



INGÉNIERIE INTELLIGENCE  
ARTIFICIELLE  
2IA



ECOLE NATIONALE SUPÉRIEURE  
D'INFORMATIQUE ET D'ANALYSE DES SYSTÈMES

Rapport de Stage de Fin de Première Année

---

# Étude et Développement d'un Modelé IA pour Assister les Contrôleurs d'État des Établissements et Entreprises Publics

---

*Soutenu le 2022 par :*  
NECHBA MOHAMMED

*Encadré par :*  
M. ABOUENOUR LAHSEN (DSI)

*Jury :*  
Pr. EL AFIA ABDELLATIF

Année Universitaire :2021-2022



# Table des matières

<b>Remerciements</b>	<b>5</b>
<b>Résumé</b>	<b>6</b>
<b>Introduction Générale</b>	<b>7</b>
<b>1 Environnement de Stage</b>	<b>8</b>
1.1 Introduction . . . . .	8
1.2 Ministère de l'Economie et des Finances . . . . .	8
1.2.1 Présentation générale du Ministère . . . . .	8
1.3 Organisations et structures du MEF . . . . .	9
1.4 Direction des Entreprises Publiques et de la Privatisation . . . . .	10
1.4.1 Présentation de la direction . . . . .	10
1.4.2 Missions . . . . .	10
1.5 Division des Systèmes d'Information . . . . .	11
<b>2 Analyse et étude du cahier des charges</b>	<b>12</b>
2.1 Introduction . . . . .	12
2.2 Étude de cahier des charges . . . . .	12
2.2.1 Problématique . . . . .	12
2.2.2 La solution proposée . . . . .	12
2.2.3 Les objectifs . . . . .	12
2.2.4 Avantages de notre solution . . . . .	13
2.3 Spécification des besoins . . . . .	13
2.3.1 Spécification des besoins fonctionnels . . . . .	13
2.3.2 Spécification des besoins non fonctionnels . . . . .	13
2.4 Planification . . . . .	14
2.5 conclusion . . . . .	14
<b>3 Conception et modélisation</b>	<b>15</b>
3.1 Introduction . . . . .	15
3.2 Présentation UML . . . . .	15
3.3 Les acteurs . . . . .	15
3.4 Diagramme de cas d'utilisation . . . . .	16
3.4.1 Diagramme de cas d'utilisation . . . . .	16
3.4.2 Description du cas d'utilisation . . . . .	16
3.5 Dictionnaire de données . . . . .	19
3.6 Diagramme de class . . . . .	19
3.7 Conclusion . . . . .	19

<b>4</b>	<b>Machines à vecteurs de support</b>	<b>20</b>
	Introduction . . . . .	20
4.1	Éléments de la théorie de l'optimisation . . . . .	20
4.1.1	Introduction à l'optimisation . . . . .	20
4.1.2	La convexité . . . . .	21
4.1.3	La dualité et le théorème de Kuhn-Tucker . . . . .	22
4.2	Machines à vecteurs de support pour données linéairement séparables . . . . .	24
4.2.1	Introduction aux machines à vecteurs de support . . . . .	24
4.2.2	Hyperplan séparateur . . . . .	24
4.2.3	Marge et hyperplan canonique . . . . .	26
4.2.4	Trouver l'hyperplan . . . . .	27
4.2.5	Représentation duale . . . . .	28
4.2.6	Vecteurs de support . . . . .	29
4.3	Machines à vecteurs de support pour données non linéairement séparables . . . . .	30
4.3.1	Transformations . . . . .	30
4.3.2	Les noyaux . . . . .	31
4.3.3	Noyaux : Construction et exemples . . . . .	32
4.4	Marges souples . . . . .	32
4.4.1	Machines à vecteurs de support et bruit . . . . .	32
4.4.2	Marge souple . . . . .	33
4.4.3	Représentation duale . . . . .	34
4.4.4	Remarques . . . . .	35
<b>5</b>	<b>Les technologies utilisées et réalisation</b>	<b>36</b>
5.1	Introduction . . . . .	36
5.2	Environnement de travail . . . . .	36
5.2.1	Environnement matériel . . . . .	36
5.2.2	Environnement logiciel : Système de gestion de base de données . . . . .	36
5.3	Choix technique . . . . .	37
5.3.1	Les langages de programmation . . . . .	37
5.3.2	Les langages de description . . . . .	38
5.4	Présentation des interfaces . . . . .	38
5.4.1	Interface d'inscription . . . . .	38
5.4.2	Interface d'authentification . . . . .	39
5.4.3	Page principale . . . . .	40
5.4.4	Page de résultat . . . . .	40
5.5	Conclusion . . . . .	41
	<b>Conclusion Générale</b>	<b>42</b>
	<b>Références</b>	<b>43</b>

# Table des figures

1.1	Organigramme de la division SI . . . . .	11
2.1	Diagramme de GANTT . . . . .	14
3.1	Diagramme de cas d'utilisation . . . . .	16
3.2	Cas d'utilisation « création de compte » . . . . .	16
3.3	Cas d'utilisation « Se connecter » . . . . .	18
3.4	Diagramme de class . . . . .	19
4.1	Une fonction convexe . . . . .	21
4.2	Des données linéairement séparables . . . . .	24
4.3	Il existe une infinité d'hyperplans pouvant séparer les données . . . . .	25
4.4	La marge . . . . .	26
4.5	Les vecteurs de support . . . . .	29
4.6	Marge souple . . . . .	33
5.1	Interface d'inscription . . . . .	38
5.2	Interface d'authentification . . . . .	39
5.3	Page principale . . . . .	40
5.4	Page de résultat . . . . .	40

# Liste des tableaux

1.1	Fiche d'identité du MEF . . . . .	8
1.2	Directions du MEF . . . . .	9
1.3	Fiche d'identité de la DEPP . . . . .	10
3.1	Création de compte . . . . .	17
3.2	Se connecter . . . . .	18
3.3	Dictionnaire de données . . . . .	19
4.1	Les noyaux les plus fréquemment utilisés . . . . .	32

# Remerciements

Je tiens à remercier, en premier lieu, Mr. Adil KOUSCOUS et Mr.Ahjir HAMZA pour la confiance qu'ils m'ont accordée en me permettant ce stage, cette agréable aventure intellectuelle et sociale, au sein de leur entreprise ainsi que pour leurs idées constructives et suivi bienveillant. J'espère qu'ils trouveront dans ce travail un hommage vivant à leurs hautes personnalités.

De surcroît, je tiens à exprimer ma sincère reconnaissance à mon maître de stage Mr.Lahsen ABOUENOUR pour tous les efforts qu'il a fournis dans mon orientation et pour le précieux temps qu'il me réservait avec générosité afin de répondre à toutes mes interrogations ; sans oublier tous les membres de la famille DEPP avec qui j'ai pu collaborer, discuter et apprendre dans les meilleures des conditions.

Je ne pourrais clore ce chapitre sans rendre mes hommages à tous les professeurs qui m'ont enseigné et qui, par leurs compétences et leurs lumières, m'ont soutenu dans la poursuite de mes études.

Mon humble gratitude aux membres du jury qui se consacreront à la lecture de ce présent rapport et à l'analyse de cet ensemble construit.

Finalement, je remercie encore ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.

# Résumé



# Introduction Générale

Dans le cadre du stage de fin de première année à l'École Nationale Supérieure d'Informatique et d'Analyse des Systèmes en filière d'Ingénierie Intelligence Artificielle, la Direction des Entreprises Publiques et de Privatisation (DEPP) m'a permis une mise en oeuvre et un développement des connaissances acquises durant ma première année d'étude à l'ENSIAS. Actuellement, la DEPP est chargée de l'exercice du contrôle financier de l'Etat sur les Etablissements et Entreprises Publiques (EEP), la surveillance du portefeuille public, la participation aux actions de rationalisation et de restructuration des EEP, le pilotage des Audits externes des EEP, la normalisation comptable et le transfert des entreprises publiques au secteur privé. Dans le monde du contrôle et audit de la finance public, les contrôleurs sont souvent confronté à des problèmes de compréhension des termes techniques des rapports de finances venants de plusieurs établissements différents et cela dû au fait qu'ils n'ont pas de connaissances au préalable dans ces domaines spécifiques, Pour pallier à cette problématique donnée nous étions ammené à mettre en place une application web dynamique, qui une fois un fichier chargé là déçu récupère tous les mots techniques du fichier et affiche leurs définitions détaillés, tiré du dictionnaire.

Ce présent rapport est divisé en cinq chapitres. Outre des introduction et conclusion générales, le premier dresse le contexte général de notre stage en présentant l'entreprise, le deuxième formule les objectifs du stage avec une planification des tâches ; sera suivi d'un troisième chapitre qui présente la conception et la modélisation de l'ensemble de notre application ; pour ensuite atteindre le quatrième chapitre expliquant le modèle (Machines à vecteurs de support) utilisé par notre application pour prédire les mots à expliquer. Pour enfin atteindre le dernier chapitre exposants les technologies utilisé pour le développement du modèle prédictif et de l'application web

# Chapitre 1

## Environnement de Stage

### 1.1 Introduction

L'objectif de ce chapitre est de fournir une présentation de l'environnement de stage qui est la Direction des Entreprises Publiques et Privatisation (DEPP).

### 1.2 Ministère de l'Economie et des Finances

#### 1.2.1 Présentation générale du Ministère

Dans le cadre de ses attributions, le Ministère d'Economie et des Finances (MEF) est en charge des questions financières et monétaires, y compris les politiques des crédits et de finances extérieures.

Le MEF assure la tutelle financière des entreprises et établissements publics ou à participation étatique et, le cas échéant, contribue dans une large mesure à leur transfert au secteur privé, dans les cadres, législatif et réglementaire du processus de privatisation.

	<b>Ministère de l'Economie et des Finances</b>
<b>Type d'activité</b>	Chargé de la politique économique et financière du Maroc
<b>Siège</b>	Boulevard Mohammed V Quartier Administratif, Rabat
<b>Ministre</b>	M. Nadia Fettah Alaoui
<b>Site Web</b>	<a href="http://www.finances.gov.ma">www.finances.gov.ma</a>

TABLE 1.1 – Fiche d'identité du MEF

### 1.3 Organisations et structures du MEF

Le Ministère de l'Economie et des Finances du Royaume présente 11 directions, précisés dans le tableau suivant :

Directions	Missions
<b>L'Inspection Générale des Finances</b>	Effectuer les vérifications des services de caisse et de comptabilité, deniers et matières, des comptables publics et des agents de l'Etat, des collectivités locales, des EEP et de tout organisme public.
<b>L'Administration des Douanes et Impôts Indirects</b>	Chargée de la perception des droits et taxes douanières, du recouvrement des impositions fiscales et parafiscales, de la lutte contre les trafics illicites et du contrôle des marchandises et des personnes aux frontières.
<b>La Trésorerie Générale du Royaume</b>	Le recouvrement des créances publiques, Le contrôle et le paiement des dépenses publiques, La gestion des finances locales, La gestion des dépôts au Trésor et la production de l'information financière et comptable.
<b>La Direction Générale des Impôts</b>	Fournir au gouvernement une expertise fiscale notamment dans le cadre de l'élaboration des lois de finances et de la négociation des conventions de non double imposition avec les états étrangers.
<b>La Direction du Budget</b>	Chargée de la mise en œuvre de la politique budgétaire nationale, faire toute proposition et de procéder à toute étude de nature à éclairer les choix du Ministre de l'Economie et des Finances en la matière.
<b>La Direction du Trésor et des Finances Extérieurs</b>	Acteur central dans le développement et la régulation du secteur financier, gestionnaire de la dette et de la trésorerie et gardien des équilibres macroéconomiques.
<b>La Direction des Entreprises Publiques et de la Privatisation</b>	L'amélioration du dispositif de gouvernance et de contrôle financier de l'Etat sur les EEP, l'amplification du recours au PPP pour diversifier les sources de financement.
<b>La Direction des Domaines de l'Etat</b>	Remplit à travers la mobilisation du foncier privé de l'Etat une mission de soutien des politiques d'investissement productifs et des programmes majeurs de l'Etat, une mission fondamentale dans le développement des équipements publics et des services sociaux.
<b>La Direction des Affaires Administratives et Générales</b>	Coordonner et veiller à la mise en œuvre des schémas d'organisation des services du ministère aux niveaux central et déconcentré, élaborer, diffuser et veiller à la mise en œuvre des normes de gestion des ressources.
<b>La Direction des Etudes et des Prévisions Financières</b>	Contribuer à l'évaluation des impacts, sur l'économie nationale, des politiques économiques, financières et sociales ainsi que des grands projets publics.
<b>L'Agence Judiciaire du Royaume</b>	Les recours en annulation pour excès de pouvoir formulés contre les décisions administratives devant les juridictions administratives et la Cour de cassation, la présentation des plaintes et la défense des fonctionnaires.

TABLE 1.2 – Directions du MEF

## 1.4 Direction des Entreprises Publiques et de la Privatisation

### 1.4.1 Présentation de la direction

La DEPP est chargée de l'exercice du contrôle financier de l'Etat sur les Etablissements et Entreprises Publics (EEP), la surveillance du portefeuille public, la participation aux actions de rationalisation et de restructuration des EEP, le pilotage des Audits externes des EEP, la normalisation comptable et le transfert des entreprises publiques au secteur privé.

Consciente de ces enjeux stratégiques et des exigences du contexte actuel la Direction des Entreprises Publiques et de la Privatisation poursuit ses efforts en vue de traduire les chantiers engagés en actions concrètes visant, principalement, l'amélioration du dispositif de gouvernance et de contrôle financier de l'Etat sur les Etablissements et Entreprises publics, l'amplification du recours au Partenariat Public-Privé pour diversifier les sources de financement, la rénovation du cadre global de pilotage du Portefeuille public afin d'assurer un meilleur suivi de ses performances, le renforcement de l'information et de la communication sur les Etablissements et Entreprises Publics et l'amélioration des instruments d'intervention de la Direction.

	<b>Direction des Entreprises Publiques et Privatisation</b>
<b>Type d'activité</b>	Assistance et accompagnement des EEP et PPP
<b>Adresse</b>	Rue Hajj Ahmed Cherkaoui, Agdal, Rabat
<b>Directeur</b>	M. SEMMAR Abderrahmane
<b>Site Web</b>	<a href="http://www.finances.gov.ma">www.finances.gov.ma</a>

TABLE 1.3 – Fiche d'identité de la DEPP

### 1.4.2 Missions

La Direction des Entreprises Publiques et de la Privatisation (DEPP) est chargée de :

- L'exercice du contrôle financier de l'Etat sur les Etablissements et Entreprises Publics (EEP)
- La surveillance du portefeuille public
- La participation aux actions de rationalisation et de restructuration des EEP
- Le pilotage des Audits externes des EEP
- La normalisation comptable
- Le transfert des entreprises publiques au secteur privé

## 1.5 Division des Systèmes d'Information

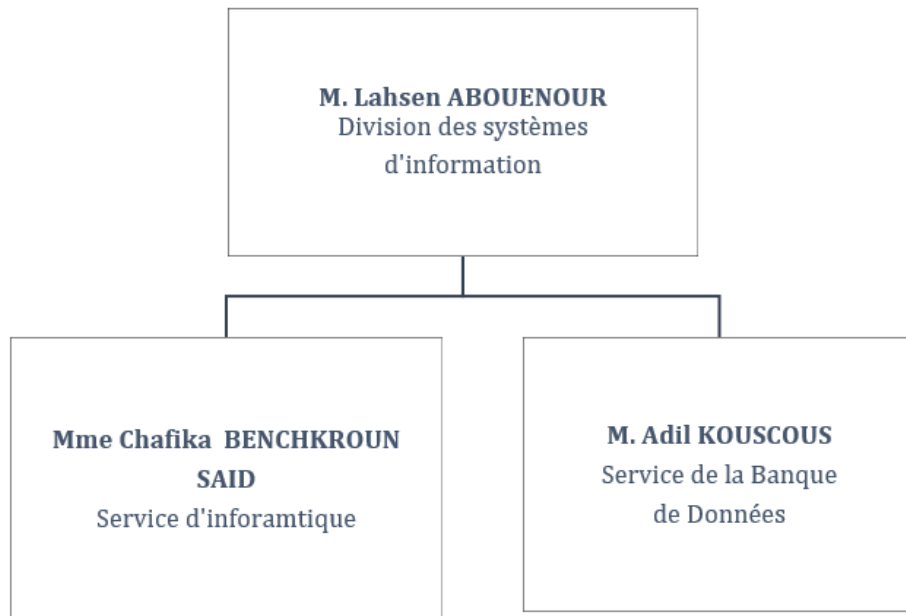


FIGURE 1.1 – Organigramme de la division SI

# Chapitre 2

## Analyse et étude du cahier des charges

### 2.1 Introduction

Ce chapitre introduit le contexte général du projet. On va présenter le projet, la problématique et la planification du projet.

### 2.2 Étude de cahier des charges

#### 2.2.1 Problématique

Dans le monde du contrôle et audit de la finance public, les contrôleurs sont souvent confronté à des problèmes de compréhension des termes techniques des rapports de finances venants de plusieurs établissements différents et cela dû au fait qu'ils n'ont pas de connaissances au préalable dans ces domaines spécifiques.

#### 2.2.2 La solution proposée

Pour pallier cette problématique donnée nous allons mettre en place une application web dynamique, qui une fois un fichier télécharger là déçu récupère tous les mots techniques dans le fichier et donne une définition tiré des dictionnaire à chacun de ces mots techniques.

#### 2.2.3 Les objectifs

Je suis amené à mettre en place une application web dynamique avec une interface intuitif prenant en compte assez d'exigences que se soit technique ou fonctionnelle. L'application web permet de :

- Gérer les comptes utilisateurs pour l'admin.
- Donner une interface d'inscription ou d'authentification pour tous les utilisateurs.
- Permettre aux utilisateurs de télécharger des fichiers.
- Collecter des mots-clés de fichier donné.
- Définition des mots-clés collectés.
- Afficher à l'utilisateur la liste des fichiers de simulation d'un fichier particulier.

### 2.2.4 Avantages de notre solution

Notre solution répond exactement au cahier de charge, et elle apporte plusieurs avantages, parmi ses avantages :

- Définition de chaque mot technique.
- Amélioration des procédures de contrôle de gestions de finance de l'Etat.
- Faire économiser le temps du contrôleur.

## 2.3 Spécification des besoins

La spécification de besoins constitue la phase de départ de toute application à développer dans laquelle nous allons identifier les besoins de notre application. Nous distinguons des besoins fonctionnels qui présentent les fonctionnalités attendues de notre application et les besoins non fonctionnels pour éviter le développement d'une application non satisfaisante.

### 2.3.1 Spécification des besoins fonctionnels

Dans cette partie nous désignons les différents services offerts par notre application :

**Application serveur :**

- Intégrité des données.
- Access réduit aux données.
- La sécurité des données.
- La mises à jour.

**Interface web :**

- L'authentification de gestionnaire.
- L'inscription.
- Analyser les fichiers en fonction de mots clés.
- Découvrir des fichiers similaires à un fichier particulière

### 2.3.2 Spécification des besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Et en ce qui concerne nos applications, nous avons dégagé les besoins suivants :

- **La sécurité de l'accès aux informations critiques :** nous devons prendre en considération la confidentialité des données de clients surtout au niveau de l'authentification.
- **La convivialité de l'interface graphique :** l'application doit fournir une interface conviviale et simple pour tout type d'utilisateur.
- **Une solution ouverte et évoluée :** l'application peut être améliorée par l'ajout d'autres fonctionnalités pour garantir la souplesse, l'évolutivité et l'ouverture de la solution.

## 2.4 Planification

Pour mieux planifier, analyser et contrôler le bon déroulement de la réalisation du projet, ainsi qu'assurer une bonne qualité du produit dans des délais fixés et une conformité entre ce qui est définie et ce qui est obtenu, nous avons utilisé les notions de la gestion des projets informatiques et le respect des normes qualité. Dans ce cadre nous allons présenter le diagramme de GANTT.

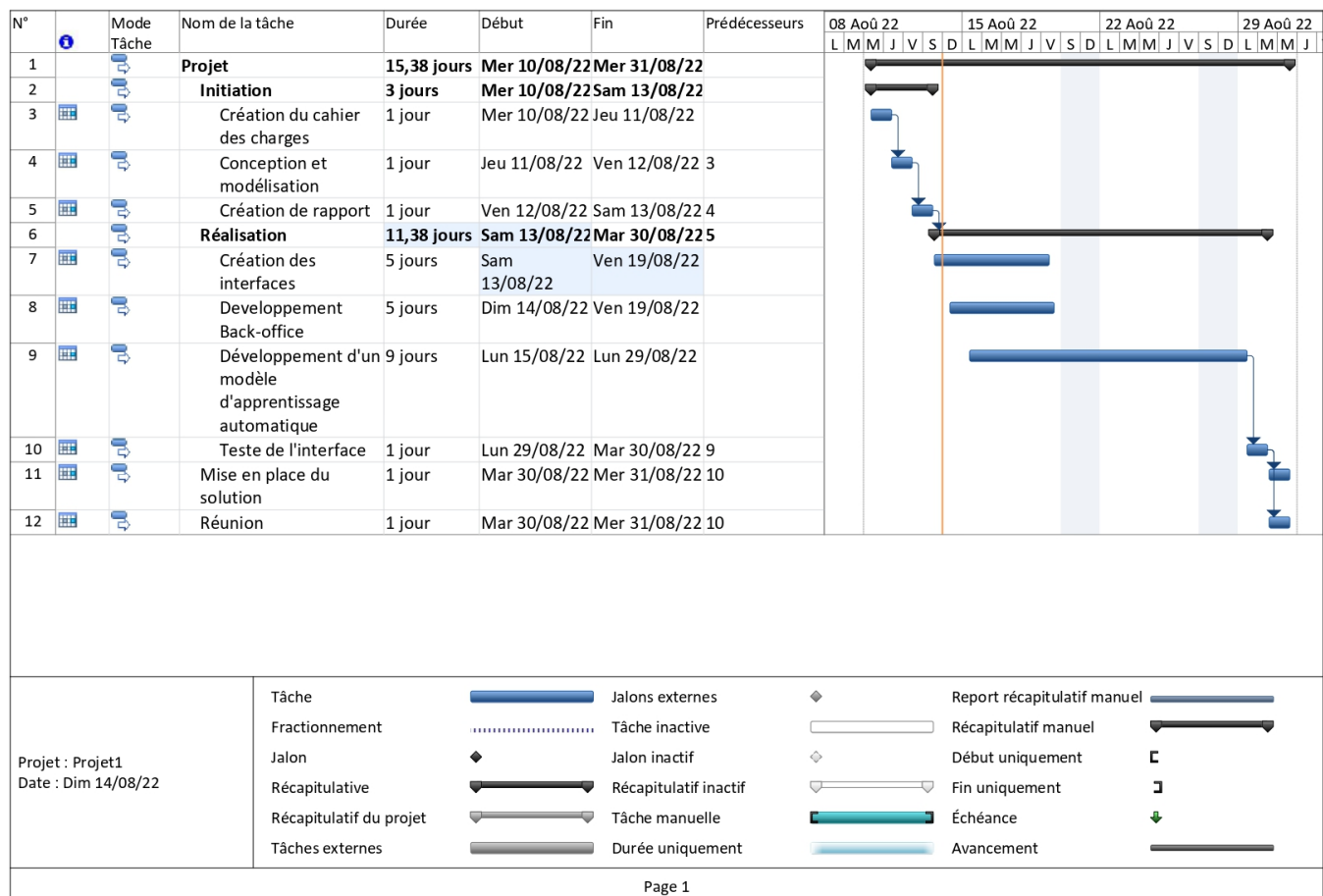


FIGURE 2.1 – Diagramme de GANTT

## 2.5 conclusion

En guise de conclusion, notre site web répond à un problème majeur dont la société fait face. il joue le rôle d'aide-mémoire pour toute personne se trouvant dans le besoin de compréhension des mots techniques dont il n'est pas familier dans un document



# Chapitre 3

## Conception et modélisation

### 3.1 Introduction

Dans cette partie on présente quelques diagrammes qui schématisent les fonctionnalités offertes par notre solution ainsi que leurs déroulements. Loin du code, cette représentation est un moyen de communication entre le maître d'ouvrage et le développeur.

### 3.2 Présentation UML

**UML ( Unified Modelling Language )** est un langage standard, pour spécifier, visualiser, concevoir et documenter tous les aspects d'un système d'information, il fournit un support de communication : un langage graphique comportant 13 diagrammes standards (pour UML 2.0) représentant des 'vues' d'un système d'information, et il permet aussi d'exprimer et d'élaborer des modèles objet, indépendamment de tout langage de programmation.

Nous allons présenter comme exemple de cette modélisation un ensemble de diagramme de cas d'utilisation et diagramme de classe.

**Diagramme de cas d'utilisation** : Un diagramme de cas d'utilisation capture le comportement d'un système, permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique.

**Diagramme de classe** : Un diagramme de classes fournit une vue globale d'un système en présentant ses classes, interfaces et collaborations, et les relations entre elles. Les diagrammes de classes sont statiques : ils affichent ce qui interagit mais pas ce qui se passe pendant l'interaction.

### 3.3 Les acteurs

Un acteur est l'archétype de l'utilisateur (personne, processus externe, ...) qui interagit avec le système.

- **Administrateur** : son rôle a géré les comptes utilisateur et les données du site.
- **Utilisateur** : il joue un rôle dans l'ajouter et l'analyse du fichier.

## 3.4 Diagramme de cas d'utilisation

### 3.4.1 Diagramme de cas d'utilisation

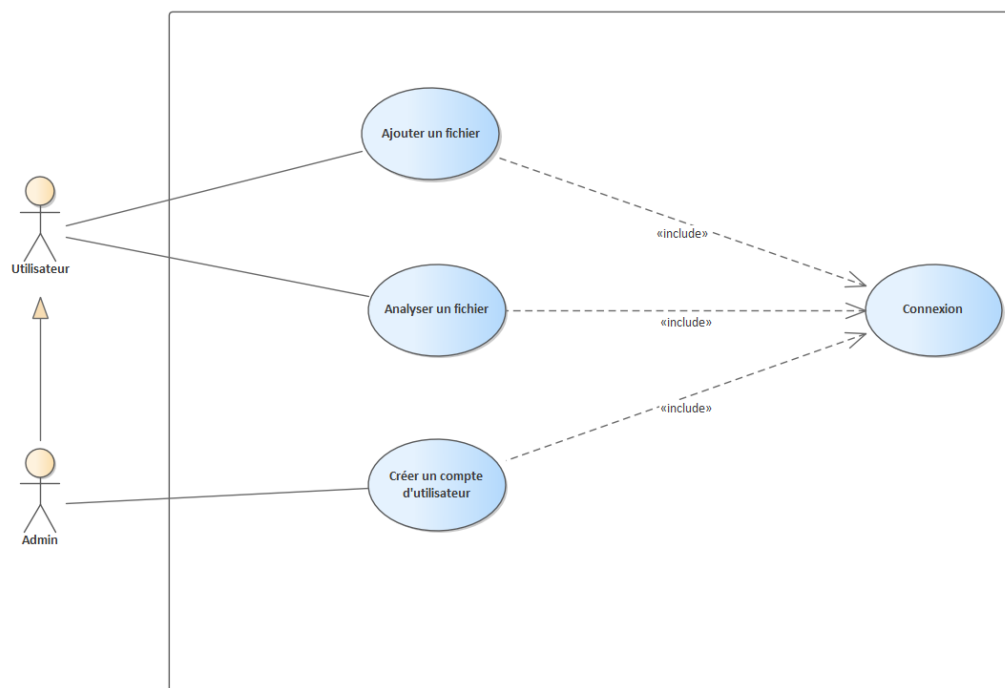


FIGURE 3.1 – Diagramme de cas d'utilisation

### 3.4.2 Description du cas d'utilisation

#### Création de compte

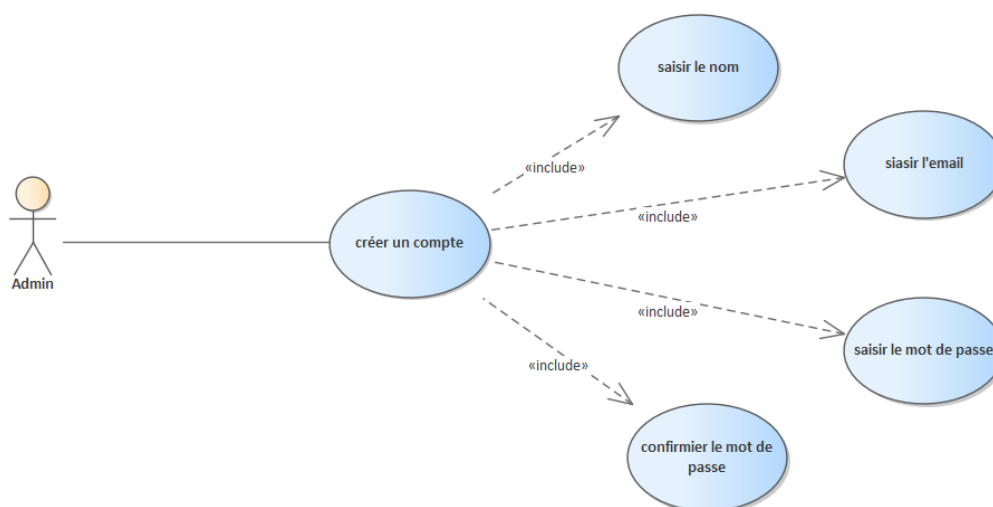


FIGURE 3.2 – Cas d'utilisation « création de compte »

<b>SOMMAIRE</b>	
<b>Titre</b>	Création de compte
<b>But</b>	Création d'un nouveau compte
<b>Résumé</b>	L'administrateur clique sur le bouton « créer un compte » pour créer un nouveau compte.
<b>Acteur</b>	Admin
<b>DESCRIPTION DES ENCHAINEMENTS</b>	
<b>Pré conditions</b>	<b>Post conditions</b>
	<ul style="list-style-type: none"> <li>- La création d'un nouveau compte et l'ajout de cet utilisateur a la base de données.</li> </ul>
<b>Scenario nominal</b>	
<ol style="list-style-type: none"> <li>1. L'administrateur clique sur le bouton « créer compte »</li> <li>2. L'administrateur remplir tous les champs qui sont affiché par le système.</li> <li>3. L'administrateur clique sur le bouton validé pour envoie ces informations.</li> <li>4. Le système va vérifier les informations des champs, et après va envoyer un message de la validation a l'utilisateur.</li> <li>5. Le système va ajouter les informations à la base de données.</li> <li>6. L'utilisateur va connecter à son espace.</li> </ol>	
<b>Enchaînement alternatif</b>	
<ol style="list-style-type: none"> <li>1. Utilisateur n'a pas rempli un champ ou les données sont incorrectes. <ol style="list-style-type: none"> <li>1. Le système affiche un message d'erreur.</li> <li>2. Retour à l'étape 1 du scénario nominal.</li> </ol> </li> </ol>	

TABLE 3.1 – Création de compte

## Se connecter

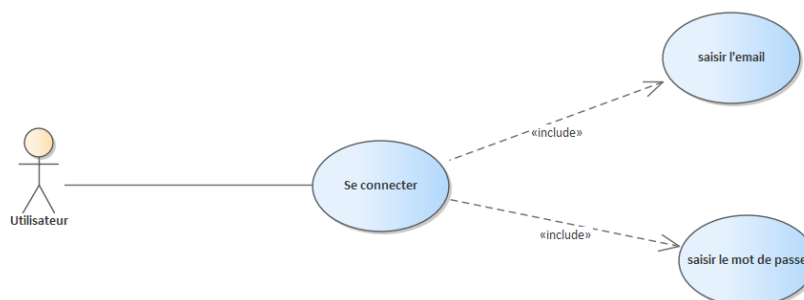


FIGURE 3.3 – Cas d'utilisation « Se connecter »

SOMMAIRE	
<b>Titre</b>	Se connecter d'utilisateur
<b>But</b>	La connexion des utilisateurs
<b>Résumé</b>	Utilisateur clique sur le bouton « Se connecter » pour la connexion a son espace.
<b>Acteur</b>	Utilisateur
DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- L'utilisateur avoir déjà un compte	
Scénario nominal	
<ol style="list-style-type: none"> <li>1. Le system afficher les champs pour faire la connexion.</li> <li>2. L'utilisateur va remplir les champs qui sont affiché.</li> <li>3. L'utilisateur clique sur le bouton « Se connecter »</li> <li>4. Le système va vérifier c'est cette espace existe ou non.</li> <li>5. Le système afficher l'espace de cet utilisateur s'il existe</li> </ol>	
Enchaînement alternatif	
<ol style="list-style-type: none"> <li>1. Utilisateur n'a pas rempli un champ ou les données sont incorrectes. <ol style="list-style-type: none"> <li>1. Le système affiche un message d'erreur.</li> <li>2. Retour à l'étape 1 du scénario nominal pour lancer à nouveau la connexion.</li> </ol> </li> </ol>	

TABLE 3.2 – Se connecter

### 3.5 Dictionnaire de données

Attribut	Désignation	Type
id_u	L'identifiant de l'utilisateur	int
email_u	L'adresse électronique de l'utilisateur	String
nom_u	Le nom de l'utilisateur	String
id_f	L'identifiant d'un fichier	int
nom_f	Le nom d'un fichier	String
data_f	C'est la date de téléchargement d'un fichier	timestamp

TABLE 3.3 – Dictionnaire de données

### 3.6 Diagramme de class

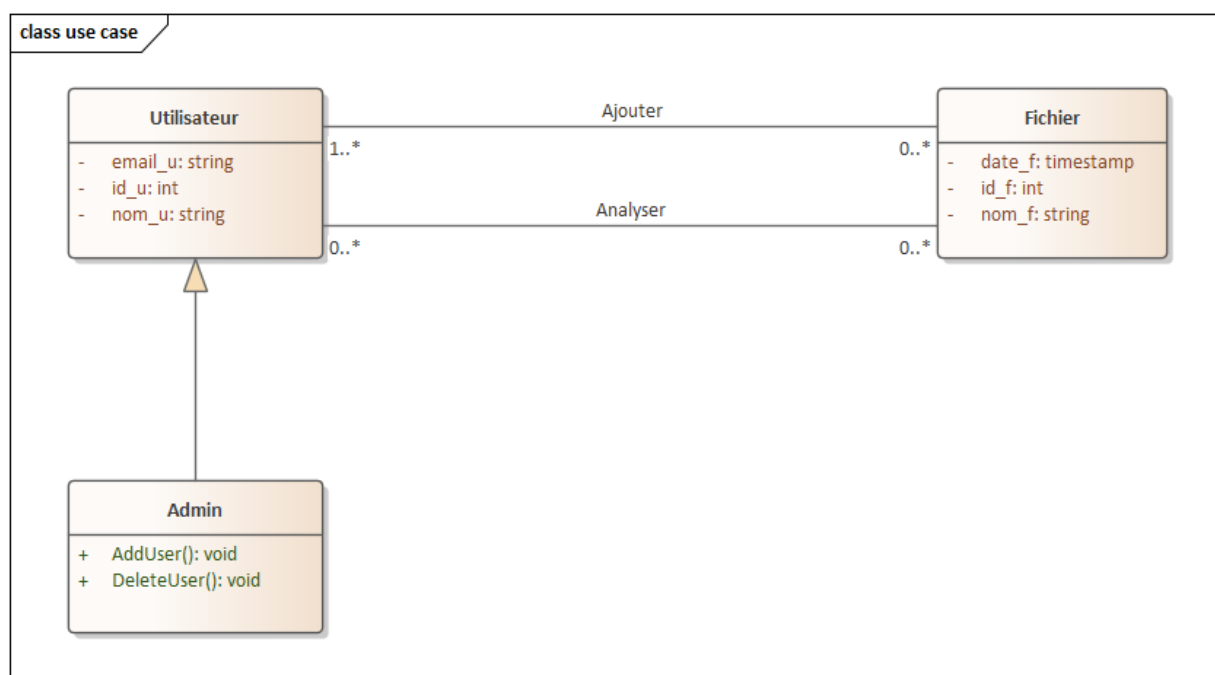


FIGURE 3.4 – Diagramme de class

### 3.7 Conclusion

Au cours de ce chapitre, nous avons conçu les différents composants de notre système. Maintenant, nos applications est prête à être codées. Le prochain chapitre concerne la mise en place de nos applications.

# Chapitre 4

## Machines à vecteurs de support

### Introduction

La classification consiste à classer des individus en fonction de certaines de leurs caractéristiques. Il existe différents types de classification, mais un des plus intuitifs et des plus utilisés est la classification supervisée. L'idée de la classification supervisée est d'apprendre une règle de classement à partir d'un ensemble de données dont le classement est déjà connu. Une fois la règle apprise, il est possible de l'appliquer pour catégoriser de nouvelles données, dont le classement est inconnu. Les machines à vecteurs de support, ou SVM (Support Vector Machines), sont une technique relativement récente (elles ont été introduites en 1992 par Vladimir Vapnik, Bernhard Boser et Isabelle Guyon) de classification supervisée qui suscite beaucoup d'intérêt pour ses bonnes performances dans un large éventail d'applications pratiques.

Cet chapitre constitue une introduction aux machines à vecteurs de support. Après la présentation de certains éléments essentiels de la théorie de l'optimisation, leur fonctionnement sera expliqué, d'abord dans le cas simple où les données sont linéairement séparables, puis dans le cas général où elles ne le sont pas nécessairement. Enfin, le concept de marge souple, très important pour l'utilisation des SVM dans la pratique, sera introduit.

### 4.1 Éléments de la théorie de l'optimisation

#### 4.1.1 Introduction à l'optimisation

Soit  $f$  une fonction dérivable définie sur un domaine ouvert  $D$  dont on cherche à connaître le minimum. Supposons que ce minimum est atteint au point  $\mathbf{x}^*$ , c'est-à-dire que  $f(\mathbf{x}^*) \leq f(\mathbf{x})$  pour tout  $\mathbf{x} \in D$ . Alors, nécessairement,  $f'(\mathbf{x}^*) = 0$ . Par conséquent, afin de déterminer le minimum de la fonction  $f$ , il suffit de considérer les points où la dérivée de  $f$  s'annule.

Cependant, supposons maintenant que le problème ne soit plus simplement de déterminer le minimum de la fonction  $f$ , mais plutôt de trouver le point  $\mathbf{x}$  qui minimise la valeur de  $f(\mathbf{x})$  tout en respectant aussi la contrainte  $g(\mathbf{x}) = 0$ , où  $g$  est aussi une fonction définie sur le domaine  $D$ . Dans ce cas, ce point n'annulera pas nécessairement la dérivée de  $f$ , puisque l'ajout de la contrainte restreint  $f$  à un ensemble  $S = \{\mathbf{x} \in D : g(\mathbf{x}) = 0\}$  et que le point qui minimise  $f$  dans  $S$  ne la minimise pas nécessairement dans  $D$ .

Le but de l'optimisation est de trouver une solution à un tel problème. De manière générale, un problème d'optimisation est un problème dans lequel on cherche un point qui minimise ou qui maximise une certaine fonction et qui est sujet à certaines contraintes. Il s'agit donc

d'un problème s'écrivant ainsi : Minimiser  $f(\mathbf{x})$  pour  $\mathbf{x} \in D$  sujet à  $\begin{cases} g_i(\mathbf{x}) \leq 0 & i = 1, \dots, k, \\ h_j(\mathbf{x}) = 0 & j = 1, \dots, m. \end{cases}$

La fonction  $f$  est appelée la fonction objectif, les fonctions  $g_i (i = 1, \dots, k)$  sont appelées les contraintes d'inégalité et les fonctions  $h_j (j = 1, \dots, m)$  sont appelées les contraintes d'égalité. Puisque maximiser  $f(\mathbf{x})$  est équivalent à minimiser  $-f(\mathbf{x})$  et que toutes les contraintes peuvent être réécrites soit sous la forme  $g(\mathbf{x}) \leq 0$  ou sous la forme  $h(\mathbf{x}) = 0$ , tous les problèmes d'optimisation peuvent s'écrire sous la forme ci-dessus.

### 4.1.2 La convexité

Introduisons maintenant quelques définitions. Un ensemble  $D \subseteq \mathbb{R}^n$  est dit convexe si pour tout  $\mathbf{x}_1, \mathbf{x}_2 \in D$  et pour tout  $\theta \in ]0, 1[$ ,

$$(\theta \mathbf{x}_1 + (1 - \theta) \mathbf{x}_2) \in D.$$

Une fonction  $f : D \rightarrow \mathbb{R}$  est dite convexe si pour tout  $\mathbf{x}_1, \mathbf{x}_2 \in D$  (où  $D$  est un ensemble convexe) et pour tout  $\theta \in ]0, 1[$ ,

$$f(\theta \mathbf{x}_1 + (1 - \theta) \mathbf{x}_2) \leq \theta f(\mathbf{x}_1) + (1 - \theta) f(\mathbf{x}_2).$$

On dit que la fonction  $f$  est strictement convexe si l'inégalité précédente est stricte.

Enfin, un problème d'optimisation est dit convexe si le domaine  $D$ , la fonction objectif et toutes les contraintes sont convexes.

La convexité possède une propriété très intéressante qui sera essentielle plus loin : si  $f$  est une fonction convexe et  $\mathbf{x}^*$  un point tel que

$$\frac{\partial f(\mathbf{x}^*)}{\partial \mathbf{x}} = 0,$$

alors  $\mathbf{x}^*$  est un minimum global de  $f$ . De plus, si  $f$  est strictement convexe, alors  $\mathbf{x}^*$  est l'unique point où  $f$  atteint ce minimum global

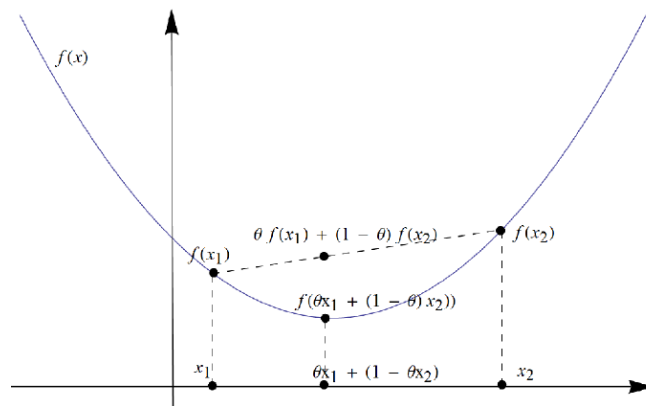


FIGURE 4.1 – Une fonction convexe

Notons quici et pour le reste de ce texte, si  $\mathbf{x} = (x_1, \dots, x_n)$ , alors

$$\frac{\partial f(\mathbf{x}_a)}{\partial \mathbf{x}} = \left( \frac{\partial f(\mathbf{x}_a)}{\partial x_1}, \dots, \frac{\partial f(\mathbf{x}_a)}{\partial x_n} \right).$$

### 4.1.3 La dualité et le théorème de Kuhn-Tucker

Considérons le problème d'optimisation suivant :

$$\text{minimiser } f(\mathbf{x}) \text{ pour } \mathbf{x} \in D \text{ sujet à } \begin{cases} g_i(\mathbf{x}) \leq 0 & i = 1, \dots, k, \\ h_j(\mathbf{x}) = 0 & j = 1, \dots, m \end{cases}$$

où  $D \subseteq \mathbb{R}^n$ ,  $f : D \rightarrow \mathbb{R}$  est la fonction objectif.  $g_i : D \rightarrow \mathbb{R} (i = 1, \dots, k)$  sont les contraintes d'inégalité et  $h_j : D \rightarrow \mathbb{R} (j = 1, \dots, m)$  sont les contraintes d'égalité. A partir de ce problème, il est possible de définir la fonction suivante, appelée le Lagrangien :

$$L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{x}) + \sum_{i=1}^k \alpha_i g_i(\mathbf{x}) + \sum_{j=1}^m \beta_j h_j(\mathbf{x}) = f(\mathbf{x}) + \boldsymbol{\alpha}^T \mathbf{g}(\mathbf{x}) + \boldsymbol{\beta}^T \mathbf{h}(\mathbf{x})$$

où  $\alpha_i, \beta_j \in \mathbb{R} (i = 1, \dots, k, j = 1, \dots, m)$ ,  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k)$ ,  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_m)$ ,  $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_k(\mathbf{x}))$  et  $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), \dots, h_m(\mathbf{x}))$ . Les constantes  $\alpha_i (i = 1, \dots, k)$  et  $\beta_j (j = 1, \dots, m)$  sont appelées les multiplicateurs de Lagrange. Il est possible d'utiliser le Lagrangien pour construire un nouveau problème d'optimisation, appelé le problème dual, qui possède des liens très intéressants avec le problème original, que nous appellerons dorénavant le problème primal.

Considérons la fonction

$$\theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \inf_{\mathbf{x} \in D} L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}).$$

Il s'agit d'une fonction qui prend la valeur minimale du Lagrangien pour un  $\boldsymbol{\alpha}$  et un  $\boldsymbol{\beta}$  donnés. Soumettons maintenant cette fonction à la contrainte  $\boldsymbol{\alpha} \geq 0$ . Alors, si  $(\boldsymbol{\alpha}, \boldsymbol{\beta})$  respecte cette contrainte.

$$\theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) \leq f(\mathbf{x})$$

pour tout  $\mathbf{x} \in D$  tel que  $g_i(\mathbf{x}) \leq 0 (i = 1, \dots, k)$  et  $h_j(\mathbf{x}) = 0 (j = 1, \dots, m)$ .

$$\begin{aligned} \text{En effet, } \theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \inf_{\mathbf{u} \in D} L(\mathbf{u}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &\leq L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &= f(\mathbf{x}) + \boldsymbol{\alpha}^T \mathbf{g}(\mathbf{x}) + \boldsymbol{\beta}^T \mathbf{h}(\mathbf{x}) \\ &\leq f(\mathbf{x}) \end{aligned}$$

puisque  $\mathbf{h}(\mathbf{x}) = 0$  et  $\mathbf{g}(\mathbf{x}) \leq 0$ , et  $\boldsymbol{\alpha} \geq 0$ .

Une conséquence immédiate de ce fait est que la valeur maximale que peut prendre  $\theta(\boldsymbol{\alpha}, \boldsymbol{\beta})$  en respectant la contrainte  $\boldsymbol{\alpha} \geq 0$  est bornée supérieurement par la valeur minimale du problème primal, c'est-à-dire que

$$\sup\{\theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) : \boldsymbol{\alpha} \geq 0\} \leq \inf\{f(\mathbf{x}) : \mathbf{g}(\mathbf{x}) \leq 0, \mathbf{h}(\mathbf{x}) = 0\}$$

Par conséquent, s'il existe une solution réalisable  $\mathbf{x}^*$  du problème primal telle que  $f(\mathbf{x}^*) = \theta(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ , avec  $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) \in \mathbb{R}^{l+m}$  et  $\boldsymbol{\alpha}^* \geq 0$ , alors  $\mathbf{x}^*$  correspond au minimum du problème primal et  $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$  correspond au maximum de  $\theta(\boldsymbol{\alpha}, \boldsymbol{\beta})$  sujette à la contrainte  $\boldsymbol{\alpha} \geq 0$ .

Ceci amène à définir ainsi un problème d'optimisation dual :

$$\text{maximiser } \theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) \text{ sujet à } \boldsymbol{\alpha} \geq 0.$$



Si la valeur maximale du problème dual correspond à la valeur minimale du problème primal, alors il est possible de résoudre le problème dual pour découvrir la solution du problème primal. Cependant, en général, il n'est pas toujours certain que les solutions du problème primal et du problème dual coïncident. Il peut en effet exister un écart de dualité (duality gap) entre les deux solutions. Cependant, le théorème de la dualité forte assure que dans le cas où la fonction  $f$  est convexe et où les contraintes sont des fonctions affines, c'est-à-dire qu'elles sont de la forme  $A\mathbf{x} + \mathbf{b}$ , avec  $A$  une matrice et  $\mathbf{b}$  un vecteur, l'écart de dualité est nul, et donc la valeur maximale du problème dual correspond à la valeur minimale du problème primal.

Ceci permet d'établir des conditions nécessaires et suffisantes pour qu'une solution du problème primal soit optimale. Ces conditions sont présentées dans le théorème de Karush-Kuhn-Tucker. Théorème de Karush-Kuhn-Tucker : Soit un problème d'optimisation convexe de domaine  $D \subseteq \mathbb{R}^n$  :

$$\text{minimiser } f(\mathbf{x}) \text{ pour } \mathbf{x} \in D \text{ sujet à } \begin{cases} g_i(\mathbf{x}) \leq 0 & i = 1, \dots, k, \\ h_j(\mathbf{x}) = 0 & j = 1, \dots, m \end{cases}$$

avec  $f \in C^1$  et toutes les contraintes affines. Alors, un point  $\mathbf{x}^*$  est un optimum si et seulement si il existe des vecteurs  $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$  tels que

$$\begin{aligned} \frac{\partial L(\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \mathbf{x}} &= 0, \\ \frac{\partial L(\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}} &= 0, \\ \alpha_i^* g_i(\mathbf{x}^*) &= 0, \quad i = 1, \dots, k \\ g_i(\mathbf{x}^*) &\leq 0, \quad i = 1, \dots, k \\ \alpha_i^* &\geq 0, \quad i = 1, \dots, k. \quad \square \end{aligned}$$

Ces conditions correspondent à imposer l'existence d'une solution réalisable du dual ayant la même valeur. En effet, imposer

$$\frac{\partial L(\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \mathbf{x}} = 0$$

revient à s'assurer que  $\theta(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = \inf_{\mathbf{x} \in D} L(\mathbf{x}, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = L(\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ . puisque le minimum du Lagrangien sans contrainte est atteint lorsque sa dérivée est nulle. En effet, puisque la fonction objectif est une fonction convexe et que les contraintes sont des fonctions affines, il est simple de vérifier que le Lagrangien est une fonction convexe. Or, on a vu que tout point qui annule la dérivée d'une fonction convexe est un minimum global de cette fonction. Par conséquent, le minimum du Lagrangien sans contraintes est bien atteint lorsque sa dérivée est nulle.

Les conditions  $\frac{\partial L(\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}} = 0$  et  $g_i(\mathbf{x}^*) \leq 0, i = 1, \dots, k$  sont simplement les contraintes du problème primal (remarquons que dériver le Lagrangien par rapport à  $\beta_j$  donne  $h_j$ ), et visent seulement à assurer que la solution est réalisable pour le problème primal. Ensuite, la condition  $\boldsymbol{\alpha}^* \geq 0$  assure que  $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$  est une solution réalisable du dual. Enfin, la condition  $\alpha_i^* g_i(\mathbf{x}^*) = 0$  pour tout  $i = 1, \dots, k$ , appelée la condition complémentaire de Karush-Kuhn-Tucker, assure que  $\theta(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = f(\mathbf{x}^*)$ . En effet, on a alors  $\beta_j h_j(\mathbf{x}^*) = 0$  pour tout  $j = 1, \dots, m$  et  $\alpha_i^* g_i(\mathbf{x}^*) = 0$  pour tout  $i = 1, \dots, k$ . Par conséquent,

$$f(\mathbf{x}^*) = f(\mathbf{x}^*) + \boldsymbol{\alpha}^{*T} \mathbf{g}(\mathbf{x}^*) + \boldsymbol{\beta}^{*T} \mathbf{h}(\mathbf{x}^*) = L(\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = \theta(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*).$$

Comme  $f(\mathbf{x}^*) = \theta(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ ,  $\mathbf{x}^*$  est une solution optimale pour le problème primal et  $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$  est une solution optimale pour le dual. Les conditions de Karush-Kuhn-Tucker peuvent être très utiles pour vérifier si une solution est optimale.

## 4.2 Machines à vecteurs de support pour données linéairement séparables

### 4.2.1 Introduction aux machines à vecteurs de support

Maintenant que certains éléments importants de la théorie de l'optimisation ont été présentés, nous sommes en mesure d'introduire les machines à vecteurs de support. Les machines à vecteurs de support (SVM) sont un algorithme dont le but est de résoudre les problèmes de discrimination à deux classes. On appelle problème de discrimination à deux classes un problème dans lequel on tente de déterminer la classe à laquelle appartient un individu (individu est ici employé au sens de constituant d'un ensemble) parmi deux choix possibles. En réalité, plusieurs méthodes ont été suggérées pour étendre l'application des SVM aux problèmes de discrimination à plus de deux classes, et il existe aussi une modification qui permet de les utiliser pour la régression. mais nous nous concentrerons ici sur les problèmes de discrimination à deux classes.

Pour ce faire, on utilise les caractéristiques connues de cet individu. Ces  $n$  caractéristiques sont représentées par un vecteur  $\mathbf{x} \in \mathbb{R}^n$ . La classe à laquelle appartient l'individu est représentée par  $y \in \{-1, 1\}$ , où une des classes possible est représentée par  $-1$  et l'autre par  $1$ . Par conséquent, avec cette notation, le problème est de déterminer la valeur de  $y$  en se servant de  $\mathbf{x}$ .

Pour  $y$  parvenir, les machines à vecteurs de support utilisent un ensemble de données pour lesquelles le classement est déjà connu et s'en servent pour construire une règle qui permet d'effectuer une bonne classification. Cet ensemble de données est appelé l'ensemble d'apprentissage. La règle trouvée avec l'ensemble d'apprentissage doit être la plus générale possible, puisqu'il faut aussi qu'elle soit bonne pour de nouvelles données qui n'étaient pas dans l'ensemble d'apprentissage. Nous présentons ici comment les SVM font pour trouver cette règle dans le cas le plus simple possible, c'est-à-dire le cas où les données sont linéairement séparables. Les autres cas seront traités dans les sections 3 et 4.

### 4.2.2 Hyperplan séparateur

Supposons que nous disposons d'un ensemble d'apprentissage de  $l$  données de la forme  $(\mathbf{x}_i, y_i) \in \mathbb{R}^n \times \{-1, 1\} (i = 1, \dots, l)$ , dont nous voulons nous servir pour déterminer une règle permettant de classer les données. Supposons aussi que ces données sont linéairement séparables, c'est-à-dire qu'il existe un hyperplan dans  $\mathbb{R}^n$  tel que toutes les données appartenant à la classe  $1$  se retrouvent d'un côté de l'hyperplan alors que celles de la classe  $-1$  se situent de l'autre côté.

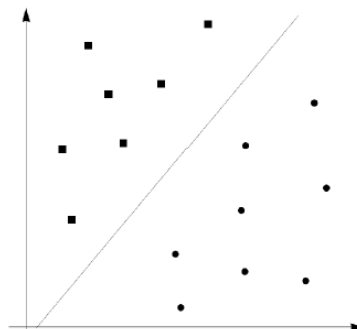


FIGURE 4.2 – Des données linéairement séparables

Plus formellement, les données sont dites linéairement séparables s'il existe un hyperplan

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

tel que  $\mathbf{w} \cdot \mathbf{x} + b > 0$  pour tout  $\mathbf{x}$  appartenant à la classe 1, et  $\mathbf{w} \cdot \mathbf{x} + b < 0$  pour tout  $\mathbf{x}$  appartenant à la classe  $-1$ , avec  $\mathbf{w} = (w_1, \dots, w_n) \in \mathbb{R}^n$  le vecteur des coefficients de l'hyperplan et  $b \in \mathbb{R}$  un scalaire appelé le biais (remarquons que tout hyperplan peut s'écrire sous cette forme). Nous dirons d'un tel hyperplan qu'il sépare les données.

Sous l'hypothèse que les données sont linéairement séparables, trouver une règle pour les classer est très simple. En effet, il suffit de prendre un hyperplan qui sépare les classes, puis de classer les données selon le côté de l'hyperplan où elles se trouvent. Plus formellement, soit

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

un hyperplan qui sépare les données. Alors, il suffit d'utiliser la fonction suivante (parfois appelée la fonction indicatrice) pour effectuer la classification :

$$\text{Classe}(\mathbf{x}) = \text{signe}(\mathbf{w} \cdot \mathbf{x} + b),$$

où

$$\text{signe}(\mathbf{w} \cdot \mathbf{x} + b) = \begin{cases} -1 & \text{si } \mathbf{w} \cdot \mathbf{x} + b < 0 \\ 0 & \text{si } \mathbf{w} \cdot \mathbf{x} + b = 0 \\ 1 & \text{si } \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$

Cette fonction classe les données par rapport au côté de l'hyperplan où elles se trouvent. On remarque que si un ensemble de données est séparé par un hyperplan, il sera parfaitement classé par cette fonction. Notons que si une donnée est directement sur l'hyperplan (ce qui peut arriver en considérant des données qui ne sont pas dans l'ensemble d'apprentissage), elle sera assignée à la classe 0, ce qui signifie qu'elle ne peut être classée par le modèle actuel. Dans ce cas, il est possible de la laisser inclassée, d'utiliser une autre règle ou de l'assigner aléatoirement à l'une des deux classes.

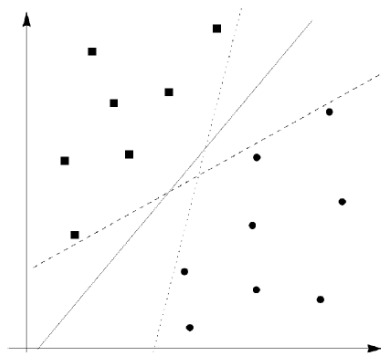


FIGURE 4.3 – Il existe une infinité d'hyperplans pouvant séparer les données

Grace à la fonction indicatrice, on constate qu'il suffit de trouver un hyperplan qui sépare les données pour déterminer une règle permettant de les classer. Cependant, si les données sont linéairement séparables, il existe une infinité d'hyperplans qui peuvent servir de séparateurs. L'idée des machines à vecteurs de support est de choisir le meilleur hyperplan, c'est-à-dire celui qui donnera la règle

qui se généralisera le mieux à d'autres données que celles de l'ensemble d'apprentissage. Afin de déterminer ce qui caractérise le meilleur hyperplan, introduisons le concept de marge.

### 4.2.3 Marge et hyperplan canonique

Définissons la marge d'un hyperplan comme étant la distance entre l'hyperplan et la donnée la plus proche. Plus formellement, si  $\text{dist}(\mathbf{x}, \mathbf{w}, b)$  représente la distance euclidienne entre le point  $\mathbf{x}$  et l'hyperplan  $\mathbf{w} \cdot \mathbf{x} + b = 0$ , alors la marge  $M$  est définie ainsi :

$$M = \min \{ \text{dist}(\mathbf{x}_i, \mathbf{w}, b) : i = 1, \dots, l \}$$

où les  $\mathbf{x}_i$  sont les données de l'ensemble d'apprentissage. Par abus de langage, nous dirons dans ce texte qu'un point se trouve sur la marge si sa distance avec l'hyperplan correspond exactement à la marge. D'après un résultat de la théorie de l'apprentissage statistique, l'hyperplan qui aura

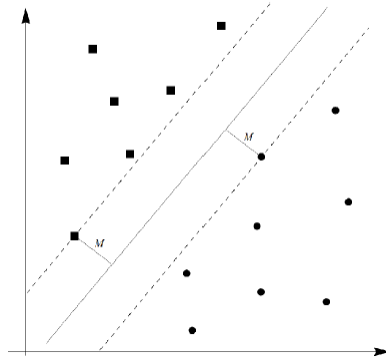


FIGURE 4.4 – La marge

la meilleure généralisation est celui qui possède la plus grande marge. Ce concept est à la base des machines à vecteurs de support. Dans le cas le plus simple, c'est-à-dire celui où les données sont linéairement séparables, les SVM trouvent l'hyperplan qui sépare les données avec la plus vaste marge possible, puis utilisent cet hyperplan pour classer de nouvelles données à l'aide de la fonction indicatrice donnée plus haut.

Toutefois, le problème de trouver l'hyperplan avec la marge maximale est mal posé, puisqu'il existe en réalité une infinité de manières différentes d'écrire le même hyperplan. En effet, supposons que l'hyperplan

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

soit un hyperplan dont la marge est maximale, et soit  $\lambda \in \mathbb{R}^+ \setminus \{0\}$ . Alors, l'hyperplan

$$\lambda \mathbf{w} \cdot \mathbf{x} + \lambda b = 0$$

est en réalité le même hyperplan et sépare les données, puisque  $\lambda$  est positif. Par conséquent,  $\lambda \mathbf{w} \cdot \mathbf{x} + \lambda b = 0$  correspond aussi à l'hyperplan dont la marge est maximale, mais possède un vecteur des coefficients et un biais différents (si  $\lambda \neq 1$ )

Le nombre infini de manières d'écrire la solution du problème de l'hyperplan avec la plus vaste marge complique sa résolution. Afin de rendre le problème bien posé, introduisons le concept d'hyperplan canonique. Un hyperplan  $\mathbf{w} \cdot \mathbf{x} + b = 0$  est dit canonique si

$$\min \{ |\mathbf{w} \cdot \mathbf{x}_i + b| : i = 1, \dots, l \} = 1,$$

où les  $\mathbf{x}_i$  sont les données d'apprentissage. On peut montrer que ce minimum correspond aux données qui sont directement sur la marge.

On peut aussi montrer que tout hyperplan qui sépare les données peut s'écrire sous forme canonique et qu'il n'existe qu'une seule façon d'écrire un hyperplan pour qu'il soit canonique.

Ainsi, en ne considérant que les hyperplans canoniques, chaque hyperplan s'écrit de manière unique. De plus, il n'existe qu'un seul hyperplan pour lequel la marge est maximale. Ceci deviendra évident un peu plus loin, puisque le vecteur des coefficients de l'hyperplan sera exprimé comme étant le point qui minimise une fonction strictement convexe (rappelons que les fonctions strictement convexes n'ont qu'un unique minimum global). Par conséquent, en ne considérant que les hyperplans canoniques, le problème de trouver l'hyperplan avec la plus grande marge est bien posé.

#### 4.2.4 Trouver l'hyperplan

On peut montrer que pour un hyperplan canonique  $\mathbf{w} \cdot \mathbf{x} + b = 0$ , la marge  $M$  est donnée par l'expression

$$M = \frac{1}{\|\mathbf{w}\|},$$

où  $\|\mathbf{w}\| = \sqrt{w_1^2 + \dots + w_n^2}$ . On voit donc que plus  $\|\mathbf{w}\|$  est petite, plus la marge de l'hyperplan canonique correspondant est grande. Ainsi, afin de trouver l'hyperplan qui sépare le mieux les données, il faut trouver celui qui respecte les conditions d'un hyperplan canonique et pour lequel  $\|\mathbf{w}\|$  est minimale.

La recherche du meilleur hyperplan peut donc s'écrire sous la forme du problème d'optimisation suivant :

$$\text{minimiser } \|\mathbf{w}\| \text{ sujet à } y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, \dots, l.$$

Les contraintes  $y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, i = 1, \dots, l$  assurent d'une part que l'hyperplan sépare les données correctement, et d'autre part qu'il est canonique. En effet,  $y_i (\mathbf{w} \cdot \mathbf{x}_i + b) > 0$  si et seulement si  $\text{signe}(\mathbf{w} \cdot \mathbf{x}_i + b) = y_i$ , donc si et seulement si  $\mathbf{x}_i$  est du bon côté de l'hyperplan. Ainsi, l'hyperplan doit correctement séparer les données. Ensuite, on peut montrer qu'imposer  $y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$  assure que pour toutes les données qui ne sont pas sur la marge,  $|\mathbf{w} \cdot \mathbf{x}_i + b| > 1$  et que  $|\mathbf{w} \cdot \mathbf{x}_i + b| = 1$  pour les données sur la marge, donc que l'hyperplan est canonique.

Nous avons ainsi formulé un problème d'optimisation dont la solution optimale est l'hyperplan canonique séparant les données avec la plus vaste marge possible. Cependant, il est possible de formuler un problème équivalent, mais avec une fonction objectif plus simple. En effet, comme

$$\|\mathbf{w}\| = \sqrt{\mathbf{w} \cdot \mathbf{w}},$$

minimiser  $\|\mathbf{w}\|$  est équivalent à minimiser  $\mathbf{w} \cdot \mathbf{w}$ . Évidemment, minimiser  $\mathbf{w} \cdot \mathbf{w}$  est équivalent à minimiser  $\frac{1}{2} \mathbf{w} \cdot \mathbf{w}$  (cette petite modification permet d'éviter d'avoir une constante dans la représentation duale du problème, comme nous le verrons un peu plus loin). Par conséquent, afin de trouver l'hyperplan canonique qui sépare les données avec la plus grande marge possible, il suffit de résoudre le problème d'optimisation suivant :

$$\text{minimiser } \frac{1}{2} \mathbf{w} \cdot \mathbf{w} \text{ sujet à } y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, \dots, l.$$

Une propriété très intéressante de ce problème est que

$$f(\mathbf{w}) = \mathbf{w} \cdot \mathbf{w} = w_1^2 + \dots + w_n^2$$

est une fonction strictement convexe. Ceci assure qu'il n'y a pas de minimum relatif et qu'il n'existe qu'une unique solution optimale.

### 4.2.5 Représentation duale

Il serait possible de résoudre le problème d'optimisation ci-dessus directement. Toutefois, sa représentation duale possède des propriétés très intéressantes qui auront des répercussions majeures lorsque nous considérerons les machines à vecteurs de supports pour le cas où les données ne sont pas linéairement séparables.

Commençons tout d'abord par écrire le Lagrangien. Pour ce faire, il est nécessaire de récrire les contraintes ainsi :

$$-(y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1) \leq 0, \quad i = 1, \dots, l.$$

Le Lagrangien est

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^l \alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1).$$

Il faut maintenant calculer la fonction objectif du problème dual. Rappelons que cette fonction correspond à la valeur minimale du Lagrangien pour un  $\alpha$  donné. Or, ce minimum correspond au point où la dérivée du Lagrangien par rapport aux variables du primal est nulle. On a donc ainsi

$$\begin{aligned} \frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i = 0 \\ \frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} &= - \sum_{i=1}^l \alpha_i y_i = 0 \end{aligned}$$

ce qu'il est possible de récrire de cette manière :

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i \text{ et } \sum_{i=1}^l \alpha_i y_i = 0$$

Utilisons ces équations pour récrire le Lagrangien minimal uniquement en fonction des variables duales :

$$\begin{aligned} \min_{\mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2} \mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^l \alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1) \\ &= \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j) \end{aligned}$$

Ainsi, nous avons le problème dual suivant :

$$\begin{aligned} &\text{maximiser } \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j) \\ &\text{sujet à } \begin{cases} \sum_{i=1}^l \alpha_i y_i = 0, \\ \alpha_i \geq 0 \end{cases} \quad i = 1, \dots, l. \end{aligned}$$

On remarque qu'il est nécessaire d'ajouter la contrainte  $\sum_{i=1}^l \alpha_i y_i = 0$  pour s'assurer que  $\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j)$  correspond bien au minimum du Lagrangien. En effet, alors que la contrainte  $\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i$  est automatiquement satisfaite par la construction de la fonction, rien n'assure que  $\sum_{i=1}^l \alpha_i y_i = 0$  est elle aussi respectée.

La solution de ce problème d'optimisation sera bien sûr un vecteur  $\alpha^* = (\alpha_1^*, \dots, \alpha_l^*)$ , alors que c'est l'équation d'un hyperplan qu'il faut pour classer les données à l'aide de la fonction indicatrice

$$\text{Classe}(\mathbf{x}) = \text{signe}(\mathbf{w} \cdot \mathbf{x} + b).$$

Il est toutefois possible de réécrire la fonction indicatrice ainsi, puisque  $\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i$

$$\text{Classe}(\mathbf{x}) = \text{signe}\left(\sum_{i=1}^l (\alpha_i y_i \mathbf{x}_i \cdot \mathbf{x}) + b\right).$$

Tout ce qu'il manque pour pouvoir utiliser cette fonction est la valeur de  $b$ . Or, comme l'hyperplan est canonique, et d'après les contraintes du problème primal, si une donnée  $\mathbf{x}_m$  se trouve sur la marge, alors on sait que

$$y_m(\mathbf{w} \cdot \mathbf{x}_m + b) = 1.$$

Donc,

$$b = \frac{1}{y_m} - \mathbf{w} \cdot \mathbf{x}_m = y_m - \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x}_m \text{ puisque } y_m \in \{-1, 1\}.$$

Ainsi, la résolution du problème dual permet de construire l'hyperplan canonique séparant les données avec la plus grande marge et de l'utiliser pour classer des données, tout comme la résolution du problème primal.

#### 4.2.6 Vecteurs de support

Comme la fonction  $\mathbf{w} \cdot \mathbf{w}$  est une fonction convexe continue et dérivable, que les contraintes  $-(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1) \leq 0$  sont des fonctions affines et que le domaine du problème est  $\mathbb{R}^n$ , la solution optimale trouvée respecte nécessairement les conditions de Karush-Kuhn-Tucker. En particulier, elle respecte la condition complémentaire de Karush-Kuhn-Tucker, c'est-à-dire que

$$\alpha_i^* (y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1) = 0, \quad i = 1, \dots, l,$$

où  $\alpha^*$  représente la solution optimale du problème dual et  $(\mathbf{w}^*, b^*)$  représente celle du problème primal.

Cette condition implique que si  $y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1 \neq 0$ , alors  $\alpha_i = 0$ . Par conséquent, les seuls cas où  $\alpha_i$  peut ne pas être nul sont ceux où  $y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1 = 0$ , c'est-à-dire ceux où

$$y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) = 1.$$

Or, les seuls points où  $y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) = 1$  sont ceux qui sont sur la marge. Par conséquent, seuls les points sur la marge peuvent avoir des  $\alpha_i$  non nuls. Ces points sont appelés les vecteurs de support.

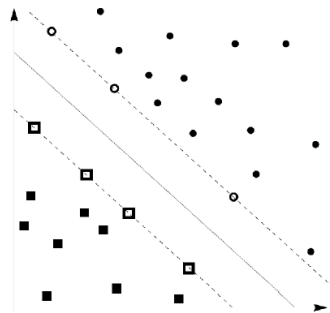


FIGURE 4.5 – Les vecteurs de support

La raison de ce nom est que ce sont les seuls points utiles pour déterminer l'hyperplan. En effet, rappelons que le vecteur des coefficients de l'hyperplan est donné par

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i$$

Ainsi, tout point qui n'est pas sur la marge n'apporte aucune contribution. puisque  $\alpha_i$  est alors nul. Si tous les points sauf les vecteurs de support étaient retirés de l'ensemble d'apprentissage. on retrouverait le même hyperplan.

Les vecteurs de support peuvent donc être vus comme les points contenant toute l'information essentielle du problème.

## 4.3 Machines à vecteurs de support pour données non linéairement séparables

### 4.3.1 Transformations

Jusqu'à présent, les machines à vecteurs de support permettent de trouver une règle pour classer les données lorsque celles-ci sont linéairement séparables. Cependant, il existe bien des cas pour lesquels il est impossible de séparer entièrement les données avec un hyperplan. Telles qu'elles ont été présentées jusqu'à présent, les SVM sont incapables de traiter un tel problème, puisqu'il est alors impossible que les contraintes

$$y_i (\mathbf{w}^* \cdot \mathbf{x}_i + b^*) \geq 1$$

soient toutes respectées. Afin de régler ce problème, il est possible d'appliquer une transformation aux données de sorte qu'une fois transformées, elles soient linéairement séparables. L'espace où se trouvent les données avant d'être transformées est appelé l'espace d'entrée (input space), alors qu'après avoir appliqué la transformation. les données se trouvent dans ce qu'on appelle l'espace de redescription (feature space). Il suffit alors de trouver l'hyperplan dans l'espace de redescription qui sépare le mieux ces données transformées. De retour dans l'espace d'entrée, le séparateur n'est pas linéaire.

$$\begin{array}{l} \Phi : \mathbb{R}^n \rightarrow \mathbb{R}^r \\ \text{Soit} \quad \mathbf{x} \rightarrow \Phi(\mathbf{x}) \end{array}$$

la transformation appliquée aux données pour les rendre linéairement séparables, avec  $r$  la dimension de l'espace de redescription. Très souvent,  $r > n$ . ce qui signifie que la transformation amène les données dans un espace de dimension supérieure afin de mieux pouvoir les séparer.

Pour trouver le séparateur, on procède de la même manière que précédemment, mais en substituant  $\Phi(\mathbf{x}_i)$  à  $\mathbf{x}_i (i = 1, \dots, l)$ . Il s'agit donc de résoudre le problème suivant :

$$\text{minimiser } \frac{1}{2} \mathbf{w} \cdot \mathbf{w} \text{ sujet à } y_i (\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1, \quad i = 1, \dots, l.$$

Le dual de ce problème est

$$\begin{array}{l} \text{maximiser } \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) \\ \text{sujet à } \begin{cases} \sum_{i=1}^l \alpha_i y_i = 0, \\ \alpha_i \geq 0 \end{cases} \quad i = 1, \dots, l. \end{array}$$



La fonction indicatrice associée à ce problème dual est par conséquent

$$\text{Classe}(\mathbf{x}) = \text{signe} \left( \sum_{i=1}^l (\alpha_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})) + b \right),$$

$$\text{où } b = y_m - \sum_{i=1}^l \alpha_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_m)$$

avec  $\Phi(\mathbf{x}_m)$  une donnée sur la marge de l'hyperplan dans l'espace de redescription.

Si la transformation utilisée est appropriée, la résolution d'un de ces problèmes (le primal ou le dual) permet de trouver un séparateur non linéaire avec la marge la plus grande possible, permettant ainsi d'utiliser les machines à vecteurs de support dans le cas où les données ne peuvent pas être séparées linéairement.

### 4.3.2 Les noyaux

Toutefois, l'utilisation des transformations pose certains problèmes. En effet, outre le fait qu'il faille choisir une bonne transformation, il faut l'appliquer à toutes les données, puis effectuer les calculs avec ces données transformées, c'est-à-dire dans l'espace de redescription. Or, comme la dimension de cet espace est bien souvent beaucoup plus grande que celle de l'espace d'entrée, les calculs requis peuvent devenir extrêmement longs à effectuer.

C'est ici que la formulation duale du problème d'optimisation prend toute son importance. En effet, on remarque que lorsque le problème est sous sa forme duale, les données de l'ensemble d'apprentissage n'apparaissent que dans un produit scalaire avec d'autres données du même ensemble. Il en est de même dans la fonction indicatrice duale. Ceci amène à définir comme suit une fonction appelée noyau (kernel) :

$$K : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$$

$$(\mathbf{x}_i, \mathbf{x}_j) \rightarrow \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j).$$

Cette fonction prend en entrée deux points dans l'espace d'entrée et calcule leur produit scalaire dans l'espace de redescription. L'avantage d'une telle fonction est qu'il n'est pas nécessaire d'appliquer une transformation aux données afin de calculer leur produit scalaire dans l'espace de redescription. Ce calcul peut se faire directement à partir des données de l'espace d'entrée.

Grâce au concept de noyau, il est possible de récrire le problème dual de cette manière :

$$\begin{aligned} & \text{maximiser } \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)) \\ & \text{sujet à } \begin{cases} \sum_{i=1}^l \alpha_i y_i = 0, \\ \alpha_i \geq 0, \end{cases} \quad i = 1, \dots, l. \end{aligned}$$

La fonction indicatrice peut elle aussi être réécrite :

$$\text{Classe}(\mathbf{x}) = \text{signe} \left( \sum_{i=1}^l (\alpha_i y_i K(\mathbf{x}_i, \mathbf{x})) + b \right),$$

$$\text{où } b = y_m - \sum_{i=1}^l \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_m)$$

avec  $\mathbf{x}_m$  un vecteur de support.

On remarque que de cette manière, lorsque la fonction noyau est connue, la transformation  $\Phi(\mathbf{x})$  n'apparaît nulle part, ni dans le problème, ni dans l'application de la solution. Par conséquent, grâce à la fonction noyau, il n'est pas nécessaire d'effectuer la transformation sur les données. Cette fonction permet

donc de faire tous les calculs nécessaires sans avoir à se préoccuper de la dimension de l'espace de redescription.

### 4.3.3 Noyaux : Construction et exemples

Il est bien de savoir qu'un noyau est tout ce qui est nécessaire pour utiliser les SVM dans le cas non linéaire, mais cette information est inutile sans la connaissance des noyaux qu'il est possible d'utiliser. Nous présentons maintenant les manières de construire des noyaux, ainsi que les noyaux les plus fréquemment utilisés pour les machines à vecteurs de support.

La première méthode pour construire un noyau est de choisir une transformation, de calculer le produit scalaire de deux éléments quelconques transformés, puis d'en faire une fonction. Cette méthode permet de déterminer le noyau d'une transformation bien spécifique, mais peut s'avérer difficile à utiliser, surtout lorsque le nombre de dimensions augmente. Une autre méthode consiste à utiliser le théorème de Mercer. D'après ce théorème, une fonction est un noyau si et seulement si elle est symétrique et semi-définie positive (voir [2] pour plus de détails). Ainsi, au lieu de choisir une transformation puis de calculer son noyau, on choisit plutôt une fonction symétrique et semi-définie positive, ce qui nous assure qu'elle correspond au produit scalaire d'une quelconque transformation. La transformation elle-même est généralement inconnue pour les noyaux construits de cette façon.

Le théorème de Mercer permet aussi de construire des noyaux à partir de noyaux déjà connus. En effet, si  $K_1$  et  $K_2$  sont des noyaux, alors les fonctions suivantes sont aussi des noyaux :

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= K_1(\mathbf{x}_i, \mathbf{x}_j) + K_2(\mathbf{x}_i, \mathbf{x}_j); \\ K(\mathbf{x}_i, \mathbf{x}_j) &= aK_1(\mathbf{x}_i, \mathbf{x}_j) \quad a \in \mathbb{R} \\ K(\mathbf{x}_i, \mathbf{x}_j) &= K_1(\mathbf{x}_i, \mathbf{x}_j) K_2(\mathbf{x}_i, \mathbf{x}_j). \end{aligned}$$

Les noyaux présentés dans la figure 6 sont les plus fréquemment utilisés. Le choix du noyau a un impact majeur sur la performance des SVM. Quelques méthodes ont été suggérées pour sélectionner un bon noyau, mais il s'agit encore d'un sujet de recherche actif. En général, le noyau gaussien est souvent préféré, puisqu'il donne de bonnes performances dans toutes sortes de contextes.

Nom	Noyau
Linéaire	$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)$
Polynomial de degré $d$	$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$
Gaussien	$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{(\mathbf{x}_i - \mathbf{x}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j)}{2\sigma^2}}$
Multiquadratique inverse	$K(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\sqrt{(\mathbf{x}_i - \mathbf{x}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j) + \beta}}$

TABLE 4.1 – Les noyaux les plus fréquemment utilisés

## 4.4 Marges souples

### 4.4.1 Machines à vecteurs de support et bruit

En pratique, les données sont rarement parfaites. Il y a souvent du \* bruit \*, c'est-à-dire des données qui sont mal classées par un modèle qui est toutefois excellent en général. Il s'agit donc d'erreurs qui sont inévitables, même pour les meilleurs modèles. Toutefois, les machines à vecteurs de support ne permettent pas de tenir compte de ce phénomène, puisque dans les contraintes, toutes les données doivent être correctement classées. Supposons par exemple qu'un ensemble de données serait très bien séparé par un hyperplan, mais qu'il n'est pas linéairement

séparable dû à la présence d'un certain bruit dans les données. Dans un tel cas, il serait impossible de construire une SVM linéaire, car il est impossible que toutes les contraintes soient respectées.

Afin de contourner ce problème, il peut être tentant d'utiliser un noyau quelconque afin de rendre les données linéairement séparables. Ceci est en effet toujours possible en utilisant un noyau polynomial avec un degré suffisamment élevé. Toutefois, bien que les données de l'ensemble d'apprentissage seront parfaitement séparées, la règle trouvée risque de très mal se généraliser. puisqu'elle va tenir compte de toutes les petites variations et ainsi généraliser des phénomènes qui sont en réalité bien spécifiques à l'ensemble de données actuel.

#### 4.4.2 Marge souple

Un meilleur moyen serait de permettre à quelques données d'être à l'intérieur de la marge ou du mauvais côté de l'hyperplan. Il s'agit du concept de marge souple (soft margin). Une première idée serait de tenter de maximiser la marge tout en minimisant le nombre de données mal classées. Toutefois, le nombre de données mal classées peut être trompeur, puisqu'il ne permet pas de déterminer si une donnée était presque correctement classée ou si elle était en réalité très loin de l'hyperplan.

Une meilleure idée est d'attribuer à chaque donnée  $\mathbf{x}_i$  une valeur  $\xi_i$  qui représente à quel point la donnée est éloignée d'un bon classement, puis de tenter de minimiser la somme des  $\xi_i$ . Plus formellement, au lieu d'imposer

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, \dots, l$$

ce qui oblige les données à être bien classées, les contraintes seront plutôt

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \dots, l \text{ avec } \xi_i \geq 0.$$

Par conséquent, il est possible pour une donnée d'être du mauvais côté de la marge, si  $\xi_i$  est non nul. On dira d'une donnée qu'elle est du mauvais côté de la marge si elle est mal classée ou si sa distance par rapport à l'hyperplan séparateur est plus petite que la marge (remarquons que les points pour lesquels  $\xi_i \neq 0$  ne sont pas considérés dans le calcul de la marge). L'objectif est ainsi de maximiser la marge tout en minimisant la somme des  $\xi_i$ . Le problème d'optimisation devient alors

$$\text{minimiser } \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^l \xi_i \text{ sujet à } \begin{cases} y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i & i = 1, \dots, l, \\ \xi_i \geq 0 \end{cases}$$

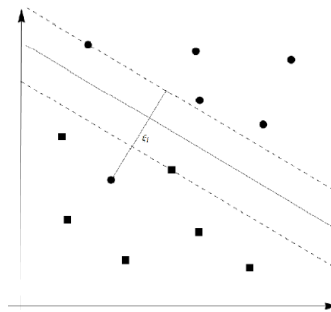


FIGURE 4.6 – Marge souple

où  $C > 0$  est une constante qui représente la pénalité d'avoir des données mal classées. Lorsque  $C$  est très élevée, il y aura très peu de données mal classées, alors qu'il y en aura plus pour une valeur plus faible de cette constante. Le choix de  $C$  a une grande influence sur le modèle. En pratique, plusieurs modèles sont souvent construits, avec différentes valeurs de  $C$ , puis le meilleur est choisi.

### 4.4.3 Représentation duale

Il est possible de construire le dual de ce problème de la même manière que précédemment. Le Lagrangien est

$$L(\mathbf{w}, b, \xi, \alpha, \mathbf{r}) = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^l \xi_i - \sum_{i=1}^l \alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^l r_i \xi_i.$$

Afin de trouver le Lagrangien minimal pour un  $(\alpha, r)$  donné, il faut le dériver par rapport aux variables primales. On obtient alors

$$\begin{aligned} \frac{\partial L(\mathbf{w}, b, \xi, \alpha, \mathbf{r})}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^l y_i \alpha_i \mathbf{x}_i = 0; \\ \frac{\partial L(\mathbf{w}, b, \xi, \alpha, \mathbf{r})}{\partial b} &= - \sum_{i=1}^l y_i \alpha_i = 0; \\ \frac{\partial L(\mathbf{w}, b, \xi, \alpha, \mathbf{r})}{\partial \xi_i} &= C - \alpha_i - r_i = 0. \end{aligned}$$

De ceci, on obtient

$$\mathbf{w} = \sum_{i=1}^l y_i \alpha_i \mathbf{x}_i, \quad \sum_{i=1}^l y_i \alpha_i = 0 \text{ et } C = \alpha_i + r_i \quad \text{pour tout } i \in \{1, \dots, l\}.$$

Utilisons ces expressions pour réécrire le Lagrangien uniquement en fonction des variables duales :

$$\begin{aligned} L(\mathbf{w}, b, \xi, \alpha, \mathbf{r}) &= \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^l \xi_i - \sum_{i=1}^l \alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^l r_i \xi_i \\ &= \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j). \end{aligned}$$

Il s'agit de la fonction objectif du problème dual pour la marge souple. On remarque que cette fonction est exactement la même que celle obtenue précédemment. La différence se situe au niveau des contraintes.

En effet, rappelons que dans le problème dual, les multiplicateurs de Lagrange qui sont associés à des contraintes d'inégalités doivent être supérieurs ou égaux à zéro (voir la sous-section 1.3). Par conséquent,  $\alpha_i \geq 0$  et  $r_i \geq 0$ . Toutefois,  $r_i$  n'apparaît pas dans le problème dual, mais on sait que  $C - \alpha_i - r_i = 0$ . La contrainte  $r_i \geq 0$  s'écrit donc aussi  $C - \alpha_i \geq 0$ . Ceci implique que  $\alpha_i \leq C$ . Le problème dual est ainsi

$$\begin{aligned} &\text{maximiser } \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j) \\ &\text{sujet à } \begin{cases} \sum_{i=1}^l \alpha_i y_i = 0, \\ 0 \leq \alpha_i \leq C \end{cases} \quad i = 1, \dots, l. \end{aligned}$$

#### 4.4.4 Remarques

Les conditions de Karush-Kuhn-Tucker tiennent toujours dans le cas de la marge souple. Par conséquent, d'après la condition complémentaire, pour la solution optimale, les égalités suivantes sont vérifiées :

$$\begin{aligned}\alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i) &= 0 \text{ pour } i = 1, \dots, l; \\ r_i \xi_i &= (C - \alpha_i) \xi_i = 0 \text{ pour } i = 1, \dots, l.\end{aligned}$$

Ceci implique que si  $\xi_i \neq 0$ , alors  $C - \alpha_i = 0$  (puisque  $(C - \alpha_i) \xi_i = 0$ ), et donc  $\alpha_i = C$ . De plus, si un point est tel que  $\xi_i \neq 0$ , alors il est du mauvais côté de la marge, ce qui découle directement du rôle de  $\xi_i$  dans le problème d'optimisation. À l'opposé, tous les points pour lesquels  $\xi_i = 0$  sont du bon côté de la marge, et ainsi nécessairement bien classés.

D'autre part, si, pour une certaine donnée, on a  $0 < \alpha_i < C$ , alors celle-ci est exactement sur la marge. En effet, on a alors  $0 < \alpha_i < C$ ,  $\alpha_i \neq C$ , et donc il faut que  $\xi_i = 0$  pour que  $(C - \alpha_i) \xi_i = 0$ . De plus,  $\alpha_i \neq 0$ , ce qui implique que  $y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i = 0$  afin de respecter l'égalité  $\alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i) = 0$ . Comme  $\xi_i = 0$ , il s'ensuit que

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) = 1$$

et donc que  $\mathbf{x}_i$  est directement sur la marge. Enfin, les points pour lesquels  $\xi_i = 0$  et  $y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \neq 1$  ont un  $\alpha_i$  nul, afin de respecter l'égalité  $\alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i) = 0$ .

Les points directement sur la marge sont appelés vecteurs de support libres (free support vectors), ou encore vecteurs de support non-bornés (unbounded support vectors). Les points pour lesquels  $\alpha_i = C$  sont quant à eux appelés vecteurs de support bornés (bounded support vectors).

Ici encore, les vecteurs de support sont les seuls points qui sont vraiment importants pour déterminer l'hyperplan optimal, puisque ce sont les seuls points pour lesquels  $\alpha_i \neq 0$ .

Enfin, remarquons que bien que la marge souple ait été présentée pour le cas linéaire, il est possible de l'utiliser aussi dans le cas non linéaire exactement de la même manière, puisque la fonction objectif du dual est parfaitement identique à celle de la marge non souple. Il suffit donc encore de remplacer tous les produits scalaires par une fonction noyau.

# Chapitre 5

## Les technologies utilisées et réalisation

### 5.1 Introduction

L'étape de la réalisation est une étape cruciale dans le cycle de vie de nos applications. Cette étape permet de concrétiser notre projet par le développement des interfaces et par des réalisations concrètes des fonctionnalités du système. Pour réaliser ces applications nous avons recouru à plusieurs outils de développement.

### 5.2 Environnement de travail

#### 5.2.1 Environnement matériel

<b>Marque</b>	Hp
<b>Processeur</b>	11th Gen Intel(R) Core(TM) i7-1165G7
<b>RAM</b>	8GO
<b>Disque dur</b>	256GO SSD
<b>Système d'exploitation</b>	Windows 10

#### 5.2.2 Environnement logiciel : Système de gestion de base de données

**MySQL** est un système de gestion de base de données (SGBD). Selon le type d'application, sa licence est libre ou propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle et Microsoft SQL Server.

**MySQL** est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées.



## 5.3 Choix technique

### 5.3.1 Les langages de programmation

**Python** est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprété, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes.



**Flask** est un micro framework open-source de développement web en Python. Il est classé comme microframework car il est très léger. Flask a pour objectif de garder un noyau simple mais extensible. Il n'intègre pas de système d'authentification, pas de couche d'abstraction de base de données, ni d'outil de validation de formulaires. Cependant, de nombreuses extensions permettent d'ajouter facilement des fonctionnalités.



**JavaScript** est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs web. Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript pour l'interpréter.



**JavaScript** est un langage orienté objet à prototype : les bases du langage et ses principales interfaces sont fournies par des objets. Cependant, à la différence d'un langage orienté objets, les objets de base ne sont pas des instances de classes. En outre, les fonctions sont des objets de première classe. Le langage supporte le paradigme objet, impératif et fonctionnel.

**Bootstrap** est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub. Avant l'arrivée de Bootstrap, plusieurs bibliothèques existaient, ce qui menait à des incohérences et à un coût de maintenance élevé.



**Bootstrap**

### 5.3.2 Les langages de description

**HTML** : désigne un type de langage informatique descriptif. Il s'agit plus précisément d'un format de données utilisé dans l'univers d'Internet pour la mise en forme des pages Web. Il permet, entre autres, d'écrire de l'hypertexte, mais aussi d'introduire des ressources multimédias dans un contenu.



**CSS** : est un acronyme pour Cascading Style Sheets ou feuille de style en cascade en français., c'est le langage de balisage basé sur le Web utilisé pour décrire l'apparence et la mise en forme d'un site Web dans le navigateur, le plus couramment utilisé en HTML



## 5.4 Présentation des interfaces

Nous exposerons quelques interfaces de notre application, en essayant à chaque fois de décrire les différents objets interactifs mis à la disposition de l'utilisateur.

### 5.4.1 Interface d'inscription

Seuls les administrateurs peuvent accéder à cette page et peuvent ajouter des comptes d'utilisateur ou d'administrateur.

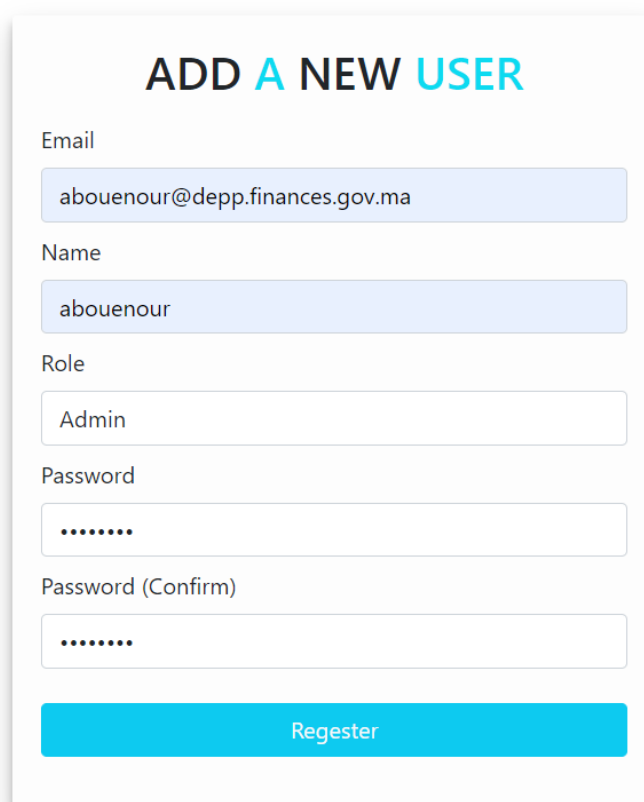
A screenshot of a web registration form titled "ADD A NEW USER" in bold black text. The form is white with a light blue border and a shadow. It contains several input fields: "Email" with the value "abouenour@depp.finances.gov.ma", "Name" with the value "abouenour", "Role" with the value "Admin", "Password" with masked characters ".....", and "Password (Confirm)" with masked characters ".....". A blue "Register" button is at the bottom.

FIGURE 5.1 – Interface d'inscription



### 5.4.2 Interface d'authentification

Après une inscription réussie l'utilisateur pour accéder à la page principale doit s'authentifier cette page d'authentification 'Login' lui demande d'entrer son mail et son mot de pass qu'il a utilisé pendant l'inscription une fois validé il sera automatiquement redirigé vers la page principale.

The login interface is displayed within a white rectangular box. At the top, there are three logos: the Moroccan Kingdom logo on the left, the national coat of arms in the center, and the DEPP logo on the right. Below the logos, the text 'Email' is followed by a text input field containing 'abouenour@depp.finances.gov.ma'. Below the email field, the text 'Password' is followed by a password input field with masked characters. A checkbox labeled 'Remember Me' is checked. At the bottom, there is a large blue button labeled 'Login'.

FIGURE 5.2 – Interface d'authentification

### 5.4.3 Page principale

la **page principale** donne la possibilité à l'utilisateur de télécharger un fichier (.pdf, .doc, .docx). La navbar est composé d'un bouton Logout qui lui permet de se déconnecter, un bouton **Add new user** donne la possibilité à l'admin d'ajouter un nouvel utilisateur, un bouton **Home** lui permet de revenir de même à la page d'accueil.



---

FIGURE 5.3 – Page principale

### 5.4.4 Page de résultat

Cette page affiche pour chaque mot technique sa définition et donne la possibilité de cliquer sur le lien **Voir détail** qui envoie vers la page source wekipidai si jamais l'utilisateur a besoin plus d'explication.

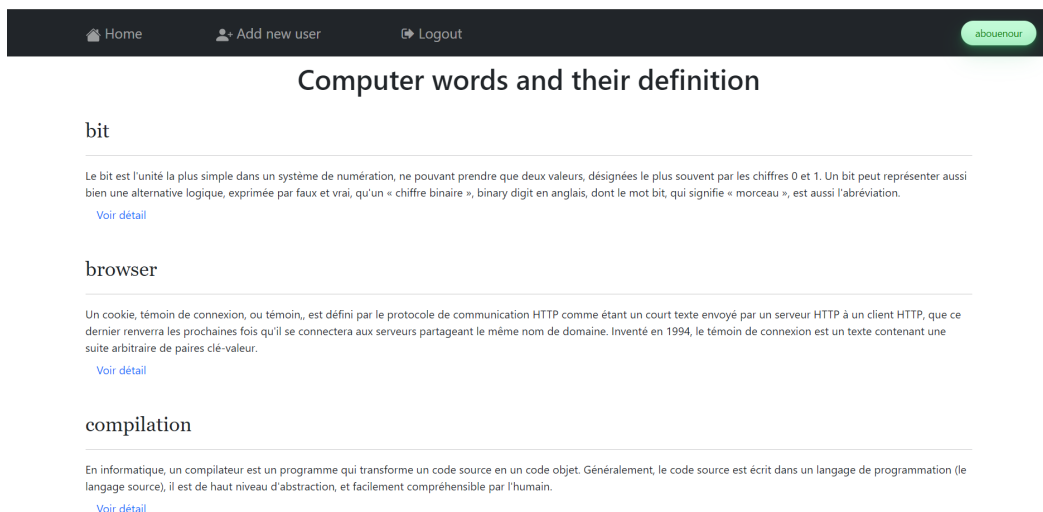


FIGURE 5.4 – Page de résultat

## 5.5 Conclusion

Dans ce chapitre nous avons présenté en détails le développement de notre système, Nous avons commencé par présenter l'environnement matériel et logiciel, les choix techniques et les outils de travail sur lesquels se base notre application, et enfin nous avons conclu par les scénarios de test et de validation de l'application.

# Conclusion Générale

Ce rapport est le fruit d'un travail étalé sur une durée de 30 jours.

D'un côté, j'ai pris connaissance de la plupart des départements de la DEPP, pour essayer de comprendre les problèmes vécus par les employés. Après énumération de plusieurs de ces problèmes, on a décidé de partir sur la détection et l'explication des mots techniques. Puis, mon encadrant et moi avons effectué une étude de faisabilité de l'application web à développer, ensuite et avec la collaboration des employés de la DEPP j'ai pu construire une base de données de plus de 10000 exemples de mots techniques et non techniques qui a servi à entraîner mon modèle de classification binaire. Étant novice dans le domaine d'apprentissage automatique je suis allé me documenter sur les modèles disponibles et adaptés à ma problématique. Le Vecteur support machines s'est avéré le meilleur vu la taille de ma base de données. Après test et validation du modèle, j'ai commencé à développer le front-end et le back-end de l'application qui depuis son élaboration est devenue un outil de compréhension indispensable pour les employés de la DEPP.

Ce stage a fait l'objet d'une expérience à la fois intéressante et enrichissante, et nous a permis d'améliorer nos connaissances et nos compétences dans le domaine des sciences des données, du développement web, de la gestion de projets. Certes, des difficultés se sont présentées à nous, à savoir trouver la base de données regroupant le plus d'exemples possible pour garantir un bon apprentissage et une meilleure généralisation ; cependant, nous avons pu, avec des recherches dans plusieurs documentations et articles et de la demande d'information auprès de mon encadrant, trouver le meilleur parti.

# Références

- Dominik Francoeur : Machines à vecteurs de support : une introduction, CaMUS (Cahiers Mathématiques de l'Université de Sherbrooke), 2010, 19 p.