École Polytechnique de Montréal

Département de génie informatique et génie logiciel

Rapport Assignment 2

MACHINE LEARNING

Antoine Pichon, 2489005

Table des matières

1 K-Nearest Neighbors (KNN)

1.1 Algorithm de KNN naif

Voici les echantillons de réponses que nous avons obtenus en faisant varier k et la distance. (A noter que mon ordianteur n'ayant que 16 Go de Ram, les testes sont fait sur des échantillons réduits mais suffisament grands pour être significatifs : le code complet a tout de meme été testé sur une machine à distance pour verrifier que celui ci fonctionnait.) :

TABLE 1 – Résultats de validation avec la distance euclidienne

k	**Précision validation (%)**
1	91.4
2	91.4
3	92.8
4	**93.0**
5	93.0
10	91.4
20	90.2

TABLE 2 – Résultats de validation avec la distance cosine

k	**Précision validation (%)**
1	91.6
2	92.2
3	**93.8**
4	93.6
5	92.8
10	90.6
20	91.0

Dans le cas des deux tableaux, les meilleurs résultats sont obtenus pour des k 3, 4 ou 5. Cela s'explique car pour k = 1, notre modèle est très sensible au bruit et aux valeurs aberrantes. En augmentant k, nous réduisons cette sensibilité, ce qui améliore la précision jusqu'à un certain point. Pour les k trop grands, les voisins éloignées peuvent etre dans les plus proches voisins, mais n'ont pas de rapport avec le point à classer. Ces points peuvent donc brouiller l'analyse.

1.2 Algorithme des KNN structuré avec KD-Tree

1.2.1 construction de l'arbre

Nous avons implémenté la construction de l'arbre dans la classe KDTree. Faisons une brève analyse de la complexité :
Complexité de temporelle : Le tri des données à chaque niveau nécessite $\mathcal{O}(n \log n)$ et il y a $\mathcal{O}(\log n)$ niveaux, donnant une complexité totale de $\mathcal{O}(n \log^2 n)$.
Complexité Spatiale : Chaque nœud de l'arbre nécessite un espace constant, donc l'espace requis est $\mathcal{O}(n)$ pour stocker les nœuds.

1.2.2 Classification 1NN

Puis, nous avons implémenté la recherche du plus proche voisin en partant de cette structure d'arbre. Voici ce que nous avons comme complexité :
Complexité temporelle : La recherche du plus proche voisin présente une complexité moyenne de l'ordre d'un $\mathcal{O}(\log n)$. Cela est dû au fait que nous ne travaillons qu'avec des arbres binaires équilibrés.
Il est essentiel de stocker la dimension de division car nous ne savons comparer des nombres par une relation d'ordre total que dans $\mathbb{R}$. En cela, nous permettons donc de savoir par quelle dimension du point nous allons comparer et donc de quel coté de l'arbre nous allons continuer la recherche.

1.3 Question Bonus

Pour obtenir les k plus proches voisins plutôt que le plus proche voisin, nous devons retenir les k points les plus proches plutôt que le point le plus proche.
On pourrait réaliser cela avec une liste triée contenant les k plus proches voisins courants. Dans ce cas, nous n'avons besoin que de comparer le point à l'extrémité qui est la plus éloignée. Si le point que l'on étudie est plus proche que celui-ci, on retire l'ancien point le plus éloigné de la liste, on ajoute le point étudié, puis on retrie la liste de manière intelligente (car elle sera normalement bien triée sauf pour un point).
On pourrait imaginer le pseudocode suivant :

```
function kNN_kdtree(node, query_point, k, voisins):
    if node == NULL:
        return voisins

    dist = distance(node.data, query_point)

    if len(voisins) < k:
        voisins.append((node, dist))
        voisins.sort()
    elif dist < voisins[-1].distance:
        voisins[-1] = (node, dist)
        // Ici on pourrait optimiser le tri (insertion triée)
        voisins.sort()

    if est_feuille(node):
        return voisins

    // Choisir la branche selon la dimension de division
    if query_point[node.dim] < node.data[node.dim]:
        prio, secc = node.left, node.right
    else:
        prio, secc = node.right, node.left

    voisins = kNN_kdtree(prio, query, k, voisins)

    // érifier l'autre branche si nécessaire
    dist_plan = abs(query_point[node.dim] - node.data[node.dim])
```

```
    if len(voisins) < k or dist_plan < voisins[-1].distance:
        voisins = kNN_kdtree(secc, query, k, voisins)

    return voisins
```

2 Theory : a special case of Stone's theorem

2.1 Question 4

2.1.1 Question 4.a

Montrons que $R(G) = E[\eta(X)1_{G(X)=0} + (1-\eta(X))1_{G(X)=1}]$.

Démonstration. On peut premièrement écrire la décomposition suivante d'erreur (car $g \in \{0,1\}$) :

$$1_{G(X)\neq g} = 1_{G(X)=0,g=1} + 1_{G(X)=1,g=0}$$

On a donc en passant le tout à l'espérance (fonction linéaire) :

$$\begin{aligned} E[1_{G(X)\neq g}] = R(G) &= E[1_{G(X)=0,g=1}] + E[1_{G(X)=1,g=0}] && (1)\\ &= P(G(X)=0, g=1) + P(G(X)=1, g=0) && (2)\end{aligned}$$

De plus, on sait que pour toute variable aléatoire A et B, on a $E(A) = E(E(A|B))$ (formule de l'espérance totale). D'où, en conditionnant sur X, on obtient :

$$\begin{aligned} P(G(X)=0, g=1) &= E[E[1_{G(X)=0,g=1}|X]] && (3)\\ &= E[1_{G(X)=0} \cdot E[1_{g=1}|X]] \quad (\text{car } G(X) \text{ ne dépend que de } X) && (4)\\ &= E[1_{G(X)=0} \cdot P(g=1|X)] && (5)\\ &= E[1_{G(X)=0} \cdot \eta(X)] \quad (\text{par définition de } \eta) && (6)\end{aligned}$$

De manière similaire :

$$\begin{aligned} P(G(X)=1, g=0) &= E[1_{G(X)=1} \cdot P(g=0|X)] && (7)\\ &= E[1_{G(X)=1} \cdot (1-\eta(X))] \quad (\text{car } P(g=0|X) = 1 - P(g=1|X)) && (8)\end{aligned}$$

On obtient donc finalement :

$$R(G) = E[\eta(X)1_{G(X)=0} + (1-\eta(X))1_{G(X)=1}] \tag{9}$$

□

2.1.2 Question 4.b (Classifieur de Bayes)

Dans cette question, on nous présente le classifieur de Bayes G^* défini par :

$$G^*(X) = \begin{cases} 1 & \text{si } \eta(X) > 1/2 \\ 0 & \text{sinon} \end{cases}$$

Montrons que $R^* = E[\min(\eta(X), 1-\eta(X))]$ et que pour tout classifieur G, on a $R(G) - R^* = E[|2\eta(X) - 1|1_{G(X)\neq G^*(X)}]$.

Démonstration. En utilisant le résultat démontré ci-dessus, on a :

$$R(G^*) = E[\eta(X)1_{G^*(X)=0} + (1-\eta(X))1_{G^*(X)=1}] \tag{10}$$
$$= E[\eta(X)1_{\eta(X)\leq 1/2} + (1-\eta(X))1_{\eta(X)>1/2}] \tag{11}$$

Deplus, on a que :
- Si $\eta(X) \leq 1/2$, alors $\eta(X) \leq 1-\eta(X)$, donc $\min(\eta(X), 1-\eta(X)) = \eta(X)$
- Si $\eta(X) > 1/2$, alors $\eta(X) > 1-\eta(X)$, donc $\min(\eta(X), 1-\eta(X)) = 1-\eta(X)$

d'ou le premier résultat :

$$R(G^*) = E[\min(\eta(X), 1-\eta(X))] \tag{12}$$

□

On veut maintenant montrer que $R(G) - R^* = E[|2\eta(X)-1|1_{G(X)\neq G^*(X)}]$.

Démonstration. En utilisant le résultat démontré à la question (a), on a :

$$R(G) - R(G^*) = E[\eta(X)1_{G(X)=0} + (1-\eta(X))1_{G(X)=1}] - E[\eta(X)1_{G^*(X)=0} + (1-\eta(X))1_{G^*(X)=1}] \tag{13}$$
$$= E[\eta(X)(1_{G(X)=0} - 1_{G^*(X)=0}) + (1-\eta(X))(1_{G(X)=1} - 1_{G^*(X)=1})] \tag{14}$$
$$= E[(2\eta(X)-1)(1_{G(X)=0} - 1_{G^*(X)=0})] \quad (\text{car } 1_{G(X)=1} = 1 - 1_{G(X)=0}) \tag{15}$$
$$= E[|2\eta(X)-1| \cdot \text{sgn}(2\eta(X)-1)(1_{G(X)=0} - 1_{G^*(X)=0})] \tag{16}$$

On remarque que $\text{sgn}(2\eta(X)-1)$ est positif si $\eta(X) > 1/2$ et négatif sinon. De plus, $G^*(X)$ est défini de telle sorte que $G^*(X) = 1$ si $\eta(X) > 1/2$ et $G^*(X) = 0$ sinon. Donc, on a :
- Si $G(X) = G^*(X)$, alors $(1_{G(X)=0} - 1_{G^*(X)=0}) = 0$
- Si $G(X) \neq G^*(X)$, alors $(1_{G(X)=0} - 1_{G^*(X)=0}) = \pm 1$, et la valeur dépend de si $G^*(X)$ est 0 ou 1.

Ainsi, on peut écrire :

$$R(G) - R(G^*) = E[|2\eta(X)-1| \cdot 1_{G(X)\neq G^*(X)}] \tag{17}$$

D'ou le résultat □

Cela nous permet de dire que le classifieur de Bayes minimise le risque de classification, car pour tout classifieur G, on a $R(G) - R^* \geq 0$, donc $R(G) \geq R^*$.

2.1.3 Question 4.c

Montrons que $\forall n \in \mathbb{N}, R(\hat{G}_n) - R^* \leq 2E[|\eta(X) - \eta_n(X)|]$.

Démonstration. En utilisant le résultat démontré à la question (b), on a :

$$R(\hat{G}_n) - R^* = E[|2\eta(X)-1|1_{\hat{G}_n(X)\neq G^*(X)}] \tag{18}$$

En partant du postulat que $\hat{G}_n(X) \neq G^*(X)$ faisons une disjonction de cas :

Premier cas : $\eta(X) \leq 1/2 < \hat{\eta}_n(X)$

$$|2\eta(X) - 1| = 1 - 2\eta(X) = 2(1/2 - \eta(X)) \tag{19}$$
$$\leq 2(\hat{\eta}_n(X) - \eta(X)) = 2|\eta(X) - \hat{\eta}_n(X)| \tag{20}$$

Deuxième cas : $\eta(X) > 1/2 \geq \hat{\eta}_n(X)$

$$|2\eta(X) - 1| = 2\eta(X) - 1 = 2(\eta(X) - 1/2) \tag{21}$$
$$\leq 2(\eta(X) - \hat{\eta}_n(X)) = 2|\eta(X) - \hat{\eta}_n(X)| \tag{22}$$

On remarque ainsi que dans tout les cas, nous avons le résultat suivant $|2\eta(X) - 1| \leq 2|\eta(X) - \hat{\eta}_n(X)|$ quand $\hat{G}_n(X) \neq G^*(X)$.
Ainsi :

$$R(\hat{G}_n) - R^* = E[|2\eta(X) - 1|1_{\hat{G}_n(X) \neq G^*(X)}] \tag{23}$$
$$\leq E[2|\eta(X) - \hat{\eta}_n(X)|1_{\hat{G}_n(X) \neq G^*(X)}] \tag{24}$$
$$\leq 2E[|\eta(X) - \hat{\eta}_n(X)|] \tag{25}$$

□

D'où le résultat

2.2 Question 5

2.2.1 Question 5.a

Montrons que $E[(\eta(x) - \hat{\eta}_n(x))^2] \leq 2E[(\eta(x) - \tilde{\eta}_n(x))^2] + 2E[(\tilde{\eta}_n(x) - \hat{\eta}_n(x))^2]$.

Démonstration. Premièrement, on peut écrire la décomposition suivante :

$$\eta(x) - \hat{\eta}_n(x) = [\eta(x) - \tilde{\eta}_n(x)] + [\tilde{\eta}_n(x) - \hat{\eta}_n(x)] \tag{26}$$

Appliquons ensuite l'inégalité de $((a+b)^2 \leq 2a^2 + 2b^2)$:

$$[\eta(x) - \hat{\eta}_n(x)]^2 \leq 2[\eta(x) - \tilde{\eta}_n(x)]^2 + 2[\tilde{\eta}_n(x) - \hat{\eta}_n(x)]^2 \tag{27}$$

En prenant l'espérance des deux côtés :

$$E[(\eta(x) - \hat{\eta}_n(x))^2] \leq 2E[(\eta(x) - \tilde{\eta}_n(x))^2] + 2E[(\tilde{\eta}_n(x) - \hat{\eta}_n(x))^2] \tag{28}$$

□

2.2.2 Question 5.b

Supposons maintenant que $k_n \to \infty$ et $k_n/n \to 0$. Et posons $a > 0$.

$$E\left[\sum_{i=1}^{n} w_{n,i}(X) 1_{\|X_i - X\| \geq a}\right] \to 0$$

En écrivant $(\eta(x) - \tilde{\eta}_n(X))^2$ comme une somme unique et en utilisant l'inégalité de Jensen, la continuité absolue de η et l'hypothèse ci-dessus, on obtient pour un certain $\delta > 0$:

$$E[(\eta(x) - \tilde{\eta}_n(X))^2] \leq \epsilon + E\left[\sum_{i=1}^{n} w_{n,i}(X) 1_{\|X_i - X\| \geq \delta}\right]$$

Ainsi, prenons $a = \delta$: On a alors $E\left[\sum_{i=1}^{n} w_{n,i}(X)1_{\|X_i - X\| \geq \delta}\right] \to 0$ quand $n \to \infty$. Ainsi, puisque l'inégalité est vrai pour tout $\epsilon > 0$. Elle est donc vrai en particulier pour $\epsilon_n = \frac{1}{n}$ On a donc n'inégalité suivante vérifiée pour tout n :

$$E[(\eta(x) - \tilde{\eta}_n(X))^2] \leq \frac{1}{n} + E\left[\sum_{i=1}^{n} w_{n,i}(X)1_{\|X_i - X\| \geq \delta = a}\right]$$

Cela montre que : $E[(\eta(x) - \tilde{\eta}_n(X))^2] \to 0$ quand $n \to \infty$.

2.2.3 Question 5.c

Montrons premierement que :

$$E[(\tilde{\eta}(x) - \hat{\eta}_n(X))^2] = E\left[\sum_{i=1}^{n} w_{n,i}^2(X)(g_i - \eta(X_i))^2\right]$$

Démonstration. Premièrement, on a par définiton :

$$\tilde{\eta}_n(x) = \sum_{i=1}^{n} w_{n,i}(x)\eta(X_i) \tag{29}$$

$$\hat{\eta}_n(x) = \sum_{i=1}^{n} w_{n,i}(x)g_i \tag{30}$$

Donc :

$$\tilde{\eta}_n(x) - \hat{\eta}_n(x) = \sum_{i=1}^{n} w_{n,i}(x)\eta(X_i) - \sum_{i=1}^{n} w_{n,i}(x)g_i \tag{31}$$

$$= \sum_{i=1}^{n} w_{n,i}(x)[\eta(X_i) - g_i] \tag{32}$$

Il en vient que :

$$[\tilde{\eta}_n(x) - \hat{\eta}_n(x)]^2 = \left[\sum_{i=1}^{n} w_{n,i}(x)[\eta(X_i) - g_i]\right]^2 \tag{33}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} w_{n,i}(x)w_{n,j}(x)[\eta(X_i) - g_i][\eta(X_j) - g_j] \tag{34}$$

Soit :

$$E[(\tilde{\eta}_n(x) - \hat{\eta}_n(x))^2] = E\left[\sum_{i=1}^{n}\sum_{j=1}^{n} w_{n,i}(x)w_{n,j}(x)[\eta(X_i) - g_i][\eta(X_j) - g_j]\right] \tag{35}$$

On a donc en séparant les sommes :

$$E[(\tilde{\eta}_n(x) - \hat{\eta}_n(x))^2] \tag{36}$$

$$= E\left[\sum_{i=1}^{n} w_{n,i}^2(x)(g_i - \eta(X_i))^2\right] + E\left[\sum_{i \neq j} w_{n,i}(x)w_{n,j}(x)(g_i - \eta(X_i))(g_j - \eta(X_j))\right] \tag{37}$$

Or, d'après la formule de l'espérance totale on a :

$$E\left[\sum_{i\neq j} w_{n,i}(x)w_{n,j}(x)(g_i-\eta(X_i))(g_j-\eta(X_j))\right] \tag{38}$$

$$= E\left[E\left[\sum_{i\neq j} w_{n,i}(x)w_{n,j}(x)(g_i-\eta(X_i))(g_j-\eta(X_j))\Big|X_1,\ldots,X_n\right]\right] \tag{39}$$

Soit (car les $w_{n,i}(x)$ sont connus une fois les X_k fixés) :

$$= E\left[\sum_{i\neq j} w_{n,i}(x)w_{n,j}(x)E\left[(g_i-\eta(X_i))(g_j-\eta(X_j))\big|X_1,\ldots,X_n\right]\right] \tag{40}$$

Par indépendance conditionnelle (l'indépendance des g_i et g_j est a montrée mais on l'admet ici) :

$$E[(g_i-\eta(X_i))(g_j-\eta(X_j))|X_1,\ldots,X_n] = E[g_i-\eta(X_i)|X_i]\cdot E[g_j-\eta(X_j)|X_j] \tag{41}$$

Or, on a $\forall i$, :

$$E[g_i-\eta(X_i)|X_i] = E[g_i|X_i]-\eta(X_i) \tag{42}$$

$$= \eta(X_i)-\eta(X_i) = 0 \tag{43}$$

Donc :

$$E\left[\sum_{i\neq j} w_{n,i}(x)w_{n,j}(x)(g_i-\eta(X_i))(g_j-\eta(X_j))\right] = 0 \tag{44}$$

D'où le résultat :

$$E[(\tilde{\eta}_n(x)-\hat{\eta}_n(x))^2] = E\left[\sum_{i=1}^{n} w_{n,i}^2(x)(g_i-\eta(X_i))^2\right] \tag{45}$$

□

Montrons maintenant que $E[(\tilde{\eta}(x)-\hat{\eta}_n(X))^2] \leq E[\max_{1\leq i\leq n} w_{n,i}(X)]$.

Démonstration. On a précédamment montré que :

$$E[(\tilde{\eta}_n(x)-\hat{\eta}_n(x))^2] = E\left[\sum_{i=1}^{n} w_{n,i}^2(x)(g_i-\eta(X_i))^2\right] \tag{46}$$

On a sachant que $(g_i-\eta(X_i))^2 \leq 1$ (car $g_i \in \{0,1\}$ et $\eta(X_i)\in[0,1]$) :

$$E[(\tilde{\eta}_n(x)-\hat{\eta}_n(x))^2] \leq E\left[\sum_{i=1}^{n} w_{n,i}^2(x)\right] \tag{47}$$

De plus, on remarque que $w_{n,i}^2(x) \leq w_{n,i}(x)\max_{1\leq j\leq n} w_{n,j}(x)$, donc :

$$\sum_{i=1}^{n} w_{n,i}^2(x) \leq \max_{1\leq j\leq n} w_{n,j}(x)\sum_{i=1}^{n} w_{n,i}(x) \tag{48}$$

$$= \max_{1\leq j\leq n} w_{n,j}(x) \tag{49}$$

$$\text{(car la définition de } w_{n,i}(x) \text{ implique que } \sum_{i=1}^{n} w_{n,i}(x) = 1) \tag{50}$$

□

Dans notre cas, on a $\max_{1\leq i\leq n} w_{n,i}(X) = 1/k$.
Par conséquent :

$$E[(\tilde{\eta}_n(x) - \hat{\eta}_n(X))^2] \leq \frac{1}{k}$$

Puisque par hypothèse $k \to \infty$, on a $\frac{1}{k} \to 0$, donc :

$$E[(\tilde{\eta}_n(x) - \hat{\eta}_n(X))^2] \to 0 \text{ quand } n \to \infty$$

Ce terme mesure l'erreur causé par le bruit sur les labels, soit la différence en moyenne quadratique entre :

— L'estimateur oracle $\tilde{\eta}_n$ (qui utilise les vraies probabilités $\eta(X_i)$)
— L'estimateur réel $\hat{\eta}_n$ (qui utilise les labels bruités g_i)

Sa convergence vers 0 signifie que l'effet du bruit diminue quand le nombre de voisins augmente ce qui semble être un résultat plutôt cohérent.

2.3 Question Bonus

3 Régression Logistique

3.1 Question 1

3.1.1 Question 1.a

La preuve est faite de façon manuscrite afin de gagner du temps.

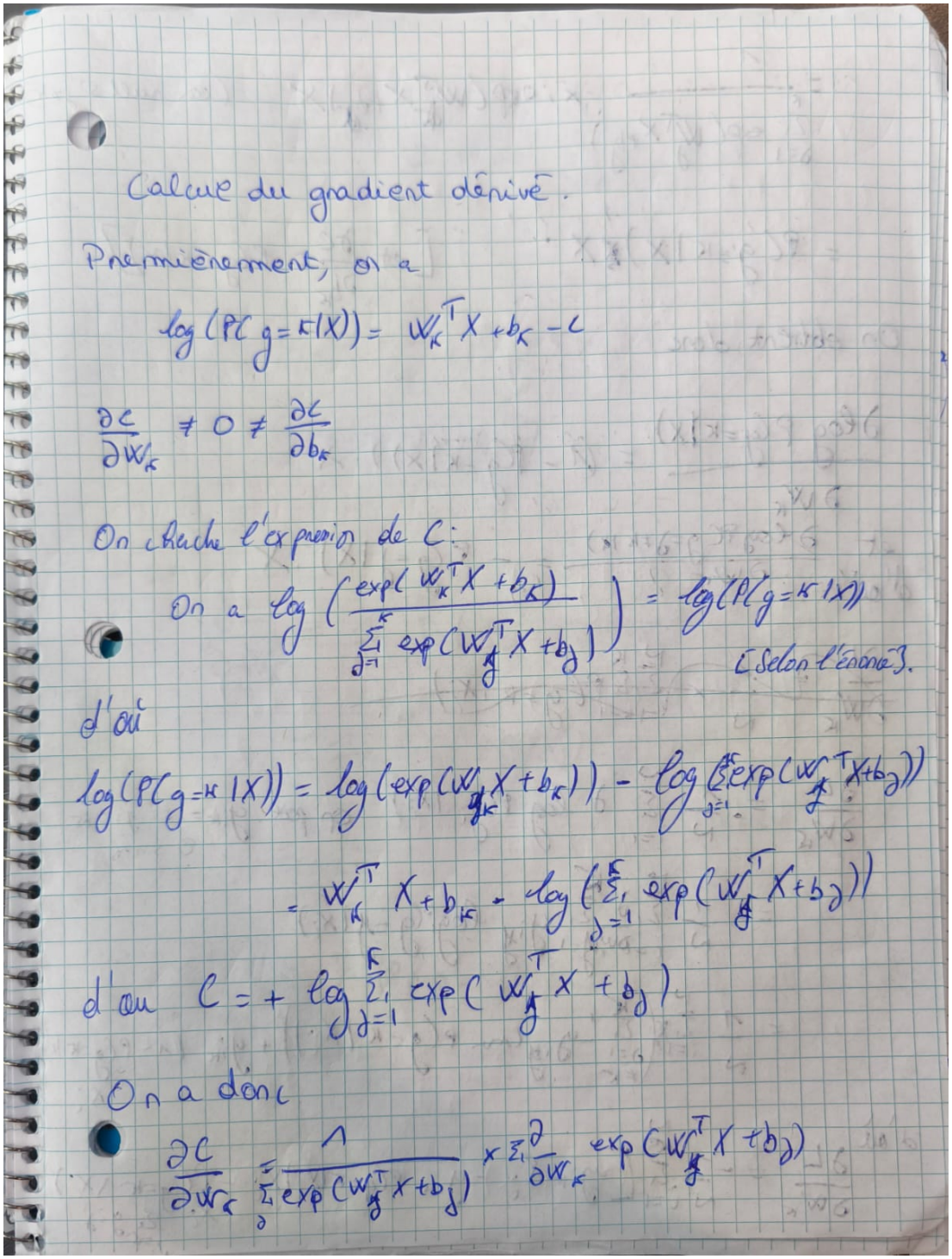

FIGURE 1 – Démonstration - Partie 1

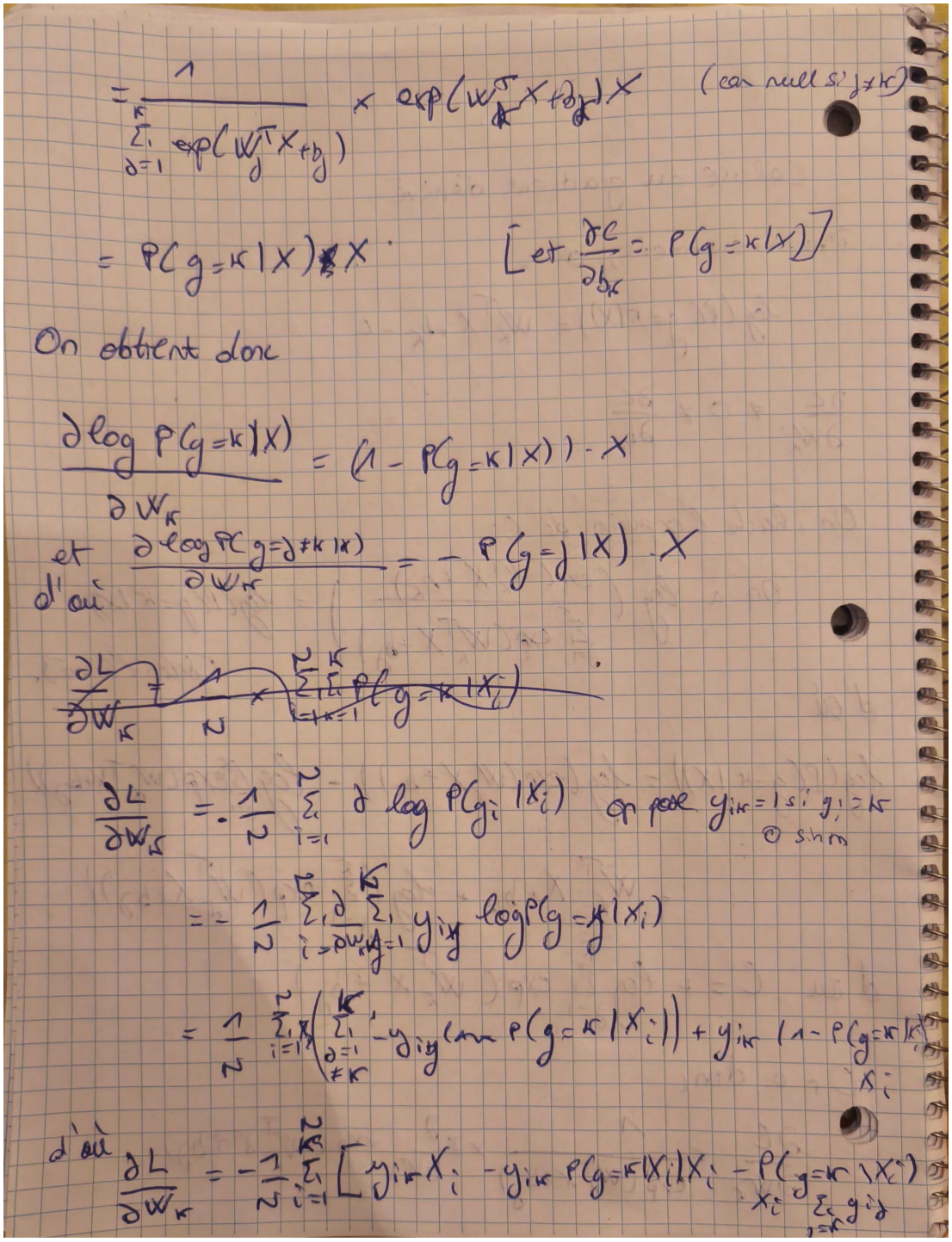

FIGURE 2 – Démonstration - Partie 2

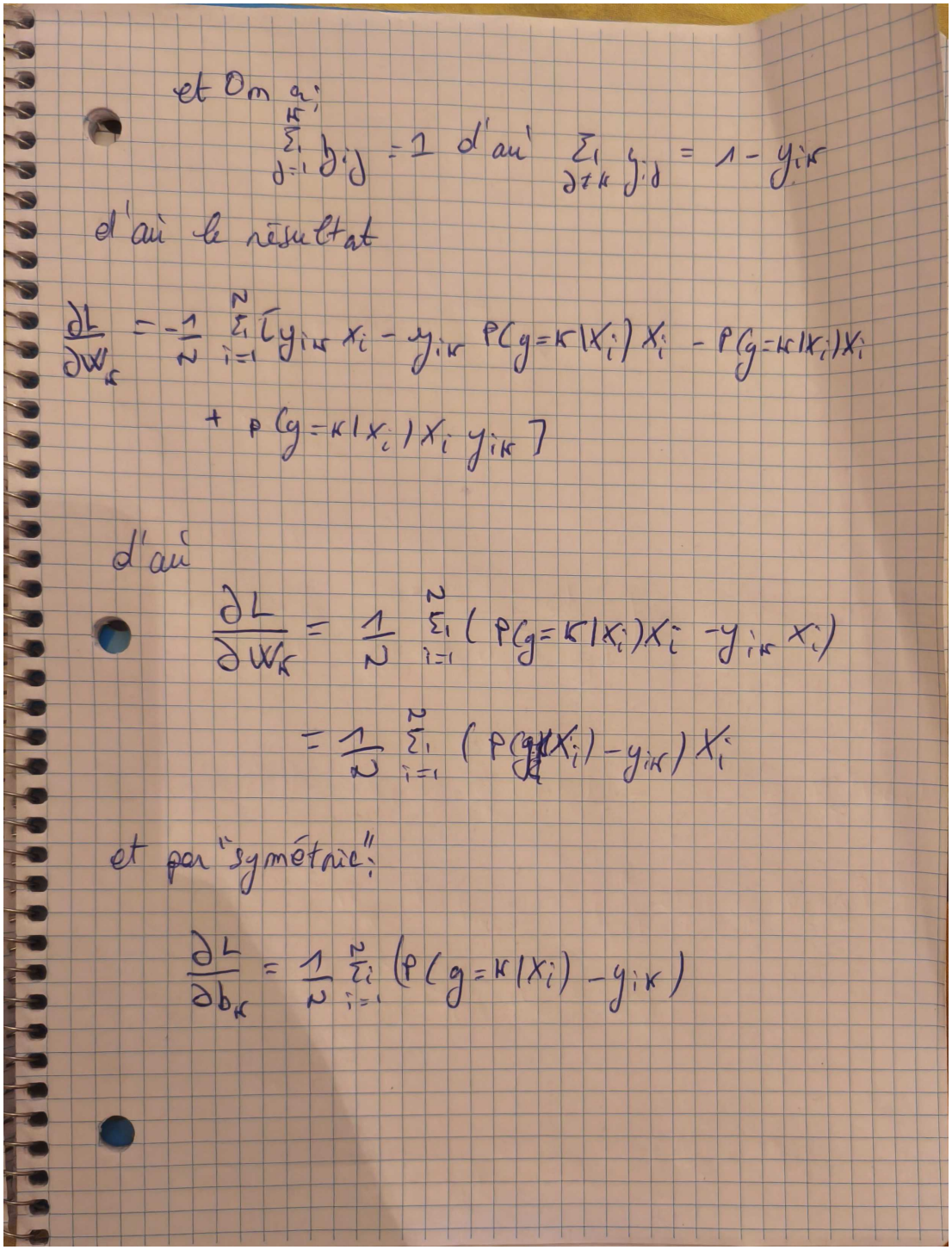

FIGURE 3 – Démonstration - Partie 3

3.1.2 Question 1.e

Le paramètre λ est concrètement un terme qui sert à la régularisation L2 du modèle de régression logistique. Il permet aussi de modifier le gradient en lui ajoutant λW. Etudions sont influence :

- **Si $\lambda = 0$:** Dans ce cas, le gradient est $\frac{\partial L}{\partial W} = \frac{1}{N} X^T (P - Y)$. Il n'y a ici pas de régularisation, ce qui signifie que les poids de W peuvent devenir très grands. Le modèle risque donc le surapprentissage. (surtout si les données sont bruités)
- **Si $\lambda > 0$:** Le gradient devient $\frac{\partial L}{\partial W} = \frac{1}{N} X^T (P - Y) + \lambda W$. Le terme λW

permet donc de rapprocher le pods vers 0 à chaque itération. Plus il sera grand, plus il sera proche de 0. On peut donc dire que si λ est trop grand, notre modèle risque le sous-apprentissage (les poids seront trop petits)

Voici le graphique des résultats obtenus en faisant varier λ :

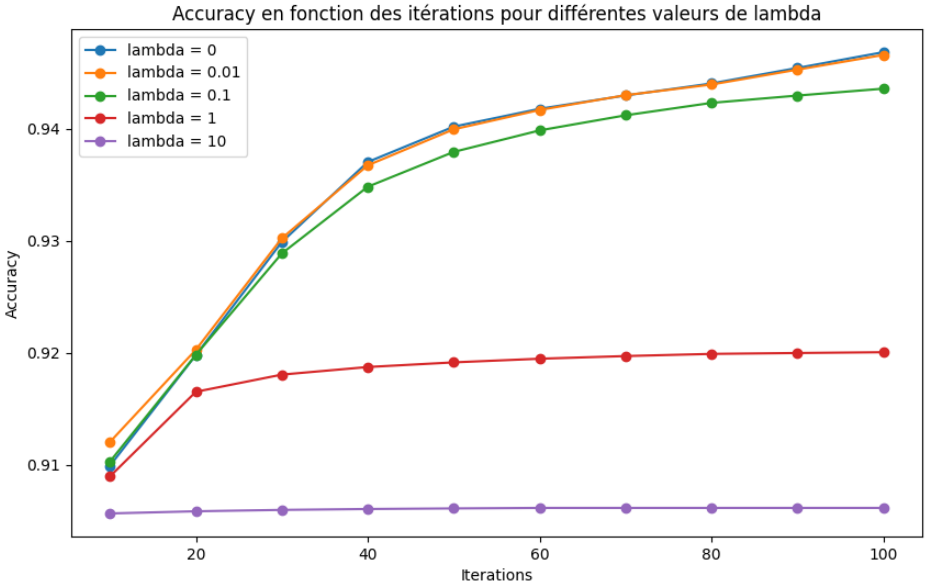

FIGURE 4 – Précision en fonction de λ

On remarque bien le sous apprentissage pour lambda trop grand. Cependant, on ne note pas forcément de sur-apprentissage pour $\lambda = 0$. Cela peut s'expliquer par le fait que les données sont relativement propres et non bruitées.

4 Gaussian Naive Bayes

4.1 Question 1

4.1.1 Question 1.a

Montrons le resultat suivant :

$$\log(P(g = k|X)) = \log(P(X|g = k)) + \log(P(g = k)) - C(X)$$

On a premierement par la formule de Bayes :

$$P(g = k|X) = \frac{P(X|g = k)P(g = k)}{P(X)} \tag{51}$$

Ainsi, en passant au log :

$$\log(P(g = k|X)) = \log(P(X|g = k)) + \log(P(g = k)) - \log(P(X)) \tag{52}$$

Or, P(X) ne dépend pas de k, on pose donc $C(X) = \log(P(X))$. D'où le résultat :

$$\log(P(g = k|X)) = \log(P(X|g = k)) + \log(P(g = k)) - C(X) \tag{53}$$

Ce que l'on cherche et de connaitre une classe sachant X : $argmax_k P(g = k|X)$. Comme $C(X)$ ne dépend pas de k, on a pas besoin de le calculer. Les paramètres a entrainer sont les paramètre qui dépendent de k soit : $P(g = X)$, μ_k et σ_k.

4.1.2 Question 1.b

On a la preuve suivantes :

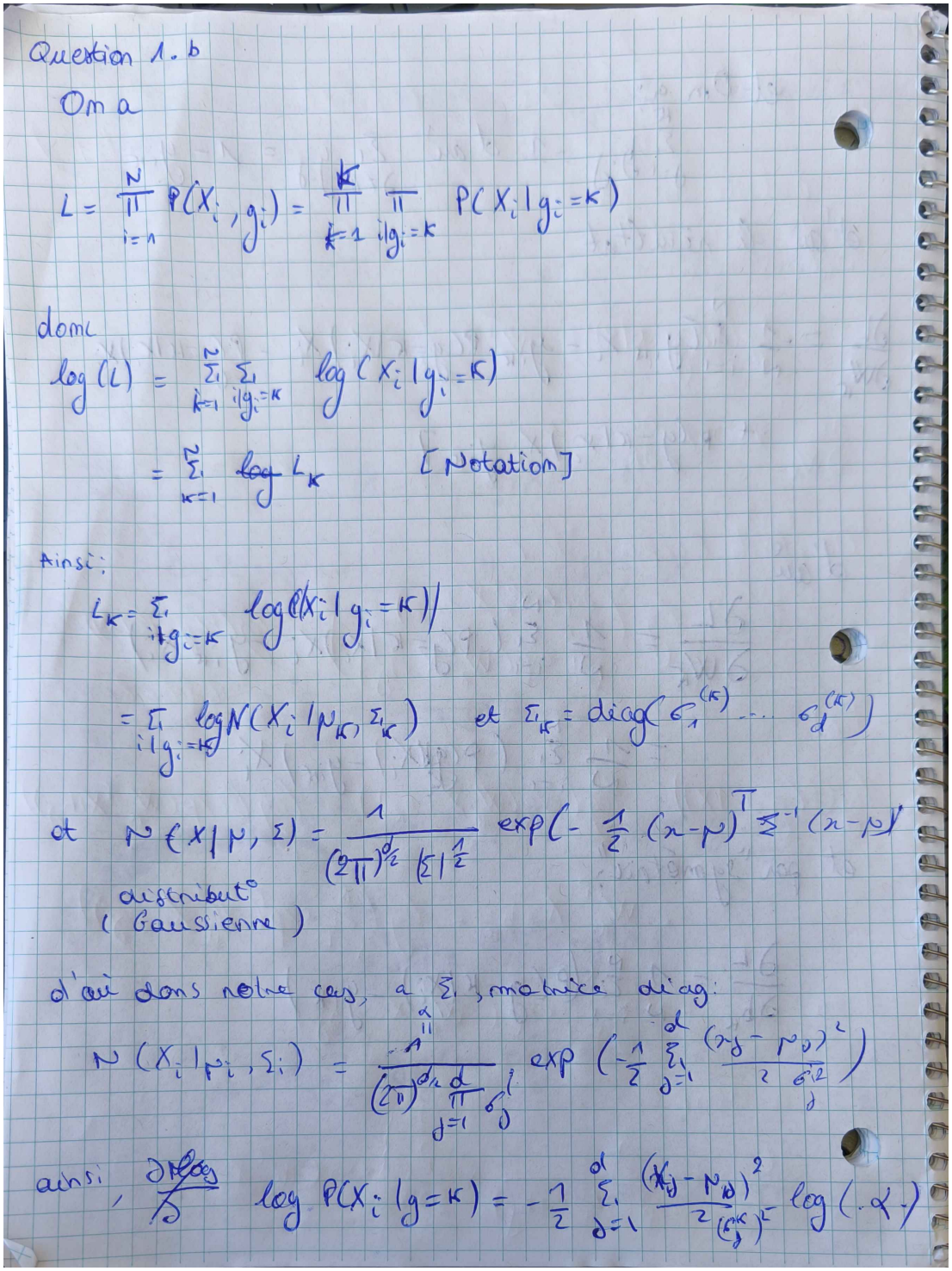

FIGURE 5 – Démonstration - Partie 1

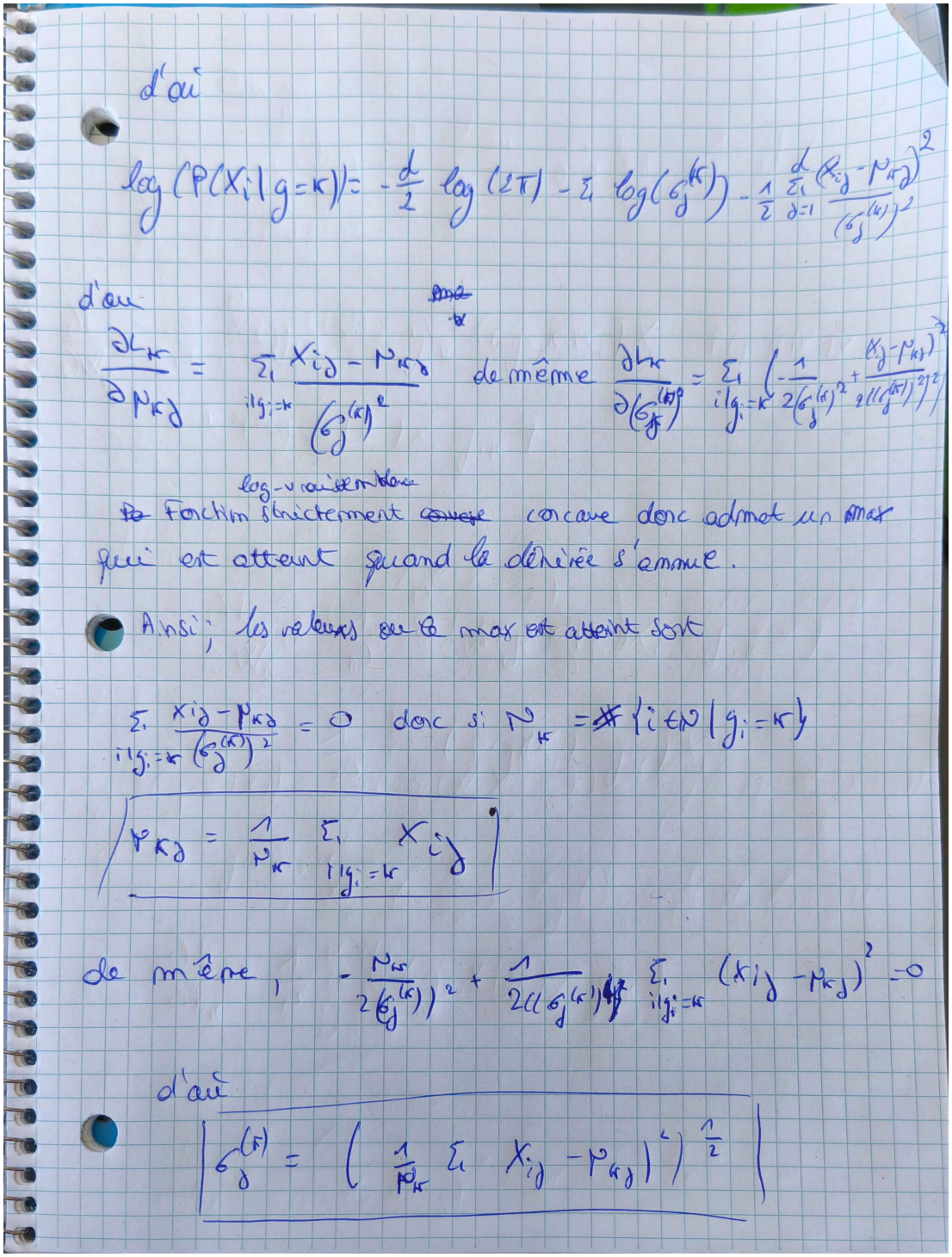

FIGURE 6 – Démonstration - Partie 2

Le résultat final est donc pour maximiser la vraissemblance :

$$\mu_k = \frac{1}{N_k} \sum_{i|g_i=k} X_i$$

$$\Sigma_k = \text{diag}\left(\frac{1}{N_k} \sum_{i|g_i=k} (X_{i,1} - \mu_{k,1})^2, \ldots, \frac{1}{N_k} \sum_{i|g_i=k} (X_{i,d} - \mu_{k,d})^2 \right)$$

4.1.3 Question 1.e

Après avoir implémenté le classifieur Gaussien naïf (GNB), nous obtenons les précisions suivantes :

— IRIS : 100,00%
— MNIST : 73,69%

Pour chaque classe, sur Iris, nous avons les résultats suivants :

Table 3 – Précision par classe (MNIST, GNB)

Classe	**Accuracy (%)**
0	77,65
1	92,86
2	66,57
3	45,54
4	89,21
5	62,89
6	88,10
7	75,58
8	89,12
9	47,67

La précision de la classe 1 plus élevée peut etre du a plusieurs choses, premierement, peut etre que cette classe est surreprésentée dans le jeu de données. Deuxièmement, peut etre que les caractéristiques de la classe 1 sont peut etre plus distinctes et mieux séparable par notre modèle. La répartition des données est surement plus simple à comprendre pour cette classe.

Concluons quant à l'éfficacité de GNB et KNN sur nos deux jeux de données :

— **IRIS** : GNB fonctionne très bien ici car les caractéristiques suivent des lois gaussiennes et sont relativement indépendantes entre elles.
— **MNIST** : GNB est moins performant parce que les pixels n'ont pas des distributions gaussiennes et sont fortement corrélés les uns aux autres dans l'espace. KNN, en revanche, profite des corrélation spatiale et de la proximité entre les images similaires, ce qui lui permet d'obtenir de meilleurs résultats pour ce jeux de données.

5 Source

A été utilisé l'IA générative pour les fautes de langues, et pour certains affichage de formule/tableau en LateX.