

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saâd DAHLAB, Blida

N° D'ordre.....



Faculté des sciences
Département d'informatique

Présenté par :

ABBAS Oussama

KOURNANE Abderrahmane

En vue d'obtenir le diplôme de master
Domaine : Mathématique et informatique

Filière : Informatique
Spécialité : Informatique
Option : Ingénierie de logiciel

Sujet :

**Conception et réalisation d'un simulateur de stationnement
permettant de décrire la personnalité d'un utilisateur en fonction
de sa manière de se stationner**

Soutenu le : 10 Octobre 2013

Mme N. REZOUG
M. A. BAOUIA
Mme M. ARKAM
M. M. R. SIDOUMOU

Présidente
Examinateur
Examinatrice
Promoteur

Promotion 2012/2013

REMERCIEMENTS

Avant d'entrer dans le vif du sujet, nous tenons à adresser nos remerciements à toutes les personnes qui nous ont soutenu et aidé dans notre travail pour la réalisation de notre projet de fin d'étude, et spécialement notre très cher ami Khebab Houssam, qui nous a soutenu et aidé par ses conseils avisés tout au long de la rédaction de ce mémoire. Nous remercions également l'ensemble des participants à la collecte de données. Pour finir, nous adressons également nos remerciements au Dr. John A. Johnson, qui a toujours Répondu à nos questions dans le but de nous aider.

DEDICACES

Abbas Oussama

*Je dédie ce travail à mes chers parents, ma femme
« Amina », mes soeurs, mes frères, mon binôme, et à mes
amis Houssame, Rafik, Nara, Peffect, Zaki, Yacine et Moh-
Bourkika et mon promoteur Reda, et cela pour remercier
leur soutien.*

DEDICACES

Kournane Abderrahmane

Je dédie spécialement ce mémoire à la mémoire de mon bien aimé père, Mohand Arezki, en priant dieu pour qu'il le récompense de tout ce qu'il a fait pour moi de son vivant. Je dédie également ce travail à ma très chère mère, qui m'a toujours soutenu, que dieu la garde.

Résumé :

Dans cette étude, nous avons conçu et développé un système qui a pour but de donner des indications sur la personnalité d'un individu selon les choix qu'il fait lors du stationnement. L'application est destinée aux terminaux sous android et propose aux utilisateurs un simulateur de stationnement représentant un parking avec plusieurs places libres. Afin de constituer un ensemble d'apprentissage, une collecte de données a été réalisée auprès d'individus dont nous connaissions le type de personnalité grâce aux résultats que ces derniers ont obtenus au questionnaire IPIP-NEO. Les participants à la collecte de données ont expérimenté le simulateur de stationnement, les choix et préférences de stationnement de chaque participant à la collecte de données ont été capturés et enregistrés, puis associé à son type de personnalités. Après traitement des résultats de la collecte de données, notamment en utilisant la classification non supervisée, nous avons eu comme résultat, des manières distinctes de se de stationner. Chacune d'entre elles correspondait à un type de personnalité particulier. L'ensemble d'apprentissage ainsi constitué, un utilisateur a pu expérimenter le système pour connaitre sa personnalité en fonction du comportement qu'il a eu et des choix qu'il a fait lors du stationnement.

Abstract :

In this study, we designed and developed a system which aims to give injections on a person's personality through his parking choice behavior. The application is destined to Android terminals and offer to users a parking simulator representing a parking lot with many free places. To constitute a training set, a data collect were done with help of some people who have their personality test results obtained after passing the IPIP-NEO questionnaire. The participants in the collect phase have tested the simulator and entered their results and IPark (the simulator) captured each participant's data and saved it. After treating the collected data, using unsupervised clustering, we have got several parking behaviors, each one of it, is linked to a particular type of personality. Through the constituted training set, a user was able to do the experiment to know its personality through his parking behavior.

ملخص:

النظام الذي طورناه ضمن هذا المشروع يهدف إلى اربط بين دراسة شخصية السائق و تأثيرها على سلوك إختيار مكان السيارة و ركنها، و دراسة التعرف عن الأنماط، و ذلك عن طريق محاكي في شكل لعبة فيديو. و قد طورنا هذا خصيصاً لأنظمة ال-Android.

بعد تطوير الفكرة، نجحنا في جمع بعض البيانات، إعتماًداً على أشخاص متطوعين. و استعملنا البيانات المجمعة لتجريب النظام الذي أعطى علامات النجاح، و كان من الممكن الحصول على نتائج أفضل في حالة وجود قاعدة بيانات أكبر.

اظهرت النتائج أنه يوجد ارتباط بين شخصية السائق و كيفية ركنه للسيارة و إختيار المكان.

Sommaire

Introduction générale.....	1
Chapitre I : Etat de l'art	
1. Introduction.....	5
2. Personnalité, big five et IPIP-NEO.....	5
2.1 Personnalité et différences individuelles.....	5
2.2 Le modèle à cinq facteurs.....	6
2.3 Les différents systèmes et questionnaires mesurant les cinq facteurs.....	7
2.3.1 IPIP-NEO	7
2.3.2 Autres système de mesures du modèle à cinq facteurs	8
3. Études sur les facteurs influençant un conducteur dans le stationnement de son véhicule.....	9
3.1 Première étude.....	9
3.2 Deuxième étude.....	12
3.3 Troisième étude.....	14
3.4 Synthèse des trois études.....	15
4. Études sur la capture et l'imitation des stratégies des joueurs humains dans les jeux de combat.....	15
4.1 Première étude.....	15
4.2 Deuxième étude.....	18
5. Conclusion.....	20
Chapitre II : Apprentissage automatique	
1. Introduction.....	23
2. Quelques définitions de base.....	23
2.1 Apprentissage automatique (Machine Learning).....	23
2.2 Fouille de données (Data mining) ou extraction de connaissances à partir des données (knowledge discovery in data).....	24
3. Quelques applications de l'apprentissage artificiel.....	24
4. Différentes formes d'apprentissage.....	26
4.1 Apprentissage supervisé.....	26
4.1.1 Classification supervisée	26
4.1.2 Algorithmes de classification supervisée.....	27
❖ Algorithme des k plus proches voisins.....	27
❖ La classification bayésienne.....	31
❖ Les arbres de décision.....	32
4.2 Apprentissage non supervisé.....	36
4.2.1 Classification non supervisée ou <i>clustering</i>	36
4.2.2 Algorithmes de classification non supervisée.....	37
4.2.2.1 Concepts de base.....	37
❖ Définition d'une partition.....	38

❖ Définition d'une hiérarchie.....	38
❖ La matrice de proximité.....	38
4.2.2.2 Clustering par partitionnement.....	39
❖ Algorithme des k-moyennes.....	39
4.2.2.3 Clustering hiérarchique.....	41
❖ Classification ascendante hiérarchique (CAH).....	42
5. Conclusion.....	48

Chapitre III : Architecture du système

1. Introduction.....	50
2. Fonctionnement générale du système.....	51
2.1 Première partie.....	51
2.2 Deuxième partie.....	53
3. Description du simulateur de stationnement et du vecteur «V».....	54
4. Collecte et traitement des données.....	58
4.1 Classification des vecteurs « V ».....	59
4.2 Le type de personnalité correspondant à chaque classe.....	61
5. Description de la personnalité d'un utilisateur selon ses choix lors du stationnement.....	65
6. Conclusion.....	68

Chapitre IV : Implémentation, test et résultats

1. Introduction.....	70
2. Langage utilisé et outils de développement.....	70
2.1 Android.....	70
2.2 Le langage de programmation JAVA.....	70
2.3 L'IDE eclipse.....	71
2.4 Le kit de développement.....	71
2.5 Les éléments d'une application android.....	72
3. Présentation d'IPark et de ses fonctionnalités.....	73
4. Tests et résultats.....	82
4.1 Collecte de données.....	82
4.2 Classification des vecteurs « V » avec la classification hiérarchique ascendante.....	83
4.3 Description des classes obtenues.....	85
4.4 Test d'IPark par un utilisateur désirant connaître sa personnalité.....	89
5. Conclusion.....	90

Conclusion générale.....	92
---------------------------------	-----------

Bibliographie.....	95
---------------------------	-----------

Introduction générale :

La capture de la manière de jouer des humains dans les jeux vidéo est un thème qui a déjà été abordé dans plusieurs études. Notamment celle conduite par S. Saini, P.W.H. Chung et C. W. Dawson, sur la capture et l'imitation des stratégies et tactiques exécutées par les joueurs humains dans les jeux vidéo de combat [21]. Mais peu d'études ont traité le cas de capturer la manière de jouer d'un utilisateur humain dans les jeux vidéo pour l'associer au comportement ou à la personnalité des joueurs.

La personnalité est la structuration cohérente de l'affect, de la cognition et des désirs (objectifs) car ils conduisent à divers comportements. Étudier la personnalité revient à étudier la façon dont les gens sentent, comment ils pensent, ce qu'ils veulent et finalement, ce qu'ils font [1]. Il est évident que des individus, peuvent faire des choix différents et avoir des comportements différents dans la même situation, en fonction des différences qu'il peut y avoir entre leurs personnalités. La personnalité est donc un facteur influant dans bon nombre de décisions que nous prenons, de choix que nous faisons et de comportements que nous avons. Connaitre de quelle manière et à quel point la personnalité d'un individu peut l'influencer dans un domaine donné, peut nous aider à prédire ses actions. Si nous arrivons à cela, nous pouvons optimiser bon nombre de systèmes afin d'avoir une meilleur expérience utilisateur ou une optimisation suivant les individus.

Le but de notre travail est de déterminer comment et à quel point, la personnalité d'un individu peut l'influencer dans le comportement qu'il peut avoir et les choix qu'il peut faire dans un environnement virtuel simulant le stationnement d'un véhicule dans un parking. Pour cela, plusieurs utilisateurs vont expérimenter le simulateur de stationnement que nous allons concevoir, leurs manières de se stationner seront capturées (en fonctions de leurs choix et comportements lors du stationnement) et associées à leurs types de personnalité. Ainsi sera obtenue une base d'apprentissage, qui nous permettra de prédire le type de personnalité d'un individu qui utilise le simulateur, la manière de se stationner de cet utilisateur sera capturée (en fonction de son comportement et de ses choix lors du stationnement) et comparée à celles des utilisateurs précédents, afin de lui attribuer le type de personnalité qui correspond à sa manière de se stationner.

▪ Problématique :

Plusieurs études et enquêtes ont été menées partout de par le monde sur les facteurs influençant un conducteur lors du choix d'un emplacement de stationnement (ex : « sureté de

la place », « proximité de la destination après le stationnement », etc.), que ce soit, hors ou à l'intérieur des parkings. La plupart des études menées sur ce sujet, ont eu pour but d'aménager des places de stationnement et des parkings adaptés aux besoins des populations et à leurs tendances dans le choix des emplacements de stationnement. Nous pouvons citer comme exemple, une étude menée en 2008, par Y. Meiping, Y. Ruisong et Y. Xiaoguang, et qui avait pour but la modélisation d'une grille de stationnement, en fonctions des facteurs privilégiés par les conducteurs lors du stationnement [11]. Les résultats de la plupart de ces études montrent qu'une dizaine de facteurs principaux sont pris en compte par le conducteur au moment de choisir son emplacement de stationnement. Lors d'une étude conduite en 2003, par A. Borgers, H. Timmermans, et P. Van Der Waerden, sous le titre de : « Travelers Micro-Behavior at Parking Lots: A Model of Parking Choice Behavior » [12], les auteurs ont tenté de déterminer quels facteurs étaient privilégiés lors du choix d'un emplacement de stationnement en fonction de l'âge, du sexe ou du but du voyage des conducteurs. Le travail que nous allons réaliser a pour but de déterminer quels facteurs sont privilégiés lors du choix d'un emplacement de stationnement en fonction du type de personnalité des conducteurs. En d'autres termes, il s'agit de répondre aux questions suivantes : *Est-ce que les facteurs privilégiés lors du choix d'un emplacement de stationnement changent en fonction du type de personnalité du conducteur ? Si oui, quels facteurs sont privilégiés par chaque type de personnalité ?*

▪ **Objectif :**

L'objectif de notre travail est de concevoir une application qui sera un simulateur de stationnement dans un parking. L'application devra être capable de donner des indications sur la personnalité d'un utilisateur X selon les choix qu'il fait et son comportement lors du stationnement. Ceci sera réalisé par la lecture de la manière de se stationner de cet utilisateur et sa comparaison avec celles également lues et enregistrées lors d'une collecte de données, où différents utilisateurs ont également utilisé le simulateur. Lors de cette collecte de données, les utilisateurs qui y participent, introduiront la description de leur personnalité avant d'utiliser le simulateur de stationnement, ainsi chaque manière de se stationner correspondra à un certain type de personnalité. De cette façon, après comparaison de la manière de se stationner de l'utilisateur X avec celles de la collecte de données et rapprochement avec l'une d'elle, on pourra lui attribuer la personnalité qui correspond à sa manière de se stationner. Il est à noter que les données issues de la collecte seront traitées afin que le résultat renvoyé à l'utilisateur X soit le plus juste que possible.

▪ **Organisation du mémoire :**

Après avoir annoncé le thème et exposé la problématique, nous présentons maintenant le plan de rédaction de ce mémoire :

Chapitre I : Dans ce chapitre, il sera question de définir la personnalité, de parler du modèle à cinq facteurs servant à la mesurer et des différents questionnaires basés sur ce modèle. Nous aborderons ensuite, différentes études sur les facteurs influençant un conducteur lors du stationnement de son véhicule. Pour finir, nous parlerons de deux études sur l'imitation des stratégies des joueurs humains dans les jeux vidéo de combat.

Chapitre II : Dans ce second chapitre, nous parlerons, en détails de l'apprentissage automatique ; sa définition, ses domaines d'application, et ses deux principales formes qui sont l'apprentissage supervisé et non supervisé. Certains algorithmes d'apprentissage supervisé et non supervisé seront vu en détails, avec l'explication de leurs fonctionnements ainsi que les avantages et inconvénients de chacun.

Chapitre III : Ce troisième chapitre, abordera en détails la conception de l'architecture de notre application. Nous expliquerons le fonctionnement de chaque module de notre architecture et justifierons les choix quant aux algorithmes utilisés.

Chapitre IV : Dans ce dernier chapitre, nous parlerons des outils avec lesquels a été réalisé le développement de l'application. Nous décrirons également l'application et le fonctionnement de chaque partie en détails et à l'aide de captures d'écran de l'application. Pour finir nous parlerons du test de l'application, et commenterons ses résultats.

Enfin, nous arriverons à la conclusion générale dans laquelle nous parlerons des perspectives à venir.

Chapitre I :

Etat de l'art

1. Introduction :

Le but de notre recherche est de concevoir une application qui va permettre de donner des informations à un utilisateur sur sa personnalité en fonction de sa manière de se stationner, il s'agira donc d'établir préalablement des liens entre personnalité et manière de se stationner.

Dans la première partie de ce chapitre, nous allons décrire brièvement ce qu'est la personnalité et présenter le modèle à cinq grand facteurs mis en place pour la mesurer. Enfin, nous parlerons des différents questionnaires et tests basés sur ce modèle, en particulier le questionnaire IPIP-NEO.

Nous avons choisit les caractéristiques d'emplacement de stationnement à mettre en avant en se basant sur les résultats de différentes études et enquêtes menées sur les facteurs influençant un conducteur dans son choix d'emplacement de stationnement. Dans la deuxième partie de ce chapitre, nous allons présenter trois de ces études et leurs résultats.

Pour que l'on puisse comparer ou évaluer le stationnement d'un individu dans le simulateur de stationnement, il faut que l'ensemble de ses choix et comportement puissent être quantifiés dans une forme mathématique. Dans la partie (3) de ce chapitre, nous présentons deux études sur la capture et l'imitation des stratégies et tactiques exécutées par les joueurs humains dans les jeux de combat, la partie qui nous intéresse le plus étant le processus de capture et de quantification des mouvement et autres choix tactiques exécutées par les joueurs humains durant le jeu.

2. Personnalité, big five model et IPIP-NEO :

2.1 Personnalité et différences individuelles :

La personnalité est la structuration cohérente de l'affect, de la cognition et des désirs (objectifs) car ils conduisent à divers comportements. Étudier la personnalité revient à étudier la façon dont les gens sentent, comment ils pensent, ce qu'ils veulent, et finalement, ce qu'ils font. Il est évident que les gens différent les uns des autres dans chacun des quatre domaines cités précédemment. Comment et pourquoi ils diffèrent est moins claire. C'est là, une partie importante de l'étude de la personnalité. C'est donc la structuration cohérente, dans le temps et l'espace des sentiments, des pensées, des désirs et des actions que nous identifions comme personnalité. La psychologie de la personnalité aborde les questions de la nature humaine, les aspects des différences individuelles et les caractéristiques propres à chaque individu [1].

2.2 Le modèle à cinq facteur « big five model » :

Les Big Five (cinq grands facteur) sont cinq traits centraux de la personnalité empiriquement mis en évidence par Golberg (1990). Ils constituent non une théorie mais un repère pour la description et l'étude théorique de la personnalité [2].

Il est parfois question du « modèle OCEAN » suivant les différentes dimensions du modèle (**O**penness to **E**xperience, **C**onscientiousness, **E**xtraversion, **A**greeableness, **N**euroticism) [2] en français respectivement (**O**uverture, **C**onscience, **E**xtraversion, **A**gréabilité, **N**évrotisme). Chacun des cinq grands facteurs est assez large et se compose d'une gamme de traits plus spécifiques [3]. Voici un tableau (Tableau 1.1) qui présente les cinq facteurs avec une brève description de ce que chacun mesure et les facettes ou traits qui le composent :

Facteur	Ce qu'il mesure	Facettes ou traits qui le composent
Ouverture	Appréciation de l'art, de l'émotion, de l'aventure, des idées peu communes, curiosité et imagination.	<ol style="list-style-type: none"> 1. Rêverie-Imagination 2. Esthétique 3. Sentiments 4. Actions-Nouveauté 5. Idées 6. Valeurs-Non conformisme.
Conscience (caractère consciencieux)	Autodiscipline, respect des obligations, organisation plutôt que spontanéité, orientation vers des buts.	<ol style="list-style-type: none"> 1. Compétence-Efficacité 2. Ordre 3. Sens du devoir 4. Recherche de la réussite 5. Autodiscipline 6. Délibération-Réflexion
Extraversion	Énergie, émotions positives, tendance à chercher la stimulation et la compagnie des autres.	<ol style="list-style-type: none"> 1. Chaleur-Convivialité 2. Grégarité-Sociabilité 3. Assertivité-Assurance 4. Activité-Réactivité 5. Recherche de sensations-Goût du risque 6. Emotions positives-Gaieté
Agréabilité	Tendance à être compatissant et coopératif plutôt que soupçonneux et antagonique envers les autres.	<ol style="list-style-type: none"> 1. Confiance 2. Droiture 3. Altruisme 4. Compliance-Consensualité 5. Modestie 6. Sensibilité

Névrotisme	Tendance à éprouver facilement des émotions désagréables comme la colère, inquiétude ou dépression, vulnérabilité.	<ol style="list-style-type: none"> 1. Anxiété 2. Colère-Hostilité 3. Dépression-Abattement 4. Timidité sociale 5. Impulsivité 6. Vulnérabilité
-------------------	--	--

Tableau 1.1. Tableau représentant les cinq grands facteurs et les facettes (sous facteurs) qui les composent (ce tableau est un recueil d'informations prise sur le site [63]).

➤ **Remarque :**

Beaucoup d'études font apparaître une corrélation négative entre extraversion et névrosisme : qui éprouve aisément des émotions négatives se montre moins social, moins enclin au dialogue [2].

2.3 Les différents systèmes et questionnaires mesurant les cinq facteurs :

2.3.1 IPIP-NEO :

Le IPIP (International Personality Item Pool), développé Lewis Goldberg, il possède un système de notation analogue à celui du NEO PI-R (non libre) et du NEO-FFI (voir plus bas). Le système de notation IPIP fait partie du domaine public et aucune autorisation n'est requise pour sa consultation. [3]

L'IPIP-NEO a pour but essentiel de faire découvrir le plus largement possible le modèle de personnalité à cinq facteurs. Il a été développé principalement par les deux chercheurs Lewis R. Goldberg et John A. Johnson (voir le site ipip.ori.org). Il existe une version courte (120 questions) et longue (300 questions) de ce test. Les deux versions du test sont gratuites.

Pour chaque question, l'utilisateur doit évaluer, à quel point, l'affirmation qui lui est proposée, correspond ou non à son comportement habituel. Il choisit, ensuite, la réponse qui lui convient le mieux, parmi les cinq possibilités qui lui sont proposées en cochant la case correspondante. Par exemple, pour l'affirmation : *je me fais facilement des amis*. L'utilisateur aura cinq choix de réponse qui sont : - - (pas du tout d'accord), - (modérément en désaccord), = (ni d'accord ni pas d'accord), + (modérément en accord), ++ (tout à fait d'accord).

▪ **Notation du test « IPIP-NEO » et interprétation des résultats :**

Le test IPIP-NEO note, à partir des réponses aux 120 questions posé (300 pour sa version longue), chacun des cinq facteurs sur une échelle de 0 à 100. La note générale d'un facteur est calculée à partir des notes obtenues dans les 6 sous facteurs qui le composent et qui sont

également notés de 0 à 100, il est à noter que la note générale d'un facteur ne correspond pas à la moyenne des 6 sous facteurs qui le composent. Après avoir fini le test, l'utilisateur obtient une fiche avec les scores qu'il a obtenus pour chaque facteur et les 6 facettes qui le composent, la fiche contient également les interprétations de ces scores ainsi que différents commentaires.

Toutes les informations sur la notation de ces deux versions et de l'interprétation de leurs résultats sont disponibles sur le site (ipip.ori.org).

2.3.2 Autres système de mesures du modèle à cinq facteurs :

▪ Le « Big Five Inventory » :

Le Big Five Inventory (BFI) est un questionnaire d'auto-évaluation visant à mesurer les cinq grands facteurs. Il est assez bref pour un questionnaire de personnalité multidimensionnel (44 questions au total), il se compose de phrases courtes avec un vocabulaire simple. Une copie du BFI, avec des instructions de notation, est reproduite dans cet article [4] (les deux dernières pages). Il est également disponible sur le site Web de Oliver P. John [23]. Le BFI est disponible gratuitement et aucune autorisation n'est requise pour l'utiliser dans des recherches à but non lucratif [3].

▪ NEO PI-R :

Le NEO PI-R est un test de 240 questions mise au point par Paul Costa et Jeff McCrae. Il mesure non seulement les cinq facteurs, mais aussi les six facettes de chacun des cinq grands facteurs. Le NEO PI-R est un produit commercial, contrôlé par une société à but lucratif, son utilisation requiert des autorisations et dans de nombreux cas, il faut payer pour l'utiliser.

▪ NEO-FFI :

Costa et McCrae ont également créé le NEO-FFI (60 questions), une version tronquée du NEO PI-R qui ne mesure que cinq les facteurs et non les facettes qui composent chacun d'entre eux. Le NEO-FFI est également non libre et son utilisation requiert des autorisations [3].

3. Études sur les facteurs influençant un conducteur dans le stationnement de son véhicule :

Nous allons présenter ici trois études sur les facteurs influençant un conducteur dans le stationnement de son véhicule, en décrivant pour chacune d'elles, le déroulement et les résultats obtenus.

3.1 Première étude :

Cette première étude que nous allons présenter a été conduite en 2003, par A. Borgers, H. Timmermans, P. Van Der Waerden [12]. Elle décrit une analyse du comportement des conducteurs lors du choix de l'emplacement de stationnement. L'analyse a été divisée en deux parties. Tout d'abord, certaines questions liées à l'occupation des places de stationnement et des choix de l'emplacement ont été abordées. Ensuite, un modèle logit imbriqué de comportement de choix de parking est estimé à partir des données recueillies d'une aire de stationnement dans la ville d'Eindhoven, Pays-Bas. Les variables les plus importantes sont la distance entre l'emplacement de stationnement et la machine à billet, la distance entre le parking et les sorties vers les commerces et les zones de travail, et si oui ou non l'espace de stationnement est un espace de coin. Nous n'aborderons pas les détails quant à la conception du modèle logit imbriqué.

L'étude a fourni des informations utilisées pour améliorer l'aménagement d'un parking de stationnement de la ville d'Eindhoven. Ce parking est utilisé par des journaliers et par des personnes qui visitent le quartier commerçant. L'objectif est de présenter des résultats descriptifs pour mieux comprendre le comportement des usagers de ce parking.

▪ Comportement lors du choix d'un emplacement de stationnement :

Dans l'article qu'ils ont publié, Borgers et ses collègues ont parlé d'une étude antérieure à la leur. Il s'agit de celle conduite par Young et ses collègues qui ont développé le système PARKSIM qui peut être utilisé pour concevoir et évaluer les différentes dispositions de stationnement dans un parking [13 ; 14 ; 15]. Le système retrace les mouvements des usagers d'un parking, simule le processus de recherches, l'interaction avec d'autres véhicules et les manœuvres de stationnement. L'ensemble des attributs qui influencent la décision qui concerne le déplacement et le choix d'une place particulière de parking est considérable et comprend le temps de déplacement vers le lieu de stationnement, le temps de marche de la place de parking jusqu'à la destination souhaitée, la facilité de stationnement, la facilité de

sortie du véhicule, la disponibilité de l'ombre [13]. Young a seulement mis en œuvre le temps de déplacement vers la place de parking et le temps de marche vers le lieu désiré dans le système PARKSIM.

Cette conceptualisation est présentée dans la figure 1.1, qui traite le micro-comportement de stationnement. On suppose qu'un automobiliste accède à un parking et décide de la place où il désire stationner sa voiture ainsi que la manière dont il pourra la stationner. De cette façon, si un automobiliste rentre dans un parking, il commence d'abord par le processus de recherche de place où stationner sa voiture. Tout en se déplaçant dans le parking, il vérifie les places libres. Quand il ne trouve pas d'espace libre, l'automobiliste peut décider d'attendre qu'une place se libère ou de quitter le parking (comportement de choix adaptif).

Lorsque l'automobiliste trouve un espace libre, il évalue cet espace puis décide s'il souhaite vraiment l'occuper ou pas. Occuper l'espace libre signifie stationner la voiture et marcher ensuite jusqu'à la destination finale. Si l'automobiliste ne choisit pas l'espace libre, il devra se déplacer à nouveau dans le parking à la recherche d'un autre espace libre. Ce processus peut être influencé par l'état de la grille de stationnement (la disposition des places libres), ou par les caractéristiques des espaces de stationnement (par exemple, la distance entre l'entrée du parking et les espaces de stationnement, la distance entre les espaces de stationnement et l'endroit où l'automobiliste souhaite se rendre pour ses activités et la distance entre les places de stationnement et le point où il doit payer le ticket de stationnement) et aussi, par les caractéristiques de l'automobiliste (par exemple, le sexe, l'âge, le type de véhicule, le nombre de personnes qui l'accompagnent).

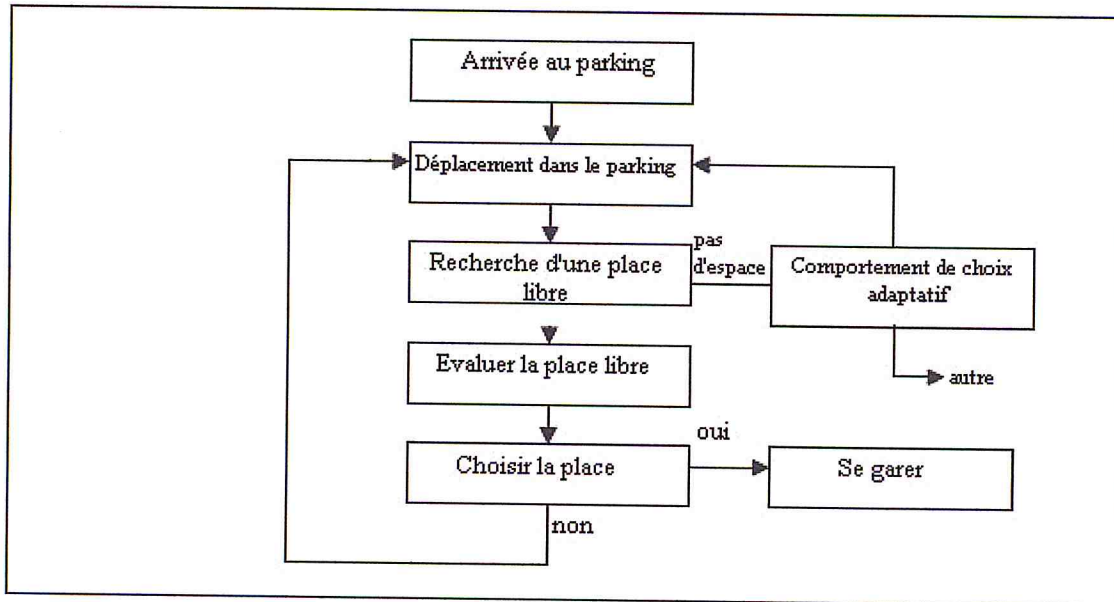


Figure 1.1. Cadre conceptuel pour un micro-comportement de stationnement (dérivé depuis Thomson et Richardson, 1998) [Notre traduction][12]

▪ La collecte de données :

Les données concernant les comportements de stationnement des automobilistes ont été recueillies à l'automne 2000 sur une grande aire de stationnement au centre-ville d'Eindhoven, Pays-Bas [16]. En plus des données sur la densité de la circulation dans ce parking et le choix de l'automobiliste, d'autres variables contextuelles telles que le type de véhicule, le sexe et l'âge de l'automobiliste, le nombre de passagers dans la voiture, et selon si l'automobiliste paye le ticket de stationnement avant ou après sa visite ont été recueillis.

Les observations ont été faites durant deux heures un mardi et un samedi matin (de 8h30 jusqu'à 10h30). Une autre période d'observations d'une durée de deux heures a été faite un jeudi après midi (de 14h00 à 16h00). Ces périodes sont les plus intéressantes dans le cadre de cette recherche car elles correspondent aux horaires de déplacements pour le travail ou aux heures typiques de visites des magasins du centre-ville.

▪ Espaces de stationnements occupés :

Le premier aspect spatial lié au micro-comportement de stationnement concerne la répartition spatiale des différents espaces de stationnement occupés.

▪ **Le choix d'un espace de stationnement :**

La manière dont un parking est rempli de voiture dépend de plusieurs règles de décisions prises par les automobilistes. Le micro-comportement de stationnement peut être décrit par les règles de décisions suivantes :

La première règle concerne le statut de l'espace de stationnement par rapport aux espaces de stationnement adjacents. On peut distinguer quatre situations : (1) les deux espaces adjacents sont libres, (2) l'espace adjacent droit est occupé, l'autre est libre, (3) l'espace adjacent gauche est occupé et l'autre est libre, et (4) les deux espaces adjacents sont occupés. Un second ensemble de règles se concentre sur l'emplacement de stationnement choisi par rapport à l'emplacement du distributeur de tickets du parking, à l'emplacement de l'entrée du parking et à l'emplacement du point de sortie des automobilistes.

▪ **Synthèse des résultats de l'étude :**

Le but de Borgers et de ses collègues était de fournir plus d'idées sur la relation entre le choix de l'espace de stationnement et un ensemble de caractéristiques personnelles, caractéristiques de l'espace de stationnement choisi (par exemple l'occupation des places adjacentes) et les différentes variables de distances (vers le distributeur de tickets, vers l'entrées du parking, vers la destination finale). Il semble que la distance entre l'espace de stationnement et le distributeur de tickets et entre l'espace de stationnement et les différentes sorties du parking, ont un impact important sur le comportement lors du choix de l'espace de stationnement. Mais d'après les analyses descriptives, il semble que les caractéristiques personnelles jouent un rôle mineur dans ce contexte.

3.2 Deuxième étude :

Cette seconde étude que nous allons présenter [11], a été conduite par Yu Ruisong, Yun Meiping et Yang Xiaoguang du département d'ingénierie de la circulation de l'université de Tongji, à Shanghai, en Chine. Elle porte sur la modélisation d'une grille d'emplacement de stationnement publique en se basant sur le comportement des conducteurs dans le choix de leurs emplacements de stationnement.

La plupart des recherches conduites sur la détermination de la grille de stationnement ne tiennent pas compte des différences individuelles entre les conducteurs dans leurs comportements quand il s'agit de choisir un emplacement de stationnement. Le travail de

Ruisong et de ses collègues porte sur l'optimisation d'une grille de stationnement tout en tenant compte du comportement du conducteur dans sa façon de choisir son emplacement de stationnement. Tout d'abord, les principaux facteurs influençant le conducteur sont analysés, comme la distance parcourue à pied pour se rendre à la destination voulue une fois que le véhicule est stationné, la durée du temps de stationnement, le but du voyage. Avec l'analyse des données et les résultats des enquêtes menées, les 3 chercheurs ont conclu que la distance parcourue à pied après stationnement du véhicule est un facteur clé. Ensuite, Un modèle de grille de stationnement a été conçu en se basant sur une loi de probabilité qui permet de sélectionner les places optimales en se basant sur les facteurs les plus influant sur le comportement d'un conducteur dans le choix de son emplacement de stationnement.

▪ **Les facteurs influençant le choix de l'emplacement de stationnement :**

Le comportement d'un conducteur dans le choix de son emplacement de stationnement est défini par tous les facteurs qu'il prend en compte lors du choix d'un emplacement ou non. La figure 1.2 montre les résultats de l'enquête conduite dans la ville de Zhuhai (dans le sud de la Chine La province du Guangdong) sur les facteurs influençant les conducteurs dans leurs choix d'emplacements de stationnement. Nous pouvons voir que la sécurité de stationnement est le facteur le plus important pour 36%, la distance parcourue à pied après le stationnement est également un facteur important avec 20%.

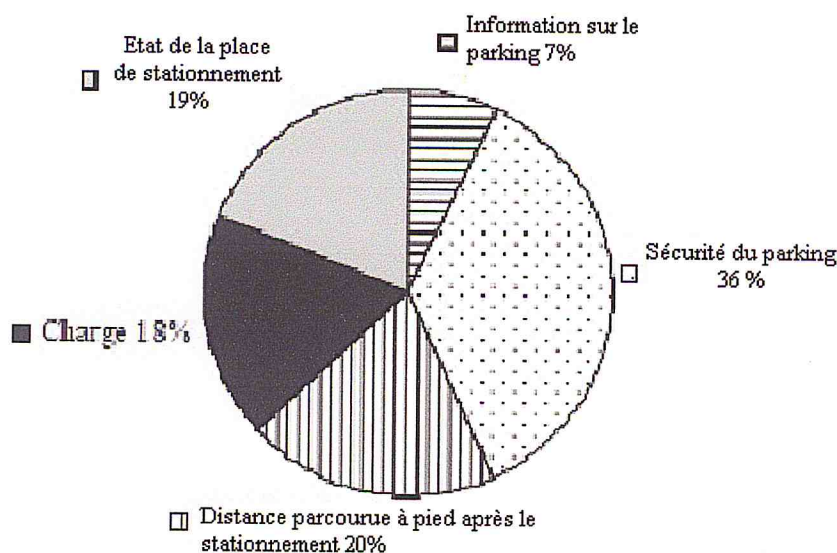


Figure 1.2. Principaux facteurs influençant le choix du conducteur [Notre traduction] [11].

3.3 Troisième étude :

Cette troisième étude [5], est des mêmes auteurs que la précédente. Cette fois ci, Ruisong et ses collègues ont tenté de connaître sur les critères sur lesquels se base un conducteur pour choisir un emplacement de stationnement, compte tenu des informations qu'il a sur la zone qui l'entoure. Il a été supposé que les conducteurs peuvent choisir entre se stationner dans un parking (sécurisé et payant) ou dans la rue. Un premier ensemble de données a été recueillis à partir de conducteurs ayant toutes les informations sur les parkings existants (leurs emplacements et leurs aménagements), le deuxième ensemble de données a été recueillis à partir de conducteurs disposant d'informations incomplètes sur les emplacements des parkings et leurs aménagements. Les facteurs qui influencent les conducteurs dans leurs choix sont :

- **Les caractéristiques du voyage (comme le but du voyage) :**

Ces caractéristiques n'ont pas été considérées comme facteurs principaux influençant le choix du conducteur dans cette étude.

- **Les caractéristiques des voyageurs (âge, genre, revenu etc..) :**

En se basant sur les recherches [6] [7] [8] ; l'âge, le genre et le remboursement ont été pris comme facteurs principaux parmi les caractéristiques des voyageurs dans cette étude.

- **Les caractéristiques de l'emplacement de stationnement :**

Ils sont considérés comme les facteurs les plus importants. Compte tenu des différences entre le stationnement sur rue et dans un parking. Cette étude considère les frais de stationnement, le temps de marche et la sûreté (pour le véhicule) comme les principaux facteurs influençant le conducteur dans son choix. Théoriquement, les frais de parking sont toujours parmi les facteurs les plus influant dans le choix d'un emplacement de stationnement [7] [9] [10].

Pour résumer les caractéristiques de places de stationnement en parking et en rue, il y a donc les frais de stationnement, le temps de marche à pieds et la sûreté de l'emplacement choisi.

D'une manière générale, pour le stationnement en parking, le prix de stationnement est plus élevé et le temps de marche est souvent plus long. Cependant, la sûreté tend à être meilleure. D'autre part, quand il s'agit de stationnement sur rue, la taxe de stationnement est souvent moins chère et la distance de marche à pieds (après stationnement) tend à être plus courte, mais la sûreté est relativement mauvaise.

Les résultats des deux sortes de comportement (lors du choix de l'emplacement) de conducteurs (bien informés et mal informés) ont été analysés. Les résultats de ces analyses ont montré que, par rapport aux conducteurs complètement informés, les conducteurs partiellement informés accordent moins d'importance au temps de marche à pieds et à la sûreté, mais plus aux frais de stationnement et d'autres facteurs.

3.4 Synthèse des trois études :

Après s'être intéressé aux trois études précédemment citées, on peut constater que les trois s'accordent sur le fait qu'une demi-douzaine de facteurs principaux est toujours prise en compte par le conducteur au moment de faire ce choix. Parmi ces facteurs, on peut citer : « La sûreté de l'emplacement choisi », « l'état de l'emplacement choisi et ses caractéristiques », « la distance parcourue à pied après avoir stationné le véhicule », « coût à payer pour stationner son véhicule », « la connaissance du parking ou des alentours ». D'autres facteurs peuvent avoir de l'importance comme le but du voyage ou les caractéristiques personnelles des conducteurs.

4. Etudes sur la capture et l'imitation des stratégies des joueurs humains dans les jeux de combats :

Nous allons maintenant présenter deux études sur la capture et la reproduction des stratégies et tactiques des joueurs humains dans les jeux vidéo de combat. Ces deux études ont été conduites par S. Saini, P.W.H. Chung et C. W. Dawson. Dans les deux études, il y a eu capture et classification des tactiques et stratégies en employant des méthodes différentes de la première étude à la seconde. Mais le point en commun entre les deux études est l'implémentation de machines à états finis conduites par des données pour que les tactiques capturées soient reproduites par des agents machines.

L'architecture proposée dans la deuxième étude diffère de la première principalement par le fait qu'elle est adaptée aux jeux de combat multi-paramètres.

4.1 Première étude :

Dans cette première étude [17], Saini et ses collègues présentent donc une approche pour l'imitation de la manière de jouer d'un humain par un agent machine. Plusieurs parties ou combats se déroulent d'abord entre deux joueurs humains. Durant ces combats, la manière de jouer d'un des deux humains est copiée puis l'agent machine la reproduit dans une partie contre un autre joueur humain. Pour cela, les données sont enregistrées et analysées avant

donne un changement de comportement de l'IA qui change en fonction du passage de l'automate fini d'un état à l'autre. Pour chaque état, il existe une fonction de transition, le passage entre les états se fait en respectant les critères de transition de chaque état. La machine à états finis est alimentée continuellement aussi longtemps que le jeu est actif [18].

L'utilisation de machines à états finis dans les jeux vidéo est promue par de nombreux développeurs. Cependant, leur principal inconvénient réside dans leur prévisibilité [19].

La côté statique et prévisible des machines à états finis peut être corrigé par l'implémentation de machines à états finis conduite par des données (*Data Driven Finite State Machine*). Cette approche utilise des données publiées pour alimenter la machine à états finis. Une machine à états finis conduite par des données (*Data Driven Finite State Machine*) est utile pour l'instanciation de machines à états finis personnalisées dont les états et la logique de transition sont définies dans un fichier externe [20]. En ce qui concerne la simulation de la stratégie des joueurs, les données contenues dans le fichier peuvent être écrites en temps réel pendant le jeu et ensuite utilisées pour compiler une machine à états finis.

- **Implémentation :**

L'approche utilisée pour résoudre le problème repose sur une combinaison des techniques décrites précédemment.

L'utilisation de l'architecture peut être divisée en deux. La première utilisation serait pendant la phase de capture de données, lors du combat entre les deux joueurs humains. C'est au cours de la phase de saisie de données que les informations sur les mouvements effectués, ainsi que l'état du jeu (à savoir la santé et la distance entre les combattants) sont collectées. Une fois que ces informations sont connues, les mouvements qui sont effectués sont affectés à différents états prédéterminés à l'aide du classificateur bayésien naïf qui a été formé à les classifier. La transition d'un état à un autre est basée sur le paramètre « santé » du joueur.

- **Conclusion :**

Pour démontrer l'efficacité de l'approche proposée, une stratégie a été élaborée avant de jouer un match entre deux joueurs humains. Les résultats de l'expérimentation démontrent que l'approche peut être utilisée pour imiter les stratégies humaines avec succès. Bien qu'elle fonctionne, cette approche a des limites, la principale d'entre elles réside dans le fait qu'elle ne convient pas aux jeux multi-paramètres.

4.2 Deuxième étude :

Cette deuxième étude [21], est des mêmes auteurs que la précédente. Ces derniers ont amélioré l'approche en l'adaptant aux jeux de combat multi-paramètres. Ils ont, en effet, inclus deux paramètres de plus dans le jeu de combat servant de démonstration de faisabilité.

Au paramètre « santé », ils ont ajouté ceux de l'endurance et du moral, ce qui donne 3 paramètres de jeu avec lesquelles le joueur doit composer pour construire sa stratégie. Ce qui suit est une brève description de leur travail :

▪ Démonstration de faisabilité :

Comme dans le précédent article [17], la démonstration de faisabilité est un simple jeu de combat en un contre un avec quelques modifications par rapport au premier.

Les mouvements des joueurs sont limités à un ensemble de techniques d'attaque et de coups qui varient dans la portée, les dommages causés et la vitesse. Le jeu comporte trois paramètres par joueur qui sont : La santé, le moral et l'endurance. Ceci donne un total de six paramètres de jeu. La santé du joueur s'affaiblit au fur et à mesure des coups qu'il reçoit. Son endurance diminue quand il effectue des blocages ou porte des coups. Quand au moral, il augmente à chaque esquive réussie. Quand le moral d'un joueur dépasse un certain seuil, ses coups provoquent proportionnellement plus de dégâts chez son adversaire (ils lui coutent plus en points de santé). La santé et l'endurance de chaque joueur sont initialisées à 100, alors que le moral l'est à 50.

▪ Conception du système :

La conception du système utilise plusieurs techniques pour imiter le joueur humain. Le jeu se déroule initialement entre deux joueurs humains qui combattent l'un contre l'autre, et seul l'un d'entre eux sera imité. Ceci est répété plusieurs fois, pendant lesquelles, les données concernant chacun des six paramètres de jeu, ainsi que les mouvements effectués sont répertoriées dans un fichier texte. Il y a également une hypothèse sous-jacente selon laquelle le joueur imité use de la même stratégie à chaque fois. La conception aborde chacune des couches de prise de décision avec une technique différente ; une machine à états définis conduite par des données est utilisée au niveau stratégique, la classification hiérarchique est employée au niveau tactique, et la méthode des k plus proches voisins est employée au niveau opérationnel.

Les données sont collectées pendant les premiers combats, Ils contiennent diverses statistiques sur les paramètres de jeu et les mouvements de jeu exécutés. Les mouvements et combinaisons de mouvements effectués sont représentés sous forme de vecteurs X , tel que $X = (x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7)$, où $x_0 \dots x_7$ représentent les divers mouvements et leurs effets dans le jeu. Ces vecteurs sont alors regroupés en utilisant la classification hiérarchique (La méthode du lien complet) :

$$D(X_i, X_j) = \max_{x \in X_i} \max_{y \in X_j} d(x, y) \quad [21]$$

La distance entre deux ensembles (clusters) est définie par la distance entre les deux éléments les plus éloignés [22]. Les ensembles de données regroupées en clusters, $s_0, S_1, s_2, \dots, S_n$, forment les états d'une machine à états finis conduite par des données, avec des mouvements et des combinaisons de mouvements dans chaque état.

Après avoir établi les états, les données brutes du combat entre les deux joueurs humains sont ré-analysées et des transitions d'états sont déterminées. En ré-analysant les données, les transitions d'états semblables sont identifiées. Les valeurs de chacun des six paramètres de jeu pour chacune des transitions semblables sont évaluées, et la variance entre les valeurs d'un paramètre dans une transition d'états et les transitions d'états qui lui sont semblables est calculée. Si la variance entre les valeurs du même paramètre est en-dessous du seuil de 10, le paramètre et sa valeur moyenne seront considérés comme fonction de transition pour cet état particulier.

Durant le jeu contre la machine, quand la machine à états finis conduite par des données entre dans un nouvel état, une action est sélectionnée. Ceci est réalisé en calculant la distance euclidienne entre le vecteur requête r , qui représente les paramètres de jeu en temps réel, et chaque vecteur de l'ensemble V , qui représente l'ensemble des vecteurs contenant les données sur les paramètres de jeu collectées pendant les combats entre joueurs humains.

Chaque état a un fichier correspondant contenant les mouvements qui doivent être exécutés ainsi que les valeurs des paramètres sous lesquels ils avaient été exécutés lors des combats entre joueurs humains. Les valeurs des paramètres sont représentées par les vecteurs appartenant à V , tandis que les mouvements correspondants pour chaque vecteur forment l'ensemble des outputs, O . La distance euclidienne entre r et chaque élément de V est calculée en utilisant l'équation suivante :

$$d(\mathbf{r}, \mathbf{v}) = \sqrt{\sum_{i=1}^n (r_i - v_i)^2} \quad [21]$$

Le vecteur de l'ensemble V ayant la distance la plus courte par rapport à r est déterminé et les mouvements (outputs) correspondants de l'ensemble O sont exécutés. Le calcul de la distance euclidienne et la sélection des outputs (mouvement à faire) sont effectués pendant le jeu.

▪ **En conclusion :**

Les résultats montrent que la tactique et la stratégie ont été imitées avec succès. Il n'y a aucune restriction sur le nombre d'états qui peuvent être utilisés. L'architecture proposée peut être utilisée dans le cas de transitions multi-paramètres. Cependant, aucune réduction de parasites n'est actuellement mise en œuvre. Les anomalies dans les données provoquées par l'erreur humaine pendant les combats entre joueurs humains peuvent empêcher l'application réussie de cette approche.

Le calcul vectoriel traite chaque attribut du vecteur d'entrée avec la même importance. Il n'existe actuellement aucune pondération signifiant les facteurs qui sont plus influents que les autres. Il est possible de remédier à cela en pondérant chaque valeur des vecteurs.

D'autres recherches pourraient impliquer l'utilisation d'un classificateur pour connaître les fonctions de transition, plutôt que la variance. Ceci peut conduire à une architecture plus robuste, et moins sujette à la contamination par les anomalies.

5. Conclusion :

Dans ce chapitre, nous avons donc choisi le moyen de mesurer la personnalité des utilisateurs de notre système, et avons également eu une idée sur les caractéristiques d'emplacement et autres facteurs à mettre en avant pour décrire le comportement d'un conducteur qui stationne en se basant sur les études citées.

Nous avons aussi eu une idée des moyens utilisés pour capturer et représenter mathématiquement (pour ensuite les traiter et les exploiter) les mouvements et choix que fait un humain dans un jeu vidéo.

Donc, notre tache dans le chapitre suivant, consiste à s'intéresser au concept d'apprentissage automatique et à ces différentes techniques. Chose nécessaire, avant de se lancer dans la conception de l'architecture de notre système où nous utiliserons certaines de ces techniques.

Chapitre II :

Apprentissage

automatique

1. Introduction :

La faculté d'apprendre est essentielle à l'être humain pour reconnaître une voix, une personne, un objet... On distingue en général deux types d'apprentissage : l'apprentissage «par cœur» qui consiste à mémoriser telles quelles des informations, et l'apprentissage par généralisation où l'on apprend à partir d'exemples un modèle qui nous permettra de reconnaître de nouveaux exemples. Pour les systèmes informatiques, il est facile de mémoriser un grand nombre de données (textes, images, vidéos...), mais difficile de généraliser. Par exemple, il leur est difficile de construire un bon modèle d'un objet et d'être ensuite capable de reconnaître efficacement cet objet dans de nouvelles images. L'apprentissage automatique est une tentative de comprendre et de reproduire cette faculté d'apprentissage dans des systèmes artificiels [24].

Dans ce chapitre, nous allons tout d'abord parler de l'apprentissage automatique de manière générale, avec quelques définitions de concepts de bases et en citant certains des domaines d'applications de l'apprentissage automatique.

Ensuite, nous exposerons les deux principales formes d'apprentissage automatique, qui sont l'apprentissage supervisé et non supervisé, en citant et détaillant quelques uns des algorithmes de chaque forme d'apprentissage.

Nous avons choisi de ne mettre la lumière que sur l'apprentissage supervisé et non supervisé, et de ne pas s'intéresser à d'autres formes comme le semi-apprentissage et l'apprentissage par renforcement. Et ce, compte tenu de nos besoins pour la conception et la réalisation de notre application.

2. Quelques définitions de base :

2.1 Apprentissage automatique (*Machine Learning*) :

L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques [25].

Il existe deux tendances principales en apprentissage, celle issue de l'intelligence artificielle et qualifiée de symbolique, et celle issue des statistiques et qualifiée de numérique [26].

L'apprentissage automatique (machine-learning en anglais) se trouve au carrefour de nombreux domaines : intelligence artificielle, statistiques, sciences cognitives, théorie des probabilités, de l'optimisation, du signal et de l'information [24]. Voir (Figure 2.1).

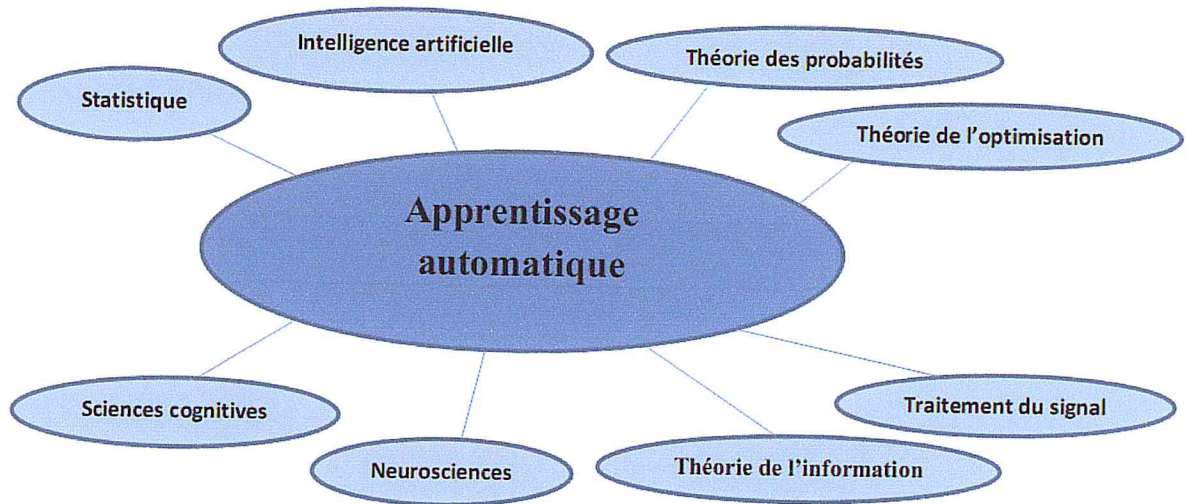


Figure 2.1. Domaines ayant un rapport avec l'apprentissage automatique [27].

2.2 Fouille de données (*Data Mining*) ou Extraction de connaissances à partir des données (*Knowledge Discovery in Data*) :

La fouille de données prend en charge le processus complet d'extraction de connaissances: stockage dans une base de données, sélection des données à étudier, si nécessaire : nettoyage des données, puis utilisation des apprentissages numériques et symboliques afin de proposer des modèles à l'utilisateur, enfin validation des modèles proposés. Si ces modèles sont invalidés par l'utilisateur, le processus complet est répété [26].

3. Quelques applications de l'apprentissage artificiel :

Un programme peut devenir plus efficace en le dotant d'une possibilité d'apprentissage. Reprenons pour cela les applications de l'intelligence artificielle et de la reconnaissance des formes citées ci-dessus :

➤ Un programme de reconnaissance de la parole augmente ses performances au fur et à mesure de son utilisation par la même personne : c'est une expérience qu'il est aujourd'hui facile de faire en pratique si on achète un logiciel personnel de dictée vocale [26].

- Un programme de détection des ressources terrestres apprend à reconnaître une zone de pollution au milieu de la mer, à partir d'une base de données d'exemples d'images de zones connues comme propres ou comme polluées : cette base de données lui sert d'expérience pour déterminer sa décision sur une zone inconnue [26].
- Un programme de diagnostic sur un ensemble d'informations évolutives prises sur un patient doit avoir été pourvu de connaissances, à partir de diagnostics de praticiens et d'experts sur des situations types. Mais il doit aussi avoir été doté d'un module de généralisation, de façon à réagir correctement à des situations auxquelles il n'a jamais été confronté exactement [26].
- Les moteurs de recherche sur le Web pourraient être munis d'un module d'adaptation au style de navigation de l'utilisateur : c'est une faculté souhaitable pour augmenter l'ergonomie de leur utilisation. Les programmes ne sont pas encore réellement agrémentés de cette propriété, mais il est clair que c'est une condition nécessaire pour franchir certains obstacles de communication si évidents actuellement [26].
- L'exploitation des fichiers client d'une entreprise est souvent faite par un expert ou un programme expert qui utilise des règles explicites pour cibler un segment de clientèle susceptible d'être intéressée par un nouveau produit. Mais ces règles peuvent être acquises automatiquement, par un apprentissage dont le but est de fournir de nouvelles connaissances expertes, à la fois efficaces et intelligibles pour l'expert [26].
- Un programme de jeu d'échecs possède en général une très bonne efficacité a priori ; mais il est naturel d'essayer de le doter d'un module où il puisse analyser ses défaites et ses victoires, pour améliorer ses performances moyennes dans ses parties futures. Ce module d'apprentissage existe dans un certain nombre de programmes de jeux [26].

4. Différentes formes d'apprentissages :

Les algorithmes d'apprentissage peuvent se catégoriser selon le type d'apprentissage qu'ils emploient, si les classes sont prédéterminées et les exemples étiquetés, on parle alors d'apprentissage supervisé. Quand le système ou l'opérateur ne disposent que d'exemples, mais non d'étiquettes, et que le nombre de classes et leur nature n'ont pas été prédéterminés, on parle d'apprentissage non supervisé [28], il existe d'autres formes d'apprentissage, comme l'apprentissage par renforcement et le semi-apprentissage, mais nous ne les aborderons pas ici car la conception de notre architecture ne requiert de connaissances qu'en apprentissage supervisé et non supervisé.

4.1 L'apprentissage supervisé :

L'apprentissage supervisé suppose qu'un oracle fournit les étiquettes de chaque donnée d'apprentissage. On distingue en général trois types de problèmes auxquels l'apprentissage supervisé est appliqué : la classification supervisée, la régression, et les séries temporelles. Ces trois types de problèmes se différencient en fonction du type d'étiquettes fournies par l'oracle [24]. Dans le cadre de notre travail, nous ne nous intéresserons qu'à la classification. Pour ce problème, les étiquettes sont des classes.

4.1.1 Classification supervisée :

La classification supervisée (appelée aussi classement ou classification inductive) a pour objectif « d'apprendre » par l'exemple. Elle cherche à expliquer et à prédire l'appartenance d'éléments à des classes connues a priori. Ainsi, c'est l'ensemble des techniques qui visent à deviner l'appartenance d'un individu à une classe en s'aidant uniquement des valeurs qu'il prend [29].

Avant de commencer à parler des algorithmes, il est nécessaire de présenter rapidement les fonctions de calcul de la distance entre les variables continues :

- **Distance entre les variables continues :**

Les fonctions de distances entre variables continues les plus connues sont représentées les suivantes :

Nom	Fonction
distance de Manhattan	$\sum_{i=1}^n x_i - y_i $
distance euclidienne	$\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
distance de Minkowski	$\sqrt[p]{\sum_{i=1}^n x_i - y_i ^p}$

Tableau 2.1. Les fonctions de différentes distances connues [57].

➤ Avec : x_i et y_i sont les différentes valeurs qui représentent les deux points x et y qui disposent de n dimensions.

4.1.2 Algorithmes de classification supervisée :

La plupart des algorithmes d'apprentissage supervisés tentent de trouver un **modèle** (une fonction mathématique) qui explique le lien entre des données d'entrée et les classes de sortie. Ces jeux d'exemples sont donc utilisés par l'algorithme [27].

Parmi les méthodes de classification supervisées les plus populaires, on peut citer par exemple [30]:

1. Les k plus proches voisins.
2. Naïve Bayes.
3. les arbres de décision.
4. les algorithmes génétiques.
5. les réseaux de neurones.
6. Les machines à support de vecteurs.

Dans ce qui suit nous n'allons détailler que les trois premières des méthodes précédemment citées :

❖ L'algorithme des k-plus proches voisins (k-ppv) :

La méthode des k plus proches voisins (noté parfois k -PPV ou k -NN pour k -Nearest-Neighbor) consiste à déterminer pour chaque nouvel individu que l'on veut classer, la liste des k plus proches voisins parmi les individus déjà classés. L'individu est affecté à la classe qui

contient le plus d'individus parmi ces k plus proches voisins. Cette méthode nécessite de choisir une distance (la plus classique est la distance euclidienne), et le nombre k de voisins à prendre en compte [24].

Contrairement aux autres méthodes de classification (arbres de décision, réseaux de neurones, algorithmes génétiques), il n'y a pas d'étape d'apprentissage consistant en la construction d'un modèle à partir d'un échantillon d'apprentissage. C'est l'échantillon d'apprentissage, associé à une fonction de distance et d'une fonction de choix de la classe en fonction des classes des voisins les plus proches, qui constitue le modèle [31].

Cependant, le temps de prédiction est très long, car il nécessite le calcul de la distance avec tous les exemples, mais il existe des heuristiques pour réduire le nombre d'exemples à prendre en compte [32,33].

Il est conseillé de se reporter à [34], [35] et [36] pour une étude approfondie des méthodes de plus proches voisins.

- **Le principe :**

La règle de décision par k -ppv est facile à illustrer, comme sur la figure 2.2 [37]:

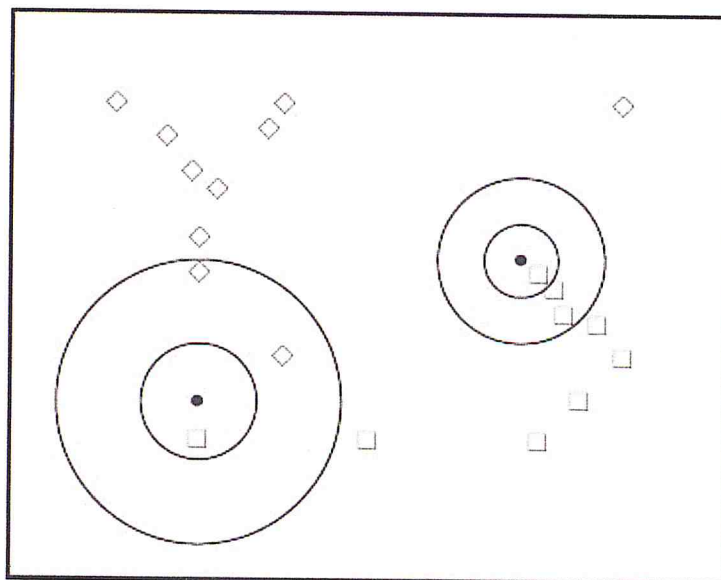


Figure 2.2. Décision par 1-ppv et 3-ppv dans un ensemble d'exemples appartenant à deux classes [37].

On y a représenté un problème à deux classes : les points à classer sont notés • et les points alentour sont les données d'apprentissage, appartenant soit à la classe notée □, soit à celle notée ◇. On cherche, au sens de la métrique choisie pour le problème (sur ce dessin,

euclidienne), les k -plus proches voisins des points \bullet ; pour $k = 1$, dans les deux cas, c'est un des points notés \square . On affecte donc les deux points \bullet à la classe \square . Pour $k = 3$, le voisinage du premier point \bullet compte deux points \diamond et un point \square : c'est la classe \diamond qui est majoritaire, et ce point est classé comme appartenant à la classe \diamond . Pour l'autre point, la décision pour $k = 3$ confirme l'appartenance à la classe \square [37].

La figure 2.3 représente la même opération pour un problème à trois classes. Pour $k = 1$, les points \bullet sont classés comme \square ; pour $k = 3$, la règle de décision produit une ambiguïté pour le premier point : on ne peut pas se décider entre les trois classes [37].

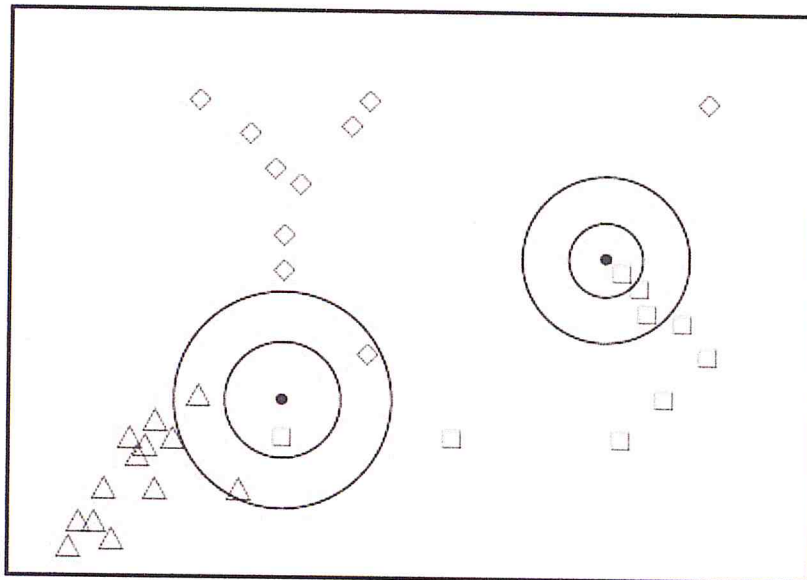


Figure 2.3. Décision par 1-ppv et 3-ppv dans un ensemble d'exemples appartenant à trois classes [37].

- **Algorithme des k -plus proches voisins [37]:**

Début

On cherche à classer le point x

pour chaque exemple (y, ω) de l'ensemble d'apprentissage faire

Calculer la distance $D(y, x)$ entre y et x

fin pour

Dans les k points les plus proches de x

Compter le nombre d'occurrences de chaque classe

Attribuer à x la classe qui apparait le plus souvent

Fin

- **Domaines d'utilisation des k plus proches voisins :**

L'algorithme des k plus proches voisins sert dans plusieurs problèmes informatiques incluant la reconnaissance des formes, la recherche dans les données multimédia, la compression

vectorielle, les statistiques informatiques et l'extraction des données [30]. Mais également, l'analyse d'images satellite et le marketing ciblé.

La méthode des k plus proches voisins pondérés est illustrée dans le cadre de la recherche de nouveaux bios marqueurs pour le diagnostic d'une pathologie complexe [38].

- **Avantages de l'algorithme des k plus proches voisins:**

Voici quelques avantages de cet algorithme:

- La méthode est facile à comprendre.
- L'algorithme est facile à mettre en œuvre [39].
- Efficace pour des classes réparties de manière irrégulière [39].
- Ne nécessite pas d'apprentissage : C'est l'échantillon qui constitue le modèle.

L'introduction de nouvelles données permet d'améliorer la qualité de la méthode sans nécessiter la reconstruction d'un modèle. C'est une différence majeure avec des méthodes telles que les arbres de décision et les réseaux de neurones [31].

- Donne des resultants clairs: Bien que la méthode ne produit pas de règle explicite, la classe attribuée à un exemple peut être expliquée en exhibant les plus proches voisins qui ont amené à ce choix [31].

- Adaptée à tout type de données : La méthode peut s'appliquer dès qu'il est possible de définir une distance sur les champs. Or, il est possible de définir des distances sur des champs complexes tels que des informations géographiques, des textes, des images, du son. C'est parfois un critère de choix de la méthode car les autres méthodes traitent difficilement les données complexes. On peut noter, également, que la méthode est robuste au bruit [31].

- **Inconvénients de l'algorithme des k plus proches voisins:**

- Nombre d'attributs : La méthode permet de traiter des problèmes avec un grand nombre d'attributs. Mais, plus le nombre d'attributs est important, plus le nombre d'exemples doit être grand. En effet, pour que la notion de proximité soit pertinente, il faut que les exemples couvrent bien l'espace et soient suffisamment proches les uns des autres. Si le nombre d'attributs pertinents est faible relativement au nombre total d'attributs, la méthode donnera de mauvais résultats car la proximité sur les attributs pertinents sera noyée par les distances sur les attributs non pertinents. Il est donc parfois utile de d'abord sélectionner les attributs pertinents [31].

- Temps de classification : Si la méthode ne nécessite pas d'apprentissage, tous les calculs doivent être effectués lors de la classification. Ceci est la contrepartie à payer par

rapport aux méthodes qui nécessitent un apprentissage (éventuellement long) mais qui sont rapides en classification (le modèle est créé, il suffit de l'appliquer à l'exemple à classer). Certaines méthodes permettent de diminuer la taille de l'échantillon en ne conservant que les exemples pertinents, mais il faut, de toute façon, un nombre d'exemples suffisamment grand relativement au nombre d'attributs [31].

➤ Stockage du modèle : Le modèle est l'échantillon, il faut donc un espace mémoire important pour le stocker ainsi que des méthodes d'accès rapides pour accélérer les calculs [31].

➤ Distance et nombre de voisins : Les performances de la méthode dépendent du choix de la distance, du nombre de voisins et du mode de combinaison des réponses des voisins. En règle générale, les distances simples fonctionnent bien. Si les distances simples ne fonctionnent pour aucune valeur de k , il faut envisager le changement de distance, ou le changement de méthode [31].

➤ La méthode est particulièrement vulnérable au fléau de la dimensionnalité.

➤ En plus de ces inconvénients, l'algorithme des k plus proches voisins utilise de nombreuses données de références (les classes de bases) pour classer les nouvelles entrées [39].

❖ La classification bayésienne [24]:

On connaît l'ensemble d'apprentissage \mathcal{A} et on cherche à classer un nouvel élément d_{new} . Le classifieur bayésien va choisir la classe C_k qui a la plus grande probabilité, on parle de règle MAP (maximum *a posteriori*) :

$$C_{MAP} = \underset{C_k \in \mathcal{C}}{\operatorname{argmax}} P(C_k | d_{\text{new}}) = \underset{C_k \in \mathcal{C}}{\operatorname{argmax}} \frac{P(d_{\text{new}} | C_k) P(C_k)}{P(d_{\text{new}})} = \underset{C_k \in \mathcal{C}}{\operatorname{argmax}} P(d_{\text{new}} | C_k) P(C_k)$$

Il faut alors estimer les probabilités $P(C_k)$ et à $P(d_{\text{new}} | C_k)$ à partir des données d'apprentissage. Les probabilités *a priori* des classes $p(C_1), p(C_2), \dots, p(C_{nc})$ peuvent être estimées facilement par :

$$P(C_k) = \frac{n_{C_k}}{n_{\mathcal{A}}}$$

où n_{C_k} est le nombre d'éléments d'apprentissage dans la classe C_k et $n_{\mathcal{A}}$ est le nombre total d'éléments dans l'ensemble d'apprentissage.

Pour estimer $P(d_{\text{new}}|C_k)$, l'approche naïve de Bayes suppose que les descripteurs de d_{new} sont indépendants. On a donc :

$$P(d_{\text{new}}|C_k) = P(f_1|C_k)P(f_2|C_k) \cdots P(f_{n_F}|C_k)$$

On peut alors estimer les probabilités $P(f_1|C_k) P(f_2|C_k) \dots P(f_{n_F}|C_k)$ en supposant qu'elles suivent un modèle connu.

- **Avantages des méthodes Naïve Bayes :**

Parmi les avantages des méthodes Naïve Bayes on peut citer par exemple [40] :

1. La facilité et la simplicité de leur implémentation.
2. Leur rapidité.
3. Les méthodes Naïve Bayes donnent de bons résultats.
4. Souvent performants même quand on a peu de données [41].

- **Inconvénients des méthodes Naïve Bayes :**

1. L'inconvénient principal de cette méthode est que bien que le coût d'entraînement du classificateur soit faible, car il ne fait que mémoriser les exemples d'entraînement, le coût de classification de nouvelles instances peut être élevé, puisque c'est à ce moment que tout le calcul se fait. Cependant, une bonne indexation des exemples aide beaucoup à pallier ce problème [42].

- ❖ **Les arbres de décision :**

Nous étudions les algorithmes de génération d'arbres de décision à partir de données. Les deux algorithmes les plus connus et les plus utilisés (l'un ou l'autre ou les deux sont présents dans les environnements de fouille de données) sont CART (Classification And Regression Trees [43]) et C5 (version la plus récente après ID3 et C4.5 [44]). Ces algorithmes sont très utilisés car performants et car ils génèrent des procédures de classification exprimables sous forme de règles.

- **Qu'est-ce qu'un arbre de décision ?**

Un arbre de décision est une représentation graphique d'une procédure de classification. Les nœuds internes de l'arbre sont des tests sur les champs ou attributs, les feuilles sont les classes.

Lorsque les tests sont binaires, le fils gauche correspond à une réponse positive au test et le fils droit à une réponse négative. Un exemple d'arbre de décision est [31]:

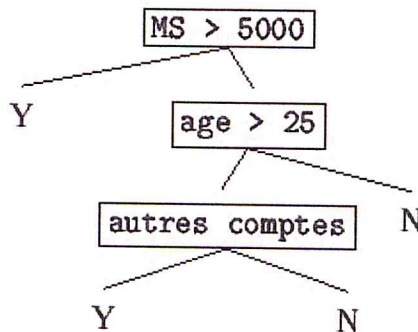


Figure 2.4 : exemple d'arbre de décision ; MS est la moyenne des soldes du compte courant, autres comptes est un champ binaire qui vaut oui si le client dispose d'autres comptes, la classe Y indique un a priori favorable pour l'attribution d'un prêt [31].

Pour classer un enregistrement, il suffit de descendre dans l'arbre selon les réponses aux différents tests pour l'enregistrement considéré. Soit l'enregistrement x défini par : nom=Digra, prénom=Omar, âge=32, MS=2550, autres comptes = oui. Cet enregistrement sera classé Y car $MS \leq 2550$ et âge > 25 et autres comptes = oui. On peut déjà remarquer quelques propriétés importantes des arbres de décision [31] :

- La procédure de classification associée est compréhensible par tout utilisateur,
- La classe associée à un enregistrement particulier peut être justifiée,
- Les attributs apparaissant dans l'arbre sont les attributs pertinents pour le problème de classification considéré.

On peut également remarquer qu'un arbre de décision est un système de règles. Il est immédiat de transformer l'arbre de la Figure 2.4 en [31]:

Si $MS > 5000$ **alors** Y

Si $MS \leq 5000$ et $age > 25$ et autres comptes = oui **alors** Y

Si $MS \leq 5000$ et $age > 25$ et autres comptes = non **alors** N

Si $MS \leq 5000$ et $age \leq 25$ **alors** N

Les systèmes de règles construits sont particuliers. En effet, pour tout enregistrement une et une seule règle s'applique, c'est-à-dire que les règles sont exhaustives et mutuellement exclusives [31].

- **Généralités sur l'apprentissage des arbres de décision :**

Idee centrale : Diviser récursivement et le plus efficacement possible les exemples de l'ensemble d'apprentissage par des tests définis à l'aide des attributs jusqu'à ce que l'on obtienne des sous-ensembles d'exemples ne contenant (presque) que des exemples appartenant tous à une même classe [45].

Dans toutes les méthodes, on trouve les trois opérateurs suivants [45]:

1. **Décider si un nœud est terminal**, c'est-à-dire décider si un nœud doit être étiqueté comme une feuille. Par exemple : tous les exemples sont dans la même classe, il y a moins d'un certain nombre d'erreurs, ...

2. **Sélectionner un test à associer à un nœud**. Par exemple : aléatoirement, utiliser des critères statistiques, ...

3. **Affecter une classe à une feuille**. On attribue la classe majoritaire sauf dans le cas où l'on utilise des fonctions coût ou risque.

Les méthodes vont différer par les choix effectués pour ces différents opérateurs, c'est-à-dire sur le choix d'un test (par exemple, utilisation du gain et de la fonction entropie) et le critère d'arrêt (quand arrêter la croissance de l'arbre, soit quand décider si un nœud est terminal). Le schéma général des algorithmes est le suivant [45]:

- **Algorithme d'apprentissage générique :**

entrée : langage de description ; échantillon S (de m enregistrements)

Début

Initialiser à l'arbre vide ; la racine est le nœud courant

Répéter

Décider si le nœud courant est terminal

Si le nœud est terminal **alors**

Affecter une classe

Sinon

Sélectionner un test et créer le sous-arbre

Fin Si

Passer au nœud suivant non exploré s'il en existe

Jusqu'à obtenir un arbre de décision

fin

- **Avantages des arbres de décision:**

- Leur capacité à travailler sur des données symboliques [46].
- Leur grande capacité et efficacité à faire de la classification [46].
- Leur facilité d'apprentissage et d'utilisation [30].
- Lisibilité du résultat : Un arbre de décision est facile à interpréter et est la représentation graphique d'un ensemble de règles. Si la taille de l'arbre est importante, il est difficile d'appréhender l'arbre dans sa globalité. Cependant, les outils actuels permettent une navigation aisée dans l'arbre (parcourir une branche, développer un nœud, élaguer une branche) et, le plus important, est certainement de pouvoir expliquer comment est classé un exemple par l'arbre, ce qui peut être fait en montrant le chemin de la racine à la feuille pour l'exemple courant [31].

- Tout type de données : L'algorithme peut prendre en compte tous les types d'attributs et les valeurs manquantes. Il est robuste au bruit [31].

- Sélection des variables : L'arbre contient les attributs utiles pour la classification. L'algorithme peut donc être utilisé comme prétraitement qui permet de sélectionner l'ensemble des variables pertinentes pour ensuite appliquer une autre méthode [31].

- Classification efficace: L'attribution d'une classe à un exemple à l'aide d'un arbre de décision est un processus très efficace (parcours d'un chemin dans un arbre) [31].

- Outil disponible: Les algorithmes de génération d'arbres de décision sont disponibles dans tous les environnements de fouille de données [31].

- Extensions et modifications : La méthode peut être adaptée pour résoudre des tâches d'estimation et de prédiction. Des améliorations des performances des algorithmes de base sont possibles grâce aux techniques de bagging et de boosting : on génère un ensemble d'arbres qui votent pour attribuer la classe [31].

- **Inconvénients des arbres de décision:**

- Sensible au nombre de classes: Les performances tendent à se dégrader lorsque le nombre de classes devient trop important [31].

➤ Évolutivité dans le temps : L'algorithme n'est pas incrémental, c'est-à-dire, que si les données évoluent avec le temps, il est nécessaire de relancer une phase d'apprentissage sur l'échantillon complet (anciens exemples et nouveaux exemples) [31].

➤ Hypothèse de non-corrélation entre attributs (les nœuds ne peuvent tester qu'un seul attribut) [47].

➤ Pas de chaînage entre les règles (règles peu lisibles et moins bien indexées) [47].

➤ Impossibilité de classer un nouvel exemple dont certains attributs sont inconnus ou imprécis [47].

➤ Ce type d'algorithmes est très sensible aux points aberrants et au bruit [46].

➤ Leur sensibilité au changement des données [30].

➤ Une détection difficile des interactions entre les variables [30].

4.2 Apprentissage non-supervisé :

Contrairement à l'apprentissage supervisé, dans l'apprentissage non-supervisé il n'y a pas d'oracle qui explicite les étiquettes. L'utilisation de ce type d'algorithme permet de trouver des structures, des dépendances entre descripteurs qui nous sont inconnues (latentes ou cachées) [24].

Le plus connu des problèmes non-supervisés est la classification non-supervisée ou *clustering*. Les classes, que nous appellerons *clusters*, sont formées par regroupement des éléments qui ont certaines caractéristiques en commun [24].

4.2.1 Classification non supervisée ou *clustering* :

Le *clustering* est un outil important pour l'analyse de données. Il vise à trouver les structures intrinsèques des données en les organisant en groupes homogènes et distincts (les *clusters*). Les objets dans un même *cluster* doivent être similaires entre eux et différents des objets des autres clusters [24].

Pour construire un regroupement de ces données, un utilisateur a trois choix méthodologiques à faire [24]:

- Choisir une mesure de ressemblance entre les données;
- Choisir le type de structure qu'il veut obtenir : partition, hiérarchie, arbre, pyramide... ;

- Choisir la méthode permettant d'obtenir la structure désirée.

On distingue généralement quatre types d'approches du *clustering* : les approches basées sur le partitionnement des données, de densités, de l'espace et les approches hiérarchiques [24].

Nous ne traiterons, dans ce qui suit que des approches basées sur le partitionnement et des approches hiérarchiques. Les algorithmes de classification non-supervisée ont de nombreux inconvénients. L'un d'eux étant le problème du choix du nombre de clusters.

Lorsque l'on souhaite découvrir de nouvelles connaissances et les regrouper en clusters, un problème est de savoir combien de clusters sont nécessaires pour bien représenter ces connaissances. Pour les classifications hiérarchiques, c'est l'utilisateur qui doit décider où couper la hiérarchie. Pour les partitions comme les k -moyennes, c'est l'utilisateur qui pourra indiquer le nombre de clusters à l'initialisation. Cependant, lorsque l'on dispose d'un grand nombre de données, l'utilisateur aura de plus en plus de mal à déterminer le nombre de clusters ou le moment où il faut arrêter le regroupement. Il existe également des critères statistiques, des heuristiques ou des formules qui permettent de déterminer ce nombre, sans toutefois pouvoir garantir que le résultat soit optimal. Pour déterminer le nombre de clusters, on peut aussi poser des contraintes : comme le nombre maximum d'éléments dans un cluster [24].

4.2.2 Algorithmes de classification non supervisée :

Nous présentons maintenant deux algorithmes classiques d'apprentissage non-supervisé l'un basés sur le partitionnement des données qui est l'algorithme des k -moyennes (ou k -means), et l'autre faisant partie des approches hiérarchiques qui est la classification ascendante hiérarchique. Nous développerons particulièrement la classification ascendante hiérarchique car nous l'utiliserons dans notre application. Avant de parler en détails de ces algorithmes, il est nécessaire de définir quelques concepts de base.

4.2.2.1 Concepts de base :

Avant de parler en détails des algorithmes cités plus haut, il est nécessaire de définir quelques concepts de base :

❖ **Définition d'une partition :**

Une **partition** est un sous-ensemble de parties deux à deux disjointes dont la réunion fait l'ensemble tout entier.

$$\begin{array}{c} \{A_1, A_2, \dots, A_k\} \text{ partition de } A \\ \Downarrow \\ i \neq j \Rightarrow A_i \cap A_j = \emptyset \\ \bigcup_{k=1}^k A_k = A \\ \{ \{a, e, f, g\}, \{b\}, \{c, d\} \} \text{ est une partition de } \{a, b, c, d, e, f, g\} \end{array}$$

Figure 2.5. Explication d'une partition [48].

❖ **Définition d'une hiérarchie :**

Une **hiérarchie** de partie de A est un ensemble de parties ayant quatre propriétés [48]:

- 1) La partie vide en fait partie
- 2) Les parties réduites à un seul élément en font partie.
- 3) L'ensemble total A lui-même en fait partie.
- 4) Si X et Y en font partie, alors soit X et Y sont disjointes, soit X contient Y, soit Y contient X.

Par exemple, l'ensemble : $\{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{a, b\}, \{e, d\}, \{a, b, c, d, e\}\}$ est une hiérarchie de parties ou encore un n-arbre [49]:

Un arbre est un graphe raciné, les feuilles sont les parties à un seul élément (qui sont toujours dans une hiérarchie), la racine est l'ensemble tout entier (qui est toujours dans la hiérarchie). Chaque partie n'a qu'un ancêtre, à l'exclusion de la racine qui n'en n'a pas (sinon on trouverait deux parties chevauchantes ce qui n'existe pas dans une hiérarchie). Si l'arbre est binaire, chaque partie a deux descendants, à l'exclusion des feuilles qui n'en n'ont pas. On dit aussi que la hiérarchie est alors **totale**ment résolue.

❖ **La matrice de proximité :**

Une matrice de proximité, P, est une matrice $m \times m$ contenant toutes les dissimilarités ou les similarités entre les objets considérés. Si p_i et p_j sont le $i^{\text{ème}}$ et le $j^{\text{ème}}$ objets, respectivement,

alors l'entrée à la *i*ème ligne et la *j*ème colonne de la matrice de proximité est la similarité, ou la dissimilarité, entre p_i et p_j .

La figure 2.6 montre quatre points, leur matrice de données et leur matrice de proximité correspondante [50].

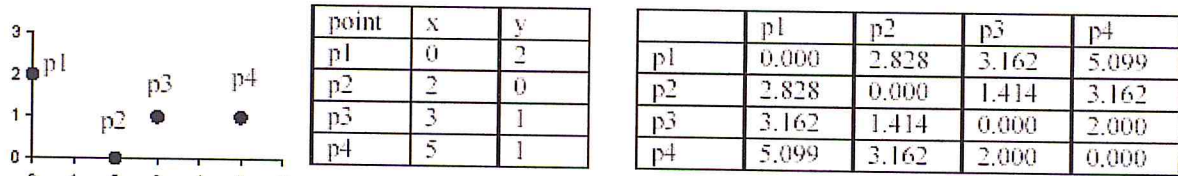


Figure 2.6. Quatre points, leur matrice de données et leur matrice de proximité [50].

4.2.2.2 Clustering par partitionnement :

Les techniques par partitionnement créent un partitionnement des points de données, d'un seul niveau. Si k est le nombre désiré de clusters, alors les approches par partitionnement trouvent typiquement tous les k clusters immédiatement [50].

Les techniques par partitionnement sont divisées en deux sous-catégories principales [51], les algorithmes basés sur les centroïdes et les algorithmes basés sur les médoïdes [50].

Ces deux techniques sont basées sur l'idée qu'un point de centre peut représenter un cluster. Pour Kmeans on emploie la notion du centroïde qui est le point de la moyenne ou la médiane d'un groupe de points [50].

❖ Algorithmes des k-moyennes :

Introduite par J. McQueen en 1971 et amélioré sous sa forme actuelle par E. Forgy, la méthode des k-moyennes est considérée comme un outil de classification efficace qui permet de diviser un ensemble de données en k classes homogènes. En effet, cette méthode initialise k clusters avec k vecteurs qui servent comme centres de gravité pour le reste des vecteurs à classifier. Chaque vecteur est ajouté dans ce cas, au cluster dont le centre est le plus proche. Les k clusters sont produits de façon à minimiser la fonction objective suivante [52,53]:

$$E = \sum_{r=1}^k \sum_{X_i \in C_r} (X_i - g_r)^2$$

[30]

Où:

C_r : représente l'ensemble des classes.

X_i : Un point qui appartient à une classe C_r .

g_r : Le point moyen de la classe C_r .

Dans le domaine de la classification non supervisée, cet algorithme cherche à partitionner l'espace des données en classes isolées les unes des autres, et cela, en minimisant la variance entre ces derniers.

On peut résumer le fonctionnement de l'algorithme des K-moyennes dans les étapes suivantes [53]:

1. On choisit k objets au hasard qu'on considère comme des centres pour les classes initiales.
2. On affecte chaque objet au centre le plus proche pour obtenir une partition de k classes.
3. On recalcule les centres de chaque classe.
4. La répétition des étapes 2 et 3 jusqu'à la stabilité des centres.

- **Avantages de l'algorithme des k moyennes :**

La méthode des K-moyennes représente plusieurs avantages, comme par exemple [30]:

- Sa complexité linéaire.
- Sa facilité.
- Sa convergence rapide.
- Son adaptation à de larges bases de données.
- L'ordre d'entrée des objets n'a aucune influence sur les résultats de cette méthode.
- Convient à tout type de données : En choisissant une bonne notion de distance, la méthode peut s'appliquer à tout type de données (mêmes textuelles) [31].
- Facile à implanter : La méthode ne nécessite que peu de transformations sur les données (excepté les normalisations de valeurs numériques), il n'y pas de champ particulier à identifier, les algorithmes sont faciles à implanter et sont, en règle générale, disponibles dans les environnements de data mining [31].

- **Inconvénients de l'algorithme des k moyennes :**

Malgré les grands avantages de cette méthode, elle a aussi des inconvénients qu'on peut résumer dans les points suivants [30]:

- Le nombre d'objets k est fixé au début, ce qui influence les résultats.
- Sa sensibilité aux éléments marginaux.
- Sa mauvaise gestion pour les clusters mal isolés.
- Problème du choix de la distance: Les performances de la méthode (la qualité des groupes constitués) sont dépendantes du choix d'une bonne mesure de similarité ce qui est une tâche délicate surtout lorsque les données sont de types différents [31].
 - Le choix des bons paramètres : La méthode est sensible au choix des bons paramètres, en particulier, le choix du nombre k de groupes à constituer. Un mauvais choix de k produit de mauvais résultats. Ce choix peut être fait en combinant différentes méthodes, mais la complexité de l'algorithme augmente [31].
 - L'interprétation des résultats : Il est difficile d'interpréter les résultats produits, i.e. d'attribuer une signification aux groupes constitués. Ceci est général pour les méthodes de segmentation [31].

4.2.2.3 Clustering hiérarchique :

Le fondement du clustering hiérarchique est de créer une hiérarchie de clusters . A la racine de l'arbre est associé un unique cluster contenant l'ensemble des objets de la base, puis plus on descend dans l'arbre, plus les clusters sont spécifiques à un certain groupe d'objets considérés comme similaires. La figure 2.8 montre une telle hiérarchie [54].

Afin de former une telle hiérarchie de clusters, il existe deux méthodes principales :

1. **La méthode ascendante** : démarrante avec autant de clusters que d'objets initiaux dans la base, puis fusionnant successivement les clusters considérés comme les plus similaires, jusqu'à ce que tous les objets soient réunis dans un unique cluster stocké à la racine de la hiérarchie formée [54].
2. **La méthode descendante** : démarrante avec un unique cluster contenant l'ensemble des objets de la base, puis divisant successivement les clusters de manière à ce que les clusters résultants soient les plus différents possible, et ce jusqu'à obtenir aux feuilles de la hiérarchie autant de clusters que d'objets dans la base [54].

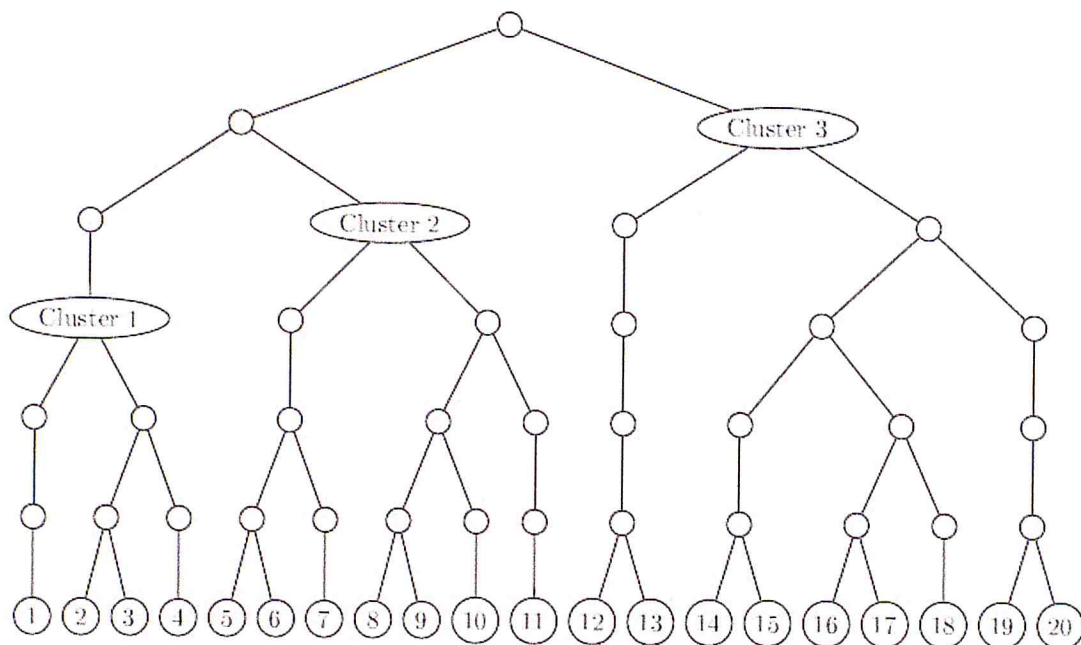


Figure 2.7. Exemple de clustering hiérarchique [54].

❖ **La Classification Ascendante Hiérarchique (CAH) :**

La Classification Ascendante Hiérarchique (CAH) [55] est un algorithme classique de *clustering* hiérarchique. Il appartient à la méthode des heuristiques [24], voici les étapes de cet algorithme [50]:

1. Calculer la matrice de proximité.
2. Fusionner les deux clusters les plus proches (les plus similaires).
3. Mettre à jour la matrice de proximité pour refléter la proximité entre le nouveau cluster et les clusters originaux.
4. Répéter les étapes 2 et 3 jusqu'à ce que seulement un seul cluster reste.

• **Choix d'un indice de dissimilarité :**

De nombreuses mesures de la "distance" entre individus ont été proposées. Le choix d'une (ou plusieurs) d'entre elles dépend des données étudiées [56]. Les plus connues sont la distance euclidienne, manhatan et Minkowski déjà citées plus haut.

• **Dendrogramme :**

Pour visualiser une classification hiérarchique, on peut utiliser un dendrogramme (arbre hiérarchique indicé). Un dendrogramme représente la hiérarchie sous la forme d'un arbre

binaire, où les données sont contenues dans les feuilles. La hauteur des nœuds de l'arbre indique généralement la distance entre les *clusters*. La figure 2.8 en donne un exemple. Cette forme de représentation facilite la visualisation de la hiérarchie [24].

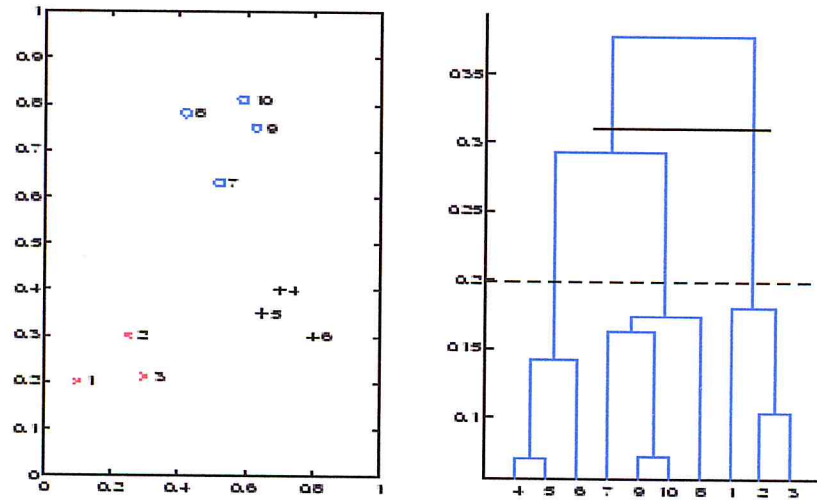


Figure 2.8. Dendrogramme (hiérarchie indicée) obtenu à partir des données (à gauche) au moyen d'une CAH utilisant le critère du lien simple. La droite en pointillé montre une coupure à la distance 0.2 (indice 0.2) donnant une partition à 3 *clusters*. La droite en trait plein propose une coupure par rapport au nombre de *clusters* (ici, on a coupé à 2 *clusters*) [24].

- **Critères d'agrégation :**

Il existe un grand nombre de distances entre deux *clusters*, distances appelées critères d'agrégation, nous exposons les critères les plus connus. En fonction du critère utilisé, les CAH obtenues pourront être très différentes. En effet, chacun de ces critères a des caractéristiques particulières, il est donc très important de bien choisir le critère en fonction du type de *clusters* que l'on souhaite obtenir [24].

- **Méthode du lien simple (Single link) :**

Pour la version du lien simple du clustering hiérarchique, la proximité de deux clusters est définie par la distance minimale entre n'importe quels deux points dans les clusters différents. Cette distance minimale entre les points appartenant aux clusters *A* et *B* est calculée avec la formule [50]:

$$d(A, B) = \min_{x \in A, x \in B} d(x, y) \quad [57]$$

Cette technique est bonne pour la manipulation de formes non elliptiques, mais elle est sensible aux bruits. La méthode du lien simple est une des méthodes les plus utilisées. La

figure 2.10 donne un échantillon de matrice de similarité pour cinq points (p1 – p5) et le dendrogramme qui montre les séries des fusions qui résultent de l'utilisation de la technique du lien simple [50].

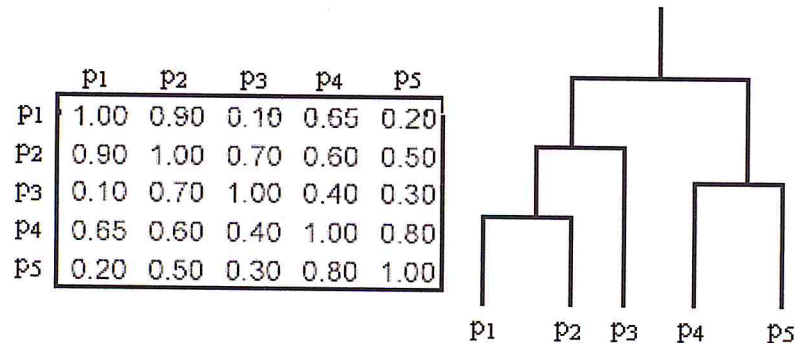


Figure 2.10. Exemple d'une matrice de similarité et le dendrogramme correspondant l'application de la méthode du lien simple [50].

➤ **Méthode du lien complet (Complete link) :**

Pour la version du lien complet du clustering hiérarchique, la proximité de deux clusters est définie par la distance maximale entre n'importe quels deux points dans les clusters différents. Cette distance maximale entre les points appartenant aux clusters A et B est calculée avec la formule [50] :

$$d(A, B) = \max_{x \in A, y \in B} d(x, y) \quad [57]$$

La méthode du lien complet est moins susceptible au bruit, mais peut fractionner de grands clusters, et a des problèmes avec les formes convexes. La figure 2.11 donne un échantillon de matrice de similarité pour cinq points (p1 – p5) et le dendrogramme qui montre les séries des fusions qui résultent de l'utilisation de la technique du lien complet [50].

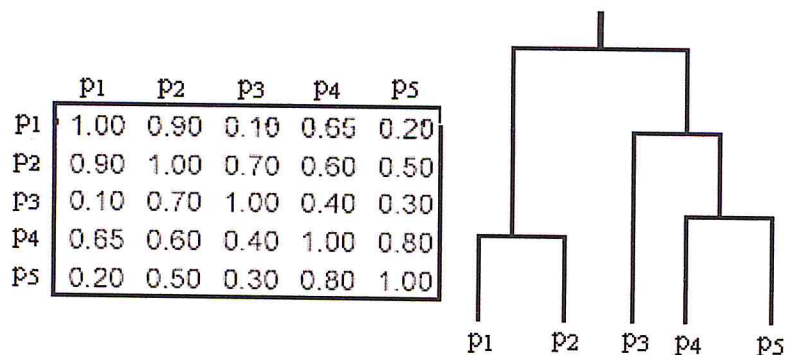


Figure 2.11. Exemple d'une matrice de similarité et le dendrogramme correspondant à l'application de la méthode du lien complet [50].

➤ **Méthode du lien moyen (Average link) :**

Pour la version du lien moyen du clustering hiérarchique, la distance de deux clusters est définie par la moyenne des distances entre toutes les paires de points dans les clusters différents. C'est une approche intermédiaire entre la méthode du lien complet et celle du lien simple. Ceci est exprimé par l'équation suivante [50]:

$$d(A, B) = \frac{1}{n_A n_B} \sum_{x \in A, y \in B} d(x, y) \quad \text{où } n_A \text{ et } n_B \text{ représente la taille des clusters } A \text{ et } B \text{ respectivement [57].}$$

La figure 2.12 donne un échantillon de matrice de similarité et le dendrogramme montre les séries des fusions qui résultent de l'utilisation de l'approche du lien moyen. Le clustering hiérarchique dans ce cas simple est le même comme produit par la méthode du lien simple [50].

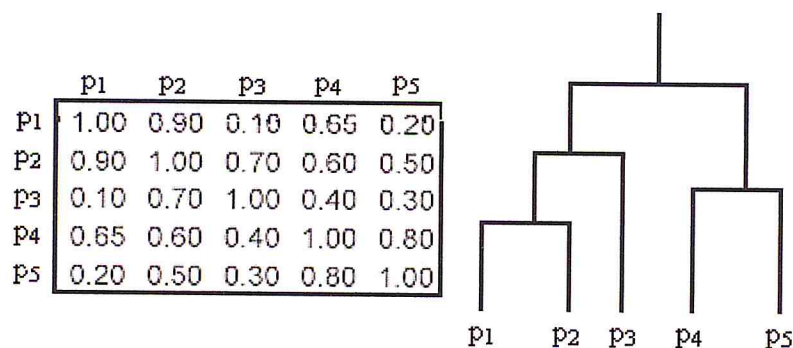
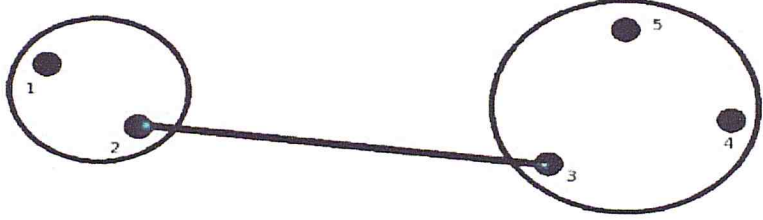
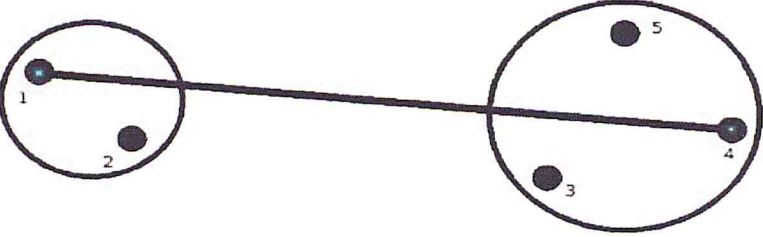


Figure 2.12. Exemple d'une matrice de similarité et le dendrogramme correspondant à l'application du lien moyen [50]

Ces critères, résumés dans le tableau 2.2, sont :

Critère	$\delta^*(C_k, C_{k'})$	Remarques
Lien simple	$\min_{d_T \in C_k, d_S \in C_{k'}} \delta(d_T, d_S)$  <p style="text-align: center;">delta(d2, d3)</p>	<p>distance simple à calculer, pas de recalcul des distances, mais sensible au bruit (effet de chaîne), crée un petit nombre de grands clusters</p>
Lien complet	$\max_{d_T \in C_k, d_S \in C_{k'}} \delta(d_T, d_S)$  <p style="text-align: center;">delta(d1, d4)</p>	<p>distance simple à calculer, pas de recalcul des distances, sensible mais moins que le lien simple, crée un grand nombre de petits clusters</p>

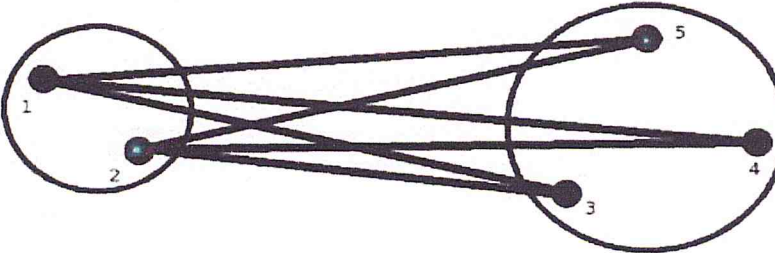
Lien moyen	$\frac{\sum_{d_r \in C_k} \sum_{d_s \in C_{k'}} \delta(d_r, d_s)}{n_{C_k} \times n_{C_{k'}}$  $1/6(\delta_{13} + \delta_{14} + \delta_{15} + \delta_{23} + \delta_{24} + \delta_{25})$	nécessite le recalcule des distance à chaque agrégation, mais bon compromis entre lien simple et lien complet
---------------	--	--

Tableau 2.2. Résumé des principaux critères d'agrégation [24].

- **Coupure de l'arbre :**

Pour obtenir une partition à partir de l'arbre hiérarchique (dendrogramme), on peut faire le choix de couper horizontalement le dendrogramme. La partition obtenue est composée des *clusters* restant sous la coupe [24].

Pour obtenir une bonne partition, on peut choisir manuellement de couper à un niveau où les branches de l'arbre sont longues, indiquant que les données contenues dans les *clusters* sont très différentes. Lorsque l'on a un grand nombre de données, il suffit de visualiser les nœuds les plus près de la racine. Un autre critère de coupe possible est de choisir en fonction de la distance entre les clusters. La droite en pointillée de la figure 2.8 montre une coupure à 0.2. Cependant, il est difficile de déterminer automatiquement quelle est la bonne valeur de coupe. Un autre critère est de couper en fonction du nombre de *clusters* que l'on souhaite obtenir. Cependant, le nombre de regroupements intrinsèques aux données n'est en général pas connu. Il est donc difficile de déterminer automatiquement quel sera le bon critère de coupe [24].

- **Avantages de la classification ascendante hiérarchique:**

- Le principal avantage des HAC réside dans leur capacité à générer des partitions emboîtées. En effet, ces méthodes proposent un ensemble de partitions (des solutions) représentées sous forme d'arbre, où l'utilisateur aura la possibilité de choisir la solution qui convient le mieux à ses besoins [58]. Cette représentation sous forme d'arbre qui met en évidence une information supplémentaire : l'augmentation de la dispersion dans un groupe produit par une agrégation. L'utilisateur peut dès lors avoir une idée du nombre adéquat de classes en choisissant la partition correspondant au saut le plus élevé dans l'augmentation de la dispersion au sein des classes [58].

➤ Si l'on compare avec les méthodes non-hiérarchiques traditionnelles, comme le *k-means*, la CAH ne nécessite pas de connaître le nombre de *clusters a priori*. De plus, il n'y a pas de fonction d'initialisation [24].

➤ La CAH est une méthode heuristique. Il est donc difficile d'apporter une justification à ce type de méthode. Cependant, dans [59], une interprétation probabiliste de la CAH, basée sur une estimation par maximum de vraisemblance des modèles de mélange, est proposée [24].

➤ Enfin, une seule construction (équivalent à une itération pour les méthodes de partitionnement) suffit pour atteindre le critère d'optimisation [24].

- **Inconvénients de la classification ascendante hiérarchique:**

➤ Le principal inconvénient de la classification ascendante hiérarchique est qu'elle nécessite le calcul des distances entre individus pris deux à deux. Ce qui est très rapidement prohibitif dès que la taille du fichier excède le millier d'individus [58].

➤ L'utilisation de plusieurs types de Métriques, pour mesurer la distance entre deux clusters, peut générer des résultats différents [30].

5. Conclusion :

Dans ce chapitre, nous avons donc présenté et défini l'apprentissage automatique, et détailler l'apprentissage supervisé et non supervisé en détaillant quelques uns des algorithmes propre à chaque type avec les avantages et les inconvénients de chaque algorithmes.

Dans le chapitre suivant, qui concerne la conception de l'architecture du système, nous allons nous servir des connaissances acquises dans ce chapitre, afin de choisir le bon algorithme à implémenter pour chaque étape en fonction de son type, de son fonctionnement, des informations qu'il requiert ainsi que de ses avantages et inconvénients.

Chapitre III :
Architecture du
systeme



1. Introduction :

Dans ce chapitre, nous allons détailler une à une les parties de l'architecture de notre application. Celle-ci propose un simulateur de stationnement à l'utilisateur, où il aura à stationner un véhicule dans un parking. Et en fonction de la manière avec laquelle il va stationner ce véhicule, le système doit pouvoir donner des informations sur sa personnalité.

L'application est destinée aux appareils sous android (smartphones, tablettes tactiles, PDA et terminaux mobiles). Et ce, afin que son utilisation soit plus pratique et pour attirer un grand nombre d'utilisateurs.

L'architecture comporte deux parties, en fonction des étapes qui permettront de réaliser le but final. Dans un premier temps, plusieurs personnes dont nous avons la description de la personnalité utiliseront le simulateur de stationnement. Après ça, *la classification hiérarchique ascendante* sera utilisée pour classifier leurs différentes manières de se stationner. Nous aurons donc comme résultat des manières de se stationner qui correspondront à des types de personnalité. Ensuite, l'utilisateur désirant connaître sa personnalité à partir de sa façon de se stationner, utilisera le simulateur. Sa manière de se stationner sera lue, et comparée à celles des utilisateurs précédents. Dans cette étape, *l'algorithme des k plus proches voisins* sera implémenté pour attribuer à la manière de se stationner de cet utilisateur une classe parmi celles résultantes de la première étape. Pour finir le type de personnalité correspondant à la classe allouée sera renvoyé à l'utilisateur comme étant le sien.

L'idée d'utiliser *la classification hiérarchique ascendante* dans la première étape, et *l'algorithme des k plus proches voisins* dans la seconde, nous est venue d'une étude sur l'imitation des stratégies des joueurs humains dans les jeux vidéo de combats [21], conduite par Saini, S.S., Dawson, C.W. et Chung, P.W.H. Cette étude a été citée et détaillée dans le premier chapitre. Les auteurs avaient utilisé *la classification hiérarchique ascendante* pour classifier les mouvements et combinaisons de mouvements exécutés par un joueur humain au cours d'un match de boxe dans un jeu vidéo. Pour pouvoir ensuite reproduire la façon de jouer du joueur humain par un agent machine, *l'algorithme des k plus proches voisins* a été utilisé au niveau opérationnel, pour déterminer selon les paramètres du jeu, en temps réel, quels actions doivent être effectuées par l'agent machine afin d'imiter le joueur humain en question. Bien que notre travail ne soit pas tout à fait dans le même registre que cette étude, nous avons cru bon de la citer ici, car nous nous en sommes inspirés.

2. Fonctionnement général du système :

Le système a donc pour but, de donner des indications sur la personnalité d'un utilisateur X en se basant sur l'analyse de son comportement lors du choix de son emplacement de stationnement dans un parking. Son comportement et ses choix seront analysés en relevant les caractéristiques de la place choisie, et d'autres détails sur le comportement qu'il a eu en choisissant cet emplacement. L'utilisateur en question se servira d'un simulateur de stationnement qui a été développé afin de reproduire les conditions d'un parking réel.

➤ **Remarque :** Dans ce qui suit, nous allons nommer « utilisateur X », celui qui se sert du système pour connaître sa personnalité en fonction de sa manière de se stationner, et ce afin de ne pas confondre avec les utilisateurs participants à la collecte de données. L'architecture du système comporte deux parties :

2.1 Première partie :

Dans cette partie, il s'agit d'établir les liens entre les personnalités (ou certaines de leurs facettes ou dimensions) d'un groupe d'individus et leurs comportements lors du choix de leurs emplacements de stationnement. Autrement dit, mettre en évidence les traits ou dimensions de la personnalité qui influencent chacun dans les choix qu'il fait en se stationnant. Donc, nous devrions avoir comme résultats de cette étape des associations comme celle-ci (1) :

Tel comportement et choix lors du stationnement correspondent à Tel type de personnalité (1)

Ces associations seront obtenues après le traitement des résultats d'une collecte de données réalisée auprès du groupe d'individus précédemment cité. Ces individus auront préalablement passé un test mesurant leurs personnalités (*le questionnaire IPIP-NEO*). Ensuite, ils expérimenteront le simulateur de stationnement après avoir entré et enregistré les résultats qu'ils ont obtenus au test. Le simulateur représente un simple parking avec plusieurs places libres, le but est de stationner une voiture à l'un des emplacements libres. Le comportement et les choix d'un individu seront quantifiés et enregistrés sous la forme d'un vecteur multi-variable V qui décrira sa manière de se stationner. Donc, chaque utilisateur aura un vecteur « V » représentant sa manière de se stationner en fonction des choix qu'il a fait et du comportement qu'il a eu. Ces vecteurs seront ensuite regroupés en clusters (en utilisant la classification hiérarchique ascendante) réunissant ainsi les comportements (et choix de stationnement) qui se ressemblent dans les mêmes classes (*ou clusters*), chaque classe représente donc une certaine manière de se stationner, ce qui est la première partie d'une

association de type (1). Dans chacune des classes résultantes, un type de personnalité commun à tous les individus sera défini en fonction des résultats de ces derniers au test de personnalité, les tendances communes entre les individus seront retenues. Ainsi, seront obtenues les associations de type (1). La figure 3.1 représente un schéma illustrant la partie dont nous venons de parler.

➤ **Remarque :** Le groupe d'individus est un échantillon ciblé qui ne comporte que des individus de sexe masculin dans la tranche d'âge allant de 25 à 30 ans.

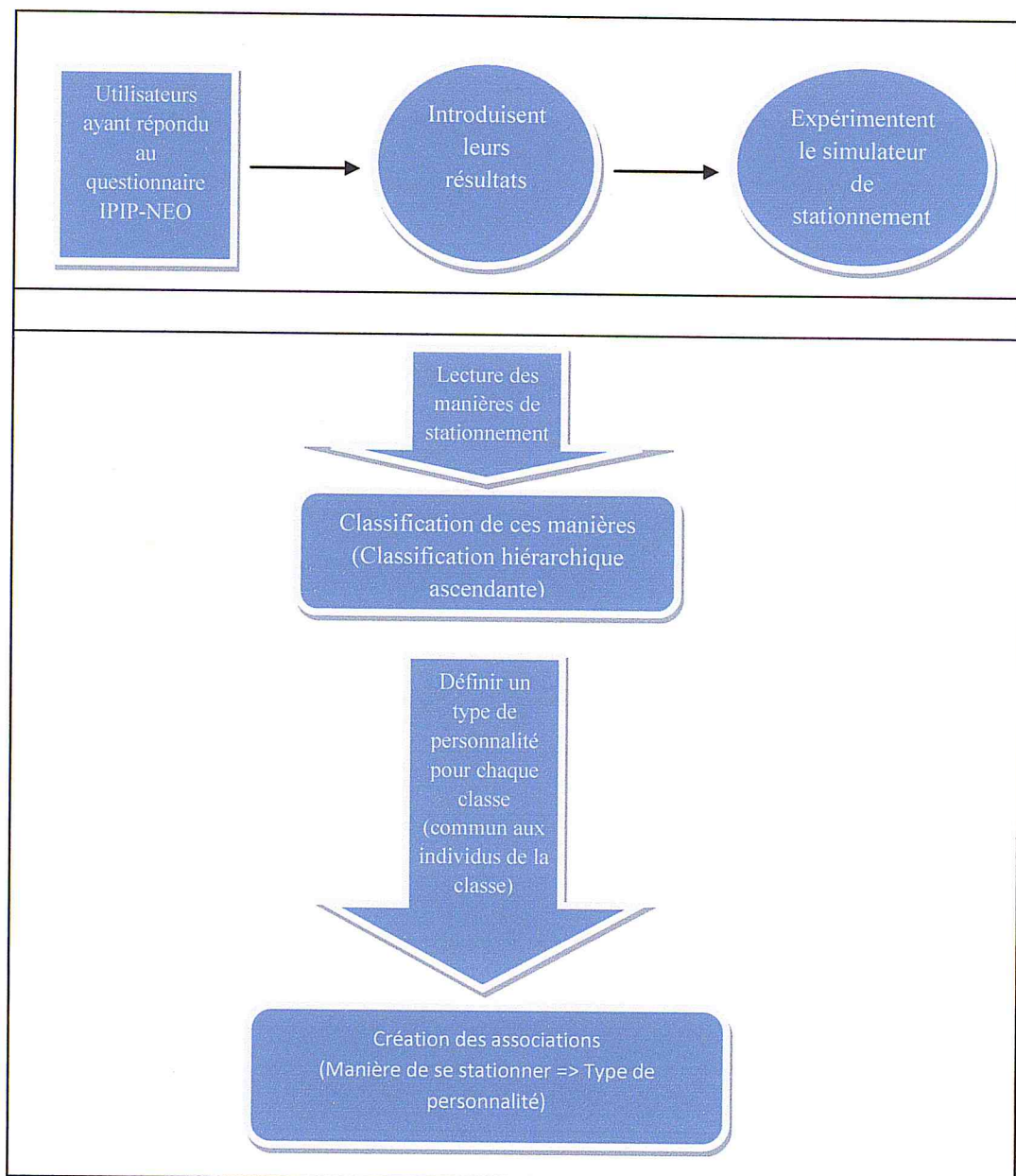


Figure 3.1. Schéma illustrant la première partie de l'architecture.

2.2 Deuxième partie :

Cette partie concerne le module qui va permettre de fournir à un utilisateur X des indications sur sa personnalité en se basant sur les choix qu'il a fait et le comportement qu'il a eu lors du stationnement. Ceci est fait en enregistrant son comportement et les choix qu'il a fait en se stationnant sous forme d'un vecteur tout comme avec les utilisateurs de la collecte de données, ce vecteur sera nommé V_x , il sera ensuite comparé à ceux du groupe d'individus ayant participé à la collecte de données. Après cette comparaison, une classe (parmi celles résultantes de la classification des vecteurs « V » obtenus après de la collecte de données) sera allouée au vecteur V_x (ceci à l'aide de l'algorithme des k-plus proches voisins). Et compte tenu de l'association de type (1) qu'aura la classe allouée au vecteur V_x , le type de personnalité correspondant au vecteur V_x sera renvoyé à l'utilisateur comme étant le sien. La figure 3.2 représente un schéma illustrant cette deuxième partie.

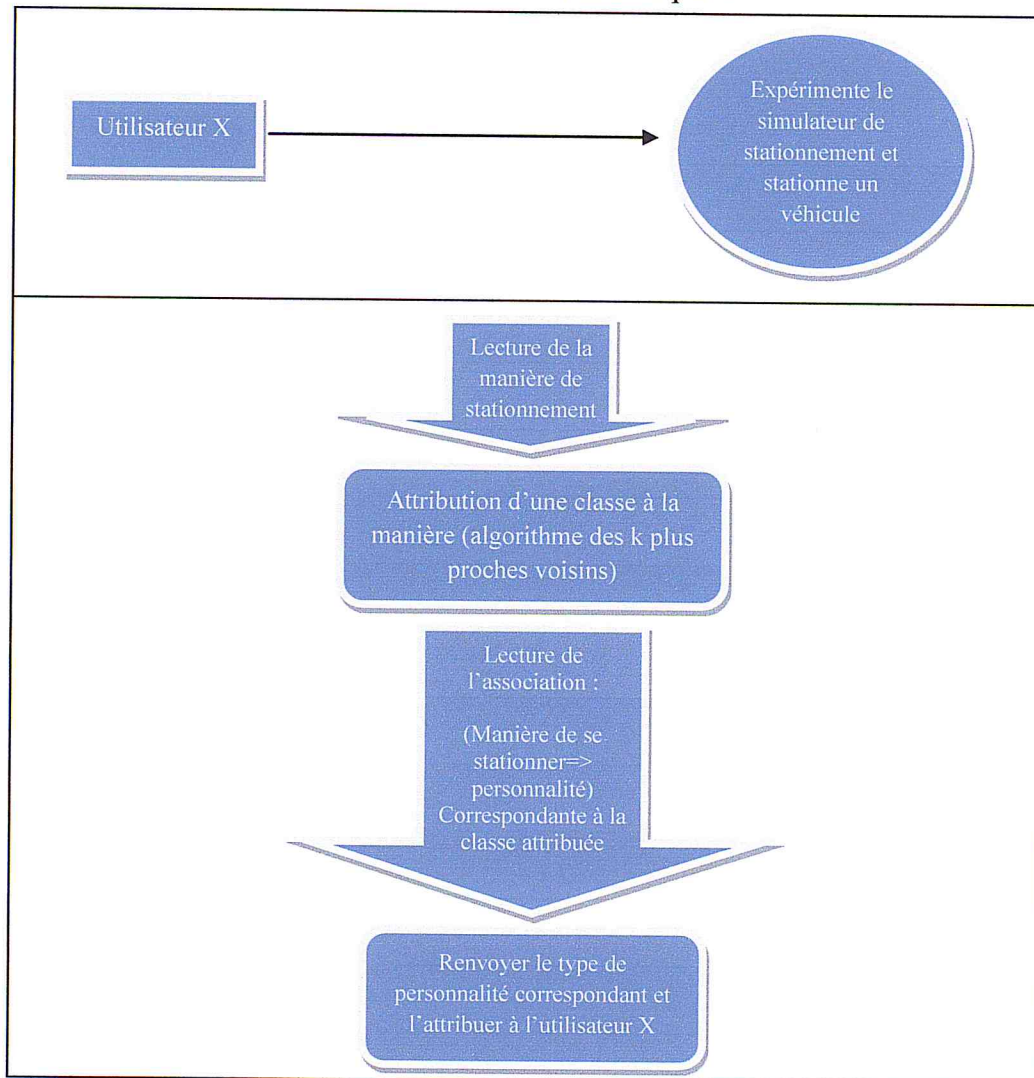


Figure 3.2. Schéma illustrant la deuxième partie de l'architecture.

Nous allons voir plus en détails ces deux parties dans ce qui suit (dans les parties 4 et 5), mais avant cela, il est nécessaire de décrire en détails le simulateur de stationnement.

3. Description du simulateur de stationnement et du vecteur « V »:

Le simulateur offre un choix à l'utilisateur entre deux parkings, l'un sécurisé et payant et l'autre gratuit mais non sécurisé. Les deux parkings sont identiques et contiennent le même nombre de places libres. Ces places varient entre elles en fonction de plusieurs caractéristiques, certaines sont étroites, d'autres non. Certaines se situent dans des coins ou autres endroits difficilement accessibles, ou d'où la sortie serait compliquée. D'autres sont commodes pour l'utilisateur mais les choisir causerait une gêne aux conducteurs des voitures proches de ces places. Et enfin, il y a des places dont les deux emplacements adjacents (gauche et droit) sont occupés, d'autres où l'un deux seulement l'est et d'autres où les deux emplacements adjacents sont libres.

Le parking n'a qu'une seule entrée, une seule sortie pour véhicules et une seule sortie pour piétons, il y aura donc des places plus éloignées de la sortie piétonne que d'autres, idem pour la sortie véhiculée et l'entrée. Le conducteur devra également composer avec cet élément quand il choisira son emplacement de stationnement.

En ce qui concerne la visibilité de l'utilisateur, il est rare que dans le monde réel, un conducteur puisse apercevoir l'ensemble des places disponibles dans un parking depuis n'importe quel point de ce dernier. Dans un souci de réalisme, le parking a été divisé en quatre parties. L'utilisateur ne pourra apercevoir que la partie (et les places libres qu'elle contient) dans laquelle il se trouve. De plus, une étude menée par Meiping, Y., Ruisong, Y et Xiaoguang, Y. en 2009 sur « *les critères sur lesquels se base un conducteur pour choisir un emplacement de stationnement, compte tenu des informations qu'il a sur la zone qui l'entoure* » [5], a établi le lien entre les choix que fait un conducteur lors du stationnement et sa connaissance du parking (ou de la zone qui l'entoure de manière générale). Donc, nous avons inclus ce paramètre parmi les caractéristiques à mettre en avant pour décrire la manière de se stationner de chacun. De plus, le nombre de déplacements entre les différentes parties visitées peut être le témoin de l'hésitation d'un utilisateur à choisir son emplacement de stationnement.

La figure 3.3 montre les quatre parties du parking. On peut y apercevoir l'entrée du parking (partie 4, en bas, à droite), la sortie piétonne (partie 1, en haut, à droite, elle est également

visible depuis la partie 2, en haut, à gauche). Il y a également des flèches au sol indiquant à l'utilisateur quelle direction prendre pour visiter d'autres parties du parking.

La disposition des places a été conçue de manière à donner le choix à l'utilisateur entre des places différentes les unes des autres de part leur caractéristiques.

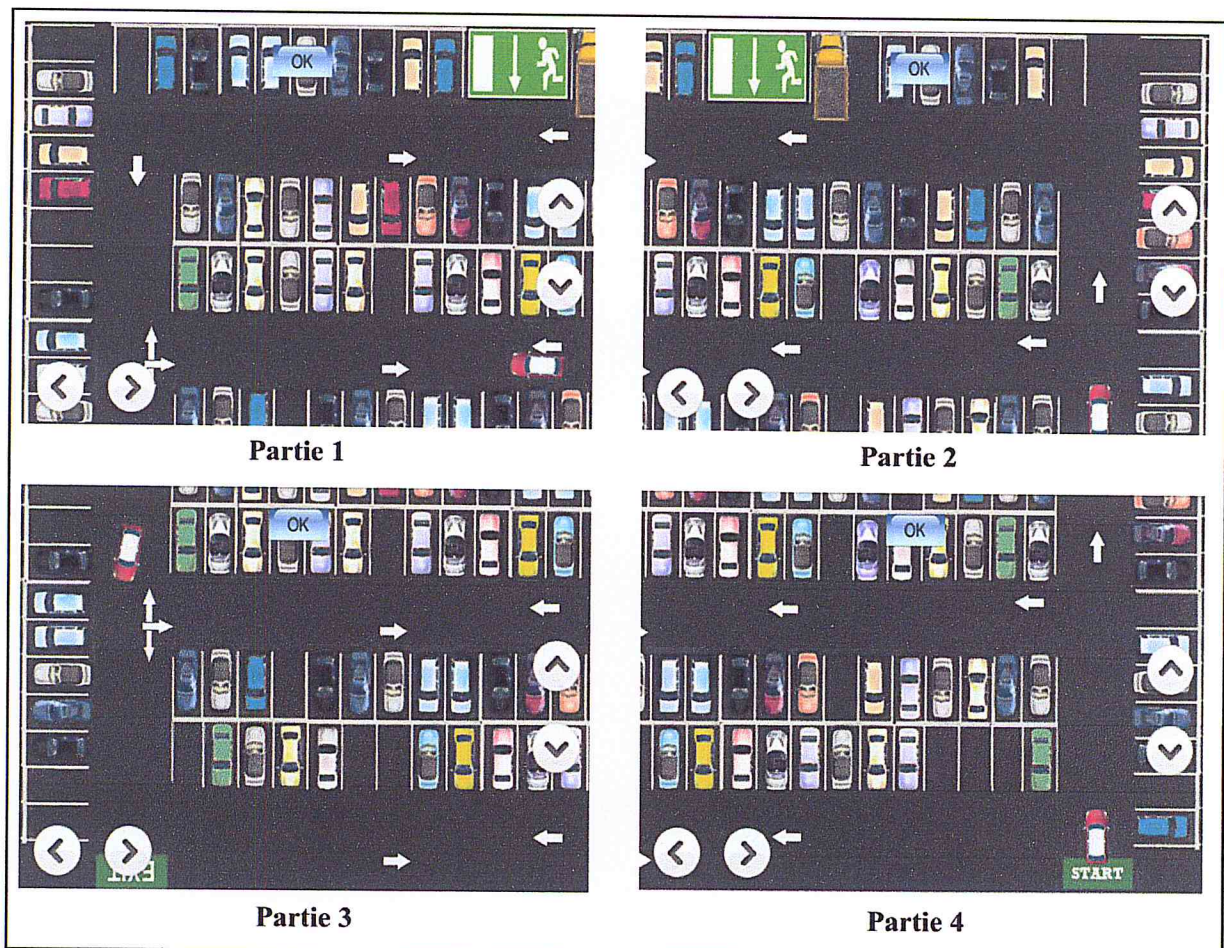


Figure 3.3. Les quatre parties du parking.

Nous avons choisi les caractéristiques d'emplacement de stationnement et les autres facteurs à mettre en avant pour décrire le comportement d'un conducteur lors du choix de son emplacement de stationnement en se basant sur les résultats de différentes études et enquêtes menées sur les facteurs influençant un conducteur dans son choix d'emplacement de stationnement et dont nous avons parlé au chapitre I. Par exemple, l'étude [11] conduite par Meiping, Y., Ruisong, Y. et Xiaoguang, Y. en 2008, a révélé que les facteurs les plus influents étaient : « la distance de marche à pied après stationnement du véhicule » (ici, la distance entre la place choisie et la sortie piétonne), « la sûreté de l'emplacement » (ici l'utilisateur a le

choix entre un parking sécurisé ou non), « les informations sur le parking » (ici, toutes les parties du parking ne sont pas visibles du premier coup), « l'état de la place » (ici représenté par l'ensemble des caractéristiques des places) et la « charge du véhicule » (qui ne nous intéresse pas, le véhicule à stationner étant le même pour tous les utilisateurs). De plus, lors d'une étude conduite en 2003, par Borgers, A., Timmermans, H. et Van Der Waerden, P. sous le titre de : « Travelers Micro-Behavior at Parking Lots: A Model of Parking Choice Behavior » [12], les auteurs ont tenté de déterminer quels facteurs étaient privilégiés lors du choix d'un emplacement de stationnement en fonction de l'âge, du sexe ou du but du voyage des conducteurs. Les facteurs considérés par les auteurs de cette étude, comme déterminant lors du choix d'un emplacement de stationnement ont été les mêmes que ceux que nous avons choisi de mettre en avant, en plus d'autres.

D'autre part, certaines des places libres du parking ont été conçues avec des caractéristiques de manière à ce que le choix de l'une ou de l'autre par l'utilisateur puisse être révélateur de certains aspects de sa personnalité. Par exemple, le choix d'une place qui pour soi est commode mais dont l'occupation rend difficile la sortie d'un autre, pourrait être celui d'une personne ayant des scores bas dans les dimensions « Conscience » ou « agréabilité ». Ceci est la supposition du Dr. John A. Johnson (Co-développeur du questionnaire IPIP-NEO), que nous avons questionné par mail au sujet des liens possibles entre manière de se stationner et personnalité.

Le parking de ce simulateur a donc été conçu de façon à ce que l'application réponde à la problématique suivante : « *Quels facteurs un conducteur prend le plus en considération en se stationnant compte tenu de sa personnalité ?* ».

La figure 3.4 représente un schéma qui résume l'ensemble des paramètres que nous avons cité. Ceux-ci seront ensuite quantifiés sous forme d'un vecteur à plusieurs variables (voir tableau 3.1).

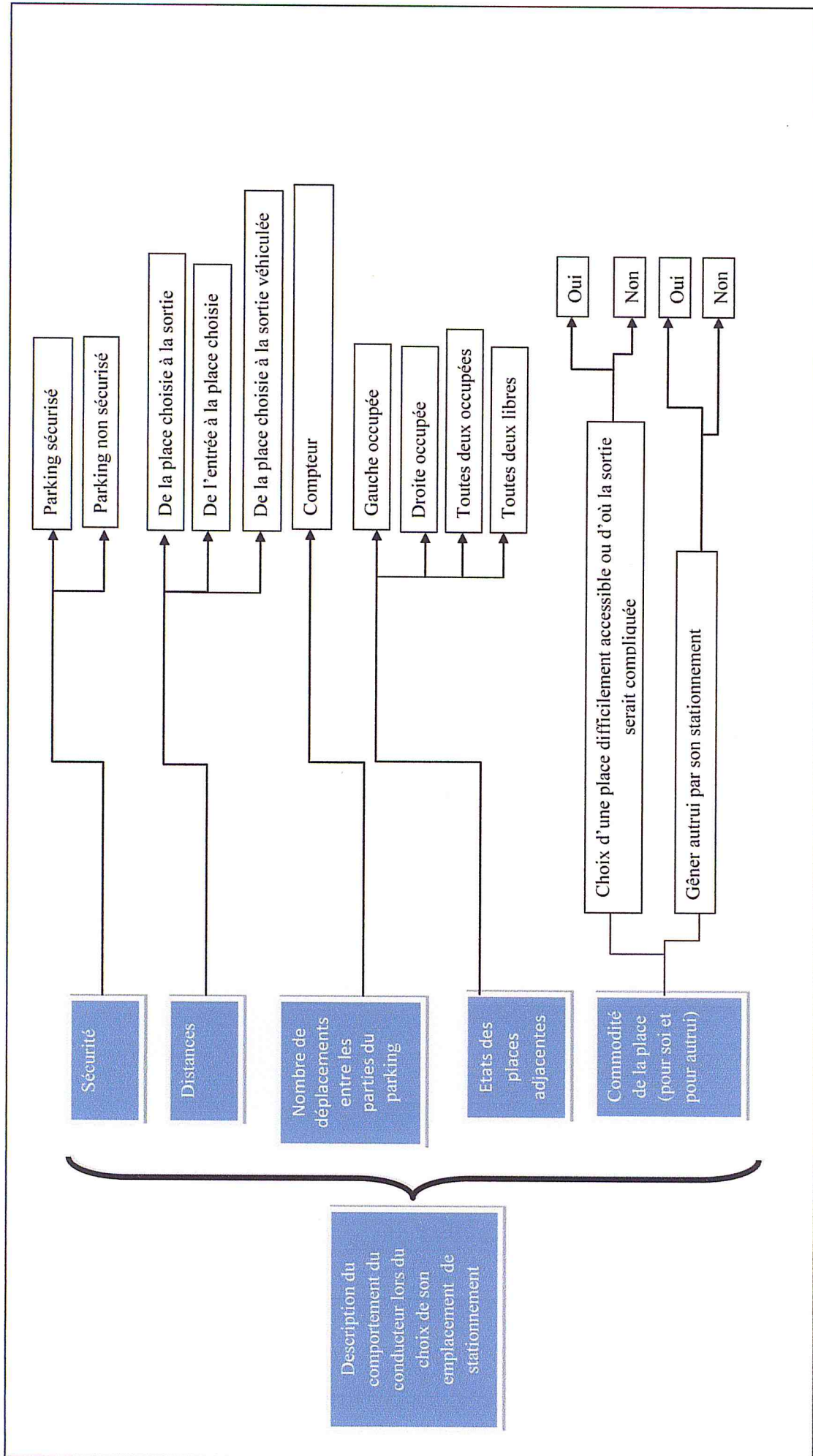


Figure 3.4. Schéma résumant l'ensemble des paramètres pris en compte pour évaluer le type de stationnement d'un utilisateur.

Les caractéristiques de la place choisie et les autres facteurs pris en compte par l'utilisateur sont quantifiés par un vecteur de valeurs réelles $V(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$. Les paramètres que représentent les valeurs (x_1, \dots, x_9) sont cités dans le tableau 3.1.

Variables (réelles)	Ce qu'elles représentent	Valeurs possibles
X_1	Parking sécurisé ou non	0 ou 1
X_2	Place adjacente gauche occupée ou non	0 ou 1
X_3	Place adjacente droite occupée ou non	0 ou 1
X_4	Place située à un emplacement difficile d'accès ou d'où la sortie serait difficile ou non	0 ou 1
X_5	Place dont le choix gêne autrui ou non	0 ou 1
X_6	Distance (entrée à place choisie)	Valeur réelle
X_7	Distance (place choisie à sortie piétonne)	Valeur réelle
X_8	Distance (place choisie à sortie véhiculée)	Valeur réelle
X_9	Nombre de déplacements entre les parties du parking	Valeur réelle (compteur)

Tableau 3.1. Liste des variables du vecteur « V » et des paramètres que chacune représente avec les valeurs possibles de chacune.

4. Collecte et traitement des données :

C'est dans cette étape, que sera constituée l'ensemble d'apprentissage. Plusieurs utilisateurs vont utiliser le système. Chacun d'entre eux devra tout d'abord répondre au questionnaire IPIP-NEO, et introduire les scores qu'il a obtenu dans chacun des cinq facteurs.

Dans notre étude, le moyen choisi pour évaluer la personnalité des individus est le questionnaire IPIP-NEO, qui est basé sur le modèle des cinq grands facteurs ou *big five model* (le test IPIP-NEO et le modèle à 5 facteurs ont été décrits en détails dans le chapitre I). Le test IPIP-NEO évalue donc la personnalité selon 5 grands facteurs qui sont : **L'ouverture, la conscience, l'extraversion, l'agréabilité et le névrotisme**. Il note chacun de ces cinq facteurs sur une échelle de 0 à 100 et interprète ensuite les résultats pour chaque facteur, comme suit :

Si la note d'un facteur est entre 0 et 33, le score est jugé comme bas dans le facteur en question, si elle est entre 34 et 67, le score est considéré comme moyen et au-delà de 67, le score est considéré comme élevé. Après avoir passé le test, l'utilisateur qui se sert de l'application introduit le score qu'il a obtenu dans chaque facteur, en chiffres (de 0 à 100), ensuite un traitement sera fait sur chaque score pour le classer comme étant bas, moyen ou élevé. Il est à noter que dans les textes et tableaux à venir, ce que nous entendrons par score d'un individu dans un facteur est l'une des mentions : « bas », « moyen » ou « élevé ».

Nous avons choisi d'utiliser le test IPIP-NEO, en raison de sa gratuité, de sa popularité et de la notoriété du modèle des cinq grands facteurs dans le domaine de la mesure de la personnalité. Le choix de sa version courte plutôt que la longue, a été orienté principalement par des raisons d'ordres pratiques, pour que le test ne soit pas trop long à passer pour les utilisateurs.

Après avoir introduit les résultats qu'il a obtenu au test IPIP-NEO, chacun des utilisateurs expérimentera le simulateur de stationnement, où il stationnera son véhicule en tenant compte d'un certains nombre de critères. Les caractéristiques de son choix seront enregistrées dans un vecteur « V » (que nous avons décrit en détails dans la partie 3).

Lorsqu'un nombre suffisant d'utilisateurs aura expérimenté le système, la collecte de données prendra fin et nous aurons donc un tableau enregistré dans un fichier externe semblable au tableau 3.2. Ce tableau contient à chaque ligne, le vecteur « V » décrivant le choix de l'utilisateur dans le simulateur de stationnement et les résultats qu'a obtenu ce même utilisateur dans le questionnaire IPIP-NEO.

4.1 Classification des vecteurs « V » :

Chaque individu qui utilise le simulateur de stationnement pour participer à la collecte de données va faire des choix : type de parking (sécurisé ou non), choix de l'emplacement (plusieurs places possibles avec diverses caractéristiques pour chacune). Et l'utilisateur peut

parcourir les quatre parties du parking autant de fois qu'il le souhaite, ces choix vont être différents d'un utilisateur à un autre. De ce fait, les vecteurs « V » décrivant les manières de se stationner des utilisateurs participant à cette collecte de données ne seront pas tous homogènes. Il est donc nécessaire, d'effectuer une classification de type non supervisée, qui aura pour but de regrouper entre eux, les vecteurs similaires dans les mêmes clusters. Ceci est une préparation à l'étape suivante, qui aura pour but d'allouer une classe parmi ces dernières au vecteur V_x , d'un utilisateur X , qui teste l'application pour connaître son type de personnalité en fonction de ses choix lors du stationnement.

Les vecteurs « V » sont regroupés en utilisant la classification hiérarchique ascendante avec la méthode du lien complet) :

$$D(X_i, X_j) = \max_{x \in X_i} \max_{y \in X_j} d(x, y) \quad [21]$$

Pour rappel, la classification hiérarchique ascendante a pour principe de démarrer avec autant de clusters que d'objets initiaux dans la base, puis de fusionner successivement les clusters considérés comme les plus similaires, jusqu'à ce que tous les objets soient réunis dans un unique cluster stocké à la racine de la hiérarchie formée [54]. Cet algorithme a été vu en détails dans le Chapitre II, consacré à l'apprentissage automatique.

La classification hiérarchique ascendante a été choisie pour cette étape, car bien connue, facile à implémenter, et ne nécessitant aucune connaissance à priori sur le nombre de clusters, ce qui convient parfaitement à notre cas.

Dans la classification hiérarchique ascendante, il existe trois méthodes pour définir la proximité entre deux clusters (voir chapitre II). Nous avons opté pour la méthode du lien complet où la proximité de deux clusters est définie par la distance maximale entre n'importe quels deux points dans les clusters [50]. Nous avons choisi cette méthode plutôt que les deux autres car avec la méthode du lien complet, *la distance est simple à calculer, pas de recalcul des distances, sensible au bruit mais moins que le lien simple, crée un grand nombre de petits clusters* [24]. En créant un grand nombre de petits clusters, les éléments de ces derniers seront plus homogènes qu'avec la méthode du lien simple et c'est ce que l'on cherche à obtenir afin de négliger le moins de variables possibles des vecteurs V , et ainsi négliger le moins de facteurs décrivant le stationnement que possible.

La distance entre les éléments initiaux (en l'occurrence les vecteurs V) pour former la matrice de proximité sera calculée en utilisant la fonction de calcul de la distance euclidienne : $\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$ également expliqué dans le chapitre II.

4.2 Le type de personnalité correspondant à chaque classe :

Comme le montre la figure 3.6, après la classification des vecteurs, le tableau initial est scindé en plusieurs tableaux en fonctions des classes obtenues. Chaque classe, aura donc son tableau avec la liste des utilisateurs, leurs vecteurs et leurs scores dans le questionnaire. Le type de personnalité de l'utilisateur X est celui qui correspond à la classe qu'on lui a attribué (*comme expliqué dans la parties 2de ce chapitre*). Un type de personnalité est décrit en fonctions des scores obtenus dans chacun des cinq facteurs. Seulement voilà, il se pourrait très bien que dans l'un ou plusieurs des tableaux correspondant aux différentes classes, il y ait une disparité entre les résultats obtenus par les individus dans un même facteur de la personnalité. C'est pour cela, que pour chaque facteur, on détermine le score majoritaire (en calculant les pourcentages d'apparitions de chaque score obtenu). Ce score sera considéré comme celui de cette classe dans ce facteur de la personnalité. Si dans un facteur donné, le pourcentage d'apparition du score majoritaire ne dépasse pas 50%, cela signifie qu'il y a une disparité trop importante entre les scores obtenus par les individus de la classe dans ce facteur, et celui-ci sera éliminé de la description de la personnalité correspondante à cette classe.

Prenons l'exemple du tableau 3.2, nous avons quatre individus ; les utilisateurs U_1, U_2, U_3 et U_4 , leurs choix de stationnement sont représentés par les vecteurs V_1, V_2, V_3 et V_4 . Leurs manières de se stationner sont semblables, vu que leurs vecteurs « V » ont été regroupés dans le même cluster, mais leurs résultats dans le questionnaire IPIP-NEO, nous montrent qu'ils n'ont pas tout à fait le même type de personnalité. Si jamais la classe allouée au vecteur V_x de l'utilisateur X est celle-ci, on saura que sa manière de se stationner ressemble à celles des utilisateurs de cette classe, mais il sera difficile de fournir à l'utilisateur X des informations sur sa personnalité vu que les individus de cette classe n'ont visiblement pas le même tous type de personnalité. Il faut donc déterminer un type de personnalité commun à tous les individus de cette classe. Pour cela, il suffit de déterminer les scores majoritaires obtenus par les individus de cette classe dans chacun des cinq facteurs. Et ce, en calculant pour chaque facteur, le pourcentage d'apparition de chaque score obtenu. Donc, pour chaque colonne du

tableau 3.2, on calcule le pourcentage d'apparition de chacun des scores (élevé, bas et moyen).

Utilisateurs	Vecteurs «V »	Résultats IPIP-NEO				
		Ouverture	Conscience	Extraversion	Agréabilité	Névrotisme
U1	V ₁	Moyen	Elevé	Moyen	Bas	Moyen
U2	V ₂	Moyen	Elevé	Moyen	Elevé	Bas
U3	V ₃	Bas	Elevé	Elevé	Bas	Elevé
U4	V ₄	Bas	Elevé	Moyen	Bas	Moyen
U5	V ₅	Bas	Elevé	Bas	Bas	Bas

Tableau 3.2. Exemple d'un tableau résultant de la collecte de données.

Dans l'exemple du tableau 3.2, on peut constater que les scores majoritaires dans chaque facteur sont les suivants : « Bas » pour l'ouverture avec 60 %, « Elevé » pour la conscience avec 100%, « Moyen » pour l'extraversion avec 60% et « Bas » pour l'agréabilité avec 80%. Pour le névrotisme, les deux scores majoritaires (à égalité, 40% chacun) sont les scores « Bas » et « Moyen », dans ce cas, on doit éliminer le facteur « Névrotisme » de la description de la personnalité de cette classe car le pourcentage d'apparition du score majoritaire n'excède pas 50%. Donc, si la classe allouée au vecteur V_x , de l'utilisateur X est celle-ci, la description de sa personnalité se fera de cette façon : « *Les individus se stationnant d'une manière semblable à la votre ont eu à 60% des scores bas en ouverture, à 100% des scores élevés en conscience, à 60% des scores bas en extraversion et à 80% des scores bas en agréabilité* ». La description de la personnalité ne contiendra donc aucune information sur le facteur névrotisme. En d'autres termes, nous considérons que se stationner d'une manière semblable à celle des individus de cette classe, n'est révélateur d'aucune tendance en ce qui concerne le névrotisme. La figure 3.5 montre un organigramme décrivant le processus dont nous avons parlé dans cette étape.

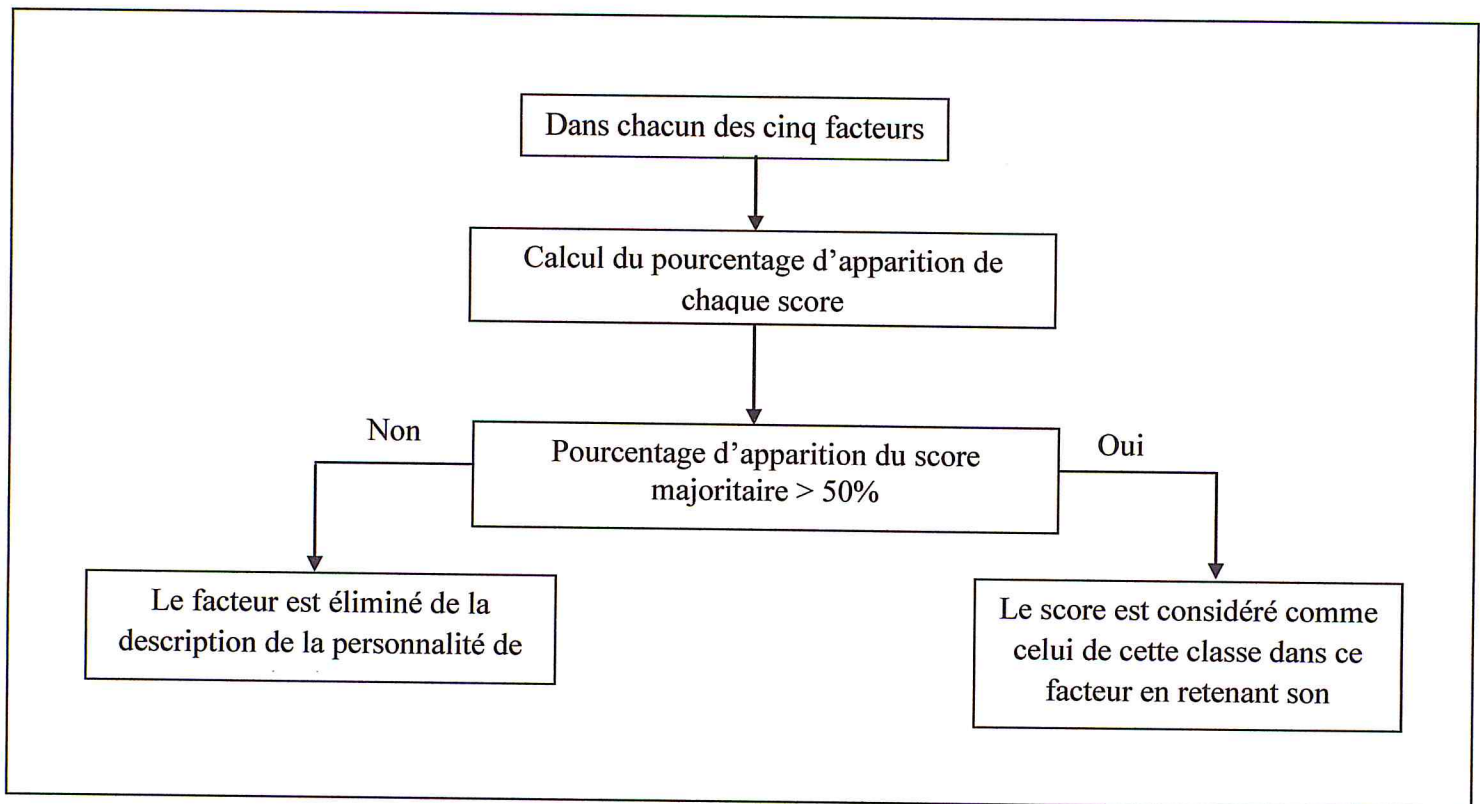
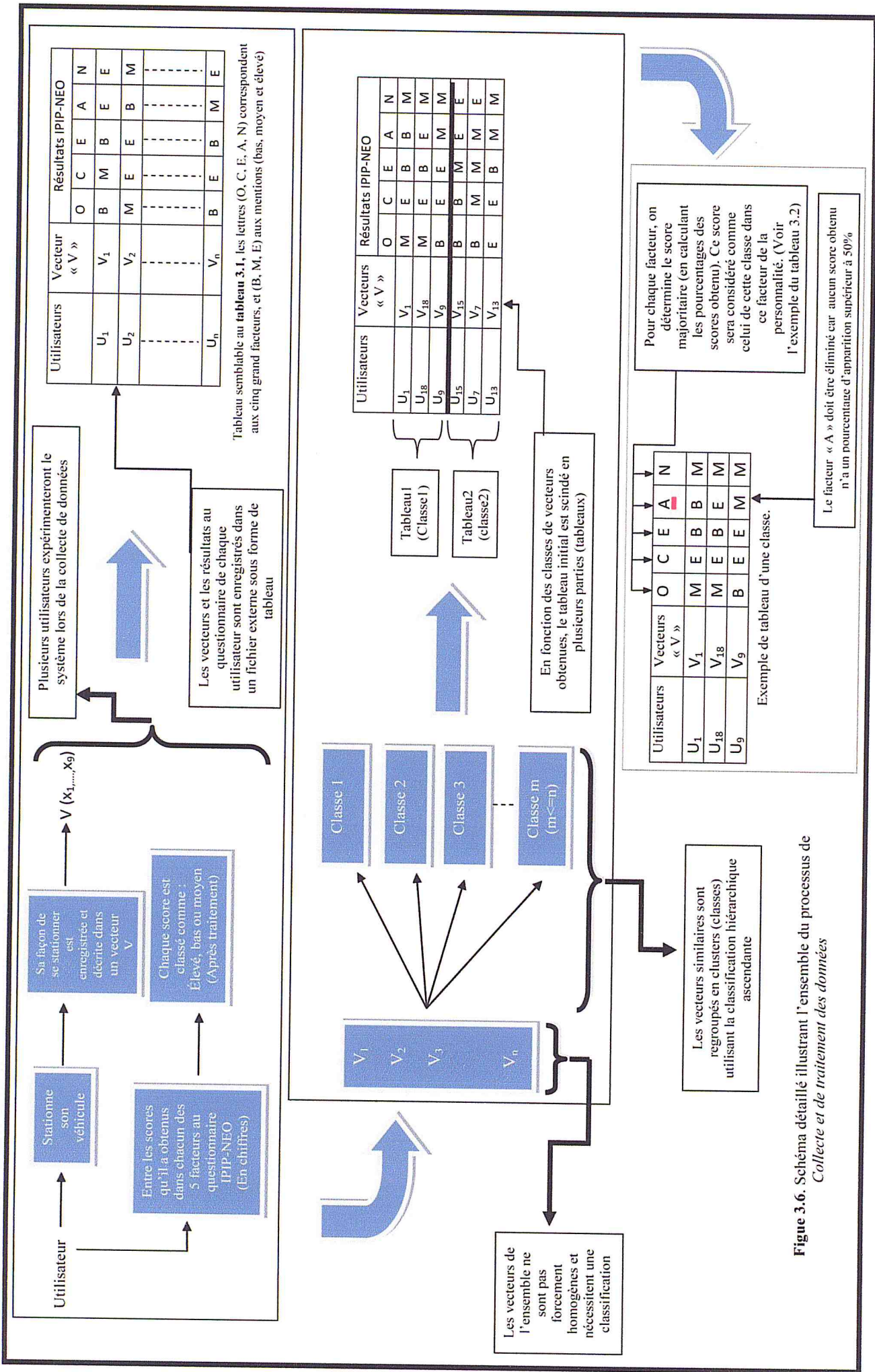


Figure 3.5. Organigramme décrivant le processus d'élimination des facteurs considérés comme non influent.

➤ **Remarque :** Si dans une classe, tous les facteurs sont éliminés comme ce fut le cas pour le névrotisme dans notre exemple, alors nous considérons que la manière de se stationner représentée par cette classe ne correspond à aucun type particulier de personnalité. Et si le vecteur V_x , de l'utilisateur X (qui utilise le simulateur pour connaître sa personnalité selon sa manière de se stationner) est alloué à cette classe, on n'aura aucune indication sur sa personnalité et voici le message qui lui sera renvoyé : « *La manière avec laquelle vous avez stationné votre véhicule ne nous donne aucune indication sur votre type de personnalité* ».

La figure 3.6 représente un schéma détaillé illustrant l'ensemble de la partie « *Collecte et traitement des données* ».



5. Description de la personnalité d'un utilisateur « X » selon ses choix lors du stationnement :

Dans la partie précédente, nous avons décrit le processus de collecte et de traitement des données, notamment par la classification des vecteurs « V » décrivant le stationnement des utilisateurs ayant participé à la collecte de données. Ces mêmes utilisateurs avaient introduit les résultats qu'ils avaient obtenus au questionnaire IPIP-NEO mesurant leur personnalité.

Nous avons également vu qu'un type de personnalité a été défini pour chaque classe, celui-ci correspond aux tendances communes entre les différentes personnalités des individus constituant une classe. Chaque classe résultante de la classification des vecteurs « V » a donc un type de personnalité qui lui correspond et qui est décrit uniquement par les facteurs dans lesquels les individus de cette classe ont eu des scores semblables.

Si nous revenons à ce que nous avons dit dans la partie 2 de ce chapitre, le but de la première étape (collecte de données et leur traitement) est d'obtenir des associations de type (1), que nous avons défini par :

Tel comportement et choix lors du stationnement correspondent à Tel type de personnalité (1)

Après la collecte et le traitement des données, de telles associations seront donc établies, elles seront représentées par des tableaux semblables au *tableau 3.1*. Ces tableaux seront enregistrés dans un fichier externe et serviront de base de données d'apprentissage. Donc, des liens ont été établis entre les manières de se stationner et les différents types de personnalité.

L'application peut à présent donner des indications à un utilisateur X en se basant sur l'analyse de son comportement lors du choix de son emplacement de stationnement dans le parking du simulateur.

Pour cela, un utilisateur X utilise le simulateur de stationnement, où il aura à stationner une voiture dans un parking, il pourra choisir entre parking sécurisé ou non et parmi une multitude de places ayant des caractéristiques différentes (voir partie 3 de ce chapitre). L'utilisateur X sera donc amené à faire un certain nombre de choix, à privilégier des facteurs plutôt que d'autres, l'ensemble des caractéristiques de la place choisie et autres facteurs pris en compte

par l'utilisateur X sera enregistré et quantifié sous forme d'un vecteur V_x de la même façon qu'avec les utilisateurs ayant participé à la collecte de données. Ce vecteur sera ensuite comparé à ceux obtenus lors de la collecte de données. Après comparaison une classe sera allouée à ce vecteur, tout ceci à l'aide de l'algorithme des k -plus proches voisins, cet algorithme a été décrit en détails dans le chapitre II.

La méthode des k plus proches voisins consiste à déterminer pour chaque nouvel élément que l'on veut classer, la liste des k plus proches voisins parmi les individus déjà classés. L'élément est affecté à la classe qui contient le plus d'élément parmi ces k plus proches voisins. Cette méthode nécessite de choisir une distance (la plus classique est la distance euclidienne), et le nombre k de voisins à prendre en compte [24].

- **Algorithme des k -plus proches voisins [37]:**

Début

On cherche à classer le point x

pour chaque exemple (y, ω) de l'ensemble d'apprentissage faire
calculer la distance $D(y, x)$ entre y et x

fin pour

Dans les k points les plus proches de x

compter le nombre d'occurrences de chaque classe

Attribuer à x la classe qui apparaît le plus souvent

Fin

Dans notre cas, nous avons choisit la distance euclidienne et pour le nombre de voisins à prendre en compte, cela va dépendre des résultats de la collecte de données, du nombre de classes obtenus à considérer et du nombre d'éléments à chaque classe.

Ainsi, et compte tenu de ses nombreux avantages, en particulier sa simplicité et le fait qu'il soit facile à implémenter, l'algorithme des k plus proches voisins a été choisi pour cette étape.

Au final, la classe qui sera allouée au vecteur V_x , de l'utilisateur X , contiendra donc des vecteurs relevés depuis des utilisateurs (parmi les participants à la collecte de données) ayant fait à peu près des choix similaires à ceux de l'utilisateur X . Chaque classe étant associée à un type de personnalité (relation de type (1)), le type de personnalité correspondant à classe allouée au vecteur V_x , sera donc renvoyer en résultat à l'utilisateur X comme étant le sien avec des commentaires et des interprétations comme expliqué dans la partie 3 de ce chapitre.

La figure 3.7 représente un schéma détaillé illustrant l'étape dont nous venant de parler.

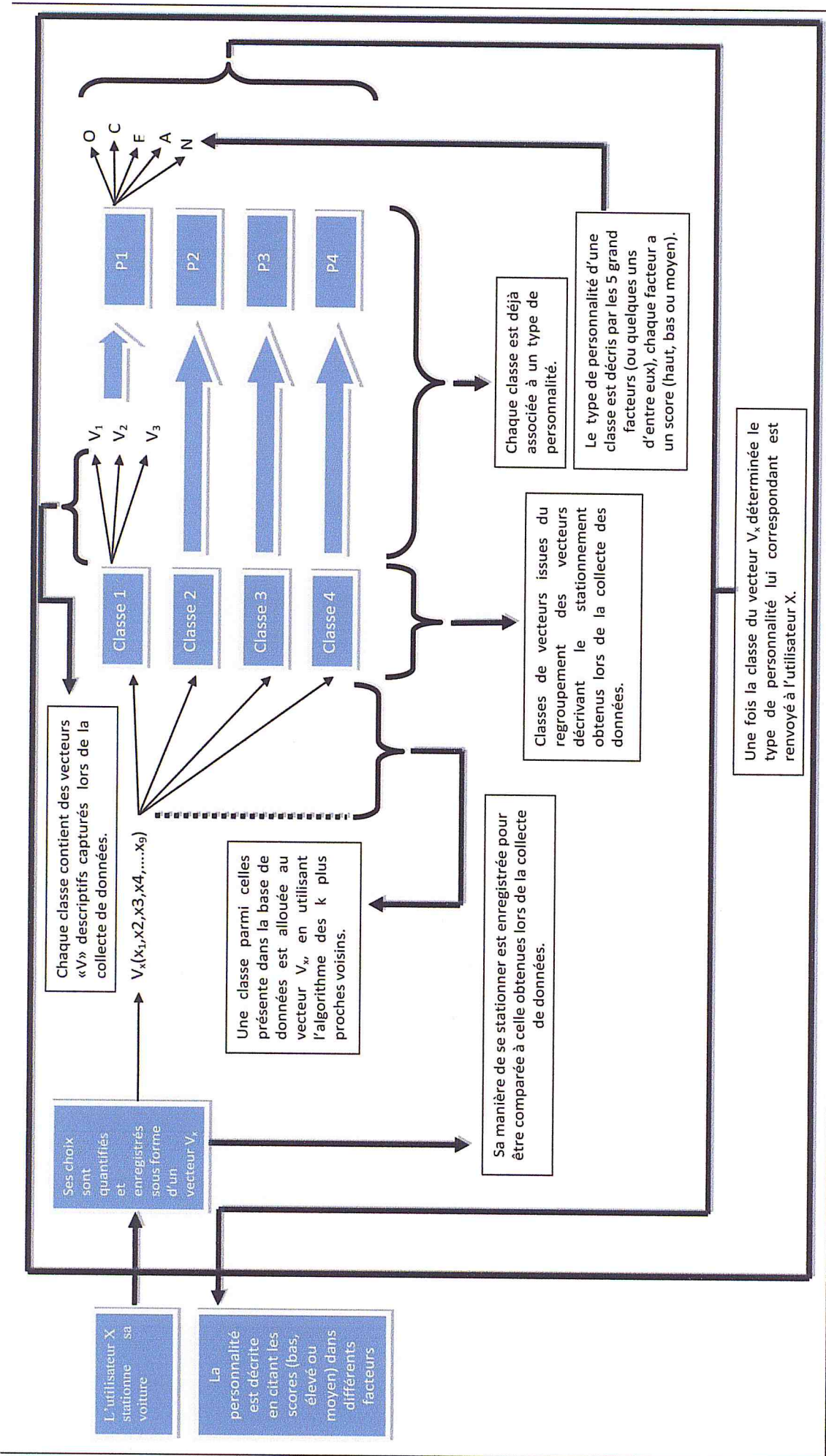


Figure 3.7. Schéma détaillé illustrant le processus de description de la personnalité de l'utilisateur X selon ses choix lors du stationnement.

6. Conclusion :

L'architecture proposée dans ce chapitre, devrait pouvoir réaliser le but de notre travail, qui est de donner des indications sur la personnalité d'un individu à partir de son comportement et des choix qu'il fait quand il stationne un véhicule dans un parking.

Les résultats peuvent dépendre du nombre d'individus ayant participé à la collecte de données, ainsi que du type de population ciblée, car le questionnaire IPIP-NEO évalue la personnalité d'un individu en fonction des réponses qu'il donne au questionnaire. Mais les personnes sont notées différemment selon leurs âges et leurs genres. Ici, et comme nous l'avons déjà précisé, la population ciblée est de sexe masculin et dans la tranche d'âge allant de 25 à 30 ans. Donc les individus souhaitant connaître leurs personnalités en utilisant le système devront être des hommes entre 25 et 30 ans.

Pour ce qui est du nombre d'individus participant à la collecte de données, dans un premier temps, et dans le but de tester ce système, la collecte de données ne visera qu'une dizaine de personnes. Après cela, un utilisateur désirant connaître sa personnalité selon sa manière de se stationner pourra utiliser l'application et constater le résultat. Mais après cette première collecte de données, d'autres utilisateurs pourront encore alimenter l'ensemble d'apprentissage, en introduisant leurs résultats obtenus au questionnaire IPIP-NEO, avant d'utiliser le simulateur de stationnement, tout comme les personnes ayant participé à la collecte de données initiale. Ainsi, le système sera, au fur et à mesure de son utilisation, plus efficace pour déterminer le type de personnalité d'un utilisateur. Et ce, grâce à un ensemble d'apprentissage qui s'agrandit. De ce fait, l'application se conforme à l'une des caractéristiques majeures des systèmes d'apprentissage automatique, qui est d'être plus efficaces au fur et à mesure de leurs utilisations.

Dans le chapitre suivant, nous allons parler des outils de travail nécessaires au développement de l'application. Nous décrirons également toutes ses fonctionnalités. Il sera aussi et surtout question du test de l'application et des résultats obtenus qui seront analysés et commentés.

Chapitre IV :

Implémentation, tests et résultats

1. Introduction :

Nous entamons à présent le dernier chapitre, où nous allons parler du langage de programmation et des principaux outils de développement utilisés pour réaliser cette application. L'application et ses fonctionnalités seront présentées décrites à l'aide de captures d'écran. Et enfin, nous parlerons du test de l'application, de la collecte de données que nous avons réalisé et des résultats obtenus.

2. Langage utilisé et outils de développement :

Notre application (IPark) est destinée aux appareils sous android. Pour son développement, nous avons utilisé le langage de programmation JAVA pour android et l'IDE eclipse.

2.1 Android :

Android, prononcé Androïd, est un système d'exploitation pour smartphones, tablettes tactiles, PDA et terminaux mobiles. C'est un système open source^{3,4} utilisant le noyau Linux. Il a été lancé en novembre 2007 par Android, une startup rachetée par Google en 2005.

L'écosystème d'Android s'appuie sur deux piliers [60]:

- Le langage Java.
- Le SDK qui permet d'avoir un environnement de développement facilitant la tâche du développeur.

Le kit de développement donne accès à des exemples, de la documentation mais surtout à l'API de programmation du système et à un émulateur pour tester ses applications [60].

Stratégiquement, Google utilise la licence Apache pour Android ce qui permet la redistribution du code sous forme libre ou non et d'en faire un usage commercial [60].

Le plugin *Android Development Tool* permet d'intégrer les fonctionnalités du SDK à Eclipse [60].

2.2 Le langage de programmation JAVA :

Le langage **Java** est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au *SunWorld*.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plate-formes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

2.3 L'IDE eclipse :

Eclipse est un éditeur, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la Fondation Eclipse visant à développer un environnement de production de logiciels libres qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java. Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment environnement de développement intégré et frameworks) mais aussi d'ATL recouvrant modélisation, conception, gestion de configuration [61].

2.4 Le kit de développement Android :

Exploiter une nouvelle plate-forme n'est jamais chose aisée. C'est pourquoi Google fournit, en plus du système d'exploitation, un kit de développement (*Software Development Toolkit* ou SDK). Ce SDK est un ensemble d'outils qui permet aux développeurs et aux entreprises de créer des applications [62].

Le SDK Android est composé de plusieurs éléments pour aider les développeurs à créer et à maintenir des applications [62]:

- Des API (interfaces de programmation) ;
- Des exemples de code ;
- De la documentation ;
- Des outils – parmi lesquels un émulateur – permettant de couvrir quasiment toutes les étapes du cycle de développement d'une application.

Le SDK Android est disponible gratuitement sur le site de Google.

2.5 Les éléments d'une application android [60]:

Une application Android peut être composée des éléments suivants:

- Des activités (**android.app.Activity**): il s'agit d'une partie de l'application présentant une vue à l'utilisateur.
- Des services (**android.app.Service**): il s'agit d'une activité tâche de fond sans vue associée.
- Des fournisseurs de contenus (**android.content.ContentProvider**): permet le partage d'informations au sein ou entre applications.
- Des widgets (**android.appwidget.***): une vue accrochée au Bureau d'Android.
- Des *Intents* (**android.content.Intent**): permettent d'envoyer un message pour un composant externe sans le nommer explicitement.
- Des récepteurs d'*Intents* (**android.content.BroadcastReceiver**): permettent de déclarer être capable de répondre à des *Intents*.
- Des notifications (**android.app.Notifications**): permettent de notifier l'utilisateur de la survenue d'événements.

3. Présentation d'IPark et de ses fonctionnalités :

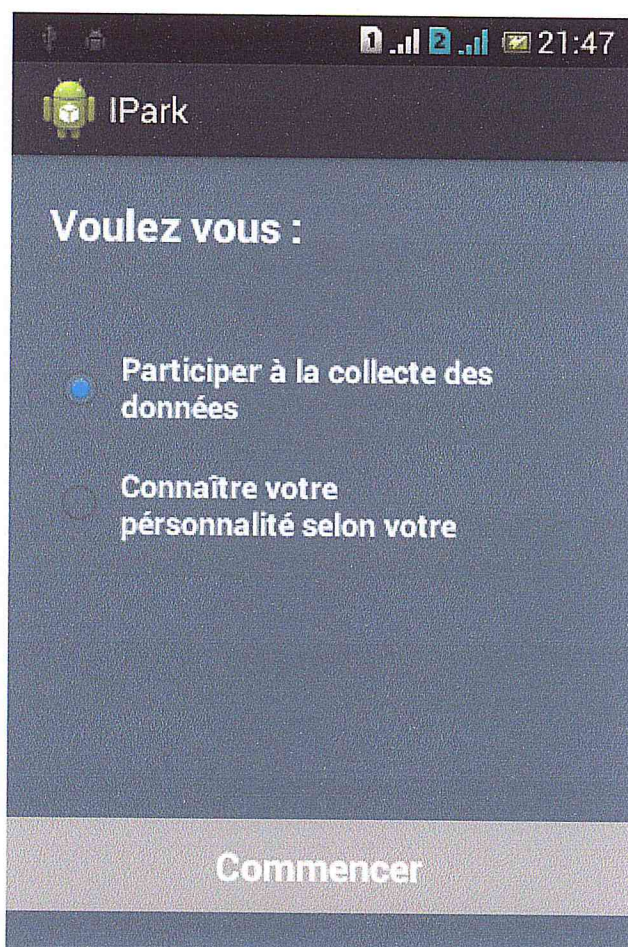


Figure 4.1. Activité principale d'IPark.

Lors du lancement de l'application, voici le premier écran qui apparaît à l'utilisateur (figure 4.1). Il lui est demandé de choisir ce qu'il veut faire, entre participer à la collecte de données ou connaître sa personnalité en fonction de son stationnement. Il peut répondre en cliquant sur l'un des deux boutons radio et en appuyant sur le bouton « Commencer » pour confirmer son choix. Si l'utilisateur répond en choisissant « Participer à la collecte de données » cela signifie qu'il a déjà répondu au questionnaire IPIP-NEO et qu'il souhaite participer à la collecte de données. S'il répond par « Connaître votre personnalité selon votre stationnement », cela signifie qu'il souhaite avoir des informations sur sa personnalité selon sa manière de se stationner. Dans le cas où il répond par le premier choix, la deuxième activité qui lui apparaîtra sera celle représenté dans la figure 4.2, où il devra introduire les résultats obtenus

au questionnaire IPIP-NEO. S'il répond par non, il passera directement à l'activité représentée par la figure 4.3, où il aura à choisir entre un parking sécurisé et payant ou non sécurisé mais gratuit, avant de lancer le simulateur de stationnement.

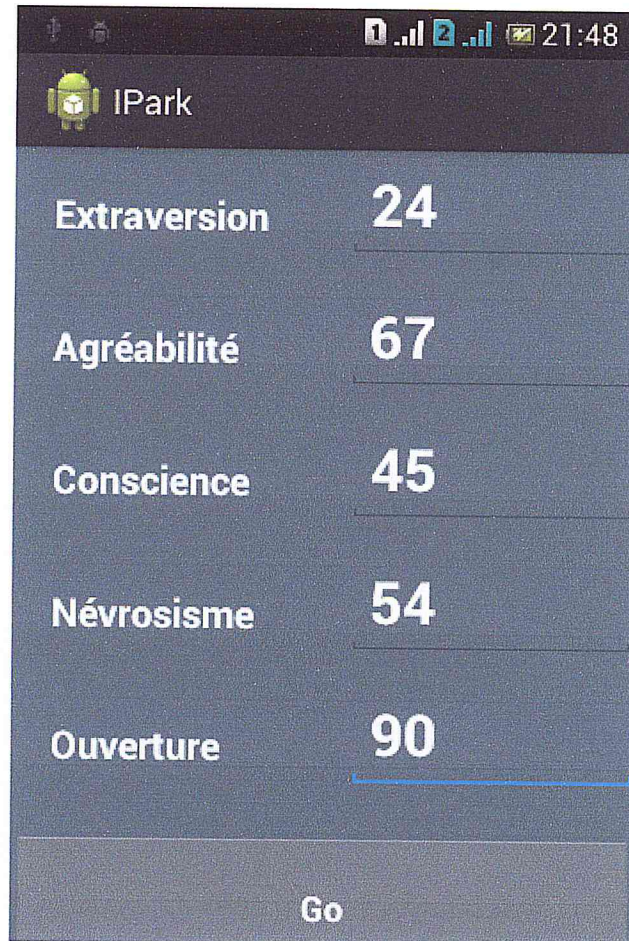


Figure 4.2. Activité où l'utilisateur peut introduire les résultats qu'il a obtenus au questionnaire IPIP-NEO.

Après que l'utilisateur ait introduit les scores (pour chacun des cinq facteurs) qu'il a obtenus au questionnaire IPIP-NEO en chiffres (de 0 à 100). Le système classera ensuite chaque score comme bas (s'il est entre 0 et 33), moyen (s'il est entre 34 et 67) ou élevé (s'il est au-delà de 67). Sur les tableaux qu'on a vu au chapitre précédents, et ce que nous allons voir dans ce chapitre les scores apparaissent sous la forme des mentions (bas, moyen ou élevé). Après avoir introduit ses scores, l'utilisateur doit confirmer en appuyant sur le bouton « Go ». Ses scores sont alors enregistrés, et il passe à l'activité suivante représentée par la figure 4.3.

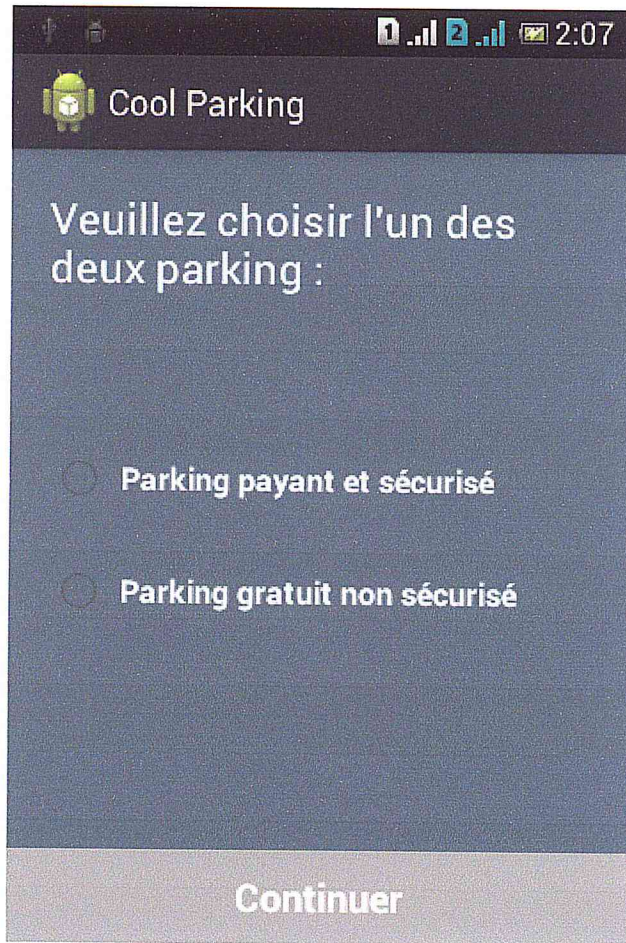


Figure 4.3. Activité où l'utilisateur doit choisir entre un parking payant et sécurisé ou gratuit mais non sécurisé.

Dans cette activité, l'utilisateur choisit donc entre un parking payant et sécurisé ou gratuit mais non sécurisé, il doit cliquer sur l'un des deux boutons radio et confirmer son choix en appuyant sur le bouton « Continuer ». En appuyant sur le bouton « Continuer » pour confirmer son choix, l'utilisateur lance le simulateur de stationnement (figure 4.4).

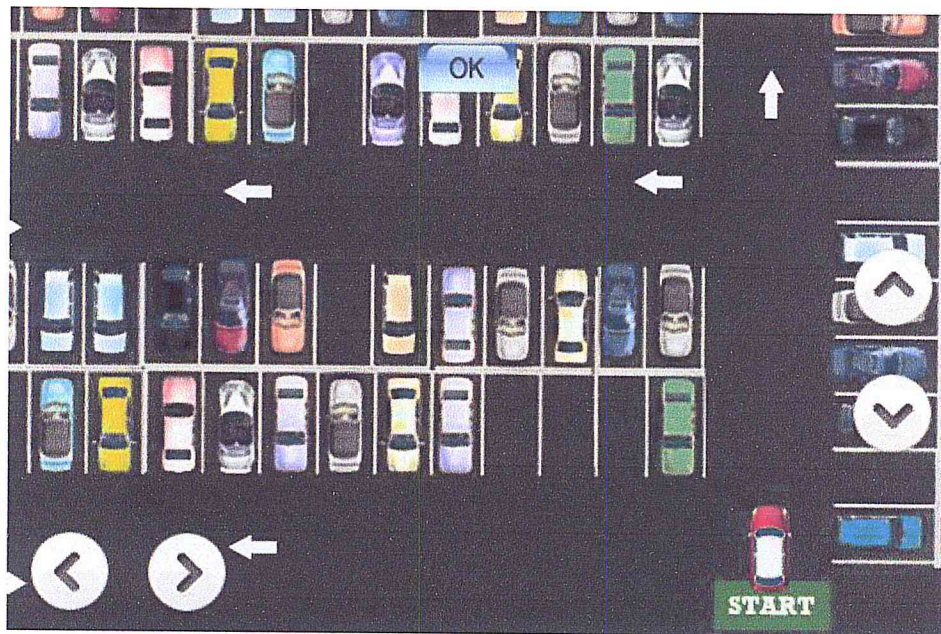


Figure 4.4. Premier écran apparaissant au lancement du simulateur.

La figure 4.4 montre l'écran qui apparaît quand l'utilisateur lance le simulateur, il s'agit de la première partie du parking, seul la partie du parking où se trouve l'utilisateur peut être visible par celui-ci. Dans cette partie, on peut apercevoir l'entrée du parking qui se situe en bas à droite. On peut également voir la voiture rouge, que l'utilisateur doit guider à l'aide des boutons de direction mis à sa disposition (avant, arrière, gauche et droite). Les deux premiers boutons (pour avancer et reculer) se trouvent à droite de l'écran, et les deux autres pour aller à gauche et à droite se situent en bas à gauche.

L'utilisateur doit donc guider le véhicule pour le stationner à l'une des places libres. Il y a également des flèches indiquant les directions que l'utilisateur doit prendre s'il souhaite visiter d'autres parties du parking. La figure 4.5 montre les quatre parties du parking, avec les flèches de directions dans chacune d'elles.

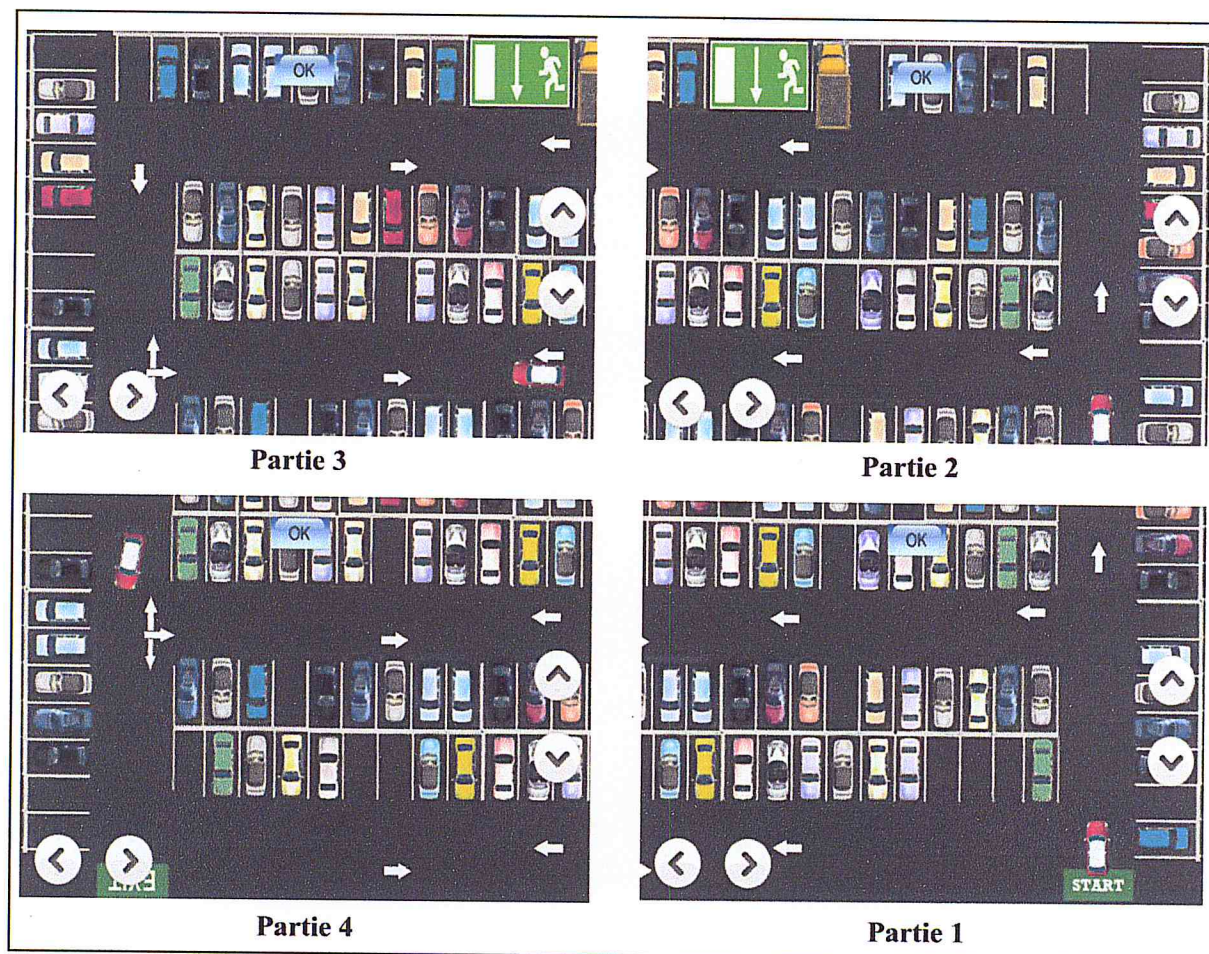


Figure 4.5. Captures d'écrans des quatre parties du parking.

Sur la figure 4.5, on peut apercevoir les quatre parties du parking. Dans la partie 4, l'utilisateur peut apercevoir la sortie véhiculée du parking, qui se situe en bas à gauche. La sortie piétonne peut être aperçue par l'utilisateur s'il se trouve dans l'une des deux parties 2 ou 3.

La disposition des places a été conçue de manière à donner le choix à l'utilisateur entre des places différentes les unes des autres de part leur caractéristiques. Après avoir choisi une place et stationner le véhicule, l'utilisateur valide son choix en appuyant sur le bouton « Ok », visible en haut de chaque écran. À ce moment, les variables du vecteur $V(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$ décrivant la manière de se stationner de l'utilisateur, prendront des valeurs en fonction du type du parking choisi (sécurisé ou non), des caractéristiques de la place choisie, du nombre de déplacements entre les parties du parking, et des distances de la

place choisie par rapport aux deux sorties (véhiculée et piétonne) et par rapport à l'entrée du parking. Ces variables et paramètres qu'ils décrivent sont représentés dans le tableau 3.1 du chapitre précédent. Si l'utilisateur appuie sur le bouton « Ok » alors que le véhicule n'est pas stationné correctement (c'est-à-dire pas dans l'une des places libres). Une activité apparaît à son écran, avec un message lui demandant de stationner correctement le véhicule. La figure 4.6 montre cette activité.

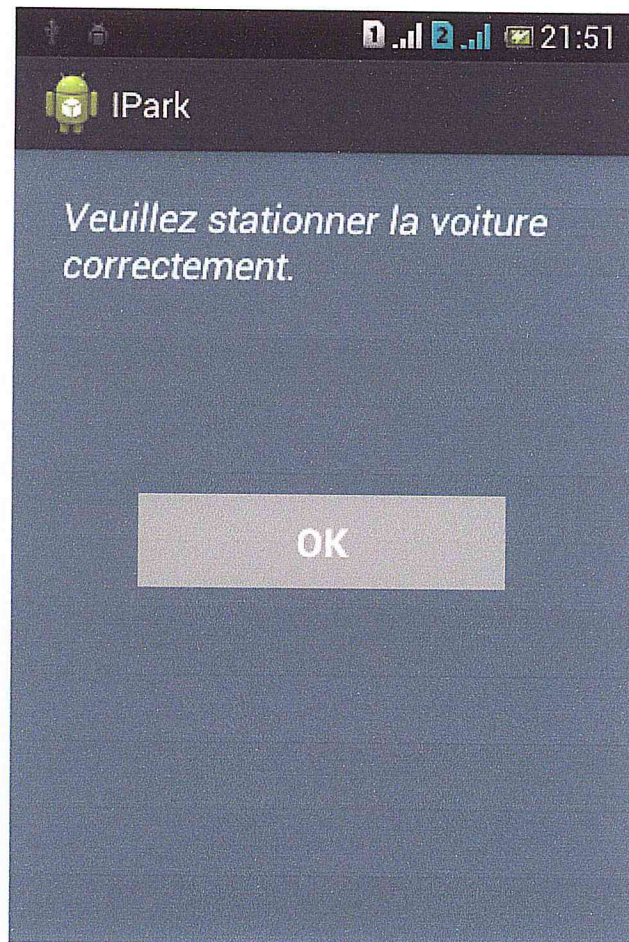


Figure 4.6. Message qui apparaît lorsque l'utilisateur appuie sur « Ok », alors que le véhicule est mal stationné.

Dans le cas où l'utilisateur aurait utilisé IPark pour participer à la collecte de données, son vecteur « V », décrivant sa façon de se stationner, sera enregistré dans un tableau avec ses scores obtenus au questionnaire IPIP-NEO (voir tableau 3.2 du chapitre précédent), ce tableau est enregistré sur un fichier texte externe. Et en appuyant sur le bouton « Ok » pour confirmer son choix, une activité apparaîtra avec un message le remerciant de sa participation (figure 4.7).

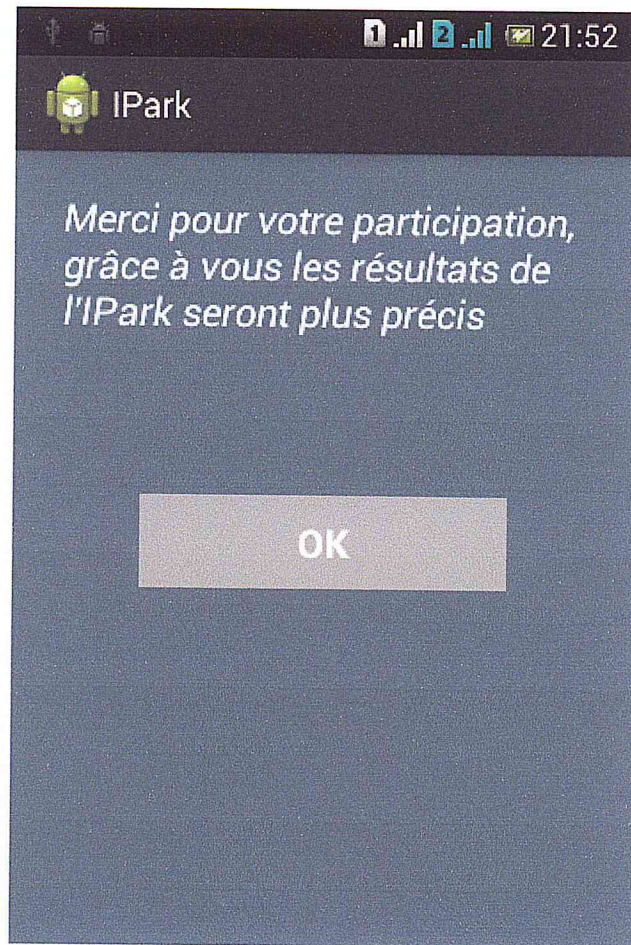


Figure 4.7. Message remerciant l'utilisateur pour avoir participé à la collecte de données.

Et dans le cas où l'utilisateur aurait utilisé IPark pour avoir des informations sur sa personnalité suivant son mode de stationnement (dans ce cas nous avons nommé cet utilisateur, utilisateur X) . Au moment où il termine de se stationner et appuie sur le bouton « Ok » (du simulateur de stationnement), son vecteur « V_x » décrivant son comportement et ses choix de stationnement sera généré et comparé à ceux de la collecte de données, et le type de personnalité qui correspond à sa manière de se stationner lui sera renvoyer et décrit par l'activité représentée dans la figure 4.8.

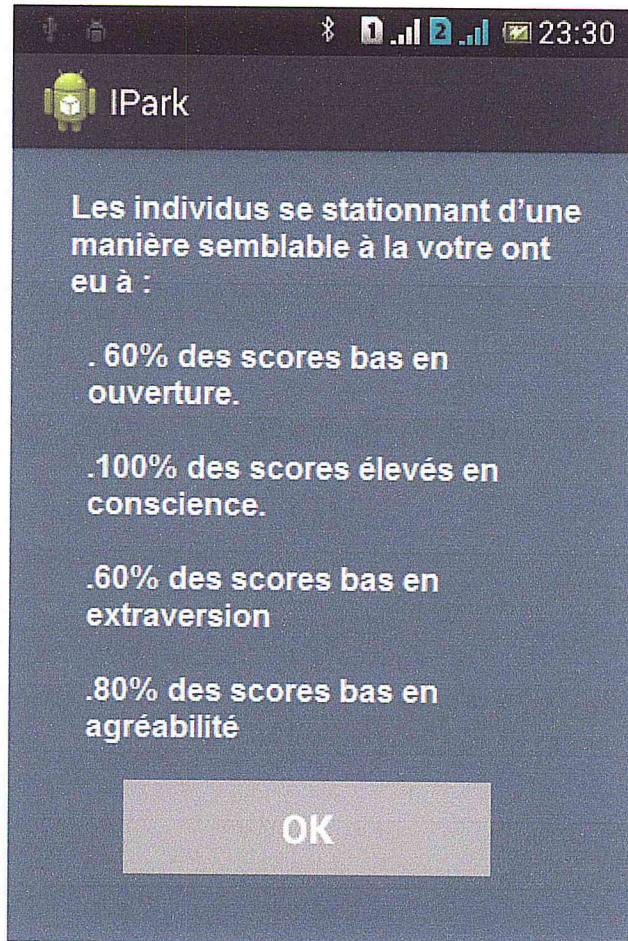


Figure 4.8. Exemple d'un résultat renvoyé à un utilisateur souhaitant connaître sa personnalité selon sa manière de se stationner.

Comme expliqué au chapitre précédent, il se pourrait que la manière de se stationner de l'utilisateur X ne corresponde à aucun type de personnalité particulier car la classe à laquelle son vecteur V_x a été alloué contient des individus dont les personnalités sont trop différentes les unes des autres, dans ce cas l'activité qui lui apparaîtra sera celle montrée par la figure 4.9.

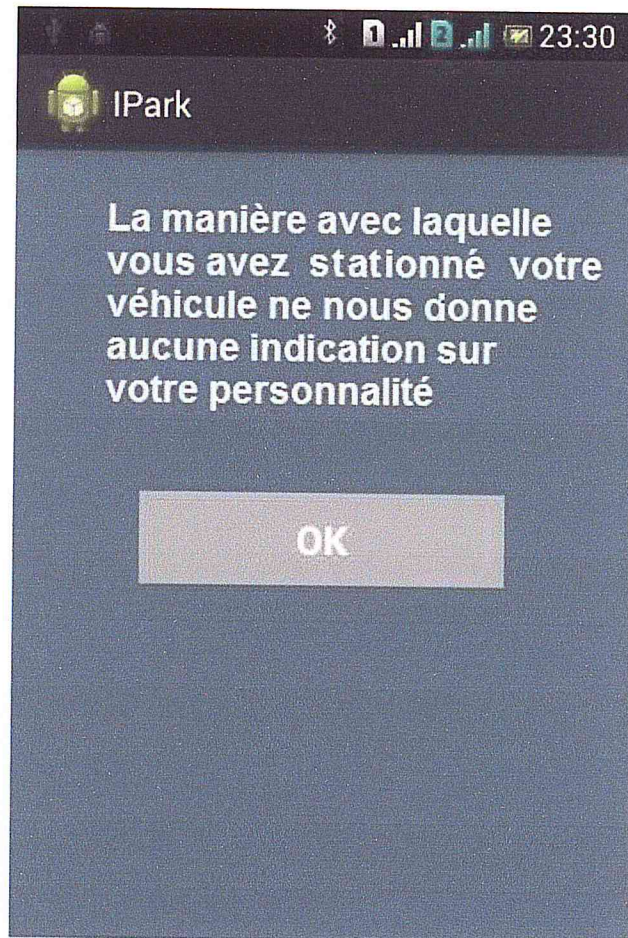


Figure 4.9. Message renvoyé à l'utilisateur si sa manière de se stationner ne correspond à aucun type de personnalité particulier.

4. Tests et résultats :

4.1 Collecte de données :

Une collecte de données a été réalisée auprès de dix individus de sexe masculin, tous à peu près du même âge (entre 25 et 30 ans). Chaque individu a suivi les étapes suivantes :

1. Evaluer sa personnalité en répondant au questionnaire IPIP-NEO qui évalue la personnalité selon cinq dimensions ou facteurs, ce test est disponible gratuitement sur internet.
2. Une fois ses résultats au test IPIP-NEO connus, chaque individu peut utiliser IPark pour participer à la collecte de données.
3. Il introduit d'abord les scores qu'il a obtenus au questionnaire en chiffres de (0 à 100) et chaque score sera ensuite classé comme « bas », « moyen » ou « élevé » par le système. Ces scores sont enregistrés.
4. Ensuite, il utilise le simulateur de stationnement (comme décrits dans l'étape précédente) pour stationner le véhicule à l'un des emplacements libres de l'une des quatre parties du parking. Rappelons qu'il a le choix entre un parking sécurisé et payant et autre non sécurisé mais gratuit. Son comportement est l'ensemble de ses choix de stationnement sont alors enregistrés dans un vecteur «V» les décrivant.

Les scores obtenus par chacun des dix utilisateurs dans le test IPIP-NEO, ainsi que les vecteurs « V » décrivant leurs comportements et choix lors du stationnement ont été enregistrés dans un fichier externe sous forme de tableau. Le tableau 4.1 est une reproduction de ce tableau, à chaque ligne de ce tableau, un utilisateur est représenté par son vecteur « V » (correspondants à sa manière de stationner le véhicule) et les résultats qu'il a obtenus questionnaire IPIP-NEO. Les variables d'un vecteur $V(x_1, \dots, x_9)$ et les paramètres qu'ils représentent sont décrits en détails dans le tableau 3.1 du chapitre précédent. Dans le tableau 4.1, les diminutifs : Ex, Ag, Co, Ne et Ou, représentent les facteurs (Extraversion, Agréabilité, Conscience, Névrotisme et Ouverture) et les lettres B, M et E correspondent aux mentions (Bas, moyen et Elevé).

- **Remarque :** La variable x_9 qui correspond au compteur du nombre de déplacements de l'utilisateur entre les différentes parties du parking est initialisé à 0 et incrémentée de 0.5 à chaque fois que l'utilisateur passe d'une partie à l'autre.

	Vecteur « V »									Résultat IPIP-NEO				
	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	Ex	Ag	Co	Ne	Ou
V ₁	1	1	0	0	1	0.973	0.976	0.050	0.5	M	B	B	E	B
V ₂	1	0	1	0	0	0.724	0.654	0.325	0.5	M	E	M	E	M
V ₃	1	1	1	1	1	0.972	0.974	0.051	1	M	E	B	M	B
V ₄	1	0	1	0	0	1.028	0.590	0.586	2.5	M	M	B	E	M
V ₅	1	1	1	0	0	0.996	1.000	0.063	3	M	E	M	B	B
V ₆	1	1	1	0	0	0.721	0.654	0.326	0	E	E	E	E	M
V ₇	0	1	1	0	0	0.730	0.684	0.297	0	E	M	M	E	M
V ₈	0	1	1	0	0	0.977	0.979	0.052	0.5	M	B	B	E	M
V ₉	1	1	1	0	0	0.723	0.656	0.324	0.5	M	B	B	M	E
V ₁₀	1	1	1	0	0	0.721	0.654	0.326	2	M	B	E	E	B

Tableau 4.1. Résultats de la collecte de données.

4.2 Classification des vecteurs « V » avec la classification hiérarchique ascendante :

Les dix vecteurs « V » obtenus lors de la collecte de données vont être répartis en classes regroupant les vecteurs les plus proches entre eux, et donc les manières de se stationner qui se ressemblent le plus. Cela a été fait en employant la classification ascendante hiérarchique, avec la méthode du lien complet. La fonction de calcul de la distance euclidienne a été utilisée pour calculer la distance entre les vecteurs « V » pour former la matrice de proximité. L'utilisation de cet algorithme pour la classification des vecteurs « V » a été expliquée et détaillée dans le chapitre précédent. Après cette classification, le tableau 4.1 sera scindé en plusieurs autres tableaux en fonction des classes obtenues (*clusters*).

Voici l'ordre de formation des différents clusters obtenus en réunissant deux à deux, les clusters les plus proches en un seul à chaque itération (à chaque mise à jour de la matrice de proximité), et les vecteurs « V » qui composent chaque cluster à chaque fois, N représente le nombre de clusters. Rappelons que les vecteurs « V » sont considérés comme clusters initiaux:

1) Clusters initiaux : C₁, C₂, C₃, C₄, C₅, C₆, C₇, C₈, C₉, C₁₀ =

$\{\{V_1\}, \{V_2\}, \{V_3\}, \{V_4\}, \{V_5\}, \{V_6\}, \{V_7\}, \{V_8\}, \{V_9\}, \{V_{10}\}\}$ (N=10)

2) Création du cluster $C_{11} = \{C_9, C_6\}$

$N=9$: $C_1, C_2, C_3, C_4, C_5, C_7, C_8, C_{10}, C_{11} = \{\{V_1\}, \{V_2\}, \{V_3\}, \{V_4\}, \{V_5\}, \{V_7\}, \{V_8\}, \{V_{10}\}, \{V_6, V_9\}\}$

3) Création du cluster $C_{12} = \{C_7, C_8\}$

$N=8$: $C_1, C_2, C_3, C_4, C_5, C_{10}, C_{12}, C_{11} = \{\{V_1\}, \{V_2\}, \{V_3\}, \{V_4\}, \{V_5\}, \{V_{10}\}, \{V_7, V_8\}, \{V_6, V_9\}\}$

4) Création du cluster $C_{13} = \{C_{11}, C_2\}$

$N=7$: $C_1, C_3, C_4, C_5, C_{10}, C_{12}, C_{13} = \{\{V_1\}, \{V_3\}, \{V_4\}, \{V_5\}, \{V_{10}\}, \{V_7, V_8\}, \{V_6, V_9, V_2\}\}$

5) Création du cluster $C_{14} = \{C_{10}, C_5\}$

$N=6$: $C_1, C_3, C_4, C_{14}, C_{12}, C_{13} = \{\{V_1\}, \{V_3\}, \{V_4\}, \{V_{10}, V_5\}, \{V_7, V_8\}, \{V_6, V_9, V_2\}\}$

6) Création du cluster $C_{15} = \{C_{14}, C_4\}$

$N=5$: $C_1, C_3, C_{15}, C_{12}, C_{13} = \{\{V_1\}, \{V_3\}, \{V_{10}, V_5, V_4\}, \{V_7, V_8\}, \{V_6, V_9, V_2\}\}$

7) Création du cluster $C_{16} = \{C_1, C_3\}$

$N=4$: $C_{16}, C_{15}, C_{12}, C_{13} = \{\{V_1, V_3\}, \{V_{10}, V_5, V_4\}, \{V_7, V_8\}, \{V_6, V_9, V_2\}\}$

8) Création du cluster $C_{17} = \{C_{13}, C_{12}\}$

$N=3$: $C_{16}, C_{15}, C_{17} = \{\{V_1, V_3\}, \{V_{10}, V_5, V_4\}, \{V_6, V_9, V_2, V_7, V_8\}\}$

9) Création du cluster $C_{18} = \{C_{17}, C_{16}\}$

$N=2$: $C_{12}, C_{18} = \{\{V_{10}, V_5, V_4\}, \{V_6, V_9, V_2, V_7, V_8, V_1, V_3\}\}$

10) Création du cluster $C_{19} = \{C_{18}, C_{15}\}$

$N=1$: $C_{19} = \{\{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10}\}\}$

La figure 4.10 montre le dendrogramme résultant de la classification hiérarchique ascendante effectuée sur les dix vecteurs « V » :

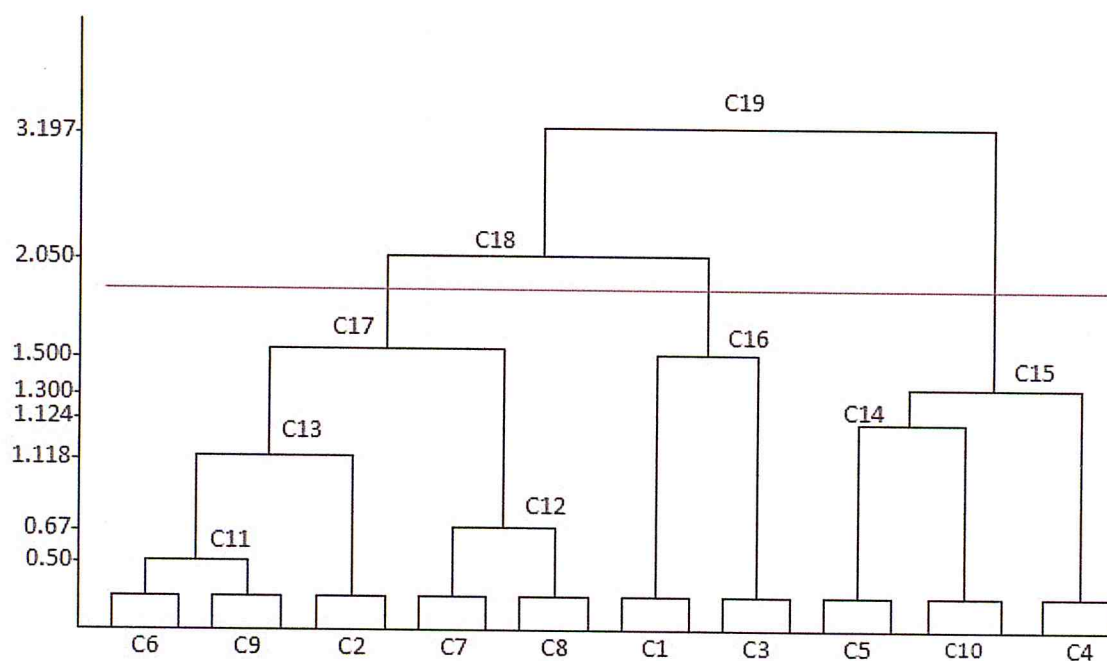


Figure 4.10. Dendrogramme résultant de la classification hiérarchique ascendante.

Pour obtenir une bonne partition, nous avons choisi de couper à un niveau où les branches de l'arbre sont longues (à la distance « 1.5005946 » précisément), ce qui indique que les données contenues dans les *clusters* sont très différentes. Ce niveau correspond à l'itération n°8 parmi celles décrites plus haut, où nous avons les classes (*clusters*) suivantes : C_{16} , C_{15} , $C_{17} = \{\{V_1, V_3\}, \{V_{10}, V_5, V_4\}, \{V_6, V_9, V_2, V_7, V_8\}\}$. Nous avons donc trois classes que nous avons renommé, classe « 1 » = $\{V_6, V_9, V_2, V_7, V_8\}$, classe « 2 » = $\{V_{10}, V_5, V_4\}$, classe « 3 » = $\{V_1, V_3\}$. Le tableau initial sera donc divisé en trois tableaux, et chaque classe aura un type de personnalité attiré qui correspondra aux tendances commune entre les individus qui la composent. Dans ce qui suit, nous allons détailler tout cela.

➤ **Remarque :** Pour obtenir la matrice de proximité, la distance entre les vecteurs « V » a été calculée en utilisant la fonction de calcul de la distance euclidienne. Mais nous avons également essayé en utilisant la distance Manhattan, et les résultats n'ont pas été très différents.

4.3 Description des classes obtenues:

1. Classe « 1 »:

Elle est composée des vecteurs $\{V_2, V_6, V_7, V_8, V_9\}$, Le tableau 4.2 montre les vecteurs « V » et les résultats IPIP-NEO des individus appartenant à cette classe :

	Vecteurs « V »									Résultats IPIP-NEO				
	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	Ex	Ag	Co	Ne	Ou
V ₂	1	0	1	0	0	0.724	0.654	0.325	0.5	M	E	M	E	M
V ₆	1	1	1	0	0	0.721	0.654	0.326	0	E	E	E	E	M
V ₇	0	1	1	0	0	0.730	0.684	0.297	0	E	M	M	E	M
V ₈	0	1	1	0	0	0.977	0.979	0.052	0.5	M	B	B	E	M
V ₉	1	1	1	0	0	0.723	0.656	0.324	0.5	M	B	B	M	E

Tableau 4.2. Vecteurs «V» et résultats IPIP-NEO de la classes « 1 ».

• **Type de personnalité décrivant les individus de la classe « 1 » :**

Comme expliqué lors du chapitre précédent, le type de personnalité qui correspond à une classe est une synthèse des tendances communes entre les résultats des individus de cette classe au questionnaire IPIP-NEO. Il s'agit d'abord de calculer les pourcentages d'apparition de chaque score (élevé, moyen et bas) dans chacun des cinq facteurs décrivant la personnalité. Ensuite, pour chaque facteur, on retient le score majoritairement obtenus, ce score sera considéré comme celui de cette classe dans ce facteur. Si toute fois, le pourcentage du score majoritairement obtenus n'excède pas 50%, le facteur est éliminé de la description de la personnalité de la classe. Nous allons donc appliquer cela à la classe « 1 » :

➤ **En extraversion :** Le score majoritairement obtenus est « Moyen » avec 60%, ce score est donc retenu comme étant celui de cette classe dans ce facteur en retenant son pourcentage d'apparition.

➤ **En agréabilité et en conscience :** Dans chacun de ces deux facteurs, aucun score n'a un pourcentage d'apparition qui excède le seuil de 50%, ils sont donc éliminés de la description de la personnalité de cette classe.

➤ **En névrotisme :** Le score majoritairement obtenus est « Elevé » avec 80%, ce score est donc retenu comme étant celui de cette classe dans ce facteur en retenant son pourcentage d'apparition.

➤ **En ouverture :** Le score majoritairement obtenus est « Moyen » avec 80%, ce score est donc retenu comme étant celui de cette classe dans ce facteur, en retenant son pourcentage d'apparition.

Si un utilisateur X se sert d'IPark pour connaître sa personnalité selon sa manière de se stationner et que celle-ci ressemble à celles des individus de cette classe, son vecteur « V_x » décrivant sa manière de se stationner sera allouée à cette classe et voici ce qu'il aura comme message en résultat : « *Les individus se stationnant de manières semblables à la votre ont généralement des scores à : 60% moyens en extraversion, à 80% élevés en névrotisme et à 80% moyens en ouverture.* ». La description de la personnalité s'est donc faite sans parler des facteurs (agréabilité et conscience).

- **Éléments privilégiés par les individus de la classe « 1 » lors du stationnement :**

En observant les valeurs des variables des vecteurs « V » décrivant les manières de se stationner des utilisateurs, on peut déduire les critères auxquels ils ont accordé le plus d'importance lors de leurs stationnements. Pour cette classe, on peut constater, d'après les valeurs de x_3 , x_4 et x_5 , (en rouge dans le tableau 4.2) que les individus ont tous choisi des places dont les emplacements adjacents droits étaient occupés, ces places étaient toutes facilement accessibles et leur occupation n'était pas gênante pour les autres véhicules. En comparant les valeurs de x_7 et x_8 (en vert dans le tableau 4.2), on constate que les utilisateurs ont choisi des places plus proches de la sortie véhiculée, en n'accordant pas trop d'importance au fait qu'elles soient éloignées de la sortie piétonne. Enfin, en observant les valeurs de x_9 (en orange dans le tableau 4.2), on remarque que les utilisateurs n'ont pas trop hésité avant de choisir une place, le nombre de déplacement entre les parties du parking étant relativement bas.

2. Classe « 2 » :

	Vecteurs « V »									Résultats IPIP-NEO				
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	Ex	Ag	Co	Ne	Ou
V_4	1	0	1	0	0	1.028	0.590	0.586	2.5	M	M	B	E	M
V_5	1	1	1	0	0	0.996	1.000	0.063	3	M	E	M	B	B
V_{10}	1	1	1	0	0	0.721	0.654	0.326	2	M	B	E	E	B

Tableau 4.3. Vecteurs «V» et résultats IPIP-NEO de la classes « 2 ».

- **Type de personnalité décrivant les individus de la classe « 2 » :**

La description de la personnalité de cette classe se fait de la même manière que pour la classe précédente, nous avons donc ces résultats:

100% des utilisateurs ont des scores « moyens » en extraversion, 60% d'entre eux ont eu des scores « élevés » en névrotisme et 60% ont eu des scores « bas » en ouverture. Les facteurs agréabilité et conscience ont été éliminés de la description de la personnalité correspondante à cette classe. On peut également constater que les individus de cette classe ne diffèrent de ceux de la première qu'on ce qui concerne les scores obtenus en « ouverture », qui ont été « bas » pour les individus de cette classe et « moyens » pour ceux de la précédente.

- **Éléments privilégiés par les individus de la classe « 2 » lors du stationnement :**

On peut constater en observant les valeurs de la variable x_1 (en vert dans le tableau 4.3) que les individus de cette classe ont tous choisi un parking sécurisé et payant. Pour ce qui est des variables x_3 , x_4 et x_5 (en rouge dans le tableau 4.3) les tendances sont les mêmes que ceux de la classe précédente c'est-à-dire que les individus ont tous choisi des places dont les emplacements adjacents droits étaient occupés, ces places étaient toutes facilement accessibles et leur occupation n'était pas gênante pour les autres véhicules. Enfin, la spécificité des individus de cette classe est le fait qu'il ait tous visité plusieurs fois les différentes parties du parking avant de faire un choix, ce qui remarquable en observant les valeurs prise par la variable x_9 (en orange dans le tableau 4.3).

3. Classe « 3 » :

	Vecteurs « V »									Résultats IPIP-NEO				
	X1	X2	X3	X4	X5	X6	X7	X8	X9	Ex	Ag	Co	Ne	Ou
V ₁	1	1	0	0	1	0.973	0.976	0.050	0.5	M	B	B	E	B
V ₃	1	1	1	1	1	0.972	0.974	0.051	1	M	E	B	M	B

Tableau 4.4. Vecteurs «V» et résultats IPIP-NEO de la classes « 3 ».

- **Type de personnalité décrivant les individus de la classe « 3 » :**

100% des utilisateurs ont des scores « moyens » en extraversion, 100% d'entre eux ont eu des scores « bas » en agréabilité et 100% ont eu des scores « bas » en conscience et en ouverture. Les facteurs agréabilité et névrotisme ont été éliminés de la description de la personnalité correspondante aux individus de cette classe.

- **Éléments privilégiés par les individus de la classe « 1 » lors du stationnement :**

Comme les utilisateurs de la classe précédente, ceux de cette classe ont tous opté pour un parking sécurisé, d'après les valeurs prises par x_1 (en couleur orange dans le tableau 4.4). Mais la spécificité des utilisateurs de cette classe lors du stationnement a été de choisir des

places dont les emplacements adjacents gauches étaient occupés, l'occupation de ces places rendaient également la sortie des véhicules adjacents difficile, ces tendances ont été déduites en observant les valeurs de x_1 et x_5 (en rouge dans le tableau 4.4). En ce qui concerne les distances des places choisies par rapport aux différentes sorties, les utilisateurs de cette classe ont opté pour des places très proches de la sortie véhiculée et très éloignée de l'entrée du parking et de la sortie piétonne (valeurs de x_6, x_7, x_8 en vert dans le tableau 4.4).

➤ **Remarque :** Dans le chapitre précédent, nous avons dit que le choix d'une place qui serait gênante pour autrui mais confortable pour soi par un utilisateur, pourrait être révélateur d'une personnalité ou le niveau de « Conscience » et d'agréabilité serait bas (en se basant sur la supposition du Dr. John A. Johnson). On peut remarquer que c'est le cas pour cette classe, mais il n'est bien évidemment pas question de généraliser en se basant sur ce cas.

4.4 Test d'IPark par un utilisateur désirant connaître sa personnalité :

Afin de tester l'efficacité d'IPark, nous avons proposé à un utilisateur de la tester pour connaître sa personnalité, cet utilisateur s'est servi le simulateur de stationnement, a choisi un emplacement libre, et après validation de son choix voici le vecteur « V » obtenu qui décrit sa manière de se stationner en fonction des choix qu'il a fait.

Vecteur « V »								
X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9
1	1	1	0	0	0.729	0.682	0.299	0

C'est « la classe 1 » a été allouée à ce vecteur en utilisant l'algorithme des k plus proches voisins, en utilisant la distance euclidienne et en prenant $k = 3$ comme nombre de voisins à prendre en compte, les vecteurs les plus proches ont été respectivement les suivants : V_6, V_9, V_7 .

On constate que la classe majoritaire est bien la « classe 1 » (les 3 vecteurs « V » les plus proches appartiennent à cette classe), le message qui lui a été renvoyé pour décrire sa personnalité a été le suivant :

« Les individus se stationnant de manières semblables à la votre ont généralement des scores à : 60% moyens en extraversion, à 80% élevés en névrotisme et à 80% moyens en ouverture. ».

Cet utilisateur avait déjà passé le test IPIP-NEO et obtenus les résultats suivant :

Facteur de la personnalité	Extraversion	Agréabilité	Conscience	Névrotisme	Ouverture
Score	50 (moyen)	43 (moyen)	20 (bas)	80 (élevé)	26 (bas)

En comparant avec le résultat renvoyé par IPark à cet utilisateur, on constate que la prédiction du type de la personnalité selon la manière de se stationner s'est faite avec succès pour les facteurs : extraversion et névrotisme (en vert), mais pas pour le facteur de l'ouverture (en rouge). On peut donc dire, que la prédiction du type de la personnalité pour cet utilisateur s'est faite avec succès sans pour autant être précise (deux tendances sur trois ont été prédites avec succès).

5. Conclusion :

Dans ce chapitre, nous avons donc parlé de nos outils de développement et du langage de programmation utilisé, nous avons également fait le tour des fonctionnalités qu'offre IPark et des étapes par lesquelles passe chaque utilisateur. Le test de l'application a donné des résultats plutôt probants, des liens ont été établis entre les manières de se stationner des participants à la collecte de données et leurs personnalités ce qui a permis de donner des résultats à un individu qui a testé IPark pour connaître son type de personnalité selon son type de stationnement.

Conclusion

générale

Conclusion générale :

Le thème traité au cours de ce mémoire a été la conception et le développement d'un simulateur de stationnement destiné aux appareils sous android et permettant de donner des informations sur la personnalité d'un utilisateur à partir de la manière avec laquelle il stationne son véhicule, c'est-à-dire de son comportement et des choix qu'il fait lors du stationnement.

Nombre d'études ont été conduites par le passé visant à définir les critères majeurs pris en compte par un conducteur lorsqu'il choisit un emplacement de stationnement. Nous nous sommes basés sur les résultats de certaines de ces études afin de définir les éléments à prendre en compte pour évaluer la manière de se stationner d'un utilisateur.

La première étape du travail a été d'effectuer une collecte de données auprès d'un petit nombre de personnes, ces personnes avaient préalablement passé un questionnaire évaluant leurs personnalités. Chacun d'entre eux a introduit les résultats qu'il a obtenus puis ils ont expérimenté le simulateur de stationnement, et la manière de se stationner de chacun a été enregistrée sous forme de vecteur qui a été associée à son type de personnalité. La classification non supervisée a été utilisée pour regrouper dans les mêmes clusters les vecteurs les plus proches. Une fois ces classes obtenues, les tendances communes entre les utilisateurs de chaque classe pour ce qui de la personnalité ont été retenues, obtenant ainsi des catégories distinctes de façon de se stationner en fonction des choix faits lors du stationnement. Nous avons assumé que chaque manière de se stationner correspond à un type de personnalité différent. Un ensemble d'apprentissage a donc été obtenu suite à cette étape, et le lien a été fait entre manière de se stationner et type de personnalité.

Dans la seconde étape, l'algorithme des k plus proches voisins a été implémenté afin d'associer la manière de se stationner d'un utilisateur testant l'application pour connaître sa personnalité, à l'une des classes précédemment obtenues, et ainsi de connaître sa personnalité en fonction de la classe qui a été allouée au vecteur décrivant sa manière de se stationner.

Faire ce travail nous a permis d'approfondir nos connaissances sur les techniques de classification supervisées et non supervisées, mais également de découvrir l'apprentissage automatique et ses différentes techniques. Après avoir effectué une collecte de données auprès de dix personnes, et avoir traité les résultats de cette collecte de données, nous avons eu des catégories distinctes de façons de stationner qui correspondaient chacune à un type de

personnalité particulier. Ainsi, nous avons établi des liens entre la manière de se stationner des individus participants à la collecte de données, et certains aspects de leurs personnalités. Après cela, un utilisateur a pu tester l'application pour connaître son type de personnalité. Le système a pu lui fournir une description de son type de personnalité. La description de la personnalité fournie par le système à cet individu, a été très proche du résultat qu'il a obtenu après avoir répondu au questionnaire IPIP-NEO. L'architecture proposée a donc permis de développer une application qui a répondu aux attentes et atteint les objectifs de départ. Cette étude a donc permis de résoudre notre problématique.

▪ **Perspectives :**

L'architecture proposée dans ce mémoire permet le développement d'une application qui répond aux objectifs de départ mais qui n'est pas sans failles et qui peut être améliorée de plusieurs manières.

Le modèle à cinq grands facteurs sur lequel est basé le questionnaire IPIP-NEO que nous avons utilisé pour mesurer la personnalité, ne note pas les individus seulement sur les cinq grands facteurs, mais attribue également des notes sur les 6 sous facteurs qui composent chacun des cinq principaux. Pour décrire la personnalité des individus auprès desquels a été effectuée la collecte de données, nous n'avons pris en compte que les notes obtenus dans chacun des cinq grands facteurs. Inclure les notes obtenus par les utilisateurs dans ces sous facteurs permettrait une description plus précise de la personnalité.

Ajouter d'autres paramètres à prendre en compte pour la description de la manière de se stationner et des caractéristiques de la place choisie pourrait également améliorer la précision des résultats, cela peut être fait par la conception d'un simulateur de stationnement dont la map de parking serait plus détaillée et qui offrirait un choix plus large à l'utilisateur. De plus, les variables des vecteurs décrivant la manière de se stationner des utilisateurs pourraient être pondérées afin de favoriser les critères les plus importants.

De plus, la collecte de données a été réalisée auprès d'une population ciblée constituée d'hommes d'une vingtaine d'années, donc les résultats ne seront corrects que si l'utilisateur désirant connaître sa personnalité en fonction de sa manière de se stationner est un homme d'une vingtaine d'années, ce qui altère l'attractivité d'IPark. Ce défaut peut être effacé en effectuant des collectes de données sur d'autres types de population et en adaptant l'architecture du système de manière à ce qu'il puisse donner des résultats en fonction de l'âge

et du sexe de l'utilisateur désirant connaître sa personnalité en fonction de sa manière de se stationner.

Il serait possible d'analyser les résultats d'une collecte de données, dans le but de trouver des interdépendances entre les variables des vecteurs décrivant les manières de se stationner des utilisateurs et les résultats de ces derniers au questionnaire IPIP-NEO. Ce qui pourrait permettre d'obtenir des indications précises sur les liens existants entre type de stationnement et personnalité. Ceci peut-être fait à l'aide d'outils d'analyse statistique tel que le logiciel SPSS.

Enfin, l'application pourrait être améliorée en y ajoutant une fonctionnalité, qui serait la simulation d'une des manières de se stationner obtenues. Cela peut être fait par un agent machine auquel on aurait préalablement alloué la personnalité correspondante à la manière de se stationner qu'on souhaite le voir reproduire.

Bibliographie

- [1] Revelle, W. "Personality Theory and Research" [en ligne]. (Page consultée le 21 juin 2013). <http://www.personality-project.org/index.html>
- [2] McCrae, R.R., Costa, P.T. (1990). Personality in adulthood: A five-factor theory perspective. New York (Etats-Unis): The Guildford Press. 198 p.
- [3] Srivastava, S. " The Personality and Social Dynamics Lab" [en ligne]. (Page consultée le 03/04/2013). <http://pages.uoregon.edu/sanjay/bigfive.html>
- [4] John, O.P., Srivastava, S. (1999). The Big Five trait taxonomy: History, measurement, and theoretical perspectives. In : Pervin, L. A., John, O. P. (Eds). Handbook of personality: Theory and research. New York (Etats-Unis) : The Guildford Press, pp.102-138.
- [5] Meiping, Y., Ruisong, Y., Xiaoguang, Y. (2009). Study on Driver's Parking Location Choice Behavior Considering Drivers' Information Acquisition. IEEE *Xplore* Digital Library [en ligne], vol.3, pp. 764-770. (Page consultée le 20/07/2013) <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5288098>
- [6] Hongzhi, G., Lanhui, L. (2003). Survey and Analysis of Parking Behavior in Metropolitan Business Quarter. Journal of Beijing University of Technology, vol.19(n°1), pp. 46-51.
- [7] Thompson, R.G., Richardson, A. J. (1998). A Parking Search Model. Transportation Research Part A, vol.32(n°3), pp. 159-170.
- [8] Dell'Orco, M., Ottomanelli, M., Sassanelli, D. (2003). Modelling uncertainty in parking choice behavior. 82nd Annual Meeting of the Transportation Research Board, Committee on Historic and Archaeological Preservation in Transportation (A1F05), 12-16/01/2003, Washington, DC (Etats-Unis), 20 p.
- [9] Hongzhi, G. (2004). Disaggregate Model : A Tool of Traffic Behavior Analysis. Pékin (Chine) : China Communications Press.
- [10] Van Der Waerden, P., Oppewal, H., Timmermans, H. (1993). Adaptive Choice Behavior of Motorists in Congested Shopping Centre Parking Lots. Transportation, vol.20(n°4), pp. 395-408.
- [11] Meiping, Y., Ruisong, Y., Xiaoguang, Y. (2008). Modeling on Scale of Public Parking Lot based on Parking Choice Behavior. IEEE *Xplore* Digital Library [en ligne], vol.2, pp. 259-262. (Page consultée le 21/07/2013) http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4659763&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4659763
- [12] Borgers, A., Timmermans, H., Van Der Waerden, P. (2003). Travelers Micro-Behavior at Parking Lots: A Model of Parking Choice Behavior. 82nd Annual Meeting of the Transportation Research Board, Committee on Historic and Archaeological Preservation in Transportation (A1F05), 12-16/01/2003, Washington, DC (Etats-Unis), 20p.

- [13] Young, W. (1986). PARKSIM/1: A Network Model for Parking Facility Design. *Traffic Engineering and Control*, vol.27(n°12), pp. 606-613.
- [14] Young, W., Thompson, R. (1987). PARKSIM/1: A Computer Graphics Approach for Parking-Lot Layouts. *Traffic Engineering and Control*, vol.28(n°3), pp. 120-123.
- [15] Young, W., Thompson, R. (1987). PARKSIM/1: Data Presentation and Evaluation. *Traffic Engineering and Control*, vol.28(n°5), pp. 294-297.
- [16] Van der Waerden, P., Timmermans, H., Borgers, A. (2002). Pamela : a Parking Analysis Model for Predicting Effects in Local Areas. 81st Annual Meeting of the Transportation Research Board, Committee on Historic and Archaeological Preservation in Transportation (A1F05), 13-17/01/2002, Washington DC (Etats-Unis), pp. 10-18.
- [17] Saini, S., Chung, P.W.H., Dawson, C.W. (2011). Mimicking Human Strategies in Fighting Games using a Data Driven Finite State Machine. *IEEE Xplore Digital Library* [en ligne], vol.2, pp. 389-393. (page consultée le 11/07/2013) http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6030356&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6030356
- [18] Rabin, S. (2003). The Ultimate Guide to FSMs in Games. In: Fu, D., Houlette, R. (eds). *A.I Game Programming Wisdom 2*. Hingham, Massachusetts (Etats-Unis) : Charles River Media, pp. 283-301.
- [19] Johnson, D., Wiles, J. (2001). Computer Games with Intelligence. *Australian Journal of Intelligent Information Processing Systems*, vol.7, pp. 61-68.
- [20] Rabin, S. (2003). Implementing a Data-Driven Finite State Machine. In : Rosado, G. (eds). *A.I Game Programming Wisdom 2*. Hingham, Massachusetts (Etats-Unis) : Charles River Media, pp. 307-317.
- [21] Saini, S.S., Dawson, C.W., Chung, P.W.H. (2011). Mimicking player strategies in fighting games. *IEEE Xplore Digital Library* [en ligne], pp. 44-47. (page consultée le 11/07/2013) http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6115128&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6115128
- [22] Fernández, A., Gómez, S. (2008). Solving Non-uniqueness in Agglomerative Hierarchical Clustering Using Multidendrograms. *Journal of Classification*, vol.25, pp. 43-65.
- [23] John, P.O. " Berkeley Personality Lab " [en ligne]. (Page consultée le 22/04/2013). <http://www.ocf.berkeley.edu/~johnlab/bfi.htm>
- [24] Tollari, S. (2006). Indexation et recherche d'images par fusion d'informations textuelles et visuelles [en ligne]. Thèse de Doctorat : Informatique. Toulon (France) : Université du Sud Toulon-Var. Disponible sur : <http://www-ia.lip6.fr/~tollaris/ARTICLES//THESE/index.html> (Page consulté le 06/08/2013).
- [25] Bonastre, J.-F. "Apprentissage Automatique" [en ligne]. (Page consultée le 01/07/).

<http://lia.univ-avignon.fr/fileadmin/documents/Users/Intranet/chercheurs/bonastre/ApprentissageAutoIntroductionI.pdf>.

[26] Cornuéjols, A., Miclet, L., Kodratoff, Y. (2003). Avant-propos. Apprentissage artificiel : Concepts et algorithmes. Paris : Editions Eyrolles, pp. v-xxv.

[27] Belhabib, A., Lagha, O. (2012). Développement d'une application à base de l'algorithme de classification k-means [en ligne]. Mémoire de licence : Informatique. Tlemcen (Algérie) : Université Abou Bakr Belkaid-Tlemcen, 41 p. Disponible sur : http://bibfac.univ-tlemcen.dz/bibfs/opac_css/doc_num.php?explnum_id=252 (Page consultée le 08/08/2013).

[28] Quinlan, J.R. (1985). Induction of Decision Trees. Machine Learning, vol.1, pp. 81-106.

[29] Govaert, G. (2003). Analyse de données. Paris (France) : Hermes Science Publications. 362 p.

[30] Hilali, H. (2009). Application de la classification textuelle pour l'extraction des règles d'association maximales [en ligne]. Mémoire de maîtrise : mathématiques et informatique appliquées. Trois-Rivières (Québec): Université du Québec, 146 p. Disponible sur : <http://depot-e.uqtr.ca/1201/> (Page consultée le 04/08/2013).

[31] Gilleron, R., Tommasi, M. "Découverte de connaissances à partir de données" [en ligne]. (Page consultée le 21/08/2013) <http://www.grappa.univ-lille3.fr/polys/fouille/index.html>

[32] Berrani, S.-A., Amsaleg, L., Gros, P. (2002). Recherche par similarités dans les bases de données multidimensionnelles : panorama des techniques d'indexation. Ingénierie des systèmes d'information (RSTI série ISI-NIS), vol.7(n°5-6), pp. 65-90.

[33] Berrani, S.-A.(2004). Recherche approximative de plus proches voisins avec contrôle probabiliste de la précision ; application à la recherche d'images par le contenu. Thèse de doctorat : Informatique. Rennes (France) : Université de Rennes 1, 210 p.

[34] Duda, R., Hart, P. (1973). Pattern Classification and Scene Analysis. Hoboken (Etats-Unis) : John Wiley & sons. 512 p.

[35] Devijver, P., Kittler, J. (1982). Pattern Recognition : a statistical approach. Upper Saddle River (Etats-Unis) : Prentice-Hall. 448 p.

[36] Caraux, G., Lechevallier, Y. (1996). Règles de décision de Bayes et méthodes statistiques de discrimination. Revue d'Intelligence Artificielle, vol.10(n°2-3), pp. 219-283.

[37] Cornuéjols, A., Miclet, L., Kodratoff, Y. (2003). Chapitre 14 : L'apprentissage bayésien et son approximation. Apprentissage artificiel : Concepts et algorithmes. Paris : Editions Eyrolles, pp. 411-449.

[38] Mathieu-Dupas, E. "Algorithme des k plus proches voisins pondérés et application en diagnostic " [en ligne]. (Page consultée le 23/08/2013). <http://hal.inria.fr/inria-00494814/fr/>

- [39] Mari, J.-L. " Jean-Luc Mari : Page personnelle" [en ligne]. (Page consultée le 25/08/2013). <http://www.dil.univ-mrs.fr/~mari/Enseignement/M1SIS/assets/Classement%20RF.pdf>.
- [40] Graham-Cumming, J. (2006). "Interview de John Graham-Cumming, l'auteur du logiciel antispam PopFile " .
- [41] Robardet, C. "Classification supervisée" [en ligne]. (Page consultée le 25/08/2013). <http://liris.cnrs.fr/celine.robardet/doc/classification.pdf>
- [42] Mitchell, T. (1997). *Machine Learning*. New York (Etats-Unis): McGraw Hill. 432 p.
- [43] Breiman, L., Friedman, J. H., Olshen, R. A. et Stone, C. J. (1984). *Classification and regression trees*. Monterey, CA (Etats-Unis): Wadsworth International Group. 358 p.
- [44] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Mateo, CA(Etats-Unis) : Morgan Kaufmann. 302 p.
- [45] Denis, F., Gillerion, R. "Apprentissage à partir d'exemples" [en ligne]. (Page consultée le 19/08/2013). <http://www.grappa.univ-lille3.fr/polys/apprentissage/index.html>
- [46] Moutarde, F. " Bienvenue chez Fabien Moutarde " [en ligne]. (Page consultée le 25/08/2013). http://perso.mines-paristech.fr/fabien.moutarde/ES_MachineLearning/Slides/slides_AD.pdf
- [47] Rouveirol, C. "Céline Rouveirol" [en ligne]. (Page consultée le 18/08/2013). http://lipn.univ-paris13.fr/~rouveirol/enseigne/MICR_AS/Arbre-decision-1-2x2.pdf
- [48] Chessel, D., Thioulouse, J., Dufour, A.B. "Introduction à la classification hiérarchique" [en ligne]. (Page consultée le 29/08/2013). <http://pbil.univ-lyon1.fr/R/pdf/stage7.pdf>
- [49] Gordon, A. D. (1999). *Classification : Second Edition*. Londres: Chapman and Hall / CRC. 272 p.
- [50] Ramdane, C. (2006). *Le clustering des données ; une nouvelle approche évolutionnaire quantique* [en ligne]. Mémoire de Magistère : Informatique Option Information et Computation. Costantine (Algérie) : Université Mentouri de Constantine, 134 p. Disponible sur : <http://bu.umc.edu.dz/theses/informatique/RAM4510.pdf>. (Page consultée le 09/08/2013).
- [51] Kotsiantis, S.B., Pintelas, P. E. (2004). *Recent Advances in Clustering: A Brief Survey*. WSEAS Transactions on Information Science and Applications, vol.1(n°1), pp. 73–81.
- [52] El Ganaoui, O., Perrot, M. "Segmentation par régions : une méthode qui utilise la classification par nuées dynamiques et le principe d'hystérésis" [en ligne]. (Page consultée le : 15/08/2013) <http://www.tsi.telecom-paristech.fr/pages/enseignement/ressources/beti/hyste-dyn/report.html>
- [53] Nakache, J.-P., Confais, J. (2005). *Approche pragmatique de la classification : arbres hiérarchiques, partitionnements*. Paris (France) : Editions TECHNIP. 262 p.



- [54] Candillier, L. (2006). *Contextualisation, visualisation et évaluation en apprentissage non supervisé* [en ligne]. Thèse de doctorat : Informatique. Lille (France) : Université Charles De Gaulle-Lille 3, 227 p. Disponible sur : <http://tel.archives-ouvertes.fr/tel-00617420/> (Page consultée le 31/07/2013).
- [55] Lance, G.N., Williams, W.T. (1967). A general theory of classificatory sorting strategies: I. Hierarchical systems. *Computer Journal*, vol.9, pp. 373-380.
- [56] Carpentier, F.-G. "Documents pédagogiques" [en ligne]. (Page consultée le 22/08/2013). <http://geai.univ-brest.fr/~carpentier/>
- [57] Pedrycz, W. (2005). *Knowledge-Based Clustering: From Data to Information Granules*. Hoboken (Etat-Unis) : John Wiley & Sons. 336 p.
- [58] Tanagra. "Tutoriels Tanagra pour le Data Mining" [en ligne]. (Page consultée le 15/08/2013) <http://tutoriels-data-mining.blogspot.com/2008/10/traitement-de-gros-volumes-cah-mixte.html>
- [59] Kamvar, S. D., Klein, D., Manning, C.D. (2002). Interpreting and Extending Classical Agglomerative Clustering Algorithms using a Model-Based approach. International Conference on Machine Learning (ICML), 8-12/07/ 2002, Sydney (Australie), pp. 283-290.
- [60] Lalande, J.-F. "Développement sous Android" [en ligne]. (Page consultée le 03/03/2013) <http://www.univ-orleans.fr/lifo/Members/Jean-Francois.Lalande/teaching.html>
- [61] Eclipse Foundation. The Eclipse Foundation open source community website [En ligne]. (Page Consultée le : 18/02/2013). <http://www.eclipse.org>
- [62] Guignard, D., Chable, J., Roblès, E. et al. (2010). *Programmation Android : De la conception au déploiement avec le SDK Google Android 2*. Paris : Editions Eyrolles. 486 p.
- [63] Visame. Test de personnalité professionnel, accessible à tous et gratuit [en ligne]. (Page consultée le 25/02/2013). <http://www.visamemo.fr/archives/1364>