MVA DM1: Probabilistic Graphical Model

Amin Dhaou

November 2019

Exercise 1

Let $(z_1, x_1), \dots, (z_N, x_N)$ be an i.i.d sample of size N of observations from (z_m^n, x_k^n) with n $\in\{1, \dots, N\}$, k $\in\{1, \dots, K\}$, m $\in\{1, \dots, M\}$ with $z_m^n = 1$ if z = m for the n^{th} sample and 0 otherwise, $x_k^n = 1$ if x = k for the n^{th} example and 0 otherwise. The log-likelihood of the model can be written as:

$$\begin{aligned}
L(\pi, \theta | z_i, x_i) &= \log \prod_{i=1}^{n} P(z = z_i \cap x = x_i) = \log \prod_{i=1}^{n} P(z = z_i) P(x = x_i | z = z_i) \\
&= \log \prod_{i=1}^{n} \prod_{m=1}^{M} \pi_m^{z_m^i} \prod_{i=1}^{n} \prod_{m=1}^{M} \prod_{k=1}^{K} \theta_{mk}^{z_m^i x_k^i} \\
&= \sum_{m=1}^{M} \log \pi_m \sum_{i=1}^{n} z_m^i + \sum_{m=1}^{M} \sum_{k=1}^{K} \log \theta_{mk} \sum_{i=1}^{n} z_m^i x_k^i.
\end{aligned}$$

Let $n_m = \sum_{i=1}^{n} z_m^i$ and $n_{mk} = \sum_{i=1}^{n} z_m^i x_k^i$.

n_m is the number of samples such that z = m and n_{mk} is the number of samples such that z = m and x = k.

Finally, we have :

$$L(\pi, \theta | z_i, x_i) = \sum_{m=1}^{M} \log \pi_m n_m + \sum_{m=1}^{M} \sum_{k=1}^{K} \log \theta_{mk} n_{mk}.$$

We want to maximize the log-likelihood to find π and θ. Since the two terms of this function are independent, they can be maximized separately.

- The first term which depends on π can be reformulated as an optimization problem:

$$max(\sum_{m=1}^{M} \log \pi_m n_m)$$

$$s.t \sum_{m=1}^{M} \pi_m = 1$$

Which is equivalent to the convex problem:

$$min(-\sum_{m=1}^{M} \log \pi_m n_m)$$

$$s.t \sum_{m=1}^{M} \pi_m = 1$$

The Lagrangian can be written as:

$$L(\pi, \lambda) = -\sum_{m=1}^{M} \log \pi_m n_m + \lambda(\sum_{m=1}^{M} \pi_m - 1)$$

Clearly, as $n_m > 0$ for m $\in\{1, \dots, M\}$, L is convex and this problem is a convex optimization problem. Moreover, it is trivial that there exist $\{\pi_1, \dots, \pi_m\}$ for m $\in\{1, \dots, M\}$ with $\pi_m > 0$ and $\sum_{m=1}^{M} \pi_m = 1$. So, by Slater's constraint qualification, the problem has strong duality property.

Finally, to find the minimum we have to take the derivative with respect to π_m because the Lagrangian is convex. This leads to:

$$\frac{\partial L}{\partial \pi_m} = -\frac{n_m}{\pi_m} + \lambda, m = 1, \dots, M$$

Using the constraints $\sum_{m=1}^{M} \pi_m = 1$, the solution to the problem is: $\boxed{\pi_m = \frac{n_m}{n}}$

• For the second term of the log-likelihood equation,we need to solve this optimization problem:

$$max(\sum_{m=1}^{M}\sum_{k=1}^{K} \log \theta_{mk} n_{mk})$$

$$s.t \sum_{k=1}^{K} \theta_{mk} = 1, m = 1, \dots, M$$

Similar to previously, we need compute the lagrangian:

$$L(\pi, \lambda) = \sum_{m=1}^{M}\sum_{k=1}^{K} \log \theta_{mk} n_{mk} + \sum_{m=1}^{M} \lambda_m \sum_{k=1}^{K} (\theta_{mk} - 1)$$

Then we compute its derivative to find the maximum value:

$$\frac{\partial L}{\partial \theta_{mk}} = \frac{n_{mk}}{\theta_{mk}} - \lambda_m, m = 1, \dots, M$$

Using the constraints $\sum_{k=1}^{K} \theta_{mk} = 1$, the solution to the problem is:

$$\boxed{\theta_{mk} = \frac{n_{mk}}{n_m}}$$

LDA

Let $(y_1, x_1), \dots, (y_n, x_n)$ be an i.i.d sample from (y, x).
The likelihood of this model is:

$$\begin{aligned}\ell(\mu_0, \mu_1, \Sigma, \pi | y_i, x_i) &= \prod_{i=1}^{n} P(x_i \cap y_i) = \prod_{i=1}^{n} P(y_i)P(x_i|y_i) \\ &= \frac{1}{((2\pi)^d |\Sigma|)^{n/2}} \prod_{i=1}^{n} \pi^{y_i} \exp(-\frac{1}{2}(x_i - \mu_{y_i})^T \Sigma^{-1} (x_i - \mu_{y_i}))(1-\pi)^{1-y_i}\end{aligned}$$

Lets n_1 be the number of samples where $y_i = 1$ and $n - n_1$ the number of samples where y_i= 0 , the log-likelihood of this model is:

$$\begin{aligned}L(\mu_0, \mu_1, \Sigma, \pi | y_i, x_i) = &-\frac{n}{2}\log((2\pi)^d) + n_1 \log \pi + (n - n_1)\log(1-\pi) \\ &-\frac{n}{2}\log|\Sigma| - \frac{1}{2}\left(\sum_{i=1}^{n}(x_i - \mu_{y_i})^T \Sigma^{-1}(x_i - \mu_{y_i})\right)\end{aligned}$$

This function can be maximized separately with respect to π and (μ_0, μ_1, Σ)

• $\pi \mapsto n_1 \log \pi + (n - n_1)\log(1-\pi)$ is concave and thus may be maximized simply by computing its derivative same as in Exercice 1: After computation, we find that $\boxed{\pi = \frac{n_1}{n}}$

• $(\mu_0, \mu_1, \Sigma) \mapsto n \log|\Sigma| + \left(\sum_{i=1}^{n}(x_i - \mu_{y_i})^T \Sigma^{-1}(x_i - \mu_{y_i})\right)$ is to be minimized. For fixed Σ, the function $\mu \mapsto (x - \mu)^T \Sigma^{-1}(x-\mu)$ has Hessian $\Sigma^{-1} \in S_d^{+}(\mathbb{R})$ and is then convex . By computing thee gradient of $\mu \mapsto (x-\mu)^T\Sigma^{-1}(x-\mu)$ we find $-2\Sigma^{-1}(x-\mu)$. The minimum of this convex function is atteined when the gradient is equal to 0, because Σ is invertible, this leads to: $\sum_{y_i} \mu_{y_i} = \sum_{y_i} x_i$ and finally:

$\boxed{\mu_1 = \frac{\sum_{i=1}^{n} y_i x_i}{n_1}}$ and similarly $\boxed{\mu_0 = \frac{\sum_{i=1}^{n}(1-y_i)x_i}{n - n_1}}$.

For these values of μ_0 and μ_1, we're left with minimizing

$$\Sigma \mapsto n \log|\Sigma| + \sum_{i=1}^{n}(x_i - \mu_{y_i})^T \Sigma^{-1}(x_i - \mu_{y_i}) \qquad (*)$$

By the change of variable $\Delta = \Sigma^{-1}$ this turns into

$$\Delta \mapsto -n \log|\Delta| + \left(\sum_{i=1}^{n}(x_i - \mu_{y_i})^T \Delta (x_i - \mu_{y_i})\right)$$

We showed in lecture 2 that $\Delta \mapsto \log|\Delta|$ is concave with gradient Δ^{-1} and since

$$\begin{aligned}(x-\mu)^T\Delta(x-\mu) &= ((x-\mu)^T\Delta(x-\mu)) = (\Delta(x-\mu)(x-\mu)^T)) \\ &= \langle \Delta, (x-\mu)(x-\mu)^T\rangle\end{aligned}$$

Then, the gradient of $\Delta \mapsto (x-\mu)^T\Delta(x-\mu)$ is $(x-\mu)(x-\mu)^T$. Thus, the minimum of the previous function verify $-n\Delta^{-1} + \left(\sum_{i=1}^{n}(x_i - \mu_{y_i})(x_i - \mu_{y_i})^T\right) = 0$, hence

$$\Delta^{-1} = \frac{1}{n}\left(\sum_{i=1}^{n}(x_i - \mu_{y_i})(x_i - \mu_{y_i})^T\right)$$

Finally, this value of Δ^{-1} minimizes $(*)$, so by equivalence we have:

$$\boxed{\Sigma = \frac{n-n_1}{n}\sum_{i=1}^{n}(1-y_i)(x_i-\mu_0)(x_i-\mu_0)^T + \frac{n_1}{n}\sum_{i=1}^{n}y_i(x_i-\mu_1)(x_i-\mu_1)^T} = \frac{n-n_1}{n}\Sigma_0 + \frac{n_1}{n}\Sigma_1$$

with $\Sigma_0 = \sum_{i=1}^{n}(1-y_i)(x_i-\mu_0)(x_i-\mu_0)^T$ and $\Sigma_1 = \sum_{i=1}^{n} y_i(x_i-\mu_1)(x_i-\mu_1)^T$

- By Bayes' theorem,

$$\begin{aligned}P(Y=1|X) &= \frac{P(X|Y=1)P(Y=1)}{P(X|Y=1)P(Y=1) + P(X|Y=0)P(Y=0)} \\ &= \frac{\frac{\pi}{\sqrt{(2\pi)^d|\Sigma|}}\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)}{\frac{\pi}{\sqrt{(2\pi)^d|\Sigma|}}\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right) + \frac{1-\pi}{\sqrt{(2\pi)^d|\Sigma|}}\exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right)} \\ &= \sigma\left(x^T\Sigma^{-1}(\mu_1-\mu_0) + \frac{1}{2}(\mu_1+\mu_0)^T\Sigma^{-1}(\mu_0-\mu_1) + \log\frac{\pi}{1-\pi}\right)\end{aligned}$$

Which is equivalent to the logistic regression form $\sigma\left(\alpha^T x + \beta\right)$ with $\alpha = \Sigma^{-1}(\mu_1-\mu_0)$ and $\beta = \frac{1}{2}(\mu_1+\mu_0)^T\Sigma^{-1}(\mu_0-\mu_1) + \log\frac{\pi}{1-\pi}$

QDA

- Similarly to the LDA model: Let $(y_1, x_1), \dots, (y_n, x_n)$ be an i.i.d sample from (y, x). The likelihood of the model is

$$\ell(\mu_0, \mu_1, \Sigma_0, \Sigma_1, \pi | y_i, x_i) = \frac{1}{(2\pi)^{nd/2}|\Sigma_0|^{n_0/2}|\Sigma_1|^{n_1/2}} \quad \prod_{i=1}^{n}\pi^{y_i}\exp(-\frac{1}{2}(x_i-\mu_{y_i})^T\Sigma_{y_i}^{-1}(x_i-\mu_{y_i}))(1-\pi)^{1-y_i}$$

- The log-likelihood consequently writes as

$$\begin{aligned}\ell(\mu_0, \mu_1, \Sigma_0, \Sigma_1, \pi | y_i, x_i) = -\frac{n}{2}\log((2\pi)^d) + n_1\log\pi + (n-n_1)\log(1-\pi) \\ -\frac{n_1}{2}\log|\Sigma_1| - \frac{n_0}{2}\log|\Sigma_0| - \frac{1}{2}\left(\sum_{i=1}^{n}(x_i-\mu_{y_i})^T\Sigma_{y_i}^{-1}(x_i-\mu_{y_i})\right)\end{aligned}$$

which may be maximized separately in π and $(\mu_0, \mu_1, \Sigma_0, \Sigma_1)$.

- Exactly the same steps as in LDA may be done in order to find:

$$\begin{cases}\pi = \frac{n_1}{n} \\ \mu_1 = \frac{1}{n_1}\sum_{i=1}^{n} y_i x_i \\ \mu_0 = \frac{1}{n-n_1}\sum_{i=1}^{n}(1-y_i)x_i \\ \Sigma_1 = \frac{1}{n_1}\sum_{i=1}^{n} y_i(x_i-\mu_1)(x_i-\mu_1)^T \\ \Sigma_0 = \frac{1}{n-n_1}\sum_{i=1}^{n}(1-y_i)(x_i-\mu_0)(x_i-\mu_0)^T\end{cases}$$

- Similar to previously by Bayes' theorem,

$$\begin{aligned} P(Y=1|X) &= \frac{P(X|Y=1)P(Y=1)}{P(X|Y=1)P(Y=1)+P(X|Y=0)P(Y=0)} \\ &= \frac{\frac{\pi}{\sqrt{(2\pi)^d|\Sigma_1|}}\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right)}{\frac{\pi}{\sqrt{(2\pi)^d|\Sigma_1|}}\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right)+\frac{1-\pi}{\sqrt{(2\pi)^d|\Sigma_0|}}\exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0)\right)} \\ &= \sigma\left(x^TQx+\beta^Tx+\alpha\right) \end{aligned}$$

With :

$$\begin{cases} Q = \frac{1}{2}(\Sigma_0^{-1}-\Sigma_1^{-1}) \\ \beta = \Sigma_1^{-1}\mu_1 - \Sigma_0^{-1}\mu_0 \\ \alpha = \frac{1}{2}(\mu_0^T\Sigma_0^{-1}\mu_0 - \mu_1^T\Sigma_1^{-1}\mu_1 + \log\frac{\pi}{1-\pi} + \frac{1}{2}\log\frac{det(\Sigma_0)}{det(\Sigma_1)}\Big) \end{cases}$$

LEARNT PARAMETERS

LDA

	α	β
Train set A	[1.95, -6.32]	32.24
Train set B	[1.68 -3.03]	7.97
Train set C	[0.30 -2.86]	20.51

Table 1: Parameters learnt with the LDA classification on the different Datasets

Logistic Regression

	ω	β
Train set A	[7.58, -9.41]	0.65
Train set B	[17.42, -20.54]	1.63
Train set C	[7.27, -10.18]	1.75

Table 2: Parameters learnt with the logistic regression classification on the different Datasets

Linear Regression

	ω	β
Train set A	[0.0037,-0.1691]	1.8893
Train set B	[0.0656] [-0.1513]	1.1127
Train set C	[-0.0098,-0.1572]	1.8977

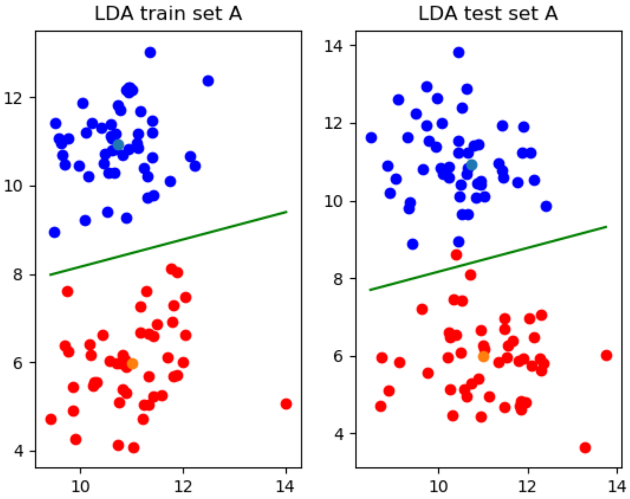
Table 3: Parameters learnt with the linear regression classification on the different Datasets

Linear Regression

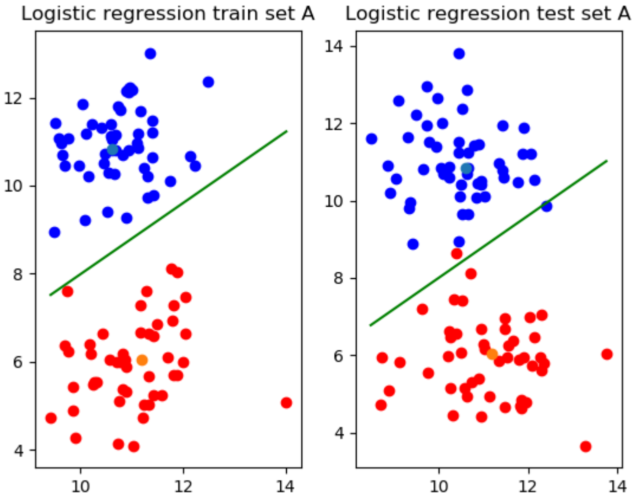
	Q	ω	β
Train set A	[[0.37, -0.02], [-0.02, 0.17]]	[-5.84,-8.90]	86.48
Train set B	[[0.39,0.08], [0.08,0.23]]	[-8.55,-9.33]	95.43
Train set C	[[-0.02,0.24], [0.24,-0.09]]	[-2.93,-7.02]	54.85

Table 4: Parameters learnt with the QDA classification on the different Datasets

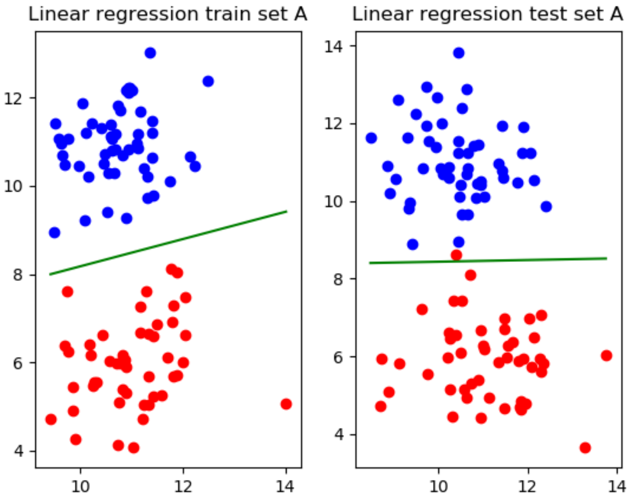
Dataset A



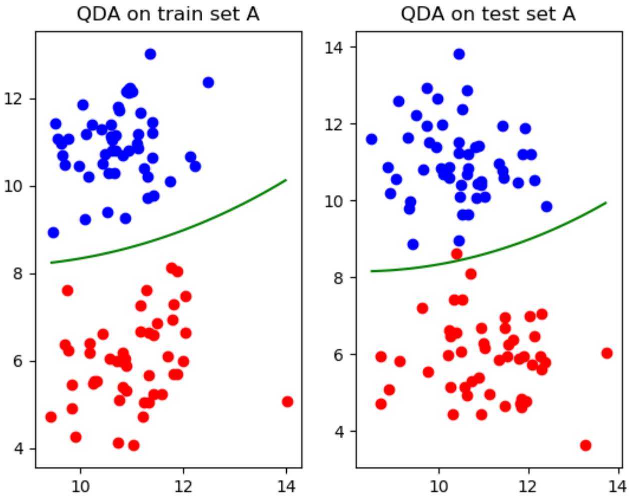
LDA



Logistic regression



Linear regression



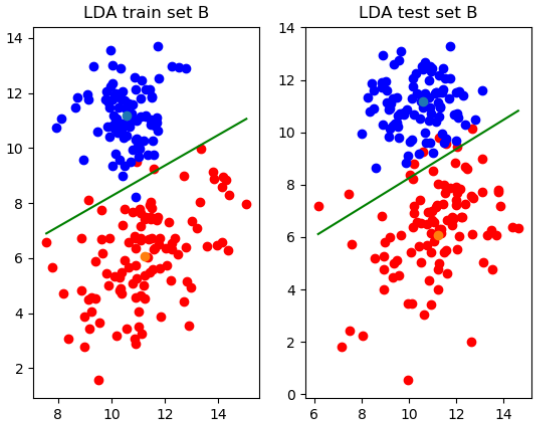
QDA

	Train	Test
LDA	0.00	1.01
Logistic	0.00	2.02
Linear	0.00	1.01
QDA	0.00	1.01

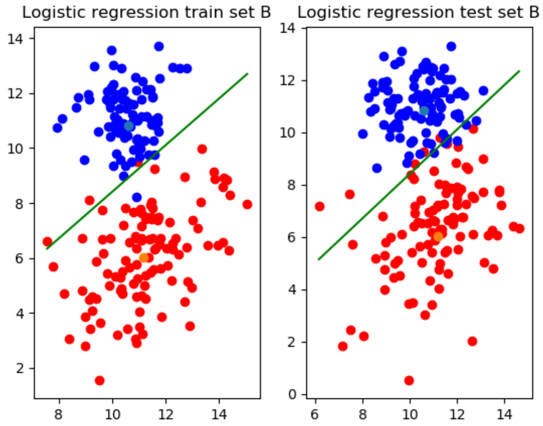
Table 5: Missclassification rates (%) on the training and test sets for Dataset A

- The training and testing datasets A are linearly separable. All four methods separate and classify perfectly on the training set. We see that the LDA, the linear regression and the QDA perform equally on the train set and on the test set whereas the logistic regression has a worse score for the test set. The logistic regression seems to overfit. That may be due to the method used for the gradient ascent which may be to simple (first order).

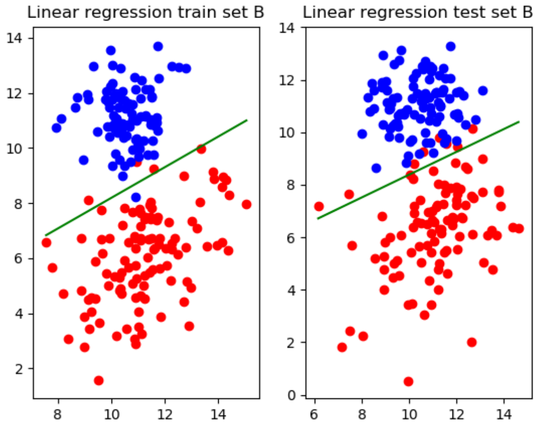
Dataset B



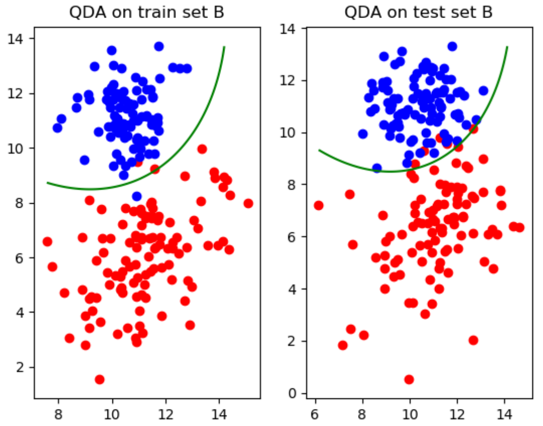
LDA



Logistic regression



Linear regression



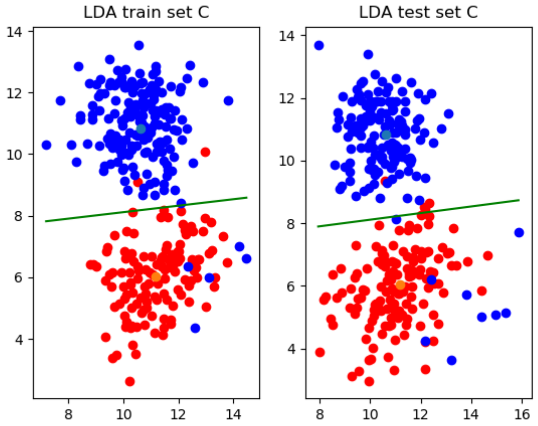
QDA

	Train	Test
LDA	2.01	5.02
Logistic	2.01	5.53
Linear	2.01	4.02
QDA	1.51	2.51

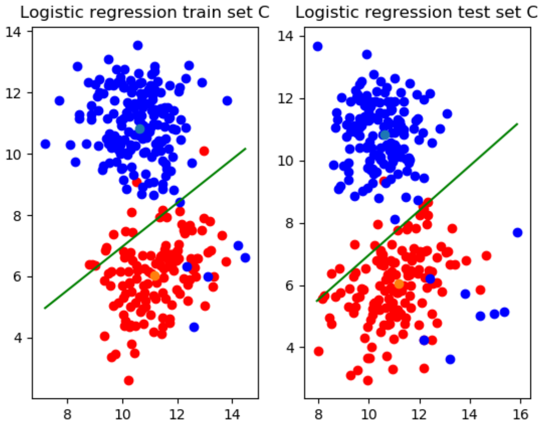
Table 6: Missclassification rates (%) on the training and test sets for Dataset B

• The training and testing set are not linearly separable. All four methods separate the data correctly with some mistakes. LDA, Logistic regression and linear regression have the performance on the training dataset, whereas QDA performs better on both training and testing datasets.
• The data seems to be generated with two Gaussian and with two covariance matrix, that's why QDA performs better. Indeed, this method take into account the different densities from the two clusters by having two different covariance matrices. Moreover, the QDA methods separates better because it has a polynomial separation which can fit the data compared to the other linear models .

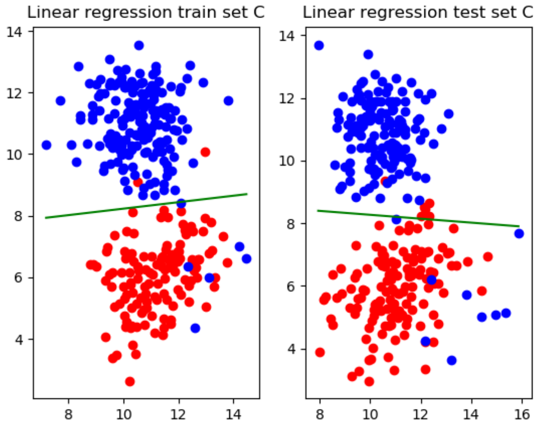
Dataset C



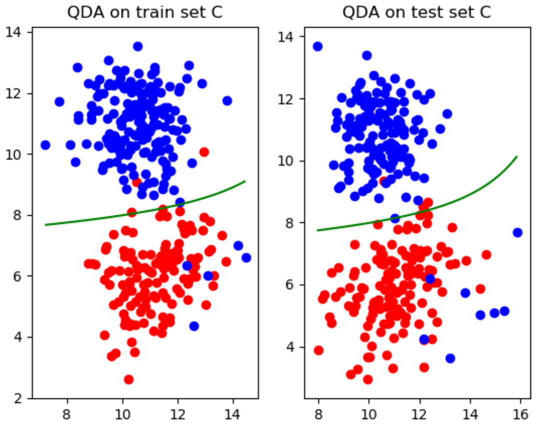
LDA



Logistic regression



Linear regression



QDA

	Train	Test
LDA	2.68	3.68
Logistic	6.35	5.35
Linear	2.68	5.02
QDA	2.68	4.35

Table 7: Missclassification rates (%) on the training and test sets for Dataset C

- In this Dataset, the two classes are not linearly separable and even with a polynomial seperation. All LDA, Linear regression and QDA performs quite well on the training set and only LDA
- Logistic regression performs worse here, that's probably due to the fact that it's a discriminative model which try to model directly $p(y|x)$.
- In all the datasets, we see that in most of the cases the training error is smaller than the testing one. In The opposite case, this could be due to an overfitting model.