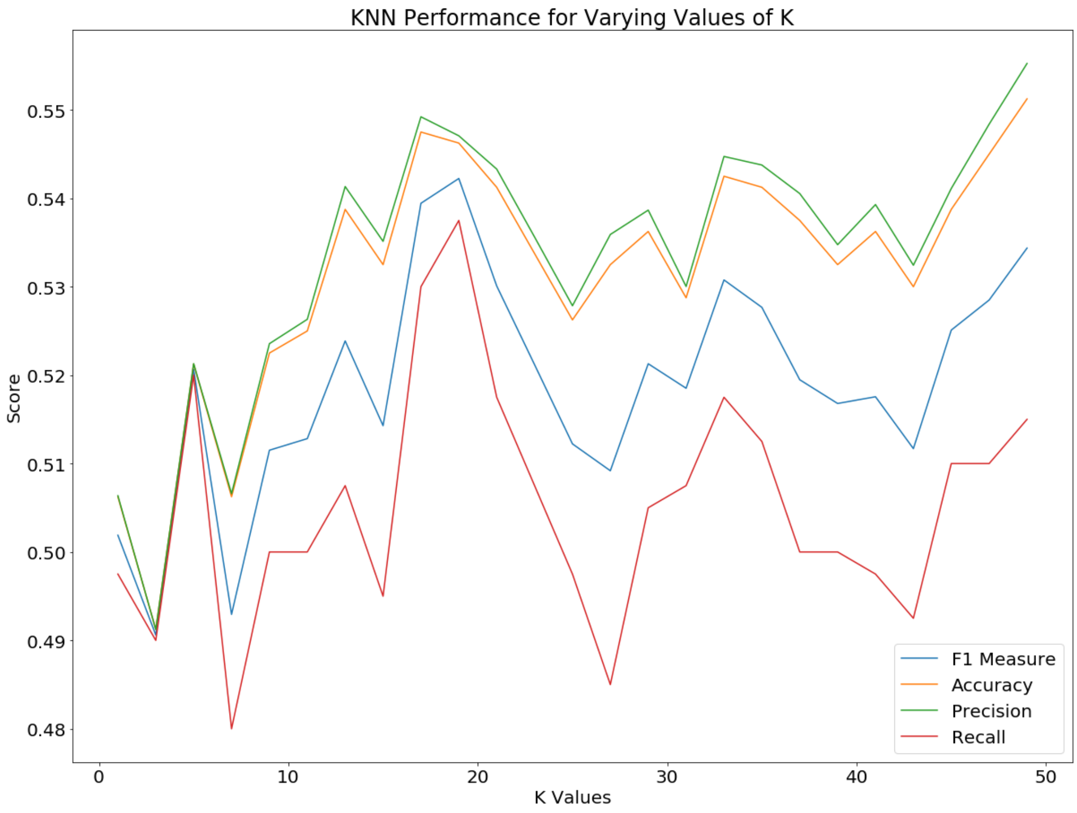
b. Please see *Assignment2_260665549_5_1_b.txt*, values:

w: [-0.05263836 0.02674428 -0.01940971 -0.03594093 0.06783143 -0.07399906 0.00366749 0.05266897 0.06628943 0.03327621 0.08143538 0.03357787 0.09192193 -0.01308588 -0.13638505 0.03466672 -0.0093443 -0.01785583 -0.12961503 0.08009116]
wo: -0.09040049709877707

2. The kNN performs better than the GDA in this case. The best performing value of k in our classification was measured using our validation set to be 19. A plot of performance vs different values of K is seen below:



From k = 1 to k = 200



From k = 1 to k = 50

This could be because as we increase our K in this case, we are not simply reducing our noise but due to the way we generated the data (different means), potentially introducing more points from another class. In this case, it is better to poll a smaller group of neighbors as they are more likely to be sharing the same mean and class value as the current point being considered and the larger the number of K, the more likely this assumption is to breaks down.

3. See *Assignment2_260665549_5_3.txt, values:*
Precision: 0.5439024390243903
Recall: 0.5575
Accuracy: 0.545
F1 Measure: 0.5506172839506173

6. On DS1, the GDA classifier performed better however, on DS2 KNN was able to outperform the GDA classifier. This is because in DS1, we generated our data in a way that was favorable to the assumptions made by GDA, meaning that it was able to classify the data very accurately. However, in DS2 as the assumptions made for the data no longer held, GDA's performance began to decrease. In both cases, KNN seemed to perform slightly better than a coin toss. This indicates that the data we generated (either DS1 or DS2) is not favorable for a KNN classifier. This could be because of the high dimensionality of the data which KNN is known to struggle with.