CSE 6363 Homework #1

Xinsheng Li 1001681730

2/7/19

Problem 1

a.

A severe computer failure is a Poisson process in which the probability of the first event to occur at time is an exponential distribution:

$$p_\lambda(x) = \lambda e^{-\lambda x}$$

Given a data set $D = \{k_1, ...k_n\}$, we want to derive the performance function and learn the parameter λ. We first calculate the probability for all the data point:

$$\prod_{i=1}^{n} \lambda e^{-\lambda k_i}$$

Then the log likelihood is:

$$l(\lambda) = \sum_{i=1}^{n} (\log \lambda - \lambda k_i) = n \log \lambda - \lambda \sum_{i=1}^{n} k_i$$

Make the log likelihood's derivative to 0:

$$l(\lambda)' = \frac{n}{\lambda} - \sum_{i=1}^{n} k_i = 0$$

Finally, we can get:

$$\lambda = \frac{1}{\bar{k}} (\bar{k} = \frac{\sum_{i=1}^{n} k_i)}{n}$$

b.

In this problem. Using the given dataset $D = \{1.5, 3, 2.5, 2.57, 2.9, 3\}$. Apply the learning algorithm from a we can calculate the result: $\lambda = 0.3834$.

c.

Show that $x \sim exp(\lambda)$ and $\lambda \sim gamma(\alpha, \beta)$, then $P(\lambda|X) \sim gamma(\alpha', \beta')$.

$$P(\lambda|X) \propto P(X|\lambda)P(\lambda) \propto \lambda^n e^{-\lambda \sum_i x_i} \lambda^{\alpha-1} e^{-\beta\lambda} \propto e^{-\lambda(\sum_i x_i - \beta)} \lambda^{n+\alpha-1}$$

So $\alpha^{'} = n + \alpha$ and $\beta^{'} = \sum_i x_i - \beta$ according to the definition of gamma distribution.

$$P(\lambda|X) \propto gamma(n+\alpha, \sum_i i - \beta)$$

Through similar steps as part a:

$$\log(P(\lambda|X) \propto -\lambda(\sum_i x_i + \beta) + (n+\alpha-1)\log\lambda$$

$$\log(P(\lambda|X)^{'} \propto -(\sum_i x_i + \beta) + \frac{(n+\alpha-1)}{\lambda} = 0$$

$$\lambda = \frac{n+\alpha-1}{\sum_i x_i + \beta}$$

Using the dataset in part b and values for α and β of 5 and 10. Then we can get the result:

$$\lambda = \frac{n+\alpha-1}{\sum_i x_i + \beta} = \frac{6+5-1}{\sum_i k_i + 10} = 0.3899$$

Problem 2
a.

First we should calculate the Cartesian distance with the data items and the data point, and according to the k value to select the prediction result.

We use $\sqrt{\sum_i^n (x_i - y_i)^2}$ to calculate the Cartesian distance, and get the following results:
(155, 40, 35)
Distance calculation
I use a python program to calculate the distance which was used in part b.
$D((170, 57, 32), W) = 22.86919325$
$D((192, 95, 28), M) = 66.65583245$
$D((150, 45, 30), W) = 8.660254038$
$D((170, 65, 29), M) = 29.76575213$
$D((175, 78, 35), M) = 42.94182111$
$D((185, 90, 32), M) = 58.38664231$
$D((170, 65, 28), W) = 29.9833287$
$D((155, 48, 31), W) = 8.94427191$
$D((160, 55, 30), W) = 16.58312395$
$D((182, 80, 30), M) = 48.51803788$
$D((175, 69, 28), W) = 35.91656999$
$D((180, 80, 27), M) = 47.84349486$
$D((160, 50, 31), W) = 11.87434209$
$D((175, 72, 30), M) = 38.06573262$

Neighbor Select
K = 1: Select: $D((150,45,30),W) = 8.660254038$
K = 3: Select: $D((150,45,30),W) = 8.660254038$, $D((155,48,31),W) = 8.94427191$, $D((160,50,31),W) = 11.87434209$
K = 5: Select: $D((150,45,30),W) = 8.660254038$, $D((155,48,31),W) = 8.94427191$, $D((160,50,31),W) = 11.87434209$, $D((160,55,30),W) = 16.58312395$, $D((170,57,32),W) = 22.86919325$
Prediction
K = 1: Select: W
K = 3: Select: W
K = 5: Select: W
(170, 70, 32)
Distance calculation
$D((170,57,32),W) = 13$
$D((192,95,28),M) = 33.5410196$
$D((150,45,30),W) = 32.07802986$
$D((170,65,29),M) = 5.830951895$
$D((175,78,35),M) = 9.899494937$
$D((185,90,32),M) = 25$
$D((170,65,28),W) = 6.403124237$
$D((155,48,31),W) = 26.64582519$
$D((160,55,30),W) = 18.13835715$
$D((182,80,30),M) = 15.74801575$
$D((175,69,28),W) = 6.480740698$
$D((180,80,27),M) = 15$
$D((160,50,31),W) = 22.38302929$
$D((175,72,30),M) = 5.744562647$
Neighbor Select
K = 1: Select: $D((175,72,30),M) = 5.744562647$
K = 3: Select: $D((175,72,30),M) = 5.744562647$, $D((170,65,29),M) = 5.830951895$, $D((170,65,28),W) = 6.403124237$
K = 5: Select: $D((175,72,30),M) = 5.744562647$, $D((170,65,29),M) = 5.830951895$, $D((170,65,28),W) = 6.403124237$, $D((175,69,28),W) = 6.480740698$, $D((175,78,35),M) = 9.899494937$
Prediction
K = 1: Select: M
K = 3: Select: M
K = 5: Select: M
(175, 70, 35)
Distance calculation
$D((170,57,32),W) = 14.24780685$
$D((192,95,28),M) = 31.0322413$
$D((150,45,30),W) = 35.70714214$
$D((170,65,29),M) = 9.273618495$
$D((175,78,35),M) = 8$
$D((185,90,32),M) = 22.56102835$
$D((170,65,28),W) = 9.949874371$

$D((155,48,31),W) = 30$
$D((160,55,30),W) = 21.79449472$
$D((182,80,30),M) = 13.19090596$
$D((175,69,28),W) = 7.071067812$
$D((180,80,27),M) = 13.74772708$
$D((160,50,31),W) = 25.3179778$
$D((175,72,30),M) = 5.385164807$

Neighbor Select

K = 1: Select: $D((175,72,30),M) = 5.385164807$
K = 3: Select: $D((175,72,30),M) = 5.385164807$, $D((175,69,28),W) = 7.071067812$, $D((175,78,35),M) = 8$
K = 5: Select: $D((175,72,30),M) = 5.385164807$, $D((175,69,28),W) = 7.071067812$, $D((175,78,35),M) = 8$, $D((170,65,29),M) = 9.273618495$, $D((170,65,28),W) = 9.949874371$

Prediction

K = 1: Select: M
K = 3: Select: M
K = 5: Select: M

(180, 90, 20)

Distance calculation

$D((170,57,32),W) = 36.51027253$
$D((192,95,28),M) = 15.26433752$
$D((150,45,30),W) = 55$
$D((170,65,29),M) = 28.39013913$
$D((175,78,35),M) = 19.84943324$
$D((185,90,32),M) = 13$
$D((170,65,28),W) = 28.08914381$
$D((155,48,31),W) = 50.0999002$
$D((160,55,30),W) = 41.53311931$
$D((182,80,30),M) = 14.28285686$
$D((175,69,28),W) = 23.02172887$
$D((180,80,27),M) = 12.20655562$
$D((160,50,31),W) = 46.05431576$
$D((175,72,30),M) = 21.1896201$

Neighbor Select

K = 1: Select: $D((180,80,27),M) = 12.20655562$
K = 3: Select: $D((180,80,27),M) = 12.20655562$, $D((185,90,32),M) = 13$, $D((182,80,30),M) = 14.28285686$
K = 5: Select: $D((180,80,27),M) = 12.20655562$, $D((185,90,32),M) = 13$, $D((182,80,30),M) = 14.28285686$, $D((192,95,28),M) = 15.26433752$, $D((175,78,35),M) = 19.84943324$

Prediction

K = 1: Select: M
K = 3: Select: M
K = 5: Select: M

b.

Please see the python files and comments.

c.
(155, 40)
Distance calculation
$D((170, 57), W) = 22.6715681$
$D((192, 95), M) = 66.28725368$
$D((150, 45), W) = 7.071067812$
$D((170, 65), M) = 29.15475947$
$D((175, 78), M) = 42.94182111$
$D((185, 90), M) = 58.30951895$
$D((170, 65), W) = 29.15475947$
$D((155, 48), W) = 8$
$D((160, 55), W) = 15.8113883$
$D((182, 80), M) = 48.25971405$
$D((175, 69), W) = 35.22782991$
$D((180, 80), M) = 47.16990566$
$D((160, 50), W) = 11.18033989$
$D((175, 72), M) = 37.73592453$
Neighbor Select
K = 1: Select: $D((150, 45), W) = 7.071067812$
K = 3: Select: $D((150, 45), W) = 7.071067812$, $D((155, 48), W) = 8$, $D((160, 50), W) = 11.18033989$

K = 5: Select: $D((150, 45), W) = 7.071067812$, $D((155, 48), W) = 8$, $D((160, 50), W) = 11.18033989$, $D((160, 55), W) = 15.8113883$, $D((170, 57), W) = 22.6715681$
Prediction
K = 1: Select: W
K = 3: Select: W
K = 5: Select: W
(170, 70)
Distance calculation
$D((170, 57), W) = 13$
$D((192, 95), M) = 33.30165161$
$D((150, 45), W) = 32.01562119$
$D((170, 65), M) = 5$
$D((175, 78), M) = 9.433981132$
$D((185, 90), M) = 25$
$D((170, 65), W) = 5$
$D((155, 48), W) = 26.62705391$
$D((160, 55), W) = 18.02775638$
$D((182, 80), M) = 15.62049935$
$D((175, 69), W) = 5.099019514$
$D((180, 80), M) = 14.14213562$
$D((160, 50), W) = 22.36067977$
$D((175, 72), M) = 5.385164807$
Neighbor Select
K = 1: Select: $D((170, 65), W) = 5$

K = 3: Select: $D((170, 65), M) = 5$, $D((170, 65), W) = 5$, $D((175, 69), W) = 5.099019514$
K = 5: Select: $D((170, 65), M) = 5$, $D((170, 65), W) = 5$, $D((175, 69), W) = 5.099019514$, $D((175, 72), M) = 5.385164807$, $D((175, 78), M) = 9.433981132$

Prediction

K = 1: Select: W
K = 3: Select: W
K = 5: Select: M

(175, 70)

Distance calculation

$D((170, 57), W) = 13.92838828$
$D((192, 95), M) = 30.23243292$
$D((150, 45), W) = 35.35533906$
$D((170, 65), M) = 7.071067812$
$D((175, 78), M) = 8$
$D((185, 90), M) = 22.36067977$
$D((170, 65), W) = 7.071067812$
$D((155, 48), W) = 29.73213749$
$D((160, 55), W) = 21.21320344$
$D((182, 80), M) = 12.20655562$
$D((175, 69), W) = 1$
$D((180, 80), M) = 11.18033989$
$D((160, 50), W) = 25$
$D((175, 72), M) = 2$

Neighbor Select

K = 1: Select: $D((175, 69), W) = 1$
K = 3: Select: $D((175, 69), W) = 1$, $D((175, 72), M) = 2$, $D((170, 65), W) = 7.071067812$
K = 5: Select: $D((175, 69), W) = 1$, $D((175, 72), M) = 2$, $D((170, 65), M) = 7.071067812$, $D((175, 78), M) = 8$, $D((170, 65), W) = 7.071067812$

Prediction

K = 1: Select: W
K = 3: Select: W
K = 5: Select: M

(180, 90)

Distance calculation

$D((170, 57), W) = 34.4818793$
$D((192, 95), M) = 13$
$D((150, 45), W) = 54.08326913$
$D((170, 65), M) = 26.92582404$
$D((175, 78), M) = 13$
$D((185, 90), M) = 5$
$D((170, 65), W) = 26.92582404$
$D((155, 48), W) = 48.87739764$
$D((160, 55), W) = 40.31128874$
$D((182, 80), M) = 10.19803903$
$D((175, 69), W) = 21.58703314$

$D((180, 80), M) = 10$
$D((160, 50), W) = 44.72135955$
$D((175, 72), M) = 18.68154169$
Neighbor Select
K = 1: Select: $D((185, 90), M) = 5$
K = 3: Select: $D((185, 90), M) = 5$, $D((180, 80), M) = 10$, $D((182, 80), M) = 10.19803903$
K = 5: Select: $D((185, 90), M) = 5$, $D((180, 80), M) = 10$, $D((182, 80), M) = 10.19803903$, $D((175, 78), M) = 13$, $D((192, 95), M) = 13$
Prediction
K = 1: Select: M
K = 3: Select: M
K = 5: Select: M
As all the above prediction shows, the data with age feature can give a better result. Because when k =1, 3, 5 the prediction is the same. But without the age feature, we seem to allow the value of k get bigger to get a better result, when k=1, 3. It seems get a wrong prediction.
Problem 3
a.
Assuming all the features obey Gaussian distribution. So we can use the data to learn the parameters μ, σ of Gaussian distribution. Use Numpy divide the data into two groups: man and woman. And calculate the μ, σ respectively.
Features of man:
$\mu(height|man) = 179.8571 \qquad \sigma(height|man) = 7.3355$
$\mu(weight|man) = 80.0 \qquad \sigma(weight|man) = 10.1489$
$\mu(age|man) = 30.1429 \qquad \sigma(age|man) = 2.6726$
Features of woman:
$\mu(height|woman) = 162.8571 \qquad \sigma(height|woman) = 9.0633$
$\mu(weight|woman) = 55.5714 \qquad \sigma(weight|woman) = 8.8667$
$\mu(age|woman) = 30.0 \qquad \sigma(age|woman) = 1.5275$
Then we can use all the μ, σ we get to build a Gaussian distribution, and use that Gaussian distribution to calculate the probability. The formula is following:

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

For example: we input 190, 70, 35, and we should calculate the probability $p(height = 190|man)p(weight = 70|man)p(age = 35|man)p(man) \approx 7.2398 * 10^{-6}$ and $p(height = 190|woman)p(weight = 70|woman)p(age = 35|woman)p(woman) \approx 3.6601 * 10^{-9}$ using the formula above. And the calculation details are in the program. So we can get a prediction that is a man.
And the predictions of all the datas given in problem 2 part a is woman, man, man, man.
b.
Please see the python files and comments.
c.
When the age data is removed, then we should calculate the probability $p(height = 190|man)p(weight = 70|man)p(man) \approx 0.0002529$ and $p(height = 190|woman)p(weight = 70|woman)p(woman) \approx 2.9727 * 10^{-6}$. So we get the result that is a man.
And the predictions of all the datas given in problem 2 part a is also woman, man, man, man

without the age feature.

d.

I think Gaussian Naive Bayes might perform better than KNN. Because Gaussian Naive Bayes has a more stable performance than KNN. To get a better result in KNN, you should learn the parameter k. Such as the problem part c, when the age date is removed. We got different prediction. Also there is a big problem when k=3, and the third and forth biggest distance is the same, but their label is different. It's hard to decide which one should be selected as the nearest neighbor. You will spend a lot time to get a better k value. And every new data come, we should repeat calculate the distance between it and all the other points. It costs a lot time. Gaussian Naive Bayes just have to calculate the mu and sigma. The calculation is done, it can be used later. And it's easier to compare two probability, it's seem never be the same, so it's easy to predict the final result. And when the number of the training set is not too big. Gaussian Naive Bayes also has a better performance than KNN.
Above all, Gaussian Naive Bayes might perform better than KNN.