CSE 6363 - *Machine Learning*

Homework 1- SampleSolution without Code - Spring 2019

MLE and MAP

1. In class we covered the derivation of basic learning algorithms to derive a model for a coin flip task. Consider a similar problems where we monitor the time of the occurrence of a severe computer failure (which requires a system reboot) and which occurs according to a Poisson process (i.e. it is equally likely to happen at any point in time with an arrival rate of λ). For a Poisson process the probability of the first event to occur at time x after a restart is described by an exponential distribution:

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

We are assuming here that the different data points we measured are independent, i.e. nothing changes between reboots.

a) Derive the performance function and the optimization result for analytic MLE optimization for a model learning algorithm that returns the MLE for the parameter λ of the model given a data set $D = \{k_1, ...k_n\}$. Make sure you show your steps.

Here the model parameter is the failure rate, λ. The performance function for MLE is thus:

$$p(D|\lambda) = \prod_{i=1}^{n} p_\lambda(k_i) = \prod_{i=1}^{n} \lambda e^{-\lambda k_i}$$

To find the maximum likelihood model we need to find:

$$\lambda^* = argmax_\lambda p(D|\lambda) = argmax_\lambda \prod_{i=1}^{n} \lambda e^{-\lambda k_i}$$

To make the solution easier the product can be converted into a sum by using log likelihoods:

$$\begin{aligned} \lambda^* &= argmax_\lambda p(D|\lambda) = argmax_\lambda \ln\left(p(D|\lambda)\right) = argmax_\lambda \textstyle\sum_{i=0}^{n} \ln\left(\lambda e^{-\lambda k_i}\right) \\ &= argmax_\lambda \textstyle\sum_{i=0}^{n} \left(\ln\lambda - \lambda k_i\right) = argmax_\lambda \left(n\ln\lambda - \lambda \textstyle\sum_{i=0}^{n} k_i\right) \end{aligned}$$

To optimize the performance and thus find λ^* we compute the derivative of the performance function:

$$\frac{d}{d\lambda}\ln p(D|\lambda) = \frac{d}{d\lambda}\left(n\ln\lambda - \lambda\sum_{i=0}^{n} k_i\right) = \frac{d}{d\lambda}n\ln\lambda - \frac{d}{d\lambda}\lambda\sum_{i=0}^{n} k_i = \frac{n}{\lambda} - \sum_{i=0}^{n} k_i$$

To find the optimum we can solve this analytically by setting the derivative to 0:

$$\begin{aligned}&\tfrac{d}{d\lambda}\ln p(D|\lambda) = \tfrac{n}{\lambda} - \textstyle\sum_{i=0}^{n} k_i = 0\\ &\Rightarrow \tfrac{n}{\lambda} = \textstyle\sum_{i=0}^{n} k_i\\ &\Rightarrow \tfrac{n}{\sum_{i=0}^{n} k_i} = \lambda\end{aligned}$$

This gives us the best model parameter λ^*.

b) Apply the learning algorithm from a) to the following dataset:

$$D = \{1.5, 3, 2.5, 2.75, 2.9, 3\}.$$

To apply the algorithm we simply need to compute the λ that maximizes the MLE performance function and thus:

$$\lambda = \frac{n}{\sum_{i=0}^{n} k_i} = \frac{6}{1.5+3+2.5+2.75+2.9+3} = \frac{6}{15.65} = 0.38339$$

c) Derive the optimization for a MAP approach using the conjugate prior, the Gamma distribution. The Gamma distribution is:

$$p_{\alpha,\beta}(\lambda) = \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}$$

Note that α and β are constants and that there still is only one parameter, λ, to be learned. Show your derivation and the result for the data in part b) and values for α and β of 5 and 10, respectively.

The performance function for MAP is $p(\lambda|D)$. To get to this we apply Bayes law and since the denominator ($p(D)$) is the same for all datasets we can replace it with a constant and we get:

$$p(\lambda|D) = \gamma p(D|\lambda)p(\lambda) = \gamma\prod_{i=1}^{n}\lambda e^{-\lambda k_i}\frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}$$

We can again find the optimal parameter using log likelihoods:

$$\begin{aligned}\lambda^* &= argmax_\lambda \gamma\textstyle\prod_{i=1}^{n}\lambda e^{-\lambda k_i}\frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}\\ &= argamx_\lambda \ln\left(\gamma\textstyle\prod_{i=1}^{n}\lambda e^{-\lambda k_i}\frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}\right)\\ &= argamx_\lambda \left(\ln\gamma + \textstyle\sum_{i=1}^{n}\ln\lambda e^{-\lambda k_i} + \ln\frac{\beta^\alpha}{\Gamma(\alpha)} + \ln\left(\lambda^{\alpha-1}e^{-\beta\lambda}\right)\right)\\ &= argamx_\lambda \left(\ln\gamma + n\ln\lambda - \lambda\textstyle\sum_{i=0}^{n}k_i + \ln\frac{\beta^\alpha}{\Gamma(\alpha)} + (\alpha-1)\ln\lambda - \beta\lambda\right)\\ &= argmax_\lambda \left(\left(\ln\gamma + \ln\frac{\beta^\alpha}{\Gamma(\alpha)}\right) + \ln\lambda\,(n+(\alpha-1)) - \lambda\,(\textstyle\sum_{i=0}^{n}k_i + \beta)\right)\end{aligned}$$

Since constants do not alter the result of $argmax$ we can drop these giving us:

$$\lambda^* = argmax_\lambda\left(\ln\lambda\,(n+(\alpha-1)) - \lambda\left(\sum_{i=0}^{n}k_i + \beta\right)\right)$$

To optimize this, we again form the derivative:

$$\frac{d}{d\lambda}\ln p(\lambda|D) = \frac{d}{d\lambda}\ln\lambda\left(n+(\alpha-1)\right) - \frac{d}{d\lambda}\lambda\left(\sum_{i=0}^{n}k_i+\beta\right) = \frac{n+\alpha-1}{\lambda} - \left(\sum_{i=0}^{n}k_i+\beta\right)$$

Analytically solving this by setting it to 0 yields:

$$\begin{aligned}
&\tfrac{d}{d\lambda}\ln p(\lambda|D) = \tfrac{n+\alpha-1}{\lambda} - (\textstyle\sum_{i=0}^{n}k_i+\beta) = 0\\
&\Rightarrow \tfrac{n+\alpha-1}{\lambda} = (\textstyle\sum_{i=0}^{n}k_i+\beta)\\
&\Rightarrow \lambda = \tfrac{n+\alpha-1}{\sum_{i=0}^{n}k_i+\beta}
\end{aligned}$$

For the example in b) with $\alpha = 5$ and $\beta = 10$ this yields:

$$\lambda = \frac{6+5-1}{15.65+10} = \frac{10}{25.65} = 0.3899$$

K Nearest Neighbor

2. Consider the problem where we want to predict the gender of a person from a set of input parameters, namely height, weight, and age. Assume our training data is given as follows:

$$\begin{aligned}
D = \{\quad &((170,57,32), \quad W),\\
&((192,95,28), \quad M),\\
&((150,45,30), \quad W),\\
&((170,65,29), \quad M),\\
&((175,78,35), \quad M),\\
&((185,90,32), \quad M),\\
&((170,65,28), \quad W),\\
&((155,48,31), \quad W),\\
&((160,55,30), \quad W),\\
&((182,80,30), \quad M),\\
&((175,69,28), \quad W),\\
&((180,80,27), \quad M),\\
&((160,50,31), \quad W),\\
&((175,72,30), \quad M), \quad \}
\end{aligned}$$

a) Using Cartesian distance as the similarity measurements show the results of the gender prediction for the following data items for values of K of 1, 3, and 5. Include the intermedia steps (i.e. distance calculation, neighbor selection, prediction).

$$(155,40,35), (170,70,32), (175,70,35), (180,90,20)$$

For each of the data points in the test set we have to calculate the distance to the points in the training set to be able to pick the closes neighbors.

$(155, 40, 35)$: The distances to the training points are (in order):
22.87, 66.66, 8.66, 29.77, 42.94, 58.39, 29.98, 8.94, 16.58, 48.52, 35.92, 47.84, 11.87, 38.07
Thus for $K = 1$ the closes point is the third data point and thus the classification is W
For $K = 3$, the 3 closest neighbors are points 3, 8, and 13, resulting in 3 W votes and thus also a classification as W
For $K = 5$, the closest neighbors are 3, 8, 13, 9, and 1, resulting in 5 W, thus also yielding a classification as W

$(170, 70, 32)$: The distances to the training points are (in order):
15.81, 28.46, 38.34, 10.72, 9.43, 20.88, 11.05, 32.57, 24.29, 10.63, 5.74, 11.22, 27.60, 4.90
Thus for $K = 1$ the closes point is the 14th point and thus the classification is M
For $K = 3$, the 3 closest neighbors are points 14, 11, and 5, resulting in 2 M and one W votes, thus yielding a classification of M
For $K = 5$, the closest neighbors are 14, 11, 5, 10, and 4, resulting in 4 M and one W vote, thus yielding a classification of M

$(175, 70, 35)$: The distances to the training points are (in order):
14.25, 31.03, 35.71, 9.27, 8.00, 22.56, 9.95, 30.00, 21.79, 13.19, 7.07, 13.75, 25.32, 5.39
Thus for $K = 1$ the closes point is the 14th point and thus the classification is M
For $K = 3$, the 3 closest neighbors are points 14, 11, and 5, resulting in 2 M and one W votes, thus yielding a classification of M
For $K = 5$, the closest neighbors are 14, 11, 5, 4, and 7, resulting 3 M and 2 W votes, thus yielding a classification of M

$(180, 90, 20)$: The distances to the training points are (in order):
36.51, 15.26, 55.00, 28.39, 19.85, 13.00, 28.09, 50.10, 41.53, 14.28, 23.02, 12.21, 46.05, 21.19
Thus for $K = 1$ the closes point is the 12th point and thus the classification is M
For $K = 3$, the 3 closest neighbors are points 12, 6, and 10, resulting in 3 M votes and thus a classification as M
For $K = 5$, the closest neighbors are 12, 6, 10, 2, and 5, resulting in 5 M votes and thus a classification as M

b) Implement the KNN algorithm for this problem. Your implementation should work with different training data sets and allow to input a data point for the prediction.

c) Repeat the prediction using KNN when the age data is removed. Try to determine (using multiple target values) which data gives you better predictions. Show your intermediate results.

To test the prediction the algorithm has to be executed on some data that has known class labels. The easiest way to do this is to use the data in the original training data set by removing a test set from it. The best way here would probably be "leave one out", i.e. treat each data point as a test point with the reminder of the set serving as the training set. This way as many predictions as data points can be made. An alternative would be to remove a small subset from the main set and test these on the remaining data.

For "leave one out" we can calculate the distance from each point to all others in the training set, pick the K nearest ones, and then classify the point, evaluating its performance.

Using all 3 attributes the distances and classification results are as follows (misclassifications are labeled with ⋆):

Data Point (170 57 32) :

Distances to points: 0.00, 44.09, 23.41, 8.54, 21.79, 36.25, 8.94, 17.52, 10.39, 26.02, 13.60, 25.57, 12.25, 15.94,
1-NN:
⋆ Real Label W, KNN Label M
3-NN: (M W W)
Real Label W, KNN Label W

Data Point (192 95 28) :

Distances to points: 44.09, 0.00, 65.33, 37.22, 25.04, 9.49, 37.20, 59.89, 51.26, 18.14, 31.06, 19.24, 55.30, 28.67,
1-NN:
Real Label M, KNN Label M
3-NN: (M M M)
Real Label M, KNN Label M

Data Point (150 45 30) :

Distances to points: 23.41, 65.33, 0.00, 28.30, 41.70, 57.04, 28.35, 5.92, 14.14, 47.42, 34.71, 46.20, 11.22, 36.80,
1-NN:
Real Label W, KNN Label W
3-NN: (W W W)
Real Label W, KNN Label W

Data Point (170 65 29) :

Distances to points: 8.54, 37.22, 28.30, 0.00, 15.17, 29.31, 1.00, 22.76, 14.18, 19.24, 6.48, 18.14, 18.14, 8.66,
1-NN:
⋆ Real Label M, KNN Label W
3-NN: (W W W)
⋆ Real Label M, KNN Label W

Data Point (175 78 35) :

Distances to points: 21.79, 25.04, 41.70, 15.17, 0.00, 15.91, 15.59, 36.28, 27.91, 8.83, 11.40, 9.64, 32.02, 7.81,
1-NN:
Real Label M, KNN Label M
3-NN: (M M M)
Real Label M, KNN Label M

Data Point (185 90 32) :

Distances to points: 36.25, 9.49, 57.04, 29.31, 15.91, 0.00, 29.43, 51.62, 43.06, 10.63, 23.60, 12.25, 47.18, 20.69,
1-NN:

Real Label M, KNN Label M
3-NN: (M M M)
Real Label M, KNN Label M

Data Point (170 65 28) :
Distances to points: 8.94, 37.20, 28.35, 1.00, 15.59, 29.43, 0.00, 22.87, 14.28, 19.31, 6.40, 18.06, 18.28, 8.83,
1-NN:
⋆ Real Label W, KNN Label M
3-NN: (M W M)
⋆ Real Label W, KNN Label M

Data Point (155 48 31) :
Distances to points: 17.52, 59.89, 5.92, 22.76, 36.28, 51.62, 22.87, 0.00, 8.66, 41.88, 29.15, 40.80, 5.39, 31.26,
1-NN:
Real Label W, KNN Label W
3-NN: (W W W)
Real Label W, KNN Label W

Data Point (160 55 30) :
Distances to points: 10.39, 51.26, 14.14, 14.18, 27.91, 43.06, 14.28, 8.66, 0.00, 33.30, 20.62, 32.16, 5.10, 22.67,
1-NN:
Real Label W, KNN Label W
3-NN: (W W W)
Real Label W, KNN Label W

Data Point (182 80 30) :
Distances to points: 26.02, 18.14, 47.42, 19.24, 8.83, 10.63, 19.31, 41.88, 33.30, 0.00, 13.19, 3.61, 37.22, 10.63,
1-NN:
Real Label M, KNN Label M
3-NN: (M M M)
Real Label M, KNN Label M

Data Point (175 69 28) :
Distances to points: 13.60, 31.06, 34.71, 6.48, 11.40, 23.60, 6.40, 29.15, 20.62, 13.19, 0.00, 12.12, 24.39, 3.61,
1-NN:
⋆ Real Label W, KNN Label M
3-NN: (M W M)
⋆ Real Label W, KNN Label M

Data Point (180 80 27) :
Distances to points: 25.57, 19.24, 46.20, 18.14, 9.64, 12.25, 18.06, 40.80, 32.16, 3.61, 12.12,

0.00, 36.28, 9.90,
1-NN:
Real Label M, KNN Label M
3-NN: (M M M)
Real Label M, KNN Label M

Data Point (160 50 31) :
Distances to points: 12.25, 55.30, 11.22, 18.14, 32.02, 47.18, 18.28, 5.39, 5.10, 37.22, 24.39, 36.28, 0.00, 26.65,
1-NN:
Real Label W, KNN Label W
3-NN: (W W W)
Real Label W, KNN Label W

Data Point (175 72 30) :
Distances to points: 15.94, 28.67, 36.80, 8.66, 7.81, 20.69, 8.83, 31.26, 22.67, 10.63, 3.61, 9.90, 26.65, 0.00,
1-NN:
⋆ Real Label M, KNN Label W
3-NN: (W M M)
Real Label M, KNN Label M

1-NN Error Rate: 5 / 14

3-NN Error Rate: 3 / 14

Ignoring the age attribute and using only the first 2, the results become:

Data Point (170 57 32) :
Distances to points: 0.00, 43.91, 23.32, 8.00, 21.59, 36.25, 8.00, 17.49, 10.20, 25.94, 13.00, 25.08, 12.21, 15.81,
1-NN:
⋆ Real Label W, KNN Label M
3-NN: (M W W)
Real Label W, KNN Label W

Data Point (192 95 28) :
Distances to points: 43.91, 0.00, 65.30, 37.20, 24.04, 8.60, 37.20, 59.82, 51.22, 18.03, 31.06, 19.21, 55.22, 28.60,
1-NN:
Real Label M, KNN Label M
3-NN: (M M M)
Real Label M, KNN Label M

Data Point (150 45 30) :
Distances to points: 23.32, 65.30, 0.00, 28.28, 41.40, 57.01, 28.28, 5.83, 14.14, 47.42, 34.66,

46.10, 11.18, 36.80,
1-NN:
Real Label W, KNN Label W
3-NN: (W W W)
Real Label W, KNN Label W

Data Point (170 65 29) :
Distances to points: 8.00, 37.20, 28.28, 0.00, 13.93, 29.15, 0.00, 22.67, 14.14, 19.21, 6.40, 18.03, 18.03, 8.60,
1-NN:
⋆ Real Label M, KNN Label W
3-NN: (W W W)
⋆ Real Label M, KNN Label W

Data Point (175 78 35) :
Distances to points: 21.59, 24.04, 41.40, 13.93, 0.00, 15.62, 13.93, 36.06, 27.46, 7.28, 9.00, 5.39, 31.76, 6.00,
1-NN:
Real Label M, KNN Label M
3-NN: (M M M)
Real Label M, KNN Label M

Data Point (185 90 32) :
Distances to points: 36.25, 8.60, 57.01, 29.15, 15.62, 0.00, 29.15, 51.61, 43.01, 10.44, 23.26, 11.18, 47.17, 20.59,
1-NN:
Real Label M, KNN Label M
3-NN: (M M M)
Real Label M, KNN Label M

Data Point (170 65 28) :
Distances to points: 8.00, 37.20, 28.28, 0.00, 13.93, 29.15, 0.00, 22.67, 14.14, 19.21, 6.40, 18.03, 18.03, 8.60,
1-NN:
⋆ Real Label W, KNN Label M
3-NN: (M W W)
Real Label W, KNN Label W

Data Point (155 48 31) :
Distances to points: 17.49, 59.82, 5.83, 22.67, 36.06, 51.61, 22.67, 0.00, 8.60, 41.87, 29.00, 40.61, 5.39, 31.24,
1-NN:
Real Label W, KNN Label W
3-NN: (W W W)
Real Label W, KNN Label W

Data Point (160 55 30) :

Distances to points: 10.20, 51.22, 14.14, 14.14, 27.46, 43.01, 14.14, 8.60, 0.00, 33.30, 20.52, 32.02, 5.00, 22.67,
1-NN:
Real Label W, KNN Label W
3-NN: (W W W)
Real Label W, KNN Label W

Data Point (182 80 30) :

Distances to points: 25.94, 18.03, 47.42, 19.21, 7.28, 10.44, 19.21, 41.87, 33.30, 0.00, 13.04, 2.00, 37.20, 10.63,
1-NN:
Real Label M, KNN Label M
3-NN: (M M M)
Real Label M, KNN Label M

Data Point (175 69 28) :

Distances to points: 13.00, 31.06, 34.66, 6.40, 9.00, 23.26, 6.40, 29.00, 20.52, 13.04, 0.00, 12.08, 24.21, 3.00,
1-NN:
⋆ Real Label W, KNN Label M
3-NN: (M M W)
⋆ Real Label W, KNN Label M

Data Point (180 80 27) :

Distances to points: 25.08, 19.21, 46.10, 18.03, 5.39, 11.18, 18.03, 40.61, 32.02, 2.00, 12.08, 0.00, 36.06, 9.43,
1-NN:
Real Label M, KNN Label M
3-NN: (M M M)
Real Label M, KNN Label M

Data Point (160 50 31) :

Distances to points: 12.21, 55.22, 11.18, 18.03, 31.76, 47.17, 18.03, 5.39, 5.00, 37.20, 24.21, 36.06, 0.00, 26.63,
1-NN:
Real Label W, KNN Label W
3-NN: (W W W)
Real Label W, KNN Label W

Data Point (175 72 30) :

Distances to points: 15.81, 28.60, 36.80, 8.60, 6.00, 20.59, 8.60, 31.24, 22.67, 10.63, 3.00, 9.43, 26.63, 0.00,
1-NN:
⋆ Real Label M, KNN Label W
3-NN: (W M M)

Real Label M, KNN Label M

1-NN Error Rate: 5 / 14

3-NN Error Rate: 2 / 14

From the results we can see that for 1-NN removing the age attribute leads to the same results as when using the attribute. In the case of 3-NN, on the other hand, removing the attribute leads to one fewer misclassifications and thus seems to slightly improve classification accuracy.

Gaussian Naïve Bayes Classification

3. Using the data from Problem 2, build a Gaussian Naïve Bayes classifier for this problem. For this you have to learn Gaussian distribution parameters for each input data feature, i.e. for $p(height|W), p(height|M)$, $p(weight|W), p(weight|M), p(age|W), p(age|M)$.

   a) Learn/derive the parameters for the Gaussian Naïve Bayes Classifier and apply them to the same target as in problem $2a)$. Show your intermediate steps.

   To determine the parameters we have to determine the prior probabilities of the classes and the parameters for the Gaussian distributions.

   Prior: P(W) :

$$P(W) = \frac{\#W}{N} = \frac{7}{14} = 0.5$$

   $p(height|W)$:

$$\mu_{height,W} = \frac{\sum_{y_i=W} height_i}{\#W} = \frac{170 + 150 + 170 + 155 + 160 + 175 + 160}{7} = 162.86$$

$$\sigma^2_{height,W} = \frac{\sum_{y_i=W} (height_i - \mu_{height,W})^2}{7-1} = 82.14$$

   $p(height|M)$:

$$\mu_{height,M} = \frac{\sum_{y_i=M} height_i}{\#M} = 179.86$$

$$\sigma^2_{height,M} = \frac{\sum_{y_i=M} (height_i - \mu_{height,M})^2}{7-1} = 53.81$$

   $p(weight|W)$:

$$\mu_{weight,W} = \frac{\sum_{y_i=W} weight_i}{\#W} = 55.57$$

$$\sigma^2_{weight,W} = \frac{\sum_{y_i=W} (weight_i - \mu_{weight,W})^2}{7-1} = 78.62$$

$p(weight|M)$:

$$\mu_{weight,M} = \frac{\sum_{y_i=M} weight_i}{\#M} = 80.00$$

$$\sigma^2_{weight,M} = \frac{\sum_{y_i=M}(weight_i - \mu_{weight,M})^2}{7-1} = 103$$

$p(age|W)$:

$$\mu_{age,W} = \frac{\sum_{y_i=W} age_i}{\#W} = 30.00$$

$$\sigma^2_{age,W} = \frac{\sum_{y_i=W}(age_i - \mu_{age,W})^2}{7-1} = 2.33$$

$p(age|M)$:

$$\mu_{age,M} = \frac{\sum_{y_i=M} age_i}{\#M} = 30.14$$

$$\sigma^2_{age,M} = \frac{\sum_{y_i=M}(age_i - \mu_{age,M})^2}{7-1} = 7.14$$

Applying this to the classification of the data points in 2a) yields:

Data Point (155 40 35) :
probability for class W: 0.000145
probability for class M: 0.000000
Predicted class: W

Data Point (170 70 32) :
probability for class W: 0.000193
probability for class M: 0.000267
Predicted class: M

Data Point (175 70 35) :
probability for class W: 0.000107
probability for class M: 0.000528
Predicted class: M

Data Point (180 90 20) :
probability for class W: 0.000000
probability for class M: 0.000658
Predicted class: M

b) Implement the Gaussian Naïve Bayes Classifier for this problem.

c) Repeat the experiment in part $2c$) with the Gaussian Naïve Bayes Classifier.

For "leave one out" we can calculate the Naïve Bayes likelihoods and then classify the point, evaluating its performance.

Using all 3 attributes the distances and classification results are as follows (misclassifications are labeled with $\star$):

Data Point (170 57 32) :
probability for class W: 0.000079
probability for class M: 0.000004
Predicted class: W Real class W

Data Point (192 95 28) :
probability for class W: 0.000000
probability for class M: 0.000010
Predicted class: M Real class M

Data Point (150 45 30) :
probability for class W: 0.000046
probability for class M: 0.000000
Predicted class: W Real class W

Data Point (170 65 29) :
probability for class W: 0.000087
probability for class M: 0.000020
∗ Predicted class: W Real class M

Data Point (175 78 35) :
probability for class W: 0.000000
probability for class M: 0.000024
Predicted class: M Real class M

Data Point (185 90 32) :
probability for class W: 0.000000
probability for class M: 0.000060
Predicted class: M Real class M

Data Point (170 65 28) :
probability for class W: 0.000046
probability for class M: 0.000016
Predicted class: W Real class W

Data Point (155 48 31) :
probability for class W: 0.000100
probability for class M: 0.000000
Predicted class: W Real class W

Data Point (160 55 30) :
probability for class W: 0.000246
probability for class M: 0.000000
Predicted class: W Real class W

Data Point (182 80 30) :
probability for class W: 0.000001
probability for class M: 0.000153
Predicted class: M Real class M

Data Point (175 69 28) :
probability for class W: 0.000014
probability for class M: 0.000052
∗ Predicted class: M Real class W

Data Point (180 80 27) :
probability for class W: 0.000000
probability for class M: 0.000080
Predicted class: M Real class M

Data Point (160 50 31) :
probability for class W: 0.000163
probability for class M: 0.000000
Predicted class: W Real class W

Data Point (175 72 30) :
probability for class W: 0.000019
probability for class M: 0.000094
Predicted class: M Real class M

Ignoring the age attribute and using only the first 2, the results become:

Data Point (170 57 32) :
probability for class W: 0.000717
probability for class M: 0.000033
Predicted class: W Real class W

Data Point (192 95 28) :
probability for class W: 0.000000
probability for class M: 0.000091
Predicted class: M Real class M

Data Point (150 45 30) :
probability for class W: 0.000178
probability for class M: 0.000000
Predicted class: W Real class W

Data Point (170 65 29) :
probability for class W: 0.000412
probability for class M: 0.000145

∗ Predicted class: W Real class M

Data Point (175 78 35) :
probability for class W: 0.000016
probability for class M: 0.000842
Predicted class: M Real class M

Data Point (185 90 32) :
probability for class W: 0.000000
probability for class M: 0.000514
Predicted class: M Real class M

Data Point (170 65 28) :
probability for class W: 0.000412
probability for class M: 0.000145
Predicted class: W Real class W

Data Point (155 48 31) :
probability for class W: 0.000472
probability for class M: 0.000000
Predicted class: W Real class W

Data Point (160 55 30) :
probability for class W: 0.000940
probability for class M: 0.000001
Predicted class: W Real class W

Data Point (182 80 30) :
probability for class W: 0.000002
probability for class M: 0.001024
Predicted class: M Real class M

Data Point (175 69 28) :
probability for class W: 0.000128
probability for class M: 0.000477
∗ Predicted class: M Real class W

Data Point (180 80 27) :
probability for class W: 0.000004
probability for class M: 0.001069
Predicted class: M Real class M

Data Point (160 50 31) :
probability for class W: 0.000773
probability for class M: 0.000000

Predicted class: W Real class W

Data Point (175 72 30) :
probability for class W: 0.000073
probability for class M: 0.000629
Predicted class: M Real class M

In this case, both conditions do make 2 incorrect classifications on the same items.

d) Compare the results of the two classifiers and discuss reasons why one might perform better than the other.

In this example, the Gaussian Naïve Bayes classifier seems to be performing slightly better. The main reason here is likely that it is able to better weigh between the different dimensions of the data vector. KNN here has to treat each of the dimensions equally since it is computing Cartesian Distance while the Gaussian Naïve Bayes can vary the variance for each dimension separately, thus making different dimensions differently sensitive to variations, thus yielding better classification results.