Machine Learning: Assigment 1

Anjali Prasad collaborated with Arpita Agrawal

September 22, 2016

1 Question 1

(a) (10 points) Suppose we have N i.i.d samples x1, x2, $\cdots$, xn. We will practice the maximum likelihood estimation techniques to estimate the parameters in each of the following cases:

Ans:

$$x^{\alpha-1}(1-x)^{\beta-1} \div \frac{\Gamma(\alpha).\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

We know that $\Gamma(\alpha) = (\alpha-1)!$ *and* $\beta = 1$ *so* $\Gamma(\beta) = 0! = 1$ *and* $\Gamma(\alpha+\beta) = (\alpha+1-1)! = \alpha!$ substituting the above equations in the main one:

$$f = x^{\alpha-1}(1-x)^{\beta-1} \div \frac{(\alpha-1)!}{\alpha!} = x^{\alpha-1} * \alpha$$

Taking log likelihood on both sides and then differentiating and equating to 0.

$$logf = log(x^{\alpha-1} * \alpha)$$

$$= (\alpha-1)logx + log\alpha = \alpha logx - logx + log\alpha$$

$$\frac{\partial f}{\partial \alpha} = logx - 0 + \frac{1}{\alpha} = 0$$

$$\frac{1}{\alpha} = -logx => \alpha = -\frac{1}{logx}$$

$$Ans : \alpha = -\frac{1}{logx}$$

Question 1.a • We assume that all samples are generated from Normal distribution N(,). Please show how to derive the maximum likelihood estimator of .

$$f(x) = N(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} exp^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

$$f(x) = N(\theta, \theta) = \frac{1}{\sqrt{2\pi\theta}} exp^{\frac{-(x-\theta)^2}{2\theta}}$$

Taking log on both sides and differentiating and equating to 0

$$logf(x) = (-log\sqrt{2\pi\theta}) - \frac{1}{2\theta}(x-\theta)^2(1)$$

$$= -log(2\pi\theta)^{\frac{1}{2}} - \frac{1}{2\theta}(x-\theta)^2$$

$$= -\frac{1}{2}log(2\pi\theta) - \frac{1}{2\theta}(x^2 + \theta^2 - 2x\theta)$$

$$= -\frac{1}{2}log(2\pi\theta) - \frac{x^2}{2\theta} - \frac{\theta}{2} + x$$

Differentiating wrt θ *and equating to* 0

$$\frac{\partial f}{\partial \theta} = -\frac{1}{2\theta} + \frac{x^2}{\theta^2} - \frac{1}{2} + 0 = 0$$

$$-\frac{1}{2\theta} + \frac{x^2}{\theta^2} = \frac{1}{2}$$

$$\theta^2 + \theta - 2x^2 = 0$$

Using Formula

$$\theta = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$\theta = \frac{1 \pm \sqrt{1 + 8x^2}}{2}$$

Question 1 .b (b) (8 points) Suppose a training set with N examples (x1, y1), (x2, y2), · · · , (xN , yN) is given, where xi = (xi1, · · · , xiD) is a D-dimensional feature vector, and yi 0, 1 is its corre- sponding label. Using the assumptions in 3.1 (not the result), provide the maximum likelihood estimation for the parameters of the Naive Bayes with Gaussian assumption. In other words, you need to provide the estimates for , jk, and j, for j = 1,...,D and k 0,1.

Ans: We are given that kernel density estimation is in the form of :

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n}K(\frac{x - X_i}{h})$$

As we know

$$E(x) = \int_{-\infty}^{\infty} xf(x)dx$$

Therefore replacing X by t and using E(x) we get

$$E[\hat{f}(x)] = \frac{1}{N} * N\int_{-\infty}^{\infty}\frac{1}{h}K(\frac{x-t}{h})f(t)dt$$

$$E[\hat{f}(x)] = \frac{1}{h}\int_{-\infty}^{\infty}K(\frac{x-t}{h})f(t)dt$$

Lets substitute

$$z = \frac{x-t}{h}$$

So

$$t = x - zh$$

and

$$dz = \frac{dt}{h}$$

Substituting all these in $\hat{f}(x)$
$E[\hat{f}(x)] = \int_{-\infty}^{\infty} K(z)f(x - zh)dz$
$TaylorExpansion$:

$$f(x-zh) = f(x)+f^{(1)}(x)(-zh)+\frac{1}{2}f^{(2)}(x)(-zh)^2+...+\frac{1}{n!}f^{n}(x)(-uh)^n+O(h^{n+1})$$

Using taylor expansion on E$[\hat{f}(x)]weget$
$\int_{-\infty}^{\infty} K(z)f(x-zh)dz = f(x)+f^{(1)}(x)(-h)\int_{-\infty}^{\infty} K(z)zdz+\frac{1}{2}f^{(2)}(x)(-h)^2\int_{-\infty}^{\infty} K(z)z^2dz+$
$O(h^3)$

we know the property $\int uK(u)du = 0$
$therefore f^{(1)}(x)(-h)\int_{-\infty}^{\infty} K(z)zdz = 0$
$E[\hat{f}(x)] = f(x) + \frac{1}{2}f^{(2)}(x)h^2\int_{-\infty}^{\infty} K(z)z^2dz + O(h^3)$

As we know bias is expected value - actual value:

$$bias = E[\hat{f}(x)] - f(x)$$

$$E[\hat{f}(x)] - f(x) = (1 \div 2) * f^{(2)}(x)h^2 \int_{-\infty}^{\infty} K(z)z^2 dz + O(h^3)$$

2 Question 2

2.a: (15 points) The binary Naive Bayes classifier has interesting connections to the logistic regression algorithm. In this exercise, you will derive the parametric form of the Naive Bayes classifier under certain assumptions and show that the likelihood function implied by the Gaussian Naive Bayes classifier for two classes is identical in form to the likelihood function for logistic regression. In the second part, you will derive the parameter estimation for the Naive Bayes algorithm.

Ans:

$$P(Y = 1|X) = \frac{P(Y = 1)P(X|Y = 1)}{P(Y = 1)P(X|Y = 1) + P(X|Y = 0)P(Y = 0)}$$

Dividing both sides by

$$P(Y = 1)P(X|Y = 1)$$

$$\frac{1}{1 + \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$

$$\frac{1}{1 + exp(ln\frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)})}$$

we know that

$$\pi = P(Y = 1)$$

Therefore

$$\frac{1}{1 + exp(ln\frac{P(Y=0)}{P(Y=1)}) * \sum_{i=1}^{n} ln\frac{P(X_i|Y=0)}{P(X_i|Y=1)})}$$

$$\frac{1}{1 + exp(ln\frac{1-\pi}{\pi}) * \sum_{i=1}^{n} ln\frac{P(X_i|Y=0)}{P(X_i|Y=1)})}$$

Therefore

$$w_1 = ln\frac{1-\pi}{\pi} - (1)$$

Lets solve the next part of the equation separately

$$\sum_{i=1}^{n} ln\frac{P(X_i|Y = 0)}{P(X_i|Y = 1)}$$

Substituting the following in above equation and simplifying

$$P(X_i|Y=k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{-\frac{(x-\mu_{jk})^2}{2\sigma_{ik}^2}}$$

We get

$$\sum_{i=1}^{n} ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)} = \sum_{i=1}^{n} ln \frac{e^{-\frac{(x_i-\mu_{jo})^2}{2\sigma_j^2}}}{e^{-\frac{(x_i-\mu_{j1})^2}{2\sigma_j^2}}}$$

Using log A/B = log A - log B

$$= \sum_{i=1}^{n} \frac{\mu_{j0}^2 - \mu_{j1}^2}{2\sigma_j^2} + \frac{\mu_{j0} - \mu_{j1}}{\sigma_j^2} X$$

therefore

$$w^T = \sum_{i=1}^{n} \frac{\mu_{j0} - \mu_{j1}}{\sigma_j^2}$$

$$w_2 = \sum_{i=1}^{n} \frac{\mu_{j0}^2 - \mu_{j1}^2}{2\sigma_j^2}$$

$$-w_0 = w_1 + w_2$$

$$-w_0 = ln \frac{1-\pi}{\pi} + \sum_{i=1}^{n} \frac{\mu_{j0}^2 - \mu_{j1}^2}{2\sigma_i^2}$$

2.b: (8 points) Suppose a training set with N examples (x1, y1), (x2, y2), · · · , (xN , yN) is given, where xi = (xi1, · · · , xiD) is a D-dimensional feature vector, and yi 0, 1 is its corre- sponding label. Using the assumptions in 3.1 (not the result), provide the maximum likelihood estimation for the parameters of the Naive Bayes with Gaussian assumption. In other words, you need to provide the estimates for , jk, and j, for j = 1,...,D and k 0,1.

Bayes rule:

$$P(Y=y_k|X_i) = \frac{P(Y=y_k)P(X_i|Y=y_k)}{\sum_{k=0}^{1} P(Y=y_k)P(X_i|Y=y_k)}$$

Assuming conditional independence among Xi

$$\sum_{i=1}^{n} P(Y=y_k|X_i) = \frac{P(Y=y_k)\prod_{i=1}^{n} P(X_i|y=y_k)}{\sum_{k=0}^{1} P(Y=y_k)\prod_{i=1}^{n} P(X_i|Y=y_k)}$$

As the denominator will be same for all yk we can ignore it in our computations.Taking log likelihood on both sides

$$logP(Y = y_k|X_i) = logP(Y = y_k) + logP(X_i|y = y_k)$$

Let

$$P(Y = y_k) = \pi^{N_k}(1-\pi)^{(N-N_k)}$$

Where N is number of points with Probability pi And Nk is the number of points with probability 1-pi Where k is the set of values that can be taken by y.
As the data follows gaussian distribution, we have

$$P(X_i|y = y_k) = \frac{1}{\sqrt{2\pi}\sigma_j} exp^{\frac{-(x_{ij}-\mu_{jk})^2}{2\sigma_j^2}}$$

Substituting in log likelihood equation:

$$logP(Y = y_k|X_i) = \sum_{k=0}^{1} log\pi^{N_k}(1-\pi)^{(N-N_k)} + \sum_{i=1}^{n} log(\frac{1}{\sqrt{2\pi}\sigma_j} exp^{\frac{-(x_{ij}-\mu_{jk})^2}{2\sigma_j^2}})$$

$$logP(Y = y_k|X_i) = \sum_{k=0}^{1} N_k log\pi + (N-N_k)log(1-\pi) + \sum_{i=1}^{n} -log\sqrt{2\pi}\sigma_j - \frac{-(x_{ij}-\mu_{jk})^2}{2\sigma_j^2})$$

Partially differentiating Log likelihood wrt σ_j

$$\frac{\partial L}{\partial \sigma_j} = \sum_{i=1}^{n} -\frac{\sqrt{2\pi}}{\sqrt{2\pi}\sigma_j} + \frac{1}{2}\frac{2}{\sigma_j^3}(x_i - \mu_{jk})^2 = 0$$

$$\frac{\partial L}{\partial \sigma_j} = \sum_{i=1}^{n}\sum_{k=0}^{1} -\frac{1}{\sigma_j} + \frac{1}{\sigma_j^3}(x_i - \mu_{jk})^2 = 0$$

$$\frac{1}{\sigma_j^3}(x_{ij} - \mu_{jk})^2 = \sum_{i=1}^{n}\sum_{k=0}^{1}\frac{1}{\sigma_{jk}}$$

$$\frac{1}{N}\sum_{i=1}^{n}\sum_{k=0}^{1}(x_{ij} - \mu_{jk})^2 = \sigma_{jk}^2$$

Partially differentiating Log likelihood wrt μ_{jk}

$$\frac{\partial L}{\partial \mu_{jk}} = \sum_{i=1}^{n} \sum_{k=0}^{1} -\frac{1}{2\sigma_{jk}^2}(2x_{ij} - 2\mu_{jk}) = 0$$

$$\mu_{jk} = \frac{1}{N_k} \sum_{i=1k=0/1}^{n} x_i$$

differentiating wrt pi

$$\frac{\partial f}{\partial \pi} = N_k log\pi + Nlog(1-\pi) - N_k log(1-\pi) = 0$$

$$\frac{N_x}{\pi} - \frac{N}{1-\pi} + \frac{N_k}{1-\pi} = 0$$

$$\frac{N_k}{\pi} = \frac{N_k - N}{1-\pi}$$

$$\pi = \frac{N_k}{N}$$

3 Question 3

$$x_{new} = \frac{x_{old} - \bar{x}}{\sigma}(1)$$

```
mean for column x and y respectively 12.76923077  12.30769231
standard dev for x and y respectively  20.71695701  25.93062737
Data before normalization where the last column is the category. 1 represents maths,
2 represents Electrical Engineering, 3 represents computer science and 4 represents
economics.
1,0,49,1
2,-7,32,1
3,-9,47,1
4,29,12,2
5,49,31,2
6,37,38,2
7,8,9,3
8,13,-1,3
9,-6,-3,3
10,-21,12,3
11,27,-32,4
12,19,-14,4
13,27,-20,4

data after normalization using formula (1)
 [ 1  -6.16366137e-01   1.41501812e+00   1]
```

```
 [ 2  -9.54253598e-01   7.59422725e-01   1]
 [ 3  -1.05079287e+00   1.33788925e+00   1]
 [ 4   7.83453343e-01  -1.18659801e-02   2]
 [ 5   1.74884609e+00   7.20858289e-01   2]
 [ 6   1.16961044e+00   9.90809336e-01   2]
 [ 7  -2.30209039e-01  -1.27559286e-01   3]
 [ 8   1.11391471e-02  -5.13203638e-01   3]
 [ 9  -9.05983961e-01  -5.90332509e-01   3]
 [ 10  -1.63002852e+00  -1.18659801e-02  3]
 [ 11   6.86914069e-01  -1.70870113e+00  4]
 [ 12   3.00756971e-01  -1.01454130e+00  4]
 [ 13   6.86914069e-01  -1.24592791e+00  4]

test set : 18,20,7,?
normalized test data
     [18   0.34902661  -0.20468816   0]
Manhattan distance L1 of test point from each training
set point respectively

2.58509902532
2.2673910871
2.94239689021
0.627248911567
2.32536592638
2.0160813259
0.656364517639
0.646402942759
1.64065492124
2.17187730399
1.84190043518
0.858122777292
1.37912721231

FOR K=1
we get 0.627248911567 as shortest distance which corresponds to 4th item in the training
set which has class "2".
Prediction : 2 (Electrical)

FOR K=5
Prediction: 3(Computer Science) tied with 4(Economics)
Computer Science wins as its nearer.

For Euclidean L2 of test point from each training set
point respectively

1.88558521056
```

1.62112587003
2.08303615977
0.47529672841
1.67813312988
1.4500248557
0.584348181825
0.457547526249
1.31292539561
1.98842641063
1.54150023132
0.811290371135
1.09469089529

for k=1 0.457547526249 is shortest distance that corresponds to class 3(Computer Science)

for k=5 we get tie between Class 3(Computer Science) and class 4 (Economics)
But we choose class 3(computer science) as it is nearer.

3.b: (10 points) Suppose now we want to derive a probabilistic K-Nearest Neighbor for cla cation of an unlabeled data point x, which is D-dimensional. We have a (multi-dimensiona D-sphere with center at x, allowing its radius to grow until it precisely contains K labe data points, irrespective of their class. At this size, the volume of the sphere is V . I there be a total of N labeled data points in the entire space (both inside and outside of sphere), with Nc data points labeled as class c, such that c Nc = N. Also, a subset of th K data points inside of the sphere belongs to class c, there are Kc of them in total. We estimated density associated with each class as p (x | Y = c) = Kc and the class prior as p (Y = c) = N

Ans: We know

$$P(X|Y) = \frac{K_c}{N_c V}$$

$$P(Y = c) = \frac{N_c}{N}$$

We have to find P(X)

$$P(X|Y = c)P(Y = c) = P(Y = c|X) * P(X)$$

$$P(X) = P(X|Y = c) * P(Y = c)$$

$$= \frac{K_c}{N_c V} * \frac{N_c}{N}$$

$$= \sum_c \frac{K_c}{VN}$$

As we know

$$= \sum_c K_c = K$$

Therefore

$$P(X) = \frac{K}{VN}$$

We know

$$P(Y = c|X) * P(X) = P(X|Y = c) * P(Y = c)$$

$$P(Y = c|X) = \frac{\frac{K_c}{N_c V} * \frac{N_c}{N}}{\frac{K}{VN}}$$

$$= \frac{K_c}{K}$$

4 Question 4

4.a: (5 points) Suppose you want to grow a decision tree to predict the accident rate based on the following accident data which provides the rate of accidents in 100 observations. Which predictor variable (weather or traffic) will you choose to split in the first step to maximize the information gain?

Ans: Using the entropy and information gain formula give below we will predict variable to split.

$$H[X] = -\sum_{k=1}^{K} P(X = a_k) log P(X = a_k)$$

$$GAIN = H[Y] * H[Y|X]$$

For Weather attribute

$$H[Weather] = -0.28 log 0.28 - 0.72 log 0.72 = 0.84146$$

where

$$P(X = sunny) = \frac{28}{100} and P(X = raining) = \frac{72}{100}$$

$$H[Sunny] = -P(X = high) log P(X = high) - P(X = low) log P(X = low)$$

$$H[Sunny] = -\frac{23}{28} log \frac{23}{28} - \frac{5}{28} log \frac{5}{28} = 0.676942$$

$$H[raining] = -P(X = high) log P(X = high) - P(X = low) log P(X = low)$$

$$H[raining] = -\frac{50}{72} log \frac{50}{72} - \frac{22}{72} log \frac{22}{72} = 0.887976$$

$$GAIN = H[Weather] - P(sunny) * H[Sunny] - P(raining) * H[raining]$$

$$GAIN = 0.84146 - \frac{28}{100} * 0.676942 - \frac{72}{100} * 0.887976 = 0.012574$$

For Traffic attribute

$$H[Traffic] = -0.73log0.73 - 0.27log0.27 = 0.84146$$

where

$$P(X = Heavy) = \frac{73}{100} and P(X = Light) = \frac{27}{100}$$

$$H[Heavy] = -P(X = high)logP(X = high)$$

$$H[Heavy] = -\frac{73}{73}log\frac{73}{73} = 0$$

$$H[Light] = -P(X = low)logP(X = low)$$

$$H[Light] = -\frac{27}{27}log\frac{27}{27} = 0$$

$$GAIN = H[Traffic] - P(Heavy) * H[Heavy] - P(Light) * H[Light]$$

$$GAIN = 0.84146 - \frac{72}{100} * 0 - \frac{27}{100} * 0 = 0.84146$$

As traffic has higher information gain than weather, we will predict traffic.

4.b: (5 points) Suppose in another dataset, two students experiment with decision trees. The first student runs the decision tree learning algorithm on the raw data and obtains a tree T1. The second student, normalizes the data by subtracting the mean and dividing by the variance of the features. Then, he runs the same decision tree algorithm with the same parameters and obtains a tree T2. How are the trees T1 and T2 related?

Ans: Normalization should have almost no impact on the performance and structure of a decision tree. It is useful when we are trying to solve equations but not when we are splitting the features. Normalization is basically scaling so that cannot change the split. Here we are trying to compare features and branching down the tree therefore normalization will not have an impact on the tree. T1 and T2 will be almost same.

4.c: In training decision trees, the ultimate goal is to minimize the classification error. However, the classification error is not a smooth function; thus, several surrogate loss functions have been proposed. Two of the most common loss functions are the Gini index and Cross- entropy, see [MLaPP, Section 16.2.2.2] or [ESL, Section 9.2.3] for the definitions. Prove that, for any discrete probability distribution p with K classes, the value of the Gini index is less than or equal to the corresponding value of the cross-entropy. This implies that the

Gini index is a better approximation of the misclassification error.
Ans: We know that gini index is

$$\sum_{k=1}^{K} p_k(1-p_k)$$

And cross entropy is

$$-\sum_{k=1}^{K} p_k log p_k$$

`we will prove that gini \le cross entropy`

$$\sum_{k=1}^{K} p_k(1-p_k) - (-\sum_{k=1}^{K} p_k log p_k)$$

$$= \sum_{k=1}^{K} p_k(1-p_k) + p_k log p_k$$

$$= \sum_{k=1}^{K} p_k(1-p_k+log p_k)(1)$$

Assuming equation 1 is ≤ 0
$Therefore$:

$$\sum_{k=1}^{K} p_k(1-p_k+log p_k) \leq 0$$

$$\sum_{k=1}^{K} (1-p_k+log p_k) \leq 0(2)$$

we know that 1 - $\mathbf{p}_k$ *is always between* 0 *and* 1. $Logp_k$ *is log of a probability and is always* <= 0

$$log p_k \leq \sum_{k=1}^{K} (1-p_k+log p_k)]$$

We came to a true solution that means the assumption we took was correct. Therefore

$$\sum_{k=1}^{K} p_k(1-p_k) - (-\sum_{k=1}^{K} p_k log p_k) \leq 0]$$

5 Question 5

Q) How many attributes are?

Ans: There are 10 + 1 .10 being attributes for measuring and 11th one being class.

Q) Do you think that all attributes are meaningful for the classification? If not, explain why?

Ans: The id column can be left out during classification as it does not tell

any quality about the data. Usually we should leave out attributes which overfits the data. I ran my code removing attributes one by one and in all cases accuracy decreased. Therefore I would only eliminate Id attribute and keep all the rest.

Q) How many classes are? Class is a type of a glass.
Ans: There are 6 classes

Q) Please explain the class distribution. Which class is majority? Do you think that it can be considered as a uniform distribution?

Ans) class : probability
The probability of the class is mentiones below.
1.0: 0.34183673469387754, 2.0: 0.37244897959183676, 3.0: 0.07142857142857142, 5.0: 0.05102040816326531, 6.0: 0.030612244897959183, 7.0: 0.1326530612244898
The class distribution seems to follow bernoulli's distribution. Class 2 and 1 form the middle of the distribution. class 3 and 6 goes on the right of the mid and the other two on the left.
Class 2 is majority. No its not a uniform distribution as all the classes have different probability.

5.b: Performance Comparison (10 points) Compare the two algorithms (kNN, Naive Bayes) on the provided dataset. kNN: Consider k = 1, 3, 5, 7. For each k, report the training and test accuracy. When computing the training accuracy of kNN, we use leave-one-out strategy, i.e. classifying each training point using the remaining training points. Note that we use this strategy only for kNN in this assignment. Naive Bayes: Report the training and test accuracy. Discussion: The results from the different classifiers are similar? If so, explain why. If not, which one is better? And please explain why.

KNN

```
K = 1
Euclidean distance:
Training set accuracy 71.4285714286
Test set accuracy 61.1111111111
WITHOUT leaving on out on training set 100.0
Manhattan distance:
Training set accuracy 75.0
Test set accuracy 66.6666666667
WITHOUT leaving on out on training set 100.0
K = 3
Euclidean distance:
Training set accuracy 71.9387755102
```

```
Test set accuracy 61.1111111111
WITHOUT leaving on out on training set 85.2040816327
Manhattan distance:
Training set accuracy 73.4693877551
Test set accuracy 61.1111111111
WITHOUT leaving on out on training set 85.7142857143
K = 5
Euclidean distance:
Training set accuracy 67.8571428571
Test set accuracy 55.5555555556
WITHOUT leaving on out on training set 80.1020408163
Manhattan distance:
Training set accuracy 67.8571428571
Test set accuracy 55.5555555556
WITHOUT leaving on out on training set 80.1020408163
K = 7
Euclidean distance:
Training set accuracy 66.8367346939
Test set accuracy 55.5555555556
WITHOUT leaving on out on training set 73.4693877551
Manhattan distance:
Training set accuracy 68.8775510204
Test set accuracy 50.0
WITHOUT leaving on out on training set 76.5306122449

naive bayes

accuracy for training set 55.1020408163
accuracy for testing set 33.3333333333
```

Q: The results from the different classifiers are similar? If so, explain why. If not, which one is better? And please explain why.

No, the results from the different classifiers are different. The KNN classifier does better with the given data set. Few reasons which I can think are

1. In the given data not all features follow gaussian distribution. When I plotted the graph for each feature, majority were not following gaussian distribution. So I feel maybe because of this reason, the naive bayes algorithm could not perform well.

2. KNN has no assumptions whereas Naive Bayes has a major assumption that the attributes are conditional independent. KNN is more flexible than Naive Bayes i.e KNN can take any decision boundary.

3. The data we used had around 2 features for particular classes which had 0 mean and standard deviation. I am guessing that this sort of data reduced the accuracy for Naive Bayes. This worked fine with KNN because we take the whole feature mean and std dev instead of class wise mean and std dev.