

Université de la Manouba

École Supérieure d'Économie Numérique



**Rapport
de projet de fin d'études**

Présenté en vue de l'obtention du diplôme de
Licence Appliquée en Commerce Electronique

Sujet

**Détection automatique de l'offre de formation
pour les agents de la société 1WayCom et
suivi de la production**

Élaboré par:

**Ben Saber Rahma
Merchaoui Mohamed Aymen**

Organisme d'accueil
1 Way Dev



ESEN
Société

Encadré par
**Mme Thabet Ines
M. Sriti Said**

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrant professionnel, **Monsieur Said SRITI**

Signature et cachet

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrant académique, **Madame Ines THABET**

Signature

Dédicaces

Nous tenons tout d'abord à exprimer nos vifs remerciements à Monsieur **Ramzi CHERIF** le directeur de **1WayDev** qui nous a ouvert les portes de son entreprise afin d'y faire notre stage. Nous tenons à lui manifester toute notre gratitude pour nous y avoir accueilli.

Nos remerciements vont aussi à Monsieur **Mr Said Sriti** Project Manager et développeur à **1WayDev** qui est notre encadrant professionnel pour ses conseils, son soutien, ses savoirs scientifiques et ses encouragements qui nous ont toujours réconfortés.

Je remercie également Madame **Ines Thabet**, notre chère encadrante académique pour ses multiples conseils et pour toutes les heures qu'elle nous'a consacrée à diriger ce projet. Je souhaite également lui dire à quel point nous avons apprécié sa grande disponibilité et lui exprimer notre respect sans faille et notre reconnaissance pour avoir été présente dans les moments les plus difficiles.

.
Nous présentons nos vifs remerciements à Mr **Mohamed Anis Bach Tobji**, le directeur de notre établissement **ESEN** Mannouba, qui nous a toujours soutenu et encouragé pour aller de l'avant.

Nous voulons aussi témoigner notre gratitude envers tous les enseignants de **l'ESEN** et toutes les personnes qui ont contribué à notre formation. Merci à tous ceux qui ont participé d'une manière ou d'une autre à nous fournir l'assistance nécessaire pour la réalisation de ce travail.

Remerciements

À ma mère,

Qui a oeuvré pour ma réussite, de par son amour, son soutien, et tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression mon profond amour éternel.

À mon père,

L'épaule solide, l'oeil attentif compréhensif et la personne la plus digne de mon estime et de mon respect. Aucune dédicace ne saurait exprimer mes sentiments, et te remercier pour tes sacrifices et pour l'affection dont tu m'as toujours entourée

À la meilleure soeur Ammoula,

Ammoula, cet inestimable cadeau de Dieu qui ne cesse d'être là dans mes moments de détresse et de joie.

À ma grande mère Que Dieu la préserve la santé et la longue vie. À L'ame pure de mima Qui nous garde au dela du ciel, paix à son ame.

À mes Amis, qui m'ont toujours supporté et réconforté
Aux personnes qui m'ont toujours aidé et encouragé, qu'étaient toujours à mes côtés, et qui m'accompagnaient durant mon chemin d'études supérieures

Aymen MERCHAQUI

À mon père Mourad,

en signe d'amour, de reconnaissance et de gratitude pour le soutien et les sacrifices dont il a fait preuve à mon égard.

À ma mère Saloua,

ma raison d'être, ma raison de vivre, la lanterne qui éclaire mon chemin et m'illumine de douceur et d'amour.

À toute ma famille, et mes amis, aucun hommage ne pourrait être à la hauteur de l'amour et du soutien qu'ils ne cessent de témoigner. Je dédie ce modeste travail.

Aux personnes qui m'ont toujours aidée et encouragée, qui étaient toujours à mes côtés, et qui m'accompagnaient durant mon chemin d'études supérieures.

Rahma BEN SABER

Table des matières

Introduction générale	1
1 Présentation du cadre du projet	3
Introduction	4
1.1 Présentation de l'organisme d'accueil	4
1.1.1 Services de 1WayDev	4
1.1.2 Relation avec 1WayCOM	5
1.2 Cadre général du projet	5
1.3 Etude de l'existant	5
1.4 Critique de l'existant	5
1.5 Problématique	6
1.6 Objectifs du projet	6
1.7 Solution proposée	6
1.7.1 Détection des offres de formation	6
1.7.2 Prédition du nombre de ventes	7
1.7.3 Visualisation des données	7
1.7.4 Implémentation	7
1.8 Choix méthodologique	7
1.8.1 La méthodologie CRISP	7
1.8.2 Les étapes de la méthodologie CRISP	8
Conclusion	9
2 Release 1 "Détection des offres de fromation"	10
Introduction	11
SPRINT1 : Compréhension du prbolème metier et des données	11
2.1 Compréhension du problème métier :	11
2.1.1 Détermination des objectifs de l'analyse	13
2.1.2 Identification des variables d'activité clés	13
2.1.3 Offre de formation	13

2.1.4	Les mesures de réussite	13
2.1.5	Identification des sources de données	14
2.2	Compréhension des données	17
2.2.1	Données quantitatives continues	17
2.2.2	Données quantitatives discrètes	17
2.2.3	Données qualitatives nominales	18
2.2.4	Données qualitatives ordinaires	18
2.2.5	Données temporelles	19
2.2.6	Représentation des données	19
2.2.7	Exploration des données	20
2.2.8	Synthèse	21
SPRINT2 : Préparation et visualisation des données	23	
2.3	Agrégation des données	23
2.3.1	Les modifications apportées sur les données	24
2.3.2	Scripts en Python	25
2.3.3	Table résultante	27
2.4	Nettoyage des données	27
2.4.1	Le nettoyage des valeurs nulles	28
2.4.2	Le nettoyage des lignes dupliquées	29
2.4.3	Le nettoyage du format des colonnes	29
2.4.4	Fichier CSV résultant	30
2.5	La visualisation des données	30
2.5.1	Visualisation du besoin en formation	30
2.5.2	Visualisation de la distribution des écoutes au cours de l'année	31
2.5.3	Visualisation de la distribution des formations sur les mois de l'année	31
2.5.4	Visualisation des corrélations entre les variables	32
SPRINT 3 : Modélisation et évaluation	35	
2.6	La modélisation	35
2.6.1	Choix de la technique de prédiction adéquate	36
2.6.2	Une base de donnée déséquilibrée	37
2.6.3	Le Tuning des hyperparamètres	37

2.6.4	La première méthode de classification multi label : La transformation du problème	38
2.6.5	La deuxième méthode de classification multi label : Les algorithmes adaptés au problème	40
2.6.6	La troisième méthode de classification multi label : Les approches ensemblistes	41
2.6.7	Les modèles d'apprentissage automatique utilisés	41
2.6.8	Synthèse	47
2.7	L'évaluation	52
2.7.1	Accuracy de classification	53
2.7.2	Précision	54
2.7.3	Rappel	54
2.7.4	F1_score	55
2.7.5	Zero one loss	55
2.7.6	Matrice de confusion	55
2.7.7	Évaluation des algorithmes utilisés	56
2.7.8	Synthèse	58
	Conclusion	62
3	Release 2 "Prédiction du nombre de ventes"	63
	Introduction	64
	SPRINT1 : Compréhension, préparation et visualisation des données	64
3.1	Compréhension du problème métier	64
3.1.1	L'environnement d'intervention	64
3.1.2	Détermination des objectifs de l'analyse	65
3.1.3	Identification des variables d'activité clés	65
3.1.4	Le nombre de ventes prédit	65
3.1.5	Les mesures de réussite	65
3.1.6	Identification des sources de données	66
3.2	Compréhension des données	69
3.2.1	Données quantitatives continues	69
3.2.2	Les données quantitatives discrètes	69
3.2.3	Les données qualitatives nominales	70

3.2.4	Les données qualitatives ordinaires	70
3.2.5	Les données qualitatives temporelles	70
3.2.6	Représentation des données	70
3.2.7	Exploration des données	71
3.2.8	Synthèse	71
3.3	Agrégation des données	73
3.3.1	Les modifications apportées sur les données :	73
3.3.2	Table résultante	77
3.4	Nettoyage des données	77
3.4.1	Nettoyage des lignes dupliquées	78
3.4.2	Nettoyage des valeurs nulles	79
3.4.3	Nettoyage du format des colonnes	80
3.4.4	Fichier CSV résultant	80
3.4.5	La visualisation des données	80
3.4.6	Visualisation de la fréquence des nombres de ventes	80
3.4.7	Visualisation de la distribution des scores et des notes :	82
3.4.8	Visualisation de la corrélation entre les variables	83
3.4.9	La matrice de corrélation de la Data Set	84
	SPRINT2 :: Modélisation et évaluation	85
3.5	La modélisation	85
3.5.1	Choix de la technique de prédiction adéquate	85
3.5.2	Le Tuning des hyperparamètres	87
3.5.3	La régression	87
3.5.4	SVR (Support Vector Regression)	87
3.5.5	Explication du choix des modèles utilisés	88
3.5.6	Synthèse	89
3.6	L'évaluation	90
3.6.1	Mean Squared error	90
3.6.2	Root mean squared error	91
3.6.3	Mean absolute error	91
3.6.4	Evaluation des algorithmes utilisés	92

3.6.5	Visualisation des résultats de prédictions pour l'algorithme retenu XGBoost	92
SPRINT3 : Déploiement		96
3.7	Analyse des besoins	96
3.7.1	Identification des acteurs de l'application	96
3.7.2	Identification des besoins fonctionnels et non fonctionnels	97
3.7.3	Diagramme des cas d'utilisation	98
3.7.4	Cas d'utilisation « Déetecter les offres de formations »	98
3.7.5	Cas d'utilisation « Prédire le nombre de ventes par agent »	100
3.7.6	Cas d'utilisation « Consulter le suivi des formations »	101
3.7.7	Cas d'utilisation « Consulter le suivi de la production »	102
3.7.8	Cas d'utilisation « Consulter le planning des formations »	103
3.8	Conception	104
3.8.1	Diagramme de classes participantes	104
3.8.2	Diagramme de séquence détaillé	105
3.8.3	Diagramme de classes	106
3.9	Déploiement	106
3.9.1	Base de données	106
3.9.2	Les interfaces	107
3.10	Phase de clôture	110
3.10.1	Architecture physique adoptée et intégration du Machine Learning	110
3.10.2	Outils utilisés dans le déploiement	111
3.10.3	Outils utilisés dans le machine learning	113
Conclusion		113
Conclusion générale		114
Bibliographie		115
Annexes		117

Table des figures

1.1 Logo Entreprise	4
2.1 Compréhension des données dans CRISP-DM	11
2.2 La base de données de 1WayCom	14
2.3 Tables sélectionnées pour le projet	15
2.4 La table absences	15
2.5 La table grid listen3 categories	16
2.6 La table grid listen3 sub categories	16
2.7 La table users	17
2.8 Matrice représentative des variables prédictives	19
2.9 Matrice représentative de la variable cible	20
2.10 Préparation des données dans CRISP-DM	23
2.11 La nature de nos données	24
2.12 La nature de nos données	25
2.13 - Ajout de l'id de l'agent	25
2.14 Ajout de la date des appels	25
2.15 code python1	26
2.16 code python2	26
2.17 La table formations résultante	27
2.18 La Data Set avant le nettoyage des données	28
2.19 La Data Set avant le nettoyage des données	28
2.20 La Data Set après le nettoyage des données	29
2.21 La transformation des colonnes dupliquées	29
2.22 Le Fichier CSV de la Data Set résultante	30
2.23 Diagramme circulaire du besoin en chaque formation	31
2.24 Distribution des écoutes sur l'année	32
2.25 Densité des formations sur les mois de l'année	33
2.26 Matrice de corrélation de notre Data Set	34
2.27 La modélisation dans CRISP-DM	35

Table des figures

2.28 Fonctionnement de la Binary Relevance	39
2.29 Fonctionnement des Classifier chains	39
2.30 Fonctionnement du Label Powerset	40
2.31 Sigmoïde de régression logistique	41
2.32 La fontion sigmoïde	42
2.33 Fonctionnement Arbres de décision	43
2.34 SVM dans un espace 2D	44
2.35 Fonctionnement de KNN	44
2.36 Fonctionnement des Random Forests	45
2.37 Calcul de probabilité avec Naïve Bayes	46
2.38 Fonctionnement du Naïve Bayes	47
2.39 L'évaluation dans la méthodologie CRISP-DMs	53
2.40 Calcul de la mesure de performance "Accuracy"	53
2.41 Histogramme de fréquence du Label formation1	54
2.42 Calcul de la mesure de performance "Précision"	54
2.43 Calcul de la mesure de performance "Rappel"	55
2.44 Calcul de la mesure de performance F1-score	55
2.45 la fonction Zero one loss	55
2.46 La matrice de confusion	56
2.47 Matrice de confusion du premier label formation1	59
2.48 Matrice de confusion du deuxième label formation2	60
2.49 Matrice de confusion du deuxième label formation3	60
2.50 Matrice de confusion du deuxième label formation4	61
 3.1 La base de données de 1WayCom	66
3.2 Tables sélectionnées pour le projet	66
3.3 La table grid_listen3_stored_calls	67
3.4 La tables sales_values	67
3.5 La table users	68
3.6 La table holiday	68
3.7 La table absences	69
3.8 Matrice représentative des variables prédictives	70

Table des figures

3.9 Matrice représentative de la variable cible	71
3.10 la table grid_listen3_stored_calls	74
3.11 Code de suppression de toutes les colonnes inutiles	74
3.12 Code d'ajout et mise à jour de la colonne niveau	74
3.13 la table grid_listen3_stored_calls	75
3.14 Code d'ajout de la colonne agent_lastname	75
3.15 Code d'ajout et mise à jour de la colonne nbr_vente	75
3.16 Code d'ajout et mise à jour de la colonne nbr_ventes_prec	76
3.17 Code d'ajout et mise à jour de la colonne nbr_heures	76
3.18 La table écoutes résultante	77
3.19 Notre Data Set avant le nettoyage des données	78
3.20 Notre Data Set après le nettoyage des lignes dupliquées	78
3.21 Notre Data Set après le nettoyage des valeurs nulles	79
3.22 Fichier CSV de notre Data Set	80
3.23 Graphe de fréquences des ventes sur deux mois consécutifs	81
3.24 Visualisation de la distribution des scores et des notes	82
3.25 Graphe pour montrer la corrélation entre le nombre de ventes et le nombre d'heures travaillées	83
3.26 Graphe de corrélation entre le niveau de l'agent et sa productivité	84
3.27 Graphe de corrélation entre le score d'un agent et sa productivité	84
3.28 Matrice de corrélation de la Data Set	85
3.29 Fonctionnement de SVR	88
3.30 Calcul de la mesure de performance MSE	91
3.31 Calcul de la mesure de performance RMSE	91
3.32 Calcul de la mesure de performance MAE	91
3.33 Graphe de comparaison entre les RMSE des algorithmes	93
3.34 Le nombre de ventes VS le nombre de ventes prédit	93
3.35 La densité des résidus de XGBoost	94
3.36 Diagramme des cas d'utilisation	98
3.37 Diagramme de séquence du cas d'utilisation "Déetecter les offres de formations"	99

3.38 Diagramme de séquence système du cas d'utilisation "Prédire le nombre de ventes par agent"	101
3.39 Diagramme de séquence système du cas d'utilisation "Consulter le suivi des formations"	102
3.40 Diagramme de séquence du cas d'utilisation "Consulter le suivi de la production"	103
3.41 Diagramme de séquence du cas d'utilisation "Consulter le planning des formations"	104
3.42 Diagramme de classes participante predire le nombre de ventes	104
3.43 Diagramme des classes participantes du cas d'utilisation détecter la formation	105
3.44 diagramme de séquence détaillé	105
3.45 Diagramme de classes	106
3.46 Interface suivi production	107
3.47 Interface suivi formations	107
3.48 Interface de détection des offres de formation du mois	108
3.49 Aucun nouveau besoin en formation dans le mois de détection	108
3.50 Interface de prédiction du nombre de ventes	109
3.51 Affichage du nombre de ventes prédit	109
3.52 Architecture adoptée	110
3.53 Logo de MySQL	111
3.54 Logo de Python :	111
3.55 Logo de Flask	112
3.56 Logo de VueJS	112
3.57 Logo d' Atom	112
3.58 Logo d'Anaconda	113
3.59 Logo de Spyder	113

Liste des tableaux

2.1	La nature de nos données.	21
2.2	Les statistiques basiques des variables.	22
2.3	les algorithmes de transformation de problème avec leurs hyperparamètres.	48
2.4	La méthode des algorithmes adaptés avec leurs hyperparamètres ajustés.	51
2.5	Les méthodes ensemblistes avec leurs hyperparamètres.	52
2.6	Évaluation des algorithmes de la méthode de transformation du problème.	57
2.7	Évaluation des algorithmes adaptés.	58
2.8	Évaluation des méthodes ensemblistes.	58
2.9	Synthèse de l'évaluation.	58
3.1	Matrice représentative de la variable cible.	72
3.2	Mesures statistiques de base.	73
3.3	Les modèles utilisés avec leurs paramètres ajustés.	89
3.4	Etude comparative des algorithmes.	92
3.5	Description textuelle du cas d'utilisation "Déetecter les offres de formations.	98
3.6	Description textuelle du cas d'utilisation "Prédire le nombre de ventes par agent.	100
3.7	Description textuelle du cas d'utilisation 'Consulter le suivi des formationst.	101
3.8	Description textuelle du cas d'utilisation "Consulter le suivi de la production.	102
3.9	Description textuelle du planning des formations.	103

Liste des abréviations

- **CRISP** = Cross Industry Standard Process
- **CRM** = Customer Relationship Management
- **DM** = Data Mining
- **ERP** = Enterprise Resource Planning
- **KNN** = K Nearest Neighbors
- **SVR** = Support Vector Machines

Introduction générale

Pouvant être pour certains une révolution aussi radicale que celle de l'internet, l'Intelligence Artificielle, apparue dès le 18ème siècle, revient en force sur le devant de la scène, revisitée par le Big Data et le Machine Learning. Absolument passionnante et exaltante, c'est un ensemble de techniques et de méthodes qui permettent de simuler le raisonnement humain, le cerveau biologique et la manière de penser. Elle a pris ses origines il y a bien longtemps, et passe par de grandes étapes : la machine de Turing, les systèmes experts, la robotique, etc. Aujourd'hui, les systèmes cognitifs prennent le relais. Ces systèmes engagent une conversation en tant qu'humains, ont le pouvoir de raisonner, et surtout d'apprendre. Ils apprennent pour aider au processus de prise de décision et nous les trouvons dans toutes les industries : la médecine, le commerce, la finance, etc. Tous les secteurs sont absolument touchés de nos jours par l'arrivée de l'intelligence artificielle dans leur domaine. Par-dessus tout, nous commençons aujourd'hui à parler d'entreprises autoapprenantes. Bien évidemment, pour qu'une entreprise s'adapte convenablement à cette technologie émergente, il faudra des cas d'usage adéquats et des solutions adaptées qui combinent plusieurs techniques. Les entreprises ne vont pas tout de même remplacer tous les systèmes d'information par des systèmes d'intelligence artificielle ; mais il y a un certain nombre de services et besoins éligibles. Quand les professionnels manipulent un corpus complexe, ou qu'ils ont besoin de visualiser une situation, s'ils le font de manière traditionnelle en programmant, ils auront des difficultés. En utilisant des principes de l'intelligence artificielle comme l'apprentissage automatique, ces professionnels vont pouvoir changer l'expérience utilisateur, en construisant des machines capables de simuler le raisonnement de l'humain, et surtout capables d'apprendre. Cela constitue un changement radical pour certaines entreprises, qui veulent à tout prix adopter les derniers procédés. En ce temps-là, et pour rallier ce changement technologique, les dirigeants de l'entreprise doivent prendre des mesures concrètes, en révisant la structure organisationnelle, et en s'assurant qu'elles puissent faire face à la montée en importance de l'intelligence artificielle. Les dirigeants devraient aussi penser à améliorer le processus de développement des stratégies. Par exemple, parmi les attributs que l'intelligence artificielle a apportés dans le monde de l'entreprise, nous citons l'amélioration de la gestion des ressources humaines, à partir de l'individualisation des formations selon des critères prédéfinis. D'autant plus, elle permet d'avoir de nouvelles perspectives et prospectives, où les algorithmes peuvent déceler des scénarios et des patterns, la plupart du temps imperceptibles par l'humain, pour visualiser des éventualités importantes dans la gestion de

Introduction générale

l'entreprise. C'est dans ce contexte que vient s'insérer notre projet de fin d'études, où nous allons construire un modèle d'apprentissage automatique qui aidera les superviseurs de notre organisme d'accueil à détecter les formations adéquates à leurs agents téléopérateurs à partir des notes qu'ils obtiennent dans leurs feuilles d'écoutes. Pour un autre motif, notamment le suivi de la production, nous allons concevoir un deuxième modèle d'apprentissage automatique permettant de prédire le nombre de vente éventuel de chaque agent de l'entreprise. Pour ce faire, nous avons organisé ce rapport comme suit :

- Le premier chapitre constitue une étude préalable qui présente l'organisme d'accueil, le cadre général du projet, l'étude de l'existant, la problématique, les objectifs du projet, la solution proposée et la méthodologie de conduite de projet adoptée.
- Le deuxième chapitre présente la construction du premier livrable, qui est le système de prédiction et détection des offres de formations adéquates à chaque agent, décomposé en trois sprints : Compréhension du problème métier et des données, ensuite Préparation et visualisation des données et enfin Modélisation et évaluation.
- Le troisième chapitre présente le deuxième livrable de notre projet, qui est le système de prédiction du nombre de ventes réalisées par un agent dans un mois déterminé, décomposé en trois sprints : Compréhension et préparation des données, ensuite Modélisation et évaluation et enfin Conception et déploiement.

Finalement, ce rapport est clôturé par une conclusion générale qui synthétise le déroulement de notre projet et qui ouvre la voie à de nouvelles perspectives de progrès et d'amélioration.

PRÉSENTATION DU CADRE DU PROJET

Plan

Introduction	4
1 Présentation de l'organisme d'accueil	4
2 Cadre général du projet	5
3 Etude de l'existant	5
4 Critique de l'existant	5
5 Problématique	6
6 Objectifs du projet	6
7 Solution proposée	6
8 Choix méthodologique	7
Conclusion	9

Introduction

Dans tout projet, une étude préalable s'impose, pour aboutir à une compréhension des objectifs, des contraintes et des modalités du projet. Dans ce chapitre, nous allons tout d'abord présenter notre organisme d'accueil, pour ensuite nous concentrer sur la décortication de notre sujet, et finalement choisir la méthodologie de travail adéquate à notre type de projet.

1.1 Présentation de l'organisme d'accueil

1WayDev est une entreprise de développement informatique leader dans la création des applications web et mobiles, opérant en Tunisie et à l'international. Elle propose son activité de service en Ingénierie Informatique en mettant à disposition ses experts pour l'externalisation des solutions informatiques et le développement d'applications spécifiques.



Figure 1.1: Logo Entreprise

1.1.1 Services de 1WayDev

L'entreprise 1WayDev fournit des services qui améliorent la croissance de ses clients tout en prenant en charge leurs besoins en produits et services informatiques pour les aider à atteindre une production optimale. Les services de 1WayDEV sont principalement :

- La création des CRM
- La création des ERP
- La proposition de solutions e-commerce
- Le développement de sites web
- Le développement d'applications mobiles

1.1.2 Relation avec 1WayCOM

La plupart des solutions proposées par 1WayDEV sont orientées vers la création de solutions informatiques pour l'entreprise 1WayCOM, qui est un centre d'appel spécialisé dans les opérations de ventes. En l'occurrence, notre projet s'insère dans l'amélioration du suivi des formations et de la production de 1WayCOM.

1.2 Cadre général du projet

Le projet est élaboré pour le centre d'appel 1WayCom, un organisme qui prend en charge l'externalisation du service client, l'optimisation de la production et l'augmentation de la compétitivité de ses clients. Le centre d'appel est une unité qui repose énormément sur l'apport de ses téléopérateurs et sur leur aptitude à faire des ventes de produits ou bien de prendre des rendez-vous pour les clients. En conséquence, l'élément ressources humaines est prépondérant. Notre projet vient s'installer au cœur de l'activité de 1WayCom puisqu'il s'intéresse à un aspect important de la gestion des ressources humaines : la formation. Bien qu'elle soit très importante, la formation des employés dans une entreprise n'est pas la seule préoccupation de ses dirigeants. Pour 1WayCom, chaque vente réalisée par un agent représente un gain financier qui doit être non seulement amélioré, mais également anticipé. Cela a créé le besoin de prédire en amont le nombre de ventes que chaque agent devrait réaliser, en considérant sa performance habituelle, ainsi que d'autres facteurs.

1.3 Etude de l'existant

Pour évaluer un agent dans un centre d'appel, son superviseur doit effectuer des écoutes régulières sur les téléchargements audio de ses appels avec les clients. Préalablement à ce projet, les agents ne faisaient pas de formations spécifiques à partir des notes qu'ils ont obtenues dans la grille d'écoute. Par conséquent, il n'y a pas de traces pour pouvoir suivre la progression de ces agents-là. Par ailleurs, les superviseurs n'avaient aucune vision sur la progression de leur groupe et sur les attentes qu'ils devraient avoir à l'égard de chaque agent en termes de nombre de ventes éventuelles.

1.4 Critique de l'existant

L'étude de l'existant nous a permis de faire du recul quant à la méthode de travail de 1wayCom. Leur procédure actuelle manque d'efficacité puisque les lacunes de leurs agents ne sont pas

traitées séparément. De cette façon, les agents cumulent les mêmes erreurs, ce qui impacte finalement leur productivité. Par ailleurs, 1WayCom ne dispose pas d'un système de suivi pour ses agents afin de pouvoir suivre leur progression et avoir une idée sur la productivité attendue de chacun d'entre eux.

1.5 Problématique

Quelles techniques utiliser pour construire un système capable de détecter automatiquement les offres de formations à suivre pour chaque agent de l'entreprise 1WayCom, et qui soit susceptible de prédire le nombre de ventes des agents pour chaque mois de l'année ?

1.6 Objectifs du projet

- Déetecter automatiquement les offres de formations adéquates pour chaque agent ;
- Prédire le nombre de ventes pour chaque agent ;
- Consulter le suivi de la productivité collective des agents ;
- Consulter le suivi des formations ;

1.7 Solution proposée

Notre solution est de construire deux modèles de Machine Learning, en apprentissage automatique supervisé, le premier pour détecter les offres de formation, et le deuxième pour prédire le nombre de ventes éventuelles pour chaque agent. Cela implique que nous devons utiliser deux techniques de prédiction différentes.

1.7.1 Détection des offres de formation

Les formations dans 1WayCom sont divisées en 4 catégories :

- une formation produit ;
- une formation métier ;
- une formation en traitement des objections ;
- une formation en souscriptionr.

Nous allons donc créer un module d'apprentissage automatique qui détectera si un agent a besoin d'une de ces formations à partir des notes qu'il a eues et du nombre de ventes qu'il a

effectuées. Suite à une réunion officielle avec le comité accompagné et après une étude budgétaire et une classification des priorités, nous avons opté pour le premier projet.

1.7.2 Prédiction du nombre de ventes

Nous allons créer un module en apprentissage automatique qui permettra de faire la prédiction du nombre de ventes qui vont être réalisées dans le mois suivant à partir de certains critères dont le score de qualification de l'agent, le nombre d'heures de travail, le niveau d'expérience de l'agent

1.7.3 Visualisation des données

Nous allons créer des tableaux de bord qui permettront de faire le suivi des formations, ainsi que le suivi de la productivité des agents.

1.7.4 Implémentation

L'implémentation permet aux utilisateurs de profiter des modules créés par le Data Scientist. Dans notre cas, nous avons choisi de créer des interfaces dans l'ERP déjà existant de l'entreprise pour permettre aux superviseurs d'avoir une idée plus précise sur la productivité de leurs agents et sur leur classement eux-même. Ces interfaces donnent une vision claire et globale qui aide dans la prise de décision.

1.8 Choix méthodologique

Pour réussir un projets, et pour qu'il soit rendu à temps et organisé, nous devons bien choisir une méthodologie de travail selon nos attentes, nos moyens et nos ressources. Nous présentons en ce qui suit les methodologies utilisés dès le debut de notre projet.

1.8.1 La méthodologie CRISP

Avant l'élaboration du CRISP, les acteurs majeurs du Data Mining travaillaient chacun avec leurs propres méthodes dans la conduite de leurs projets. Pour des raisons de standardisation, de rapidité et d'efficacité, ces acteurs majeurs ont collaboré ensemble pour établir la méthodologie CRISP-DM, acronyme pour Cross Industry Standard Process pour le Data Mining. Cette méthodologie est donc universelle, applicable dans tous les domaines et dans tous les logiciels de Data Mining. Elle permet de réaliser des projets rapides, efficaces et rentables

1.8.2 Les étapes de la méthodologie CRISP

- **Compréhension de la problématique :** La première étape est une étape très importante. C'est dans cette étape où l'on devrait se poser toutes les questions qui définissent les objectifs du projet et fixer la problématique que la Data Science viendra résoudre.
- **Compréhension des données :** Cette étape consiste à recenser les données existantes et à les explorer à travers des graphes, des tables et des statistiques pour comprendre ce que les données veulent nous dire et dans quelle mesure elles seraient utiles dans la résolution de la problématique.
- **Préparation des données :** Cette étape est la plus longue dans un projet de Data Science. En effet, pour la plupart des cas, elle représente 80 % du temps accordé à tout le projet. La préparation des données englobe des opérations qui visent à réduire le bruit dans les données, et qui aideront à exploiter des données valides pour avoir un projet réussi. C'est une étape très importante puisqu'elle influence de manière directe la qualité du modèle. Plus les données sont solides, valables et bien structurées, plus le modèle établi dans la phase suivante sera correct et performant. Le data scientist prend donc son temps pour faire plusieurs manipulations. Il s'agit de la transformation des données, du nettoyage des données, du traitement des valeurs manquantes, des valeurs nulles, des valeurs éparses, des valeurs dupliquées, etc.
- **Modélisation :** La modélisation est la phase de Data Science pure. En effet, c'est dans cette étape qu'on choisit les algorithmes à implémenter et qu'on essaye d'améliorer leur paramétrage. Les données sont divisées en deux parties, la première pour faire l'apprentissage et la deuxième pour tester.
- **Evaluation :** Les modèles réalisés dans la phase précédente passent par une étape de validation et de test où l'on retiendra le meilleur en termes de précision, de taux d'erreurs, et d'autres indicateurs qu'on détaillera dans les chapitres à suivre.
- **Déploiement :** Arrivés à cette étape, nous devrions avoir un modèle performant et répondant correctement à notre problématique. Dans cette étape, le data scientiste essaiera de générer un rapport contenant l'interprétation et la visualisation des données et des analyses qu'il a faites d'une manière claire et compréhensible, ou bien se chargera d'intégrer son module d'apprentissage automatique dans une application web par exemple, pour que son travail soit exploitable par les utilisateurs finaux du système.

Conclusion

Dans ce chapitre, nous avons présenté l'organisme d'accueil et réalisé une étude préalable du projet. De plus, nous avons présenté la méthodologie du travail CRISP-DM en expliquant ses étapes. Nous passons dans ce qui suit à la construction de la première release.

RELEASE 1 "DÉTECTION DES OFFRES DE FROMATION"

Plan

Introduction	11
SPRINT1 : Compréhension du prboléme metier et des donnéess	11
1 Compréhension du problème métier :	11
2 Compréhension des données	17
SPRINT2 : Préparation et visualisation des données	23
3 Agrégation des données	23
4 Nettoyage des données	27
5 La visualisation des données	30
SPRINT 3 : Modélisation et évaluation	35
6 La modélisation	35
7 L'évaluation	52
Conclusion	62

Introduction

Dans ce chapitre, nous allons nous concentrer sur le premier livrable de notre projet : un module d'apprentissage automatique pour la détection des offres de formation adéquates à chaque agent téléopérateur dans l'entreprise 1WayCom. Ce chapitre a été conduit à travers la méthodologie agile, et divisé sur 4 Sprints différents :

- Sprint1 : Compréhension du problème métier et des données ;
- Sprint 2 : Préparation et visualisation des données ;
- Sprint 3 : Modélisation et évaluation

SPRINT1 : Compréhension du problème métier et des données

Ce sprint se concentre sur les deux premières phases de la méthodologie CRISP-DM, qui sont la compréhension du problème métier et la compréhension des données. Nous avons fait une étude totale des données de 1WayCom et organisé plusieurs réunions avec les responsables de l'entreprise pour saisir la finalité du projet.

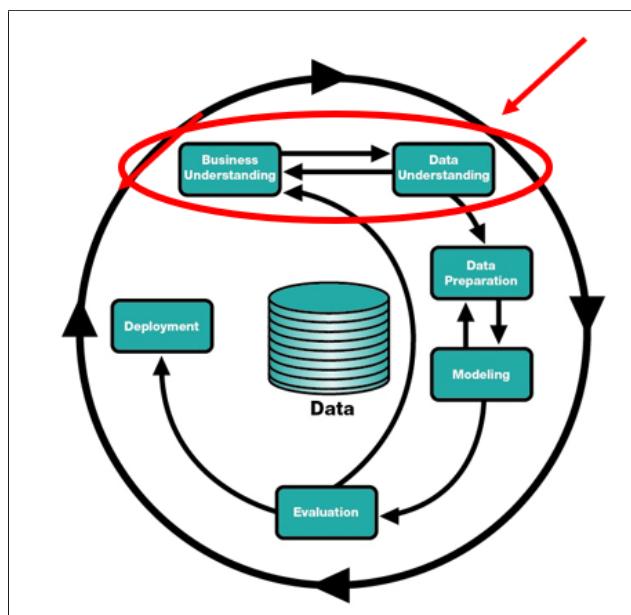


Figure 2.1: Compréhension des données dans CRISP-DM

2.1 Compréhension du problème métier :

Notre projet s'installe dans le cadre de l'amélioration de la détection des formations pour les agents de 1WayCom. Les agents sont les téléopérateurs qui ont la tâche de convaincre les

prospects d'acheter un produit à travers des appels téléphoniques. Leur façon de parler, savoir être et savoir-faire sont donc très importants dans la productivité de l'entreprise. C'est pour cette raison que le volet formation, pour 1WayCom, doit être bien élaboré. Il existe plusieurs types de formation. Le processus de choix de la formation adaptée au profil de l'agent sera détaillé ci-après.

- **Formation métier :** Cette formation vise à améliorer les savoirs liés à la communication. Le téléopérateur doit entretenir un bon discours commercial et relationnel avec son client, en prêtant attention à son intonation, en faisant de l'écoute active, en diminuant le temps mort et les hésitations dans ses réponses.
- **Formation produit :** Cette formation a une tendance didactique, c'est-à-dire qu'elle renseigne le téléopérateur sur tous les détails du produit qu'il doit vendre. Ce dernier doit connaître toutes les informations liées à son produit et doit pouvoir bien expliquer son fonctionnement, les tarifs, etc.
- **Formation en traitement des objections :** Dans un appel téléphonique de prospection, le téléopérateur risque de recevoir beaucoup d'objections de la part du prospect. Bien que toutes les objections possibles soient traitées au préalable, plusieurs agents oublient très rapidement la bonne réponse à fournir ou perdent le rythme face à un client difficile. Cette formation vise donc à consolider le savoir lié au traitement des objections et à inculquer le bon savoir-faire chez tous les agents.
- **Formation en souscription :** Beaucoup de téléopérateurs accordent une importance majeure à la prospection et oublient un côté très important de leur travail qui est la souscription. En conséquence, beaucoup de ventes ne sont pas validées. Cette formation s'adresse aux agents qui sont tellement pressés de convaincre leur client et d'atteindre leur objectif de vente qu'ils bâclent le processus de souscription comme l'obtention du RIB exact par exemple.
- **Notion de feuille d'écoute :** Chaque superviseur a un objectif journalier d'un certain nombre d'écoutes à faire. Bien entendu, tous les appels téléphoniques réalisés par les agents sont enregistrés dans des fichiers audio. Le superviseur doit effectuer des écoutes sur les agents de son groupe et attribuer les notes appropriées, dans la feuille d'écoute. La feuille d'écoute est donc une grille qui contient plusieurs catégories et sous-catégories de notation.

2.1.1 Détermination des objectifs de l'analyse

La détermination des objectifs de l'analyse est primordiale dans un projet de Data Science. C'est une étape qui va permettre au praticien de comprendre encore plus le travail demandé pour parvenir à une concrétisation adéquate du projet. La définition des variables d'activité clés, du résultat souhaité, des mesures de réussite et des sources de données vont apporter une vision claire de la stratégie à adopter.

2.1.2 Identification des variables d'activité clés

Dans chaque problème de prédiction résolu par apprentissage automatique, il y'a des variables clés qui sont les variables à prédire. Bien entendu, nous allons prédire l'offre de formation adéquate pour chaque agent. La solution que nous présentons est majeure dans l'amélioration de la production de l'entreprise 1WayCom, dont la politique est axée sur l'investissement dans l'agent à travers des formations continues.

2.1.3 Offre de formation

Nous allons essayer de prédire l'offre de formation adéquate pour chaque agent à partir des notes attribuées dans les feuilles d'écoutes d'un mois déterminé, ainsi que d'autres éléments. Ces critères seront donc nos attributs ou « Features ». Nous sommes ici en présence d'un problème de classification, en apprentissage automatique supervisé. Dans notre cas, un agent peut avoir besoin de plusieurs formations à la fois. C'est donc une classification multi label, où plusieurs étiquettes peuvent être attribuées à chaque instance.

2.1.4 Les mesures de réussite

A la fin de notre projet, nous devons livrer un système qui sera en mesure d'automatiser la détection de formation adéquate à chaque agent. Notre module d'apprentissage automatique doit prendre en compte tous les critères clés qui décident si un téléopérateur présente un problème dans une compétence donnée, aidant ainsi les superviseurs dans leur prise de décision. Pour cela, nous avons déterminé un seuil de précision de 90%. Les mesures de réussite du projet se rapportent essentiellement au dernier Sprint de ce chapitre, qui est celui de l'Évaluation.

2.1.5 Identification des sources de données

- **La base de données de 1WayCom :** Notre source de données est exclusivement la base de données de 1WayCom, qui comporte 145 tables.

		Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer
□	receptions	★	Parcourir	Structure	Rechercher	Insérer	Vider Supprimer
□	receptions_sub_categories	★	Parcourir	Structure	Rechercher	Insérer	Vider Supprimer
□	recrutements	★	Parcourir	Structure	Rechercher	Insérer	Vider Supprimer
□	recrutement_competences	★	Parcourir	Structure	Rechercher	Insérer	Vider Supprimer
□	recrutement_sub_competences	★	Parcourir	Structure	Rechercher	Insérer	Vider Supprimer
□	recrutement_tables	★	Parcourir	Structure	Rechercher	Insérer	Vider Supprimer
□	recup	★	Parcourir	Structure	Rechercher	Insérer	Vider Supprimer
□	recuperations_avances	★	Parcourir	Structure	Rechercher	Insérer	Vider Supprimer
□	regions	★	Parcourir	Structure	Rechercher	Insérer	Vider Supprimer
□	renvois	★	Parcourir	Structure	Rechercher	Insérer	Vider Supprimer
145 tables		Somme					

Figure 2.2: La base de données de 1WayCom

- Les tables que nous avons sélectionnées pour notre projet

Nous allons utiliser 12 tables de la base de données de 1WayCom pour la réalisation de notre projet.

<input type="checkbox"/> absences		Parcourir	Structure	Rechercher		Insérer		Vider		Supprimer
<input type="checkbox"/> ecoutes		Parcourir	Structure	Rechercher		Insérer		Vider		Supprimer
<input type="checkbox"/> formations		Parcourir	Structure	Rechercher		Insérer		Vider		Supprimer
<input type="checkbox"/> grid_listen		Parcourir	Structure	Rechercher		Insérer		Vider		Supprimer
<input type="checkbox"/> grid_listen3_categories		Parcourir	Structure	Rechercher		Insérer		Vider		Supprimer
<input type="checkbox"/> grid_listen3_stored_calls		Parcourir	Structure	Rechercher		Insérer		Vider		Supprimer
<input type="checkbox"/> grid_listen3_stored_call_sub_categories		Parcourir	Structure	Rechercher		Insérer		Vider		Supprimer
<input type="checkbox"/> grid_listen3_sub_categories		Parcourir	Structure	Rechercher		Insérer		Vider		Supprimer
<input type="checkbox"/> holidays		Parcourir	Structure	Rechercher		Insérer		Vider		Supprimer
<input type="checkbox"/> prediction		Parcourir	Structure	Rechercher		Insérer		Vider		Supprimer
<input type="checkbox"/> sales_values		Parcourir	Structure	Rechercher		Insérer		Vider		Supprimer
<input type="checkbox"/> users		Parcourir	Structure	Rechercher		Insérer		Vider		Supprimer
12 tables		Somme								

Figure 2.3: Tables sélectionnées pour le projet

- Les tables utilisées dans le Release 1 : 4 tables

- La table absences : Nous avons utilisé la table absences pour en ressortir le nombre d'heures travaillées pour chaque agent.

id	user_id	created_by	groupe_id	operation_id	type 1:absent, 2:retard, 3:conge, 4:suite	nombre_heures	heure_sortie	heure_retard	heure_supplementaire	date	type_conge	is_justified 0:non, 1:justifié
2	114	114	9	1	1	8	0	0	0	2020-10-01	NULL	
3	146	114	9	1	1	8	0	0	0	2020-10-01	NULL	
4	112	114	9	1	1	8	0	0	0	2020-10-01	NULL	
5	166	114	9	1	2	0	0	0	0	2020-10-01	NULL	
6	183	114	9	1	1	8	0	0	0	2020-10-01	NULL	
7	186	114	9	1	1	8	0	0	0	2020-10-01	NULL	
8	207	114	9	1	1	8	0	0	0	2020-10-01	NULL	
9	50	50	4	1	1	8	0	0	0	2020-10-01	NULL	
10	15	50	4	1	2	0	0	0	0	2020-10-01	NULL	
11	109	50	4	1	1	8	0	0	0	2020-10-01	NULL	
...
/pe_conge		is_justified 0:non, 1:justifié		notice_agent	raison		created_at		updated_at		cloture_id	
		NULL		1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1	
		NULL		1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1	
		NULL		1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1	
		NULL		1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1	
		NULL		1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1	
		NULL		1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1	
		NULL		1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1	
		NULL		1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1	
		NULL		1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1	
		NULL		1	0	NULL	2020-10-02 10:28:16		2020-10-02 10:28:16		2	
		NULL		1	0	NULL	2020-10-02 10:28:16		2020-10-02 10:28:16		2	
		NULL		1	0	NULL	2020-10-02 10:28:16		2020-10-02 10:28:16		2	
		NULL		1	0	NULL	2020-10-02 10:28:16		2020-10-02 10:28:16		2	
		NULL		1	0	NULL	2020-10-02 10:28:16		2020-10-02 10:28:16		2	

Figure 2.4: La table absences

- La table grid listen3 categories Nous avons utilisé cette table pour extraire les noms des catégories de notation.

+ Options					
id	grid_listen_id	category	created_at	updated_at	
1	1	Evaluation Métier	2020-01-09 10:58:37	2020-01-09 10:58:37	
2	1	Qualité traitement	2020-01-09 10:58:37	2020-01-09 10:58:37	
3	1	Qualité des renseignements	2020-01-09 10:58:37	2020-01-09 10:58:37	
4	1	Conformité discours	2020-01-09 10:58:37	2020-01-09 10:58:37	
5	2	Evaluation Métier	2020-01-09 11:09:19	2020-01-09 11:09:19	
6	2	Qualité traitement	2020-01-09 11:09:19	2020-01-09 11:09:19	
7	2	Qualité des renseignements	2020-01-09 11:09:19	2020-01-09 11:09:19	
8	2	Conformité discours	2020-01-09 11:09:19	2020-01-09 11:09:19	
9	3	Evaluation Métier	2020-01-10 14:04:54	2020-01-10 14:04:54	
10	3	Qualité traitement	2020-01-10 14:04:54	2020-01-10 14:04:54	
11	3	Qualité des renseignements	2020-01-10 14:04:54	2020-01-10 14:04:54	
12	3	Conformité discours	2020-01-10 14:04:54	2020-01-10 14:04:54	
Console de requêtes SQL					

Figure 2.5: La table grid listen3 categories

- La table grid listen3 sub categories : Nous avons utilisé cette table pour extraire les noms des sous-catégories de notation.

id	grid_listen_id	category_id	sub_category	created_at	updated_at
1	1	1	Présentation	2020-01-09 10:58:37	2020-01-09 10:58:37
2	1	1	Qualité de l'accroche	2020-01-09 10:58:37	2020-01-09 10:58:37
3	1	1	Qualité de clôture de l'appel	2020-01-09 10:58:37	2020-01-09 10:58:37
4	1	1	Directivité	2020-01-09 10:58:37	2020-01-09 10:58:37
5	1	2	Intonation	2020-01-09 10:58:37	2020-01-09 10:58:37
6	1	2	Ecoute active	2020-01-09 10:58:37	2020-01-09 10:58:37
7	1	2	Hésitations / temps mort	2020-01-09 10:58:37	2020-01-09 10:58:37
8	1	2	Elocution / jargon utilisé	2020-01-09 10:58:37	2020-01-09 10:58:37
9	1	3	Qualification de l'appel	2020-01-09 10:58:37	2020-01-09 10:58:37
10	1	3	traitement des objections	2020-01-09 10:58:37	2020-01-09 10:58:37
11	1	3	Explication des tarifs	2020-01-09 10:58:37	2020-01-09 10:58:37
12	1	3	Explication du fonctionnement	2020-01-09 10:58:37	2020-01-09 10:58:37
Console de requêtes SQL					

Figure 2.6: La table grid listen3 sub categories

- La table users : Nous avons utilisé cette table pour ajouter le nom et le prénom de chaque agent.

	<input type="checkbox"/> Éditer	<input type="checkbox"/> Copier	<input type="checkbox"/> Supprimer	Id	recruit_id	first_name	last_name	role_id	poste_id	name	niveau_id	email	supervisor_id
<input type="checkbox"/>				1	NULL	Said	Sriti	1	1	Sriti	2	said@fwaycom.com	NULL
<input type="checkbox"/>				3	NULL	Sofien	LABIDI	4	9	NULL	2	sofien@fwaycom.com	36
<input type="checkbox"/>				4	NULL	Abir	CHERIF	2	1		3	cherifabir1991@gmail.com	9
<input type="checkbox"/>				5	NULL	Amira	Suidi	2	1		2	amirasuidigarnaoui@gmail.com	11
<input type="checkbox"/>				6	NULL	Amira	ZAGHDOUDI	2	1		3	amirazaghoudi26@gmail.com	50
<input type="checkbox"/>				7	NULL	Khouloud	LAARBI	2	1		3	arbi.khouloud@yahoo.fr	50
<input type="checkbox"/>				8	NULL	Salma	JERBI	2	1	NULL	3	salma@fwaycom.com	9
<input type="checkbox"/>				9	NULL	Walid	CHAHBI	3	18		3	chahbi1way@gmail.com	3
<input type="checkbox"/>				10	NULL	Chaïma	AIT LACHGAR	2	1		3	chainaitlachgar13@gmail.com	9
<input type="checkbox"/>				11	NULL	Dorra	BOUCHIBA	3	4		2	dorra@fwaycom.com	3

Figure 2.7: La table users

2.2 Compréhension des données

La compréhension des données est une étape primordiale dans un projet Data Science. C'est lors de cette phase que l'on collecte, décrit, représente, explore et visualise les données, en vue de modéliser une solution optimale. Une compréhension approfondie des données permet d'éviter les problèmes que l'on pourra rencontrer lors des phases suivantes du projet.

2.2.1 Données quantitatives continues

Les valeurs continues sont des valeurs qui s'expriment dans un intervalle de nombres réels infinis[1]. Nous n'avons pas de valeurs quantitatives continues dans notre jeu de données.

2.2.2 Données quantitatives discrètes

Les valeurs discrètes sont des valeurs qui s'expriment dans un intervalle de nombres discrets finis. Dans notre jeu de données, nous disposons uniquement de valeurs quantitatives discrètes :

- Les variables NC_tobj et M_tobj qui expriment respectivement le nombre de fois où l'agent a eu la note NC (Non Conforme) et la note Moyen en traitement des objections dans un mois déterminé.
- Les variables NC_acc et M_acc qui expriment respectivement le nombre de fois où l'agent a eu la note NC (Non Conforme) et la note Moyen en qualité de l'accroche dans un mois déterminé.
- Les variables NC_dir et M_dir qui expriment respectivement le nombre de fois où l'agent a eu la note NC (Non Conforme) et la note Moyen en directivité dans un mois déterminé.

- Les variables NC_int et M_int qui expriment respectivement le nombre de fois où l'agent a eu la note NC (Non Conforme) et la note Moyen en intonation dans un mois déterminé.
- Les variables NC_ecact et M_ecact qui expriment respectivement le nombre de fois où l'agent a eu la note NC (Non Conforme) et la note Moyen en écoute active dans un mois déterminé.
- Les variables NC_hes et M_hes qui expriment respectivement le nombre de fois où l'agent a eu la note NC (Non Conforme) et la note Moyen en hésitation et temps mort dans un mois déterminé.
- Les variables NC_expfonc et M_expfonc qui expriment respectivement le nombre de fois où l'agent a eu la note NC (Non Conforme) et la note Moyen en explication du fonctionnement du produit dans un mois déterminé.
- Les variables NC_exptar et M_exptar qui expriment respectivement le nombre de fois où l'agent a eu la note NC (Non Conforme) et la note Moyen en explication des tarifs dans un mois déterminé.
- Les variables NC_clotapp et M_clotapp qui expriment respectivement le nombre de fois où l'agent a eu la note NC (Non Conforme) et la note Moyen en clôture de l'appel dans un mois déterminé.
- La variable nbr_ecoutes qui exprime le nombre de fois où l'on a effectué une écoute sur l'agent en question dans un mois déterminé.

2.2.3 Données qualitatives nominales

Les valeurs nominales sont des valeurs qualitatives exprimant le nom d'une catégorie : nom, sexe, métier, voiture, etc.[2] Les données binaires sont des données catégoriales nominales, puisque généralement, une variable binaire représente deux valeurs conceptuellement opposées. Les variables formation1 (formation en traitement des objections), formation2 (Formation métier), formation3 (formation produit) et formation4 (formation en souscription) qui sont des variables binaires catégoriales qui prennent la valeur 1 dans le cas où l'agent a besoin de la formation, et 0 dans le cas contraire.

2.2.4 Données qualitatives ordinaires

Les valeurs ordinaires sont des valeurs qualitatives qui sont naturellement ordonnées et qui peuvent être traduites par une valeur numérique, comme le rang par exemple : élevé, moyen, bas.

2.2.5 Données temporelles

Les données temporelles dans notre jeu de données sont des valeurs numériques qui suivent l'évolution du temps.

- La variable **mois** qui extrait le mois de la date de l'appel ;
- La variable **année** qui extrait l'année de la date de l'appel.

2.2.6 Représentation des données

La représentation sert à modéliser les données en vue de s'assurer qu'elles sont adéquates à la méthode de travail ultérieure. Cela permet d'avoir une vision plus claire des variables cibles, de la variable à prédire, et de la façon par laquelle elles seront formatées dans les modèles d'apprentissage automatiques que nous allons utiliser.

- **Représentation des variables prédictives :** Les variables prédictives sont appelées variables indépendantes et sont largement connues dans la Data Science sous le nom anglais « Features ».

Dans une Data Set, les variables prédictives $X = x_1, x_2, \dots, x_n$ peuvent être modélisées dans une matrice d'ordre $m \times n$ où m est la taille du training set et n est le nombre des caractéristiques de chaque instance. Dans notre étude, $n=19$ et $m=1113$ comme l'explique la figure 2.8.

$$\begin{bmatrix} x(1,1) & x(1, \dots) & \dots & x(1,19) \\ \vdots & \vdots & & \vdots \\ x(1133,1) & x(1133, \dots) & \dots & x(1133,19) \end{bmatrix}$$

Figure 2.8: Matrice représentative des variables prédictives

- **Représentation de la variable cible :** Bien entendu, la variable cible est la variable qu'on souhaite prédire à partir des variables prédictives. Elle est aussi appelée en anglais « Target variable ». Dans notre cas, un agent peut avoir besoin de plusieurs formations. Nous sommes donc en présence d'une classification multi label où chaque instance de X peut avoir plusieurs étiquettes $Y = y_1, y_2, y_3, y_4$ correspondant aux quatre formations disponibles. Notre variable cible est donc représentée par une matrice $m \times k$ où $m =$ Taille de la Data Set (1113) et $k =$ nombre d'étiquettes 2.9.

$y(1,1)$	$y(1,2)$	$y(1,3)$	$y(1,4)$
\vdots	\vdots	\vdots	\vdots
$y(1133,1)$	$y(1133,2)$	$y(1133,3)$	$y(1133,4)$

Figure 2.9: Matrice représentative de la variable cible

2.2.7 Exploration des données

Dans l'exploration des données, le Data Scientiste essaie d'extraire les connaissances statistiques qui pourraient l'aider dans ses tâches ultérieures.

- **La moyenne :** Dans une série statistique, la moyenne est le quotient de la somme des valeurs de la série sur la somme de ses effectifs.[3] La moyenne de la série statistique $[2, 3, 15, 4] = 6$
- **Le mode :** Dans une série statistique, le mode est la valeur la plus fréquente, c'est-à-dire qui y apparaît le plus. Le mode de la série statistique $[3, 5, 3, 3, 9] = 3$
- **La médiane :** La médiane est la valeur centrale d'une série statistique contenant n valeurs. La médiane vient séparer la distribution en deux groupes égaux. Pour calculer la médiane, il faut classer les valeurs de la série dans l'ordre croissant. Si n est impair, la médiane = $n+1/2$ Exemple : La médiane de $[2, 5, 1, 3, 5] = 1$ Si n est pair, la médiane est la demi somme des deux valeurs au milieu. Exemple : La médiane de $[2, 6, 8, 9] = 6+8/2 = 7$
- **Les quartiles :** Les quartiles séparent les données en 4 groupes. Il existe trois quartiles nommés respectivement Q1, Q2 et Q3.[4]
 - Le premier quartile Q1 : Appelé aussi quartile inférieur, il représente la valeur du milieu de la première partie médiane, c'est-à-dire 25% des données ordonnées N lui sont inférieurs et 75% lui sont supérieurs. $N*25\% = N/4$ Si $N/4$ est un entier, le quartile est la 0.25ème valeur de la série des données ordonnées. Si $N/4$ est un décimal, le quartile est l'arrondit à l'entier supérieur. Exemple : Dans une série de données ordonnée, si $N/4=3.2$, le premier quartile sera la 4ème valeur.
 - Le deuxième quartile Q2 : C'est la médiane puisqu'il divise les données en 50% inférieurs et 50% supérieurs à une valeur.
 - Le troisième quartile Q3 : Appelé aussi quartile supérieur, il représente la valeur d'une série de données ordonnées N qui est supérieure ou égale à 75% des valeurs statistiques de la série. $N*75\% = 3N/4$ Si $3N/4$ est un entier, le quartile est la 0.75ème valeur de la

série des données ordonnées. Si $3N/4$ est un décimal, le quartile est l'arrondit à l'entier supérieur. Exemple : Dans une série de données ordonnée, si $3N/4=10.5$, le troisième quartile sera la 11ème valeur.

2.2.8 Synthèse

Dans ce tableau, nous allons affecter les features que nous allons utiliser dans notre modèle selon leur nature et leur caractère prédictif ou cible. Nous donnerons également les valeurs statistiques des variables dans le deuxième tableau.

Tableau 2.1: La nature de nos données.

La variable	Quantitative continue	Quantitative discrète	Qualitative nominale	Qualitative ordinale	Temporelle	Préditive	Cible
agent_id		✗					
mois					✗		
annee					✗		
NC_tobj		✗				✗	
M_tobj		✗				✗	
NC_acc		✗				✗	
M_acc		✗				✗	
NC_int		✗				✗	
M_int		✗				✗	
NC_dir		✗				✗	
M_dir		✗				✗	
NC_hes		✗				✗	
M_hes		✗				✗	
NC_expfonc		✗				✗	
M_expfonc		✗				✗	
NC_exptar		✗				✗	
M_exptar		✗				✗	
NC_ecact		✗				✗	
M_ecact		✗				✗	
NC_clotapp		✗				✗	
M_clotapp		✗				✗	
nbr_ecoutes		✗					
formation1			✗				✗
formation2			✗				✗
formation3			✗				✗
formation4			✗				✗

Tableau 2.2: Les statistiques basiques des variables.

Variable	Moyenne	Mode	Médiane	Q1	Q3
NC_tobj	0.08778	0	0	0	0
M_tobj	0.533563	0	0	0	1
NC_acc	0.010327	0	0	0	0
M_acc	1.005164	0	1	0	2
NC_int	0.005164	0	0	0	0
M_int	0.638554	0	0	0	1
NC_dir	0.015491	0	0	0	0
M_dir	0.550775	0	0	0	1
NC_hes	0.092943	0	0	0	0
M_hes	1.807229	1	2	1	2
NC_expfonc	0.010327	0	0	0	0
M_expfonc	0.208262	0	0	0	0
NC_exptar	0.025818	0	0	0	0
M_exptar	0.280551	0	0	0	0
NC_ecact	0.051635	0	0	0	0
M_ecact	0.841652	0	1	0	1
NC_clotapp	0.087780	0	0	0	0
M_clotapp	1.258176	1	1	0	2
nbr_ecoutes	2.790017	2	3	2	4
formation1	0.096386	0	0	0	0
formation2	0.199656	0	0	0	0
formation3	0.049914	0	0	0	0
Formation4	0.392427	0	0	0	1

NB : les valeurs égales à 0 ne sont pas des valeurs nulles. La valeur 0 dans NC_tobj par exemple exprime le nombre de fois où l'agent reçoit la note NC (Non Conforme) en traitement des objections dans un mois déterminé. L'obtention de la note NC n'est donc pas très récurrente.

SPRINT2 : Préparation et visualisation des données

Dans ce sprint, nous nous sommes concentrés sur la préparation des données, qui englobe des opérations qui visent à réduire le bruit dans les données, et qui aideront à exploiter des données valides pour avoir un projet réussi. C'est une étape très importante puisqu'elle influence de manière directe la qualité du modèle. Ensuite, nous avons procédé à la visualisation des données, qui donne une vision concrète du jeu de données pour pouvoir en déceler les particularités.

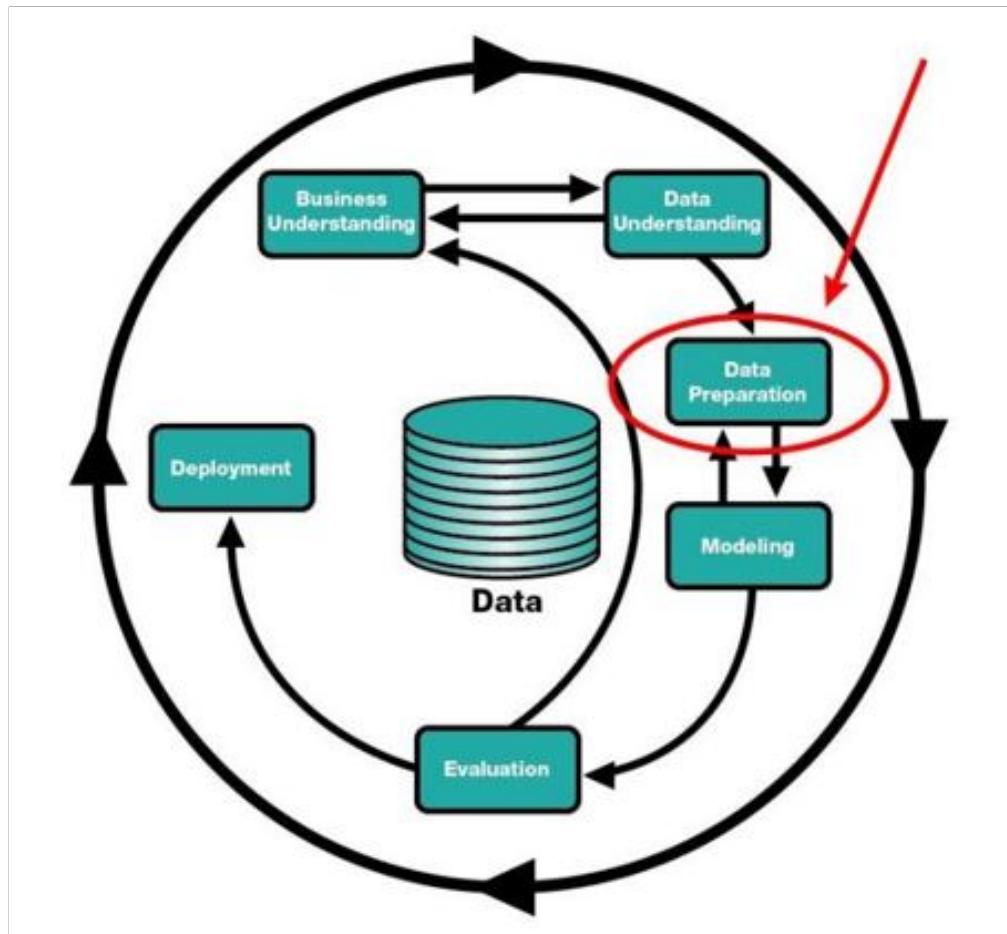


Figure 2.10: Préparation des données dans CRISP-DM

2.3 Agrégation des données

Les données nécessaires pour créer notre module d'apprentissage automatique étaient insuffisantes. Bien que l'entreprise 1WayCom enregistrait toutes les notes obtenues des agents suite aux écoutes, elle ne laissait aucune trace des formations dont leurs agents ont bénéficié. Nous n'avions donc pas de base de données historique qui nous permettrait de faire nos prédictions. Nous avons donc

organisé une réunion avec la responsable formation, pour déterminer les critères X qui induisent un besoin dans une certaine formation Y. Nous avons donc élaboré cette solution provisoire, qui nous permettrait de faire tourner nos algorithmes d'apprentissage automatique. Nous avons proposé à 1WayCom de commencer à souscrire les formations futures des agents, sur une durée minimum de 6 mois, pour avoir une base de données suffisante. A la fin de cette période, nous feront tourner sur cette nouvelle base de données historique les algorithmes que nous aurons déjà construits au cours de ce projet, et nous verrons ce qui se passera en terme de mesures de performances. L'agrégation des données permet d'utiliser les données brutes du jeu de données actif en vue d'en tirer de nouvelles observations et de nouvelles variables qui seront utilisées dans le travail de Data Science. Comme nous l'avons déjà expliqué, nous n'avons pas une table déjà prête sur laquelle nous pouvons faire tourner nos algorithmes. Il nous fallait exploiter la base de données de 1wayCom et apporter de nombreuses modifications à travers des requêtes SQL et des scripts en Python pour aboutir à une table, qui par la suite sera convertie en un fichier csv exploitable par le Machine Learning.

2.3.1 Les modifications apportées sur les données

Nous présentons dans ce qui suit quelques requêtes utilisées pour l'obtention de la base de données formations. Nous sommes partis de la table grid_listen3_stored_calls_sub_categories.

- Ajout du nom de la catégorie de notation :

```
UPDATE `grid_listen3_stored_call_sub_categories` SET  
`category_name` = (SELECT category from grid_listen3_categories where  
grid_listen3_stored_call_sub_categories.category_id=  
grid_listen3_categories.id)
```

Figure 2.11: La nature de nos données

- Ajout de la note obtenue :

```
UPDATE `grid_listen3_stored_call_sub_categories` SET `note_name`=(SELECT nom  
FROM grid_listen3_notes WHERE  
grid_listen3_notes.id=grid_listen3_stored_call_sub_categories.note_id)
```

Figure 2.12: La nature de nos données

- Ajout de l'id de l'agent :

```
UPDATE grid_listen3_stored_call_sub_categories SET agent_id=( SELECT  
agent_id FROM grid_listen3_stored_calls WHERE  
grid_listen3_stored_call_sub_categories.grid_listen3_stored_call_id=  
grid_listen3_stored_calls.id)
```

Figure 2.13: - Ajout de l'id de l'agent

- Ajout de la date des appels :

```
UPDATE grid_listen3_stored_call_sub_categories SET date_call=( SELECT  
date_call FROM grid_listen3_stored_calls WHERE  
grid_listen3_stored_call_sub_categories.id=grid_listen3_stored_calls.id)
```

Figure 2.14: Ajout de la date des appels

2.3.2 Scripts en Python

Certaines manipulations exigent le développement de quelques scripts en Python.

Calcul des occurrences de notes : Nous avons calculé pour chaque mois, le nombre de fois où un agent X a obtenu la note “NC” et le nombre de fois où il a obtenu la note “Moyen” dans chaque sous-catégorie de notation en relation avec les formations proposées. Voici un exemplaire du code utilisé pour le calcul de toutes les occurrences des notes :

- Le même code se répète pour le calcul des notes :
- Accroche
- Hésitation
- Directivité
- Ecoute Active

```

sql="""SELECT agent_id FROM formations"""
cursor.execute(sql)
id =cursor.fetchall()
for x in id:
    for y in range(1,13):
        sql1="""UPDATE formations SET M_tobj=(SELECT COUNT(*) from grid_listen3_stored_call_sub_
WHERE agent_id=%s AND sub_category_name='traitement des objections' and note_name='Moyen'
AND MONTH(date_call)=%s) WHERE agent_id=%s and mois=%s"""
        cursor.execute(sql1,(x,y,x,y))
connection.commit()
sql2="""UPDATE formations SET NC_tobj=(SELECT COUNT(*) from grid_listen3_stored_call_sub_
WHERE agent_id=%s AND sub_category_name='traitement des objections' and note_name='NC'
AND MONTH(date_call)=%s) WHERE agent_id=%s and mois=%s"""
cursor.execute(sql2,(x,y,x,y))
connection.commit()

```

Figure 2.15: code python1

- Intonation
- Explication du fonctionnement
- Explication des tarifs
- Qualité de clôture d'appel

Nous avons aussi calculé le nombre des écoutes pour chaque agent en un mois dans le code présenté dans la figure 2.16

```

cursor = mydb.cursor()
cursor.execute("""SELECT agent_id FROM formations""")
myresult = cursor.fetchall()

for x in myresult :
    #print(i)
    for y in range(1,13):
        sql1="""SELECT agent_id FROM formations"""
        cursor.execute(sql1)
        id =cursor.fetchall()
        for x in id:
            for y in range(1,13):
                sql2="""SELECT COUNT(*) FROM grid_listen3_stored_calls WHERE agent_id=%s
and MONTH(date_call)=%s and YEAR(date_call)=2020 and grid_listen_id=2 """
                cursor.execute(sql2,(x,y))
                nbr=cursor.fetchall()
                #print(nbr)
                sql="UPDATE formations SET nbr_ecoutes=%s WHERE agent_id=%s and mois=%s "
                cursor.execute(sql,(nbr,x,y))
                mydb.commit()

```

Figure 2.16: code python2

Nous sommes ensuite passés à l'ajout des colonnes formation1, formation2, formation3 et formation4 et à la mise à jour de ces colonnes

- OUI : si l'agent a besoin de la formation.
- NON : si l'agent n'a pas besoin de la formation L'agent a besoin d'une formation X s'il a obtenu, en un mois, 1 fois la note NC ou bien 3 fois la note Moyen.

2.3.3 Table résultante

Notre table résultante est représentée par la figure 2.17

id	agent_id	mois	NC_tobj	M_tobj	NC_acc	M_acc	NC_dir	M_dir	NC_int	M_int	NC_ecact	M_ecact	NC_hes	M_hes	NC_exponc	M_exponc
1	30	1	0	0	0	1	0	0	0	2	0	2	0	2	0	0
2	14	1	0	2	0	0	0	0	0	0	0	3	0	3	0	0
3	31	1	0	3	0	0	0	2	0	0	0	4	0	4	0	3
4	10	1	0	0	0	1	0	0	0	3	0	3	0	1	0	0
5	61	1	0	2	0	0	0	1	0	0	0	2	0	4	0	4
6	9	1	1	1	0	0	0	0	0	0	0	2	0	1	0	1
7	18	1	0	0	0	1	0	0	0	3	0	1	0	3	0	0
NC_exptar	M_exptar	NC_clotapp	M_clotapp	nbr_ecoutes	annee	formation1	formation2	formation3	formation4							
0	0	0	3	2	2020	0	1	0	0							
0	0	1	0	6	2020	0	1	0	0							
0	3	0	0	5	2020	1	0	1	1							
0	0	0	0	3	2020	0	0	0	0							
0	3	0	1	7	2020	0	0	1	1							
0	1	0	1	3	2020	1	0	0	0							
0	1	0	4	3	2020	0	1	0	1							

Figure 2.17: La table formations résultante

2.4 Nettoyage des données

Le Data Cleaning est le mot anglais pour nettoyage des données. Il permet d'identifier et de se débarrasser des valeurs erronées et dupliquées, de traiter les valeurs manquantes et de formater les types de données. Cela permet d'obtenir un « Training Set » plus fiable et structuré, de façon à ce que l'apprentissage automatique soit plus performant et soutienne une meilleure prise de décision.

La figure 2.18 montre notre Data Set avant le nettoyage des données avec la fonction df.info() :

...								
1128	1464	244	12	0	...	NON	NON	NON
NON								
1129	1465	241	12	0	...	NON	NON	NON
NON								
1130	1466	247	12	0	...	OUI	OUI	NON
OUI								
1131	1467	246	12	0	...	NON	NON	NON
OUI								
1132	1468	148	12	0	...	NON	NON	NON
NON								
[1133 rows x 27 columns]								

Figure 2.18: La Data Set avant le nettoyage des données

Il existe plusieurs librairies Python pour ce type de processus. Nous avons choisi la librairie Pandas, largement connue comme étant très efficace et facile pour la manipulation, l'analyse et le nettoyage des données.

2.4.1 Le nettoyage des valeurs nulles

Les valeurs nulles donnent des résultats erronés dans l'apprentissage automatique et certains algorithmes ne travaillent que si toutes les valeurs nulles du training set sont supprimées. La figure 2.19 montre le résultat de la fonction Dropna()

```
Console 3/A
NC_exptar    1133 non-null int64
M_exptar      1133 non-null int64
NC_clotapp    1133 non-null int64
M_clotapp     1133 non-null int64
nbr_ecoutes   1133 non-null int64
year          1133 non-null int64
formation1    1133 non-null object
formation2    1133 non-null object
formation3    1133 non-null object
formation4    1133 non-null object
dtypes: int64(23), object(4)
memory usage: 230.1+ KB
None
```

Le nombre de lignes est le même donc il n'y avait pas de valeurs nulles dans la table.

Figure 2.19: La Data Set avant le nettoyage des données

2.4.2 Le nettoyage des lignes dupliquées

Le nettoyage des valeurs dupliquées se fait avec la fonction `drop_duplicated()` de Pandas.

...																							
603	776	67	1	0	...																		
OUI																							
604	777	227	1	0	...																		
OUI																							
605	778	255	1	0	...																		
NON																							
606	779	212	1	0	...																		
NON																							
823	1091	29	6	1	...																		
NON																							
[581 rows x 27 columns]																							

Figure 2.20: La Data Set après le nettoyage des données

Nous constatons ainsi que la table est passée de 1133 lignes à 581 lignes.

2.4.3 Le nettoyage du format des colonnes

Dans l'apprentissage automatique, il y a des algorithmes qui ne fonctionnent qu'avec des valeurs numériques. Nous avons donc remplacé le type des colonnes `formation1`, `formation2`, `formation3` et `formation4` du type text au type int. Nous avons remplacé la valeur OUI par 1 et la valeur NON par 0.

formation1	formation2	formation3	formation4	formation1	formation2	formation3	formation4
NON	OUI	NON	NON	0	1	0	0
NON	OUI	NON	OUI	0	1	0	1
OUI	NON	OUI	OUI	1	0	1	1
NON	NON	NON	OUI	0	0	0	1
NON	NON	OUI	OUI	0	0	1	1
OUI	NON	NON	NON	1	0	0	0
NON	OUI	NON	OUI	0	1	0	1
OUI	OUI	OUI	OUI	1	1	1	1
NON	NON	NON	NON	0	0	0	0
NON	NON	NON	NON	0	0	0	0
OUI	NON	NON	NON	1	0	0	0

Figure 2.21: La transformation des colonnes dupliquées

2.4.4 Fichier CSV résultant

Notre Data Set est maintenant prête à être proprement exploitée.

id	agent_id	mois	NC_tobj	M_tobj	NC_acc	M_acc	NC_dif	M_dif	NC_int	M_int	NC_exact	M_exact	NC_hes	M_hes	NC_explor	M_explor	NC_expat	M_expat	NC_clotas	M_clotas	rbr_ecout	year	formation1	formation2	formation3	formation4	
2	1	30	1	0	0	0	1	0	0	2	0	0	2	0	0	0	0	0	3	2	2020	0	1	0	0		
3	2	14	1	0	2	0	0	0	0	0	0	0	3	0	0	0	1	0	6	2020	0	1	0	1			
4	3	31	1	0	3	0	0	0	2	0	0	0	4	0	4	0	3	0	0	5	2020	1	0	1	1		
5	4	10	1	0	0	0	1	0	0	3	0	3	0	1	0	0	0	0	3	2020	0	0	0	1			
6	5	61	1	0	2	0	0	0	1	0	0	0	2	0	4	0	4	0	3	0	1	7	2020	0	0	1	
7	6	9	1	1	1	0	0	0	0	0	0	0	2	0	1	0	1	0	1	3	2020	1	0	0	0		
8	7	18	1	0	0	0	1	0	0	0	3	0	1	0	3	0	0	1	0	4	3	2020	0	1	0	1	
9	8	25	1	0	5	0	2	0	3	0	5	0	3	0	7	0	5	0	3	0	3	8	2020	1	1	1	
10	9	29	1	0	0	0	0	0	0	0	2	0	0	0	2	0	0	0	0	0	0	1	2020	0	0	0	
11	10	12	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	2020	0	0	0	0	
12	11	20	1	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	2020	1	0	0	
13	12	21	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	2020	0	0	0	
14	13	78	1	0	4	0	0	0	0	0	0	0	4	0	5	0	5	0	0	0	4	6	2020	1	1	0	
15	14	67	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	2020	0	0	0	1	
16	15	8	1	1	1	0	1	0	3	0	5	0	5	0	6	0	5	0	3	5	2020	1	1	0	1		
17	16	39	1	1	1	0	2	0	2	0	2	0	0	4	1	0	0	0	1	3	2020	1	1	1	1		
18	17	32	1	0	1	0	0	0	3	0	0	0	2	0	0	0	1	0	1	3	2020	0	0	1	1		
19	18	53	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	2020	0	0	0	0	
20	19	56	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	2020	0	0	0	0	
21	20	46	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	2020	0	0	0	0	
22	21	95	1	0	4	0	0	0	1	0	1	0	2	1	5	0	0	0	0	3	3	2020	1	1	0	1	
23	22	94	1	0	1	0	0	0	3	0	0	0	3	0	4	0	3	0	0	0	1	4	2020	0	0	0	1
24	23	37	1	0	0	0	0	0	0	0	2	0	0	2	0	0	0	0	0	2	2	2020	0	0	0	0	
25	24	57	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	5	2020	0	0	0	0	
26	25	65	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	5	2020	0	0	0	0	
27	26	52	1	0	0	0	1	0	1	0	0	0	2	0	2	0	0	0	0	0	1	2020	0	0	0	0	
28	27	7	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	2020	0	0	0	0	
29	28	79	2	0	1	0	0	0	2	0	3	0	1	0	3	0	0	0	0	3	5	2020	0	1	0	1	
30	29	80	2	0	1	0	2	0	1	0	1	0	2	0	3	0	0	0	0	4	4	2020	0	1	0	1	
31	34	8	2	0	1	0	0	0	0	4	0	5	1	3	0	0	0	0	0	4	4	2020	0	1	0	1	
32	35	30	2	0	2	0	1	0	0	2	0	5	0	4	0	0	0	0	1	4	2020	0	0	0	1		
33	36	78	2	0	0	0	0	0	0	2	0	5	0	4	0	0	0	1	0	3	4	2020	0	1	0	1	
34	37	37	2	0	1	0	2	0	3	0	3	0	2	0	5	0	0	1	1	2	5	2020	0	1	0	1	
35	38	95	2	0	2	0	0	0	0	1	0	2	0	5	0	5	0	0	0	4	4	2020	0	1	0	1	

Figure 2.22: Le Fichier CSV de la Data Set résultante

2.5 La visualisation des données

La visualisation permet de mieux comprendre les données, puisqu'elle les résume d'une manière graphique compréhensible par quasiment tout le monde.

2.5.1 Visualisation du besoin en formation

Nous avons utilisé les bibliothèques suivantes :

Pandas : bibliothèque python déjà expliquée dans le nettoyage des données.

Matplotlib : Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques.[5]

Nos labels sont formations1, formation2, formation3 et formation4. Le diagramme circulaire de fréquence suivant nous montre que pour chaque label, nous avons un nombre déséquilibré d'échantillons dans chaque classe. Les échantillons de la classe 0 sont beaucoup plus nombreux que ceux de la classe 1. Nous sommes donc à la présence d'une Data set déséquilibrée appelée « Imbalanced ».

Nous sommes donc à la présence d'une Data set déséquilibrée appelée « Imbalanced ».

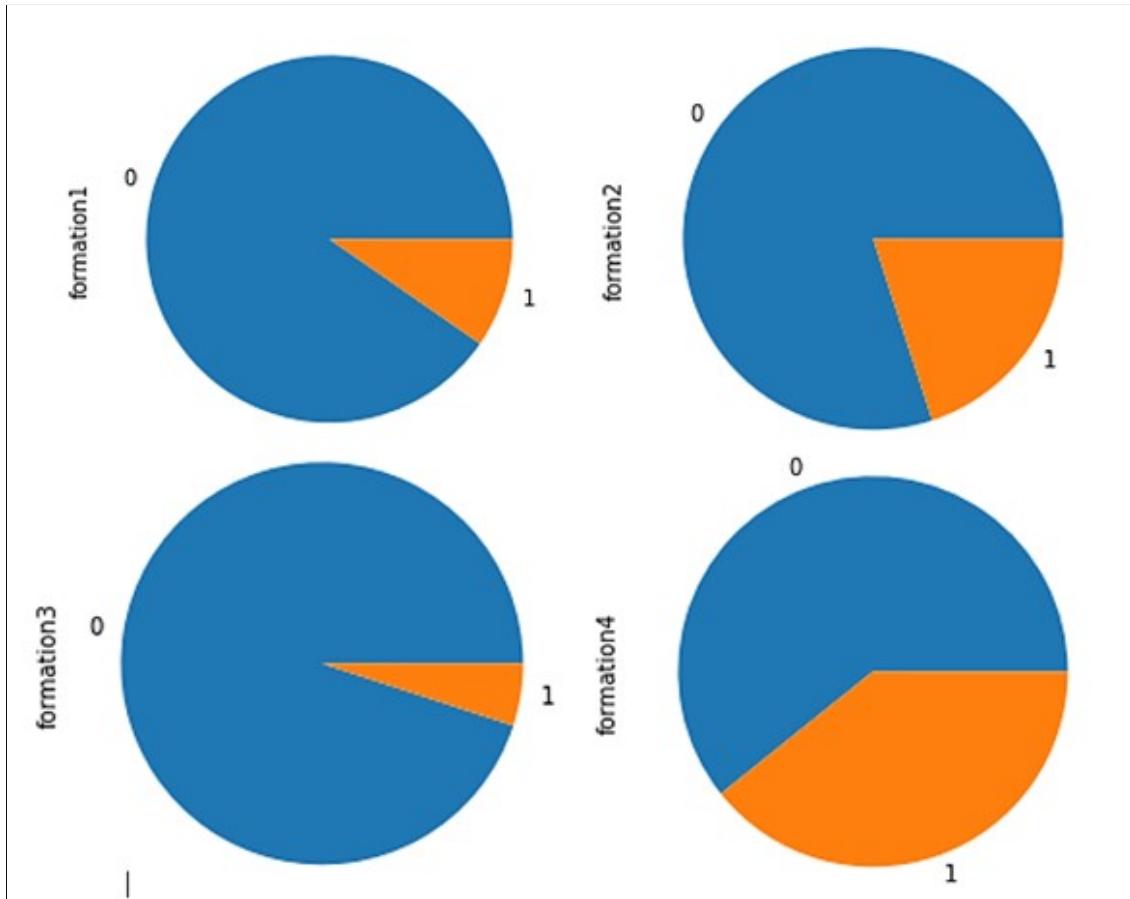


Figure 2.23: Diagramme circulaire du besoin en chaque formation

2.5.2 Visualisation de la distribution des écoutes au cours de l'année

Altair : Est une bibliothèque de visualisation statistique déclarative pour Python [6].

Le graphe de la figure 2.24 est réalisé avec la librairie Altair, avec une technique de liaison entre deux types de visualisations. Il montre la distribution des écoutes sur les mois de l'année.

Cette figure 2.24 montre que les écoutes ne sont pas également réparties sur les mois de l'année, alors qu'elles devraient l'être, pour pouvoir suivre le progrès des agents convenablement. 1WayCom doit établir un nouveau planning équilibré pour les écoutes effectuées sur les agents.

2.5.3 Visualisation de la distribution des formations sur les mois de l'année

Le graphe 2.25 montre que lorsque le besoin en une formation atteint un pic élevé, il présente juste après une diminution représentée par une pente négative. Cela veut dire que soit les formations

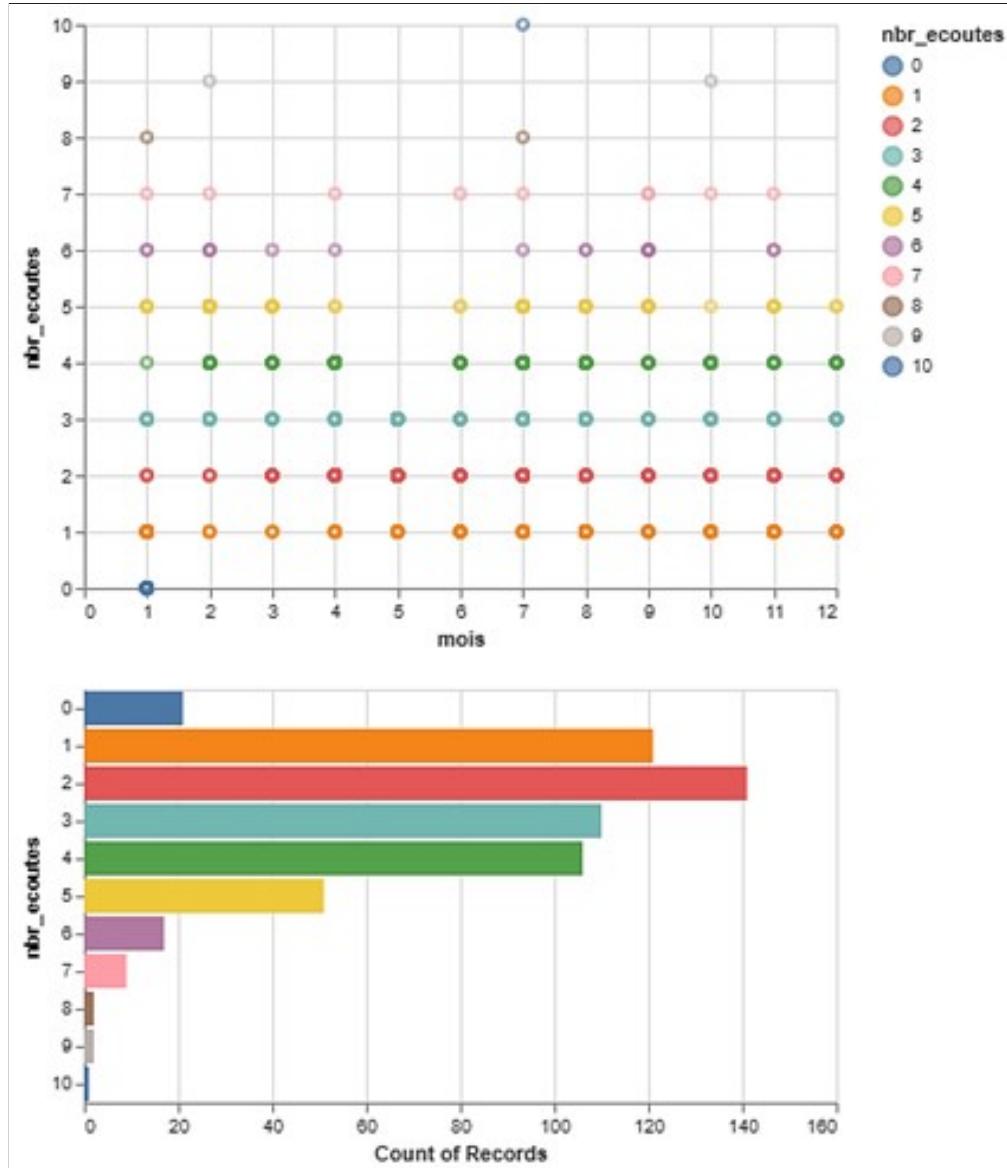


Figure 2.24: Distribution des écoutes sur l'année

sont efficaces dans le traitement des lacunes des agents, soit les écoutes dans ces mois-là ne sont pas suffisantes, résultant en une détection insuffisante des besoins en formations. Les courbes qui présentent deux pics peuvent suggérer que de nouveaux recrutements ont eu lieu.

2.5.4 Visualisation des corrélations entre les variables

La matrice 2.26 de corrélation montre que les formations 1, 2, 3 et 4 sont généralement corrélées avec les variables de non-conformité NC.

- Corrélation de 0.71 entre la variable NC_tobj et la variable formation1 ;
- Corrélation de 0.59 entre la variable NC_clotapp et la variable formation2 ;

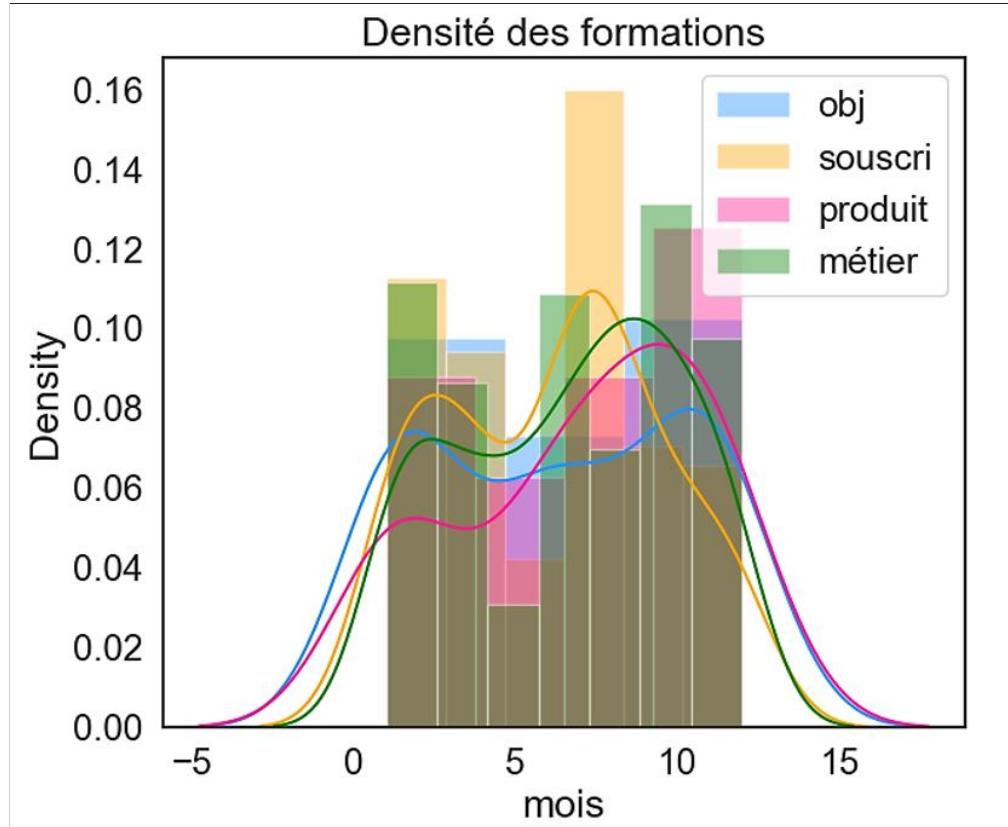


Figure 2.25: Densité des formations sur les mois de l'année

- Corrélation de 0.67 entre la variable NC_exptar et la variable formation3 ;
- Corrélation de 0.56 entre la variable M_hes et la variable formation4.

Sachant que :

- **Formation1** : formation en traitement des objections.
- **Formation2** : formation en souscription.
- **Formation3** : formation produit.
- **Formation4** : formation métier.

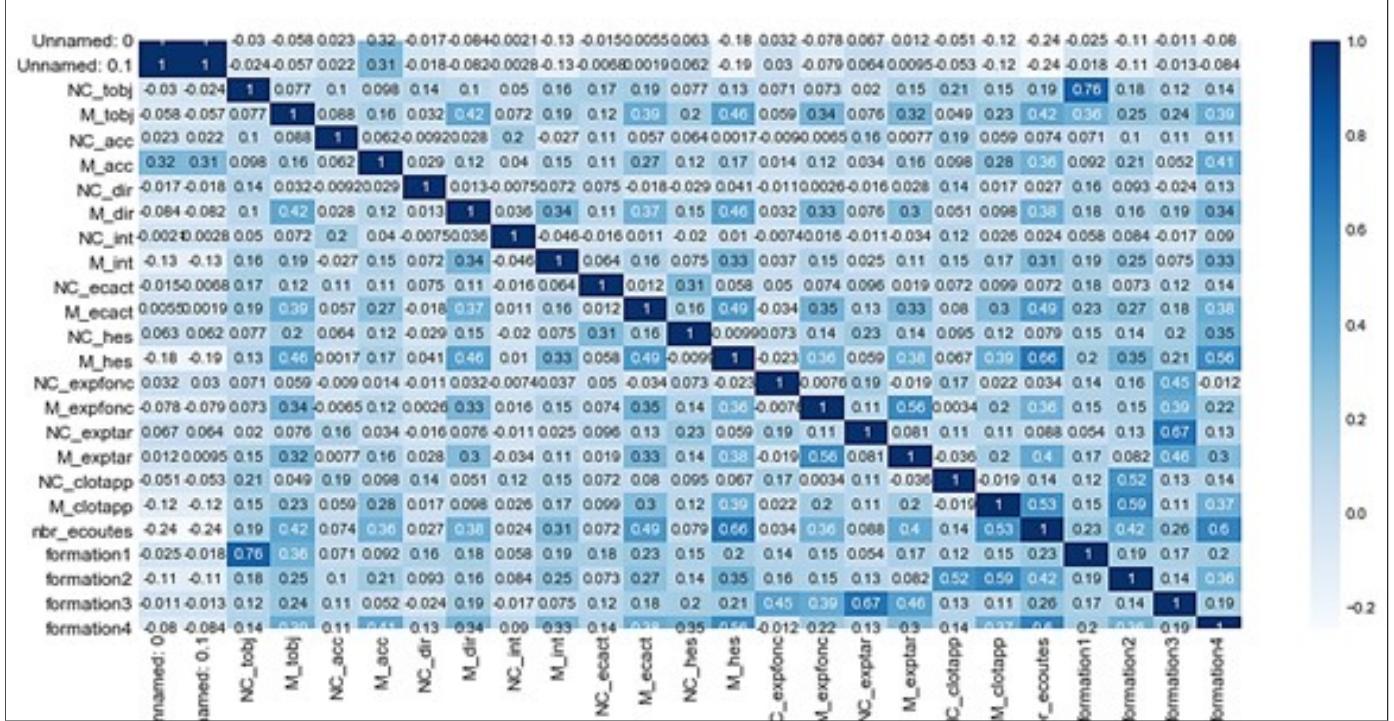


Figure 2.26: Matrice de corrélation de notre Data Set

Par consequent nous concluons que :

- La compétence traitement des objections est définitive dans l'affectation des agents à la formation en traitement des objections.
- La compétence qualité de clôture d'appel est définitive dans l'affectation des agents à la formation en souscription.
- La compétence explication des tarifs est définitive dans l'affectation des agents à la formation produit.
- La compétence temps mort et hésitation est définitive dans l'affectation des agents à la formation métier.

SPRINT3 : Modélisation et évaluation

Dans ce sprint, nous avons essayé de comprendre la technique que nous devons adopter pour modéliser notre problème, en s'adaptant aux particularités de notre Data Set et en identifiant les algorithmes d'apprentissage automatiques qui y correspondent le mieux. Ensuite, nous avons ajusté les modèles utilisés grâce au Tuning des hyperparamètres. Enfin, nous avons évalué tous les modèles et retenu le meilleur en terme de mesures de performances.

2.6 La modélisation

La modélisation est l'étape de la méthodologie CRISP-DM que la plupart des Data scientistes préfèrent le plus. Les données sont maintenant bien structurées et prêtes à être modélisées en vue de résoudre la problématique. Dans cette phase, le praticien doit sélectionner les techniques de modélisation et les algorithmes à essayer, générer une conception de test pour diviser la Data Set en une partie pour l'apprentissage et une partie pour le test, et enfin construire le modèle.

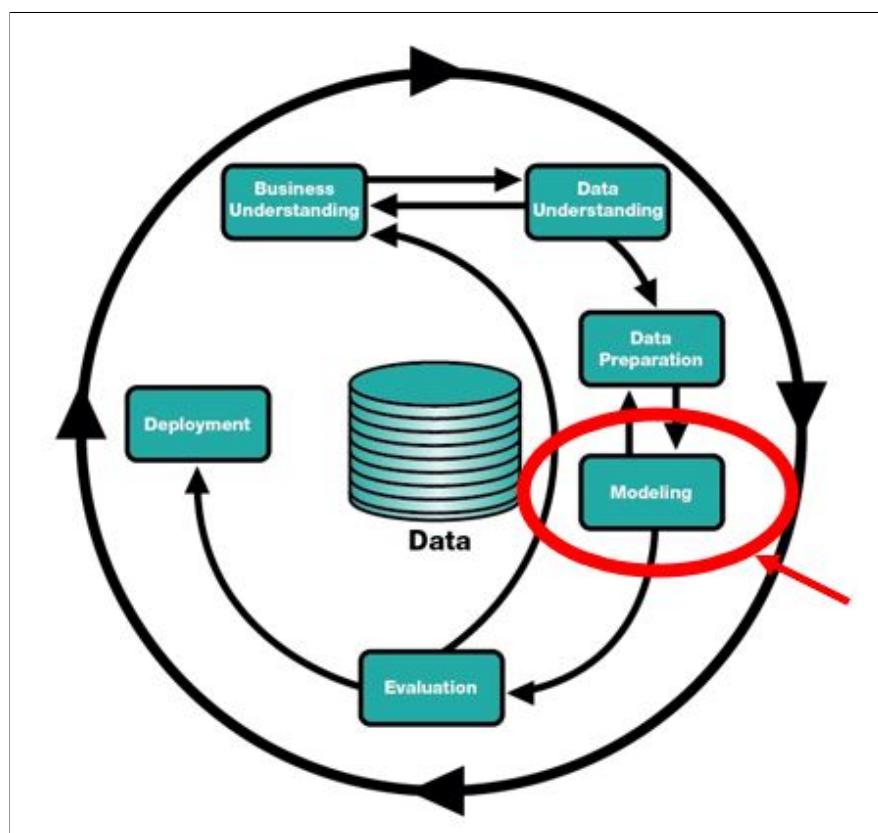


Figure 2.27: La modélisation dans CRISP-DM

2.6.1 Choix de la technique de prédition adéquate

Afin de choisir la technique de prédition adéquate à son problème, il faut se poser les questions suivantes :

- **Les performances de calcul sont-elles un problème ?**

Si oui, il est préférable de :

- Réduire la dimensionnalité.
- Utiliser des algorithmes peu couteux.
- Sélectionner seulement les attributs nécessaires à la prédition.
- Choisir des algorithmes appelés « Lazy Learners » comme KNN.

- **Quel est le type de ma variable cible ?**

Le type de la variable cible est presque définitif dans le choix de la technique de prévision pour un problème d'apprentissage automatique. En effet, il est largement reconnu que :

- Quand la variable à prédire est continue : Il s'agit d'un problème de régression.
- Quand la variable à prédire est catégoriale (nominale) : Il s'agit d'un problème de classification, qui est notre cas.
- Quand la variable à prédire est ordinaire : Il s'agit d'un problème de classification classée.
- Pas de variable à prédire, le but est de trouver une structure dans les données : Il s'agit d'un problème de clustering, Projection.

- **Est-ce que les données sont linéairement séparables ?**

La réponse à cette question est dure à connaître en amont. Pour remédier à cette contrainte d'incertitude, il est préférable de tester plusieurs modèles d'apprentissage automatique et de faire une étude comparative pour en ressortir celui qui s'ajuste le mieux à la Data Set.

- **Quelle est la taille des données ?**

Certaines Data Sets sont très larges et ne peuvent pas être stockées dans la mémoire de l'ordinateur. Dans ces cas-là, il faut utiliser :

- L'apprentissage hors noyau.
- Les systèmes distribués.

2.6.2 Une base de donnée déséquilibrée

Comme l'a montré la visualisation de données, notre Data Set est « Imbalanced » qui est le mot anglais pour déséquilibrée. Dans notre cas, nous avons beaucoup plus d'échantillons avec l'output 0 qu'avec l'output 1. Pour résoudre ce problème, nous avons implémenté la fonction suivante : `compute_sample_weight(class_weight, y, *, indices=None)`

Cette fonction va estimer le poids de chaque point de donnée, en considérant le déséquilibre mentionné. D'autant plus, dans la phase d'évaluation, nous allons nous concentrer sur une mesure de performance différente que celle des problèmes de classification non déséquilibrés. Nous allons expliquer cela plus en détail dans le sprint suivant.

2.6.3 Le Tuning des hyperparamètres

Les modèles d'apprentissage automatique résolvent des problèmes de prédiction et de classification des données. Ces modèles sont paramétrés pour que leur comportement soit ajusté le meilleur possible afin d'obtenir une meilleure performance. De ce fait, le terme « tuning » est le mot anglais pour ajustement.

- **Les paramètres :**

Un paramètre dans un modèle d'apprentissage automatique est une variable de configuration interne très importante dont la valeur peut être estimée à partir des données historiques d'apprentissage. Cette variable ne peut pas être entrée manuellement par le praticien. Quand le modèle a un nombre fixé de paramètres nous l'appelons modèle paramétré. Dans le cas contraire, nous l'appelons non paramétré [7]. Quelques exemples de paramètres :

- Les poids « weights » dans les réseaux de neurones.
- Les « support vectors » dans SVM.
- Les coefficients dans une régression linéaire ou régression logistique.

- **Les Hyperparamètres :**

Les Hyperparamètres sont des paramètres qui peuvent être ajustés afin d'améliorer la performance d'un modèle d'apprentissage automatique. Ce sont des paramètres externes au modèle, souvent définis à l'aide de l'heuristique, et dont la valeur ne peut pas être estimée à partir des données. L'importance de ces Hyperparamètres est à l'origine du concept de Hyperparamètres Tuning. C'est le processus de détermination de la meilleure combinaison d'Hyperparamètres qui permet

au modèle de maximiser sa performance [12].

- **Les stratégies du Tuning :**

Il existe plusieurs stratégies pour faire le Tuning des hyperparamètres, dont : Grid search, Random search et Cross validation. Dans la méthode Grid Search, nous créons une grille de valeurs possibles pour les hyperparamètres. Chaque itération essaie une combinaison d'hyperparamètres dans un ordre spécifique. Le modèle d'apprentissage est ajusté avec chaque combinaison et enregistre à chaque fois la performance. Le résultat retourné par Grid Search est le modèle d'apprentissage automatique avec les meilleurs paramètres. Nous avons choisi cette méthode pour le Tuning de nos algorithmes parce qu'elle est la plus adéquate en termes de taille de données.

2.6.4 La première méthode de classification multi label : La transformation du problème

La classification multi label est différente de la classification classique, puisqu'il existe plusieurs variables cibles, comme le cas de notre problématique. En effet, il existe trois techniques pour résoudre un problème de classification multi label : Les algorithmes adaptés, la transformation du problème et les approches ensemblistes. Dans la méthode de transformation du problème, nous essayons de transformer notre problème multi label en plusieurs problèmes à label unique. Cette technique s'applique avec trois façons : Binary Relevance, Classifier Chains, et Label Powerset.

- **Binary Relevance :**

La méthode Binary Relevance (Pertinence Binaire) est la méthode la plus simple de transformation de problème pour la résolution d'une classification Multi Label. Cette technique fonctionne en décomposant la tâche d'apprentissage multi label en un certain nombre de tâches d'apprentissage distincts pour un problème de classification unique. Le seul aspect négatif de cette méthode est qu'elle ne considère pas la corrélation entre les Labels puisqu'elle traite chaque variable cible indépendamment.

X	Y ₁	Y ₂	Y ₃	Y ₄
x ⁽¹⁾	0	1	1	0
x ⁽²⁾	1	0	0	0
x ⁽³⁾	0	1	0	0
x ⁽⁴⁾	1	0	0	1
x ⁽⁵⁾	0	0	0	1

X	Y ₁
x ⁽¹⁾	0
x ⁽²⁾	1
x ⁽³⁾	0
x ⁽⁴⁾	1
x ⁽⁵⁾	0

X	Y ₂
x ⁽¹⁾	1
x ⁽²⁾	0
x ⁽³⁾	1
x ⁽⁴⁾	0
x ⁽⁵⁾	0

X	Y ₃
x ⁽¹⁾	1
x ⁽²⁾	0
x ⁽³⁾	0
x ⁽⁴⁾	0
x ⁽⁵⁾	0

X	Y ₄
x ⁽¹⁾	0
x ⁽²⁾	0
x ⁽³⁾	0
x ⁽⁴⁾	1
x ⁽⁵⁾	1

Figure 2.28: Fonctionnement de la Binary Relevance

- **Classifier chains :**

Cette approche implique la liaison des classificateurs binaires prêts à l'emploi dans une structure en chaîne, de sorte que les prédictions d'étiquettes de classe deviennent des fonctionnalités pour d'autres classificateurs. De ce fait, le premier classificateur est formé uniquement sur les données d'entrée. Puis chaque classificateur suivant est entraîné sur l'espace d'entrée et tous les classificateurs précédents de la chaîne. Cette technique est similaire à la méthode Binary Relevance, en ayant l'avantage de préserver la corrélation entre les variables cible grâce à son système de chaînes.

X	y ₁	y ₂	y ₃	y ₄
x ₁	0	1	1	0
x ₂	1	0	0	0
x ₃	0	1	0	0

Classifier 1	Classifier 2	Classifier 3	Classifier 4
X y ₁	X y ₁ y ₂	X y ₁ y ₂ y ₃	X y ₁ y ₂ y ₃ y ₄
x ₁ 0	x ₁ 0 1	x ₁ 0 1 1	x ₁ 0 1 1 0
x ₂ 1	x ₂ 1 0	x ₂ 1 0 0	x ₂ 1 0 0 0
x ₃ 0	x ₃ 0 1	x ₃ 0 1 0	x ₃ 0 1 0 0

Figure 2.29: Fonctionnement des Classifier chains

- **Label Powerset :**

Cette méthode transforme le problème multi-étiquettes en un problème de classification multi-classes mono-étiquette, où les valeurs possibles pour l'attribut de classe transformé sont l'ensemble de sous-ensembles uniques distincts d'étiquettes présentes dans les données d'apprentissage d'origine. L'inconvénient de cette technique est qu'au fur et à mesure que la taille des données d'apprentissage augmente, le nombre des classes s'agrandit. Cela augmente la complexité du modèle et fait diminuer la précision.

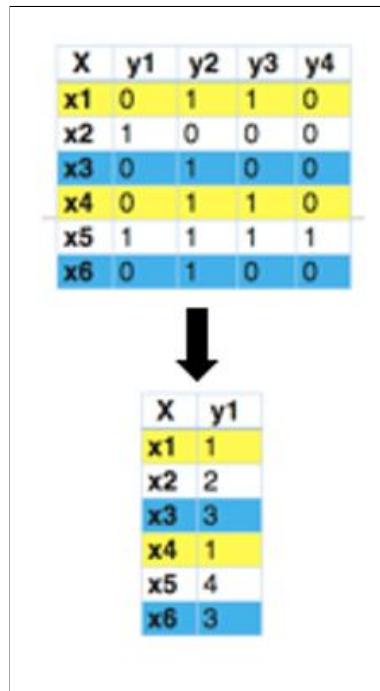


Figure 2.30: Fonctionnement du Label Powerset

2.6.5 La deuxième méthode de classification multi label : Les algorithmes adaptés au problème

La technique des algorithmes adaptés est, comme son nom l'indique, l'adaptation de l'algorithme pour l'appliquer directement sur un problème de classification multi label, au lieu de procéder par la transformation du problème en le décomposant en plusieurs classifications mono-label. La librairie Sci-kit learn fournit une prise en charge intégrée de la classification multi-label dans certains algorithmes tels que Random Forest et Ridge Regression.

2.6.6 La troisième méthode de classification multi label : Les approches ensemblistes

Les méthodes ensemblistes produisent toujours de meilleurs résultats. La bibliothèque Scikit-Multilearn fournit différentes fonctions de classification d'assemblage, qu'il est possible d'utiliser pour obtenir de meilleurs résultats.

Nous expliquerons ci-après les modèles que nous avons utilisé dans chaque méthodologie de classification multi label.

2.6.7 Les modèles d'apprentissage automatique utilisés

Nous détaillons dans cette section les modèles d'apprentissage automatique utilisés.

- **Premier modèle utilisé : La régression logistique**

En apprentissage automatique, la régression logistique est un type de modèle de classification paramétrique. Cela veut dire que ce modèle a un nombre fixe de paramètres qui dépend du nombre des attributs passés en input. Le résultat de ce modèle est une prédiction catégoriale. La régression logistique est très similaire à la régression linéaire. Au lieu de directement ajuster nos données sur une ligne droite, nous les ajustons dans une courbe sous la forme S appelée sigmoïde.

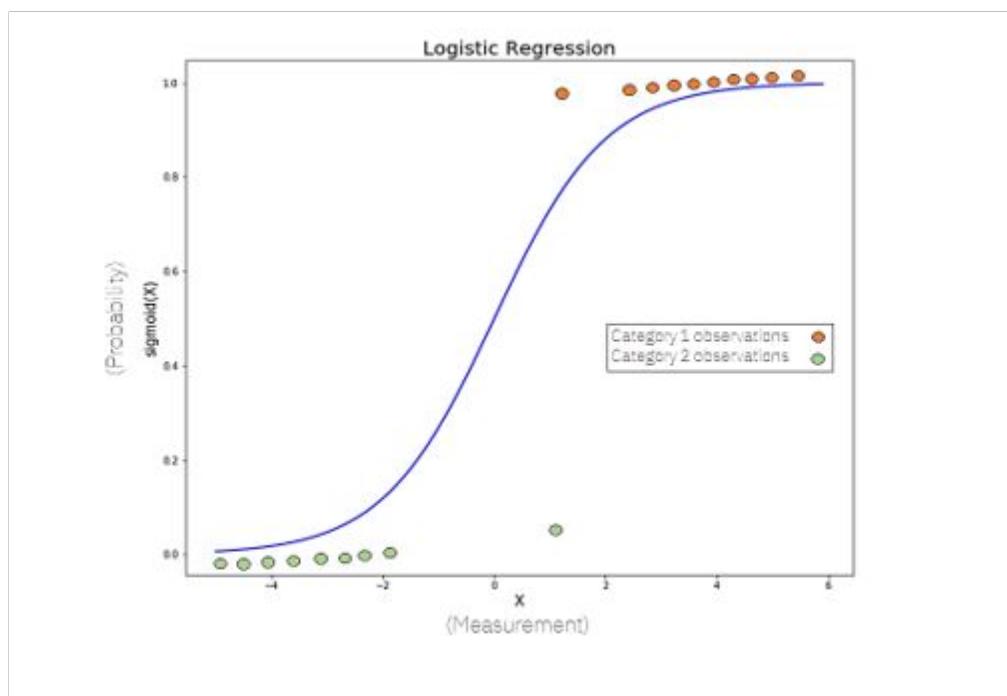


Figure 2.31: Sigmoïde de régression logistique

Tout d'abord, les modèles de régression logistique sont des modèles d'apprentissage automatique de classification, qui distinguent entre deux catégories seulement, dans lesquelles nos observations vont être classées. Comme nous pouvons le voir dans la figure, les valeurs sur l'axe Y sont dans l'intervalle $[0 \dots 1]$. La fonction sigmoïde prend toujours des valeurs entre 0 et 1, expliquant la raison pour laquelle notre modèle de classification entre deux catégories s'ajuste très bien à la courbe de cette fonction Sigmoïde. En calculant la fonction sigmoïde d'une observation X, nous obtenons une probabilité comprise entre 0 et 1 que cette observation appartienne à l'une des deux catégories. La formule de la fonction sigmoïde est la suivante :

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

Figure 2.32: La fonction sigmoïde

- **Deuxième modèle utilisé : XGBoost Classifier**

XGBoost est l'abréviation de eXtreme Gradient Boosting. C'est une implémentation open-source de l'algorithme « gradient boosted trees ». C'est un modèle très populaire de par sa force de prédiction et sa facilité d'utilisation. C'est un algorithme d'apprentissage automatique supervisé pouvant être utilisé pour des fins de régression ou de classification. Pour comprendre cet algorithme, il faut tout d'abord être familier avec les « arbres de décisions », et le « Gradient Boosting ».

- **Decision Tree :** L'algorithme des arbres de décision appartient à la famille des algorithmes d'apprentissage automatique supervisé. Cet algorithme peut être utilisé pour la régression et la classification.

Chaque nœud de l'arborescence agit comme un cas de test pour un attribut, et chaque arête descendant du nœud correspond aux réponses possibles au cas de test. Ce processus est de nature récursive et est répété pour chaque sous arbre enraciné au nouveau nœud. Les arbres de décisions utilisent beaucoup d'algorithmes pour décider de diviser un nœud en deux ou plusieurs sous-nœuds. La création de sous-nœud augmente l'homogénéité des sous-nœuds résultants. Nous pouvons dire que la pureté du nœud augmente avec le respect de la variable cible. L'arbre de décision divise les nœuds sur toutes les variables disponibles et sélectionne ensuite la division qui donne les sous-nœuds les plus homogènes.

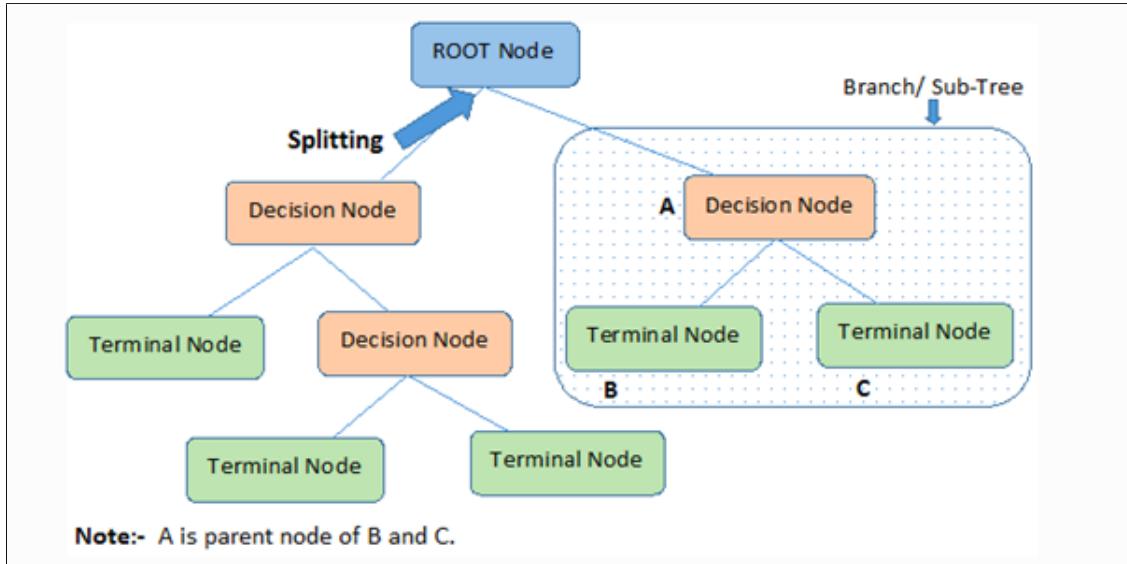


Figure 2.33: Fonctionnement Arbres de décision

- **Gradient boost :** C'est une approche ensembliste, qui combine les prédictions résultantes de plusieurs algorithmes, en prenant chaque prédicteur séquentiellement et en le modélisant en se basant sur l'erreur de son prédécesseur. Le gradient boosting minimise la fonction de pertes en utilisant l'algorithme Gradient descent. Maintenant que nous avons une idée sur le fonctionnement des arbres de décisions et du gradient boosting, nous pouvons comprendre XGBoost : C'est un algorithme qui utilise les arbres de décisions comme ses prédicteurs « faibles ».

- **Troisième modèle utilisé : SVM**

SVM est un algorithme d'apprentissage automatique supervisé qui utilise les algorithmes de classification entre deux groupes. Pour comprendre le fonctionnement de cet algorithme, il vaut mieux suivre un exemple. Imaginons que nous avons deux étiquettes : rouges et bleues. Nos données ont deux attributs : x et y . Nous voulons un classificateur qui selon une paire de (x, y) nous dit si l'output est bleu ou rouge. L'algorithme SVM prend ces points de données et retourne l'Hyperplan (qui est simplement une ligne dans un espace deux dimensions) qui sépare les étiquettes.

C'est la limite de décision, les étiquettes se situant d'un côté sont classifiées bleues, les autres rouges [8].

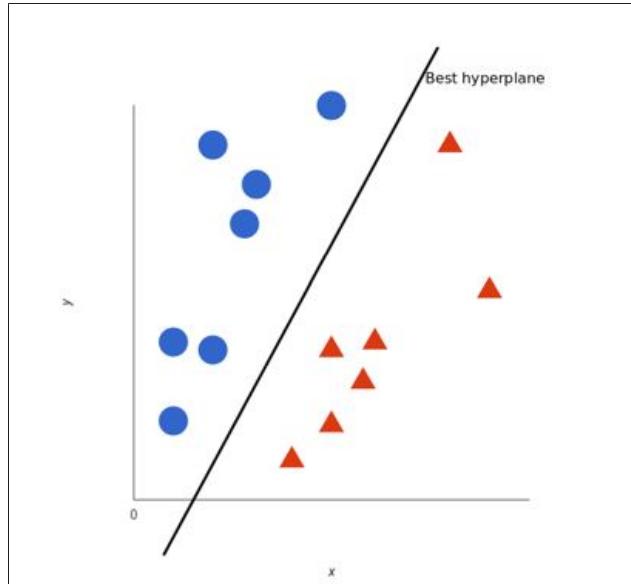


Figure 2.34: SVM dans un espace 2D

- **Quatrième modèle utilisé : KNN (BRKNN ET MLKNN)**

L'algorithme KNN classe les points de données en se basant sur ceux qui leur sont le plus similaires. Dans la figure qui suit, le point x est comparé aux points les plus proches et les plus similaires. La distance entre le point x et le point le plus proche du groupe rouge, du groupe vert et du groupe bleu est calculée.

L'algorithme KNN classe donc le point x dans le groupe le plus proche [9].

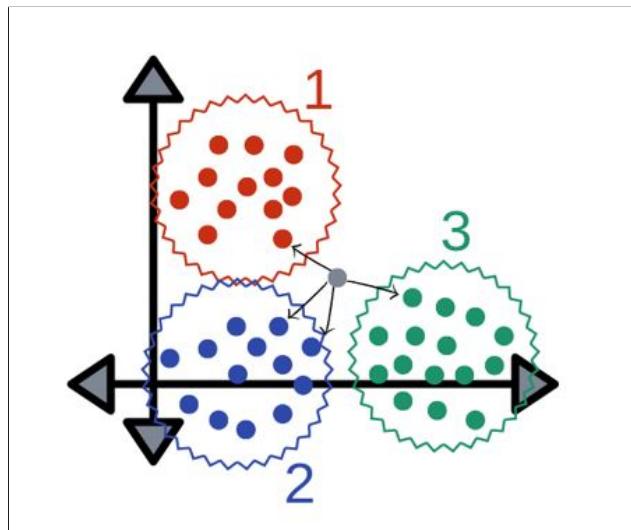


Figure 2.35: Fonctionnement de KNN

- **MLKNN :** ML-KNN est un package pour apprendre les classificateurs de voisins k-plus proches multi-étiquettes. Le package comprend le code MATLAB de l'algorithme ML-KNN, qui est conçu pour gérer l'apprentissage multi-étiquettes. Il est particulièrement utile

lorsqu'un objet du monde réel est associé à plusieurs étiquettes simultanément.

- **BRKNN** : c'est un classificateur multi-étiquettes de pertinence binaire basé sur la méthode k-Nearest Neighbors.
- **Cinquième modèle utilisé : Random Forest** Les forets aléatoires est un algorithme d'apprentissage automatique supervisé, utilisé dans des problèmes de classification et de régression. Cet algorithme construit un ensemble d'arbres de décision et les fusionne ensemble pour obtenir une prédiction plus stable et précise. Il fonctionne avec la méthode « bagging » : une approche ensembliste qui améliore le résultat obtenu. L'algorithme Random Forest ajoute un caractère

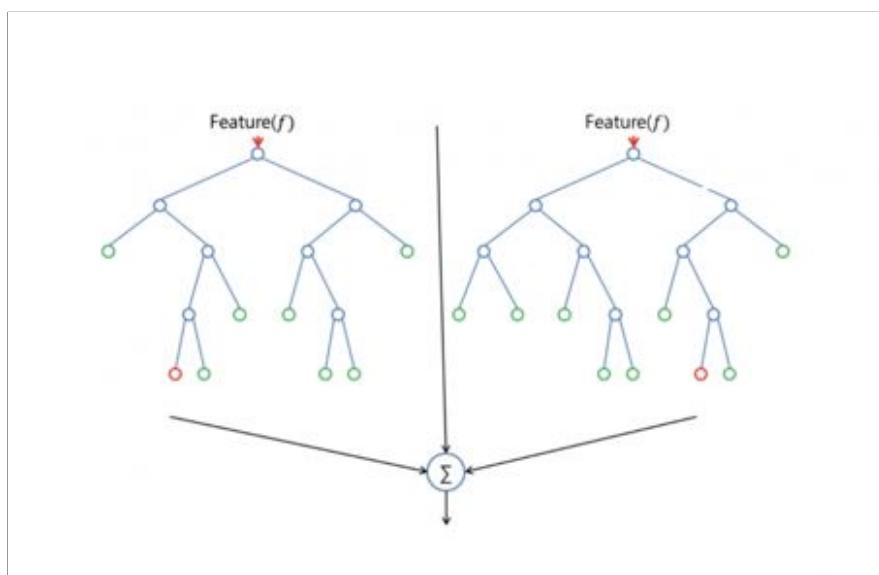


Figure 2.36: Fonctionnement des Random Forests

aléatoire au modèle. Au lieu de rechercher la fonctionnalité la plus importante lors de la division d'un noeud, il recherche la meilleure parmi un sous ensemble aléatoire de fonctionnalités. Il en résulte une grande diversité qui aboutit généralement à un meilleur modèle et résultat.

- **Sixième modèle utilisé : Gaussian Naïve Bayes**

Naïve Bayes est un groupe d'algorithmes d'apprentissage automatique supervisé basé sur le théorème de Bayes. C'est une technique simple de classification, mais qui est très fonctionnelle et performante. Ces algorithmes sont particulièrement utilisés quand la dimensionnalité des données est importante. Même les problèmes de classification complexes peuvent être implémentés et résolus grâce au classificateur Naïve Bayes [10].

— Le théorème de Bayes :

Le théorème de Bayes est utilisé pour calculer la probabilité conditionnelle. Sa qualité de

haute performance lui permet d'être utilisé dans l'apprentissage automatique. La formule du théorème de Bayes est la suivante : Avec :

$$P(A|B) = \frac{p(A \cup B)}{P(B)} = \frac{P(A).P(B|A)}{P(B)}$$

Figure 2.37: Calcul de probabilité avec Naïve Bayes

- (A)=La probabilité qu'A se produise.
- (B)= La probabilité que B se produise.
- (AB)= La probabilité qu'A se produise sachant B.
- (B|A)= La probabilité que B se produise sachant A.
- (AB)= La probabilité que A et B se produisent.

— **Naïve Bayes Classifier :**

Le classificateur Naïve Bayes est basé sur le théorème de Bayes. Ce classificateur assume que la valeur d'un attribut ou « feature » particulier est indépendante de la valeur de n'importe quel attribut. Dans un contexte d'apprentissage automatique supervisé, le classificateur Naïve Bayes est entraîné, et nécessite des données d'apprentissage de taille petite pour estimer les paramètres requis pour la classification. La conception de ce classificateur ainsi que son implémentation sont faciles et peuvent être appliqués dans des situations réelles.

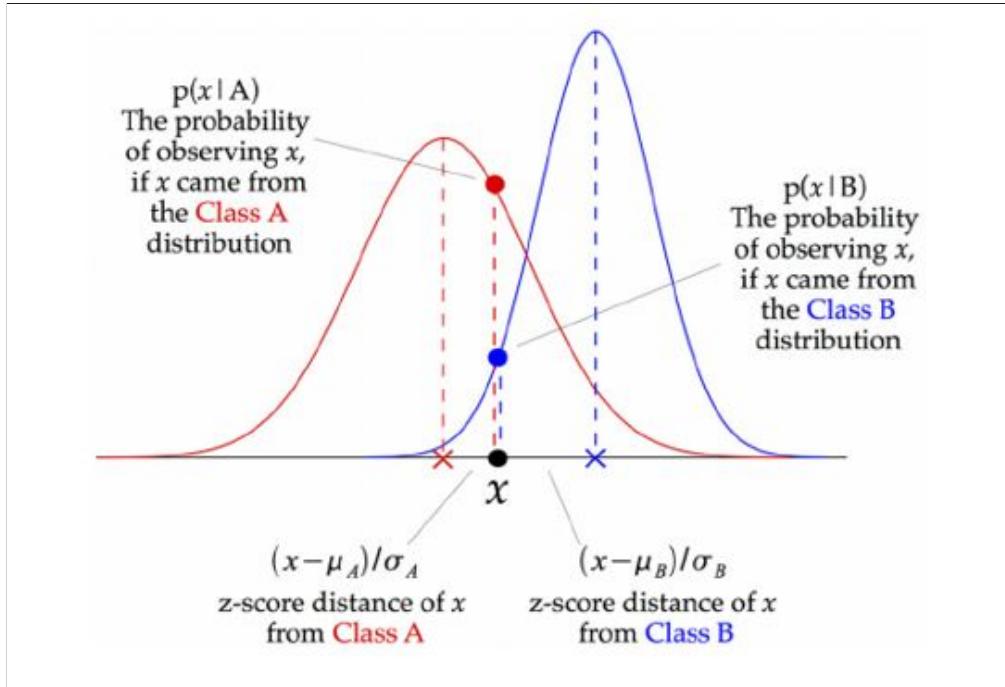


Figure 2.38: Fonctionnement du Naïve Bayes

A chaque instance, la distance z-score est calculée : la distance entre le point x et la moyenne de chaque classe.

2.6.8 Synthèse

Nous allons résumer l'étape de modélisation dans les tableaux suivants, détaillant chaque algorithme utilisé avec les Hyperparamètres ajustés (Tuning). Chaque tableau implique une méthode de classification multi label parmi les trois méthodes expliquées ci-dessus.

Première méthode : La transformation du problème

Ce tableau regroupe les modèles de transformation du problème ajustés grâce au Tuning de leurs hyperparamètres.

Tableau 2.3: les algorithmes de transformation de problème avec leurs hyperparamètres.

Méthode	Modèle	Hyperparamètres
Classifier chains	Gaussian Naïve Bayes classifier=GaussianNB()	priors = None var_smoothing = 1e-08
	XGBoost Classifier	max_depth = 5 n_estimators= 100
	Logistic Regression	solver = newton-cg penalty = L2 c_values= 100
	SVM	C = 10
	Decision Trees	splitter = best max_depth = None min_samples_split=2 max_features = None

Binary Relevance	Gaussian Naïve Bayes	priors = None var_smoothing = 1e-08
	XGBoost Classifier	max_depth = 5 n_estimators= 100
	Logistic Regression	solver = liblinear penalty = L2 c_values= 100
	SVM	kernel = rbf C = 50
	Decision Tree	splitter = best max_depth = None min_samples_split=2 max_features = None

LabelPowerset	Gaussian Naïve Bayes	priors = None var_smoothing = 1e-08
	XGBoost Classifier	max_depth=3
	Logistic Regression	solver = lbfgs penalty = L2 c_values= 10
	SVM	C = 10
	Decision Tree	splitter = best max_depth = None

Explication des hyperparamètres :

- **Gaussian Naïve bayes :**

Priors : Probabilités antérieures des classes. Si spécifié, les priors ne sont pas ajustés en fonction des données.

Var_smoothing : Portion de la plus grande variance de toutes les entités qui est ajoutée aux variances pour la stabilité des calculs.

- **XGBoost classifier :**

max_depth : la profondeur maximale d'un arbre.

n_estimators : le nombre d'arbres.

- **Logistic regression :**

Solvers : Algorithme à utiliser dans le problème d'optimisation. **_Penalty** : Utilisé pour

spécifier la norme utilisée dans la pénalisation. **C_values** : Inverse de la force de régularisation.

Comme dans SVM, des valeurs plus petites indiquent une régularisation plus forte.

- **Decision trees :**

Splitter : La stratégie utilisée pour choisir la division à chaque nœud. Les stratégies prises en charge sont « meilleures » pour choisir la meilleure division et « aléatoire » pour choisir la meilleure division aléatoire.

Max_depth : La profondeur maximale de l'arbre. Si None, les nœuds sont développés jusqu'à ce que toutes les feuilles soient pures ou jusqu'à ce que toutes les feuilles contiennent moins de min_samples_split échantillons.

Min_samples_split : Le nombre minimum d'échantillons requis pour fractionner un noeud interne.

Max_features : Le nombre de fonctionnalités à prendre en compte lors de la recherche de la meilleure répartition

- **SVM :**

Kernel : Spécifie le type de « kernel » à utiliser dans l'algorithme. Le kernel fait référence à une méthode de transformation des données sous la forme nécessaire. **C** : paramètre de régularisation.

Deuxième méthode : les algorithmes adaptés au problème

Ce tableau regroupe les modèles des algorithmes adaptés ajustés grâce au Tuning de leurs hyperparamètres.

Tableau 2.4: La méthode des algorithmes adaptés avec leurs hyperparamètres ajustés.

Modèle	Hyperparamètres
MLKnn	k=6 s =0.5
BR knn	k=3

Troisième méthode : les approches ensemblistes

Ce tableau regroupe les modèles des méthodes ensemblistes ajustés grâce au Tuning de leurs hyperparamètres.

Tableau 2.5: Les méthodes ensemblistes avec leurs hyperparamètres.

Modèle	Hyperparamètres
Random Forest	n_estimators= 100 max_features= auto max_depth= 300 min_sample_split= 5 min_sample_split_leaf = 1 bootstrap= false

2.7 L'évaluation

Dans les phases précédentes du CRISP-DM, nous avons exploré et préparé nos données, puis nous les avons modélisées à travers différents algorithmes et suivant toutes les méthodes de classification multi label. Dans cette phase, nous allons répondre aux questions suivantes : Les modèles que nous avons construits, sont-ils performants ? répondent-ils à l'objectif final de notre projet ? Pour cela, nous devons faire une évaluation détaillée de notre travail à partir d'une étude comparative de tous les algorithmes implémentés, pour en ressortir le meilleur en termes de performance et d'efficacité.

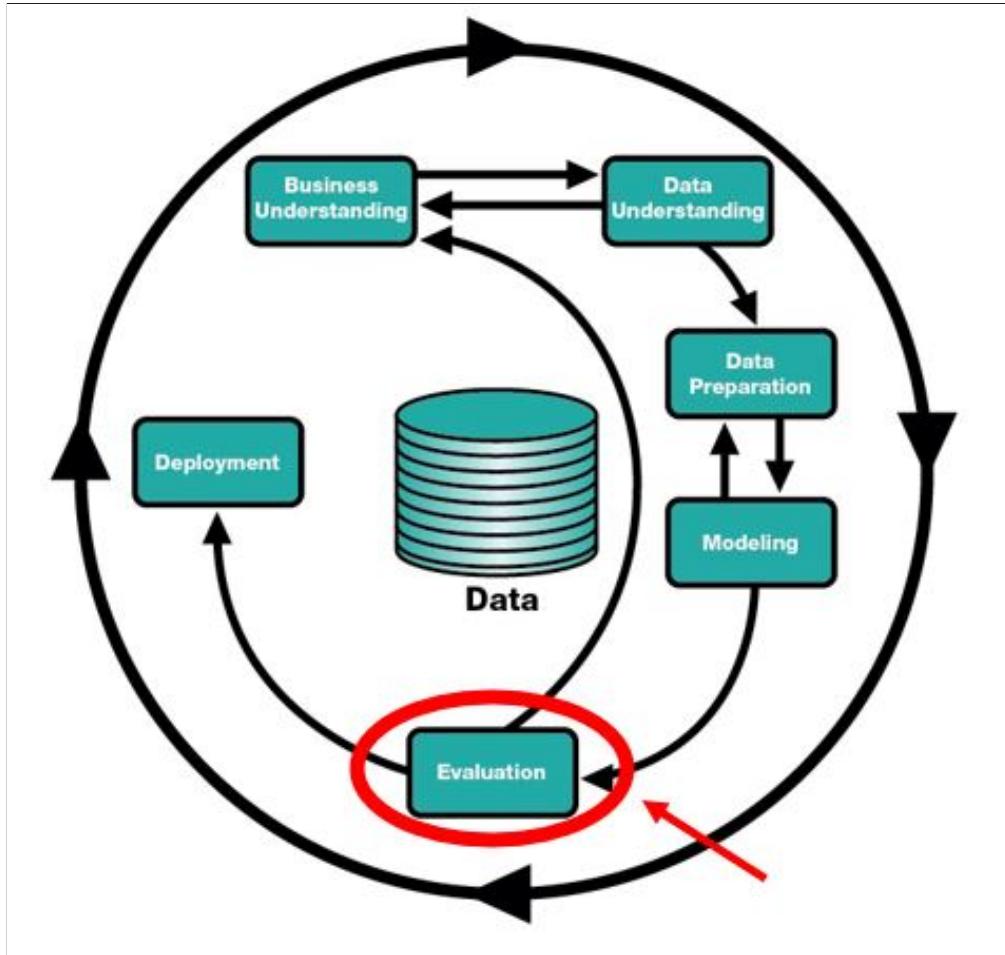


Figure 2.39: L'évaluation dans la méthodologie CRISP-DMs

2.7.1 Accuracy de classification

C'est le quotient du nombre des prédictions correctes sur le nombre total des prédictions [11].

$$\text{Accuracy} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total des prédictions}}$$

Figure 2.40: Calcul de la mesure de performance "Accuracy"

NB : Cette mesure de performance fonctionne très bien lorsque la base de données est équilibrée. Dans le cas contraire, elle sera biaisée et retournera un résultat centré sur la prédiction de la classe majoritaire. Dans notre cas, nous avons 4 labels : Formation1, formation2, formation3 et formation4. Prenons comme exemple le premier label formation1 et visualisons la fréquence des classes 0 et 1. L'histogramme de fréquence 2.41 montre que sur nos 581 lignes de données, environ 50 échantillons sont classifiés dans la classe positive 1.

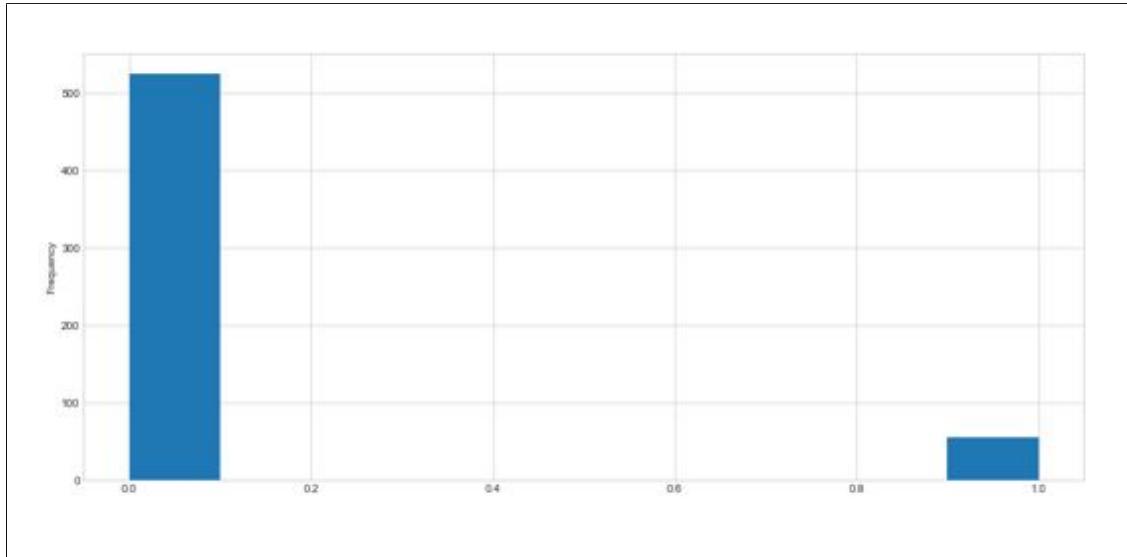


Figure 2.41: Histogramme de fréquence du Label formation1

Pourcentage de la classe positive = $50/581*100 = 8.6\%$

Cette classe est en infériorité numérique très importante, auquel cas la mesure de performance « Accuracy » n'est pas une bonne mesure pour évaluer le modèle correctement.

2.7.2 Précision

La précision est définie comme le nombre de vrais positifs divisé par le nombre de vrais positifs plus le nombre de faux positifs. Les faux positifs sont les cas où le modèle classifie un échantillon comme étant positif incorrectement. La précision exprime donc la proportion des points de données que notre modèle considère comme pertinents, et qui sont réellement pertinents [12].

$$\text{Précision} = \frac{\text{Vrais positifs}}{\text{vrais positifs} + \text{faux positifs}}$$

Figure 2.42: Calcul de la mesure de performance "Précision"

2.7.3 Rappel

La définition du Rappel (le terme anglais est Recall) est le nombre de vrais positifs divisé par le nombre de vrais positifs plus le nombre de faux négatifs. Les vrais positifs sont les échantillons classifiés par le modèle comme étant positifs et qui sont réellement positifs. Les faux négatifs sont les échantillons que le modèle classifie comme négatifs mais qui sont réellement positifs [18].

$$Rappel = \frac{Vrais\ positifs}{vrais\ positifs + faux\ négatifs}$$

Figure 2.43: Calcul de la mesure de performance "Rappel"

2.7.4 F1_score

Le F1-score est la moyenne harmonique de la précision et du rappel en tenant compte des deux métriques [18].

$$F1 - score = 2 * \frac{précision * rappel}{précision + rappel}$$

Figure 2.44: Calcul de la mesure de performance F1-score

2.7.5 Zero one loss

Pour calculer la fonction de coût, nous devons trouver la plus adéquate à notre problème de classification multi-label. Bien qu'il existe plusieurs fonctions de coût plus répandues comme Log loss, hamming loss, etc., celles-ci ne correspondent pas aux particularités de notre classification. Nous avons choisi la fonction de coût Zero one loss, implémentée dans la librairie Scikit-learn avec laquelle nous avons construit nos algorithmes. La fonction suivante décrit la mesure de performance Zero one loss avec M les classes de prédiction

$$L(i, j) = \begin{cases} 0 & \text{quand } i = j \\ 1 & \text{quand } i \neq j \end{cases} \quad i, j \in M$$

Figure 2.45: la fonction Zero one loss

2.7.6 Matrice de confusion

La matrice de confusion est une mesure de performance pour les problèmes de classification où l'output peut appartenir à une ou plusieurs classes. C'est une table avec 4 combinaisons différentes de valeurs réelles et valeurs prédictées.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 2.46: La matrice de confusion

- TP= True positives= Vrais positifs ;
- FP= False positives= Faux positifs ;
- FN= False negatives= Faux négatifs ;
- TN= True negatives = Vrais négatifs.

2.7.7 Évaluation des algorithmes utilisés

Comme nous l'avons expliqué ci-dessus, la mesure de performance « Accuracy » n'est pas adéquate pour notre problème, dû au déséquilibre dans notre Data Set. Pour ce genre de situations, il faut en quelque sorte privilégier la classe minoritaire. Intuitivement, nous pensons au Rappel, que nous voulons donc proche de 1. Par contre, au fur et à mesure que le rappel augmente, la précision diminue, ce qui donnera un modèle incapable d'identifier seulement les points de données pertinents. Nous devons donc combiner la précision et le rappel, ce qui fait que la mesure de performance sur laquelle nous allons nous baser dans l'évaluation de notre modèle est le F1-score.

Évaluation des algorithmes de la première méthode de classification multi label : Ce tableau regroupe les mesures de performances pour chaque algorithme de la méthode de transformation du problème.

Tableau 2.6: Évaluation des algorithmes de la méthode de transformation du problème.

Méthode	Algorithm	Accuracy	Precision	Recall	F1-score	Zero one Loss
Binary Relevance	Gaussian Naïve Bayes	0.6228	0.6734	0.5454	0.5902	0.3771
	Logistic Regression	0.8571	0.9137	0.8760	0.91667	0.1428
	XGBoost	0.9542	0.9913	0.9421	0.9605	0.0457
	SVM	0.9257	0.9909	0.9008	0.9427	0.074
Classifier Chains	Gaussian Naïve Bayes	0.6114	0.6320	0.5537	0.6080	0.3885
	Logistic Regression	0.88	0.9237	0.9008	0.9176	0.12
	XGBoost	0.9485	0.9912	0.9338	0.9605	0.0514
	SVM	0.9314	1	0.9008	0.9410	0.0685
Label Powerset	Gaussian Naïve Bayes	0.6171	0.6770	0.5371	0.5852	0.3828
	Logistic Regression	0.8	0.8812	0.8016	0.8700	0.1999
	XGBoost	0.8742	0.98	0.8099	0.9163	0.1257
	SVM	0.88	1	0.8099	0.8795	0.12

Évaluation des algorithmes de la deuxième méthode de classification multi label :

Ce tableau regroupe les mesures de performances pour chaque modèle de la méthode des algorithmes adaptés.

Tableau 2.7: Évaluation des algorithmes adaptés.

Algorithme	Accuracy	Precision	Recall	F1	Zero one Loss
MLKNN	0.7828	0.8425	0.7520	0.7878	0.2171
BRKNN	0.7771	1	0.595	0.7028	0.2228

Évaluation de l'algorithme de la troisième méthode de classification multi label :

Tableau 2.8: Évaluation des méthodes ensemblistes.

Algorithme	Accuracy	Precision	Recall	F1	Zero one Loss
Random Forest	0.9428	1	0.9173	0.951	0.051

2.7.8 Synthèse

Dans ce tableau, nous avons retenu les meilleurs algorithmes pour chaque méthode de classification multi-label.

Tableau 2.9: Synthèse de l'évaluation.

Méthode	Algorithme	Accuracy	Precision	Recall	F1-score	Zero one loss
Transformation du problème	XGBoost	0.9542	0.9913	0.9421	0.9605	0.0457
Algorithmes adaptés	MLKNN	0.7828	0.8425	0.7520	0.7878	0.2171
Méthodes d'ensemble	Random Forest	0.9428	1	0.9173	0.951	0.051

Le meilleur algorithme dans notre cas est celui avec le F1-score le plus élevé. Comme nous

l'avons expliqué précédemment, la mesure de performance « Accuracy » ne sera pas celle que nous adopterons pour notre évaluation, étant donné le caractère déséquilibré de notre Data Set. Comme le montre ce tableau de synthèse, le meilleur algorithme est XGBoost avec un F1-score de 0.9605

Les matrices de confusion du modèle retenu XGBoost classifier :

Comme nous l'avons déjà expliqué, la matrice de confusion est un outil permettant de mesurer les performances d'un modèle de Machine Learning en vérifiant notamment à quelle fréquence ses prédictions sont exactes par rapport à la réalité dans des problèmes de classification. Dans une classification multi label, chaque label a sa propre matrice de confusion. Notre Test Set est constitué de 175 points de données. Dans les matrices de confusion :

N=0 => N'a pas besoin d'une formation

Y=1 => A besoin d'une formation

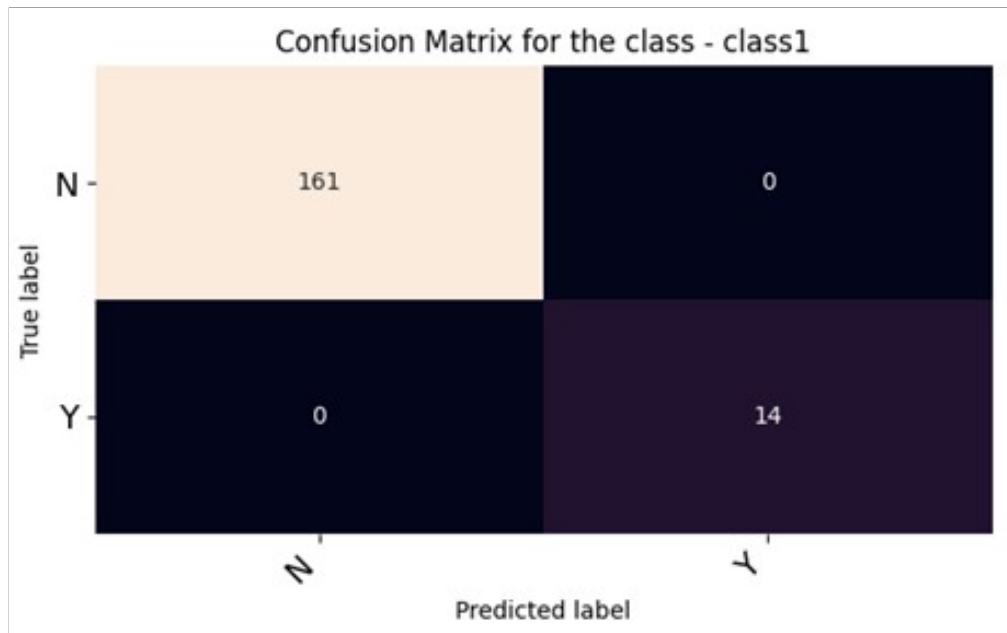


Figure 2.47: Matrice de confusion du premier label formation1

L'algorithme a prédit 161 vrais négatifs et 0 faux négatifs.

L'algorithme a prédit 14 vrais positifs et 0 faux positifs.

14 agents ont été affectés à la première formation correctement.

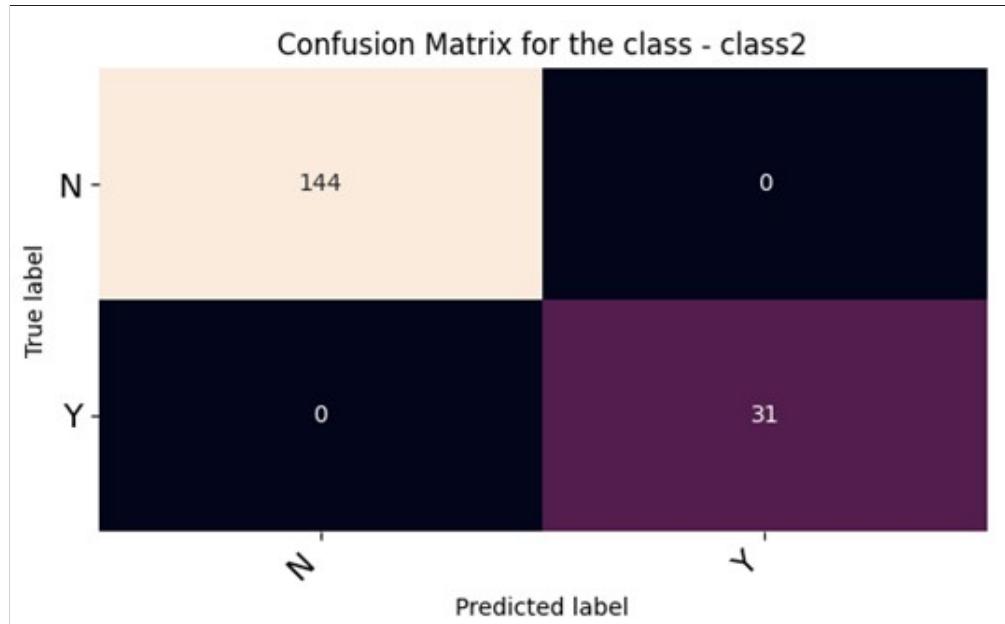


Figure 2.48: Matrice de confusion du deuxième label formation2

L'algorithme a prédit 144 vrais négatifs.

L'algorithme a prédit 31 vrais positifs et 0 faux positifs.

31 agents ont été affectés correctement à la formation 2.

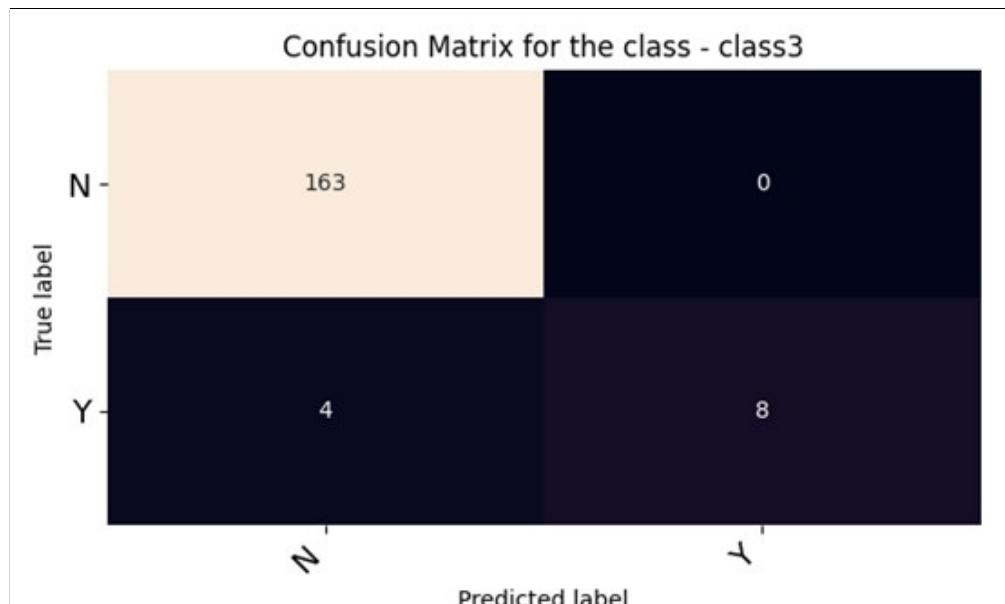


Figure 2.49: Matrice de confusion du deuxième label formation3

L'algorithme a prédit 4 faux négatifs et 163 vrais négatifs.

L'algorithme a prédit 8 vrais positifs et 0 faux positifs.

8 agent ont été affectés à la formation 3 correctement.

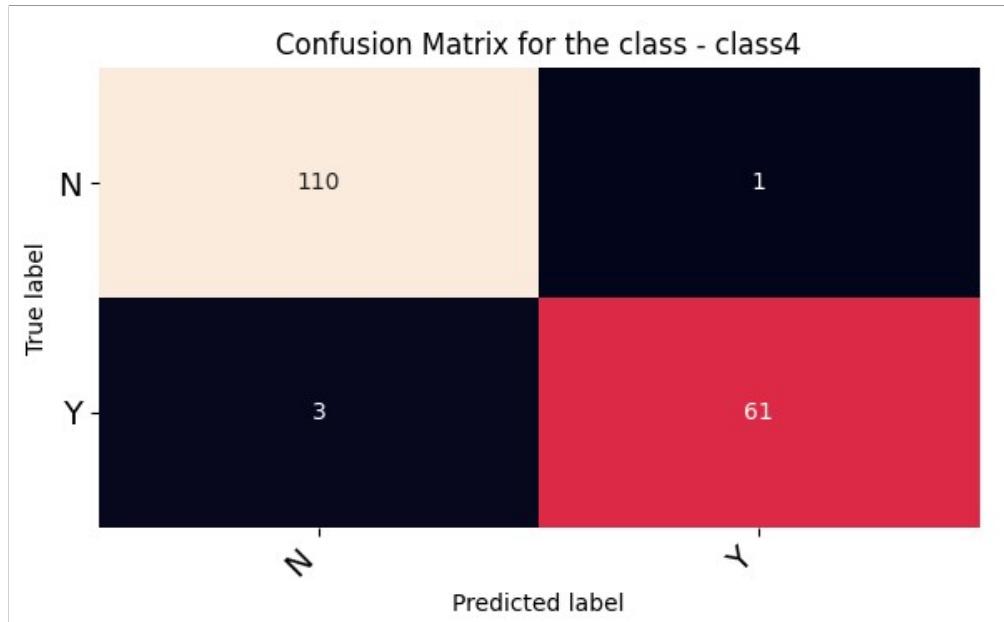


Figure 2.50: Matrice de confusion du deuxième label formation4

L'algorithme a prédit 61 vrais positifs 1 vrai négatif.

L'algorithme a prédit 3 faux négatifs et 110 vrai négatifs

62 agents affectés à la formation 4 dont une fausse affectation.

La raison pour laquelle les résultats sont aussi avantageux est que nous avons utilisé toutes les méthodes de classification multi label, ajusté nos algorithmes grâce au tuning des hyperparamètres et fait une étude comparative entre plusieurs modèles d'apprentissage automatique. Cela nous a permis de construit le modèle qui correspond le plus à notre problème, qui est le classificateur XGBoost opérant avec la méthode de transformation du problème, et ajusté avec la stratégie Grid search pour trouver la meilleure combinaison d'hyperparamètres.

Conclusion

Dans ce chapitre, nous avons construit le premier livrable de notre projet qui est le module d'apprentissage automatique permettant de détecter l'offre de formation adéquate à chaque agent. Ce module de Machine Learning va être déployé par la suite sous la forme d'alertes au superviseur lors d'un besoin de formation pour un agent de son groupe. Il sera également alimenté par une base de données historiques contenant des données plus détaillées. Le chapitre suivant sera consacré au deuxième livrable : la prédiction du nombre de ventes qu'un agent va éventuellement réaliser.

RELEASE 2 "PRÉDICTION DU NOMBRE DE VENTES"

Plan

Introduction	64
SPRINT1 : Compréhension, préparation et visualisation des données	64
1 Compréhension du problème métier	64
2 Compréhension des données	69
3 Agrégation des données	73
4 Nettoyage des données	77
SPRINT2 : : Modélisation et évaluation	85
5 La modélisation	85
6 L'évaluation	90
SPRINT3 : Déploiement	96
7 Analyse des besoins	96
8 Conception	104
9 Déploiement	106
10 Phase de clôture	110
Conclusion	113

Introduction

Introduction Dans ce chapitre, nous allons nous concentrer sur le deuxième livrable de notre projet : un module d'apprentissage automatique pour la prédiction du nombre de ventes réalisées par chaque agent. Ce chapitre a été conduit à travers la méthodologie agile, et divisé sur 3 Sprints différents : Le sprint 1 pour compréhension, la préparation et la visualisation des données. Le sprint 2 pour la modélisation et l'évaluation. Le sprint 3 pour le déploiement.

SPRINT1 : Compréhension, préparation et visualisation des données

Dans ce sprint, nous nous sommes concentrés sur la compréhension, la préparation et la visualisation des données, pour qu'elles soient proprement exploitées dans la modélisation.

3.1 Compréhension du problème métier

Dans cette section, nous allons présenter les notions qu'il faut connaître pour comprendre l'objectif de notre projet.

3.1.1 L'environnement d'intervention

- **La production :** Ce livrable de notre projet s'installe dans le cadre de l'amélioration du suivi de la production des agents de l'entreprise 1WayCom. Les prédictions apportées par notre solution permettront aux superviseurs d'avoir des attentes scientifiques de leurs agents en termes de productivité, pour pouvoir ensuite identifier ceux qui enregistrent un manque de performance. Nous allons expliquer ci-après quelques notions nécessaires à la compréhension de la visée de ce livrable
- **L'objectif principal du centre d'appel :** Le centre d'appel est une plateforme téléphonique dont l'activité clé est de gérer un nombre important d'appels entrants et sortants. L'objectif ultime du centre d'appel 1WayCom est d'augmenter le nombre de réponses favorables aux services qu'il offre, que ce soit dans la prise de rendez-vous ou dans la vente.
- **Les opérations de vente :** Lorsqu'une entreprise déterminée veut augmenter son chiffre d'affaires et développer le trafic autour de son activité, elle peut avoir recours à un centre d'appel pour l'aider à atteindre un maximum de clients. Cette tâche peut représenter pour le centre d'appel une opération de vente. Les agents téléopérateurs de la plateforme sont donc

chargés de réaliser un travail de prospection à distance pour ladite entreprise.

- **Le nombre de ventes d'un agent :** Dans l'entreprise 1WayCom, la mesure de performance la plus importante pour un agent est le nombre de ventes qu'il enregistre à la fin du mois. Cette métrique est synonyme de gain non seulement pour l'entreprise, mais aussi pour l'agent, qui obtient une prime sur objectif.

3.1.2 Détermination des objectifs de l'analyse

La détermination des objectifs de l'analyse est primordiale dans un projet de Data Science. C'est une étape qui va permettre au praticien de comprendre encore plus le travail demandé pour parvenir à une concrétisation adéquate du projet. La définition des variables d'activité clés, du résultat souhaité, des mesures de réussite et des sources de données vont apporter une vision claire de la stratégie à adopter.

3.1.3 Identification des variables d'activité clés

Dans chaque problème de prédiction résolu par un apprentissage automatique, il y'a des variables clés qui sont les variables à prédire. Notre variable cible est le nombre de ventes. Nos variables prédictives vont être définies dans la suite.

3.1.4 Le nombre de ventes prédit

Nous allons essayer de prédire le nombre de ventes qu'un agent devrait réaliser à partir de différents éléments comme le nombre d'heures pour lesquelles l'agent travaille durant un mois déterminé, ses scores dans les feuilles d'écoutes, le nombre de ventes qu'il a enregistré précédemment, etc. Ces critères seront donc nos attributs ou « Features ». Nous sommes ici en présence d'un problème de régression en apprentissage automatique supervisé. Dans notre cas, on veut prédire une seule variable cible qui est le nombre de ventes, à partir de plusieurs variables prédictives. Nous sommes donc en présence d'un problème de **régression multiple**.

3.1.5 Les mesures de réussite

A la fin de notre projet, nous devons livrer un système qui sera en mesure de prédire le nombre de ventes qu'un agent téléopérateur devrait effectuer dans un mois défini. Les superviseurs de l'entreprise pourront donc comparer le rendement de leurs groupes aux attentes scientifiques du système réalisé. Pour cela, le taux d'erreur que nous avons fixé ne doit pas dépasser les **15%**.

Les mesures de réussite du projet se rapportent essentiellement au dernier Sprint de ce chapitre, qui est celui de l'Evaluation.

3.1.6 Identification des sources de données

Notre source de données est exclusivement la base de données de 1WayCom, qui comporte 145 tables.

- La base de données de 1WayCom

<input type="checkbox"/> receptions	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer
<input type="checkbox"/> receptions_sub_categories	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer
<input type="checkbox"/> recrutements	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer
<input type="checkbox"/> recrutement_competences	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer
<input type="checkbox"/> recrutement_sub_competences	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer
<input type="checkbox"/> recrutement_tables	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer
<input type="checkbox"/> recuper	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer
<input type="checkbox"/> recuperations_avances	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer
<input type="checkbox"/> regions	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer
<input type="checkbox"/> renvois	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer
145 tables							
Somme							

Figure 3.1: La base de données de 1WayCom

- Les tables sélectionnées pour notre projet

Nous allons utiliser 12 tables de la base de données de 1WayCom pour la réalisation de notre projet. Ces tables contiennent les informations nécessaires à la prédiction des formations et également la prédiction des ventes.

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> formations	★ Parcourir Structure Rechercher Insérer Vider Supprimer	1 137	InnoDB	utf8mb4_general_ci	176,0 kio	-
<input type="checkbox"/> grid_listen3_categories	★ Parcourir Structure Rechercher Insérer Vider Supprimer	21	InnoDB	utf8_general_ci	16,0 kio	-
<input type="checkbox"/> grid_listen3_stored_calls	★ Parcourir Structure Rechercher Insérer Vider Supprimer	2 592	InnoDB	utf8_general_ci	240,0 kio	-
<input type="checkbox"/> grid_listen3_stored_call_sub_categories	★ Parcourir Structure Rechercher Insérer Vider Supprimer	~50 222	InnoDB	utf8_general_ci	9,5 Mio	-
<input type="checkbox"/> grid_listen3_sub_categories	★ Parcourir Structure Rechercher Insérer Vider Supprimer	100	InnoDB	utf8_general_ci	16,0 kio	-
<input type="checkbox"/> holidays	★ Parcourir Structure Rechercher Insérer Vider Supprimer	21	InnoDB	latin1_swedish_ci	16,0 kio	-
<input type="checkbox"/> mytable	★ Parcourir Structure Rechercher Insérer Vider Supprimer	600	InnoDB	utf8mb4_general_ci	96,0 kio	-
<input type="checkbox"/> planning	★ Parcourir Structure Rechercher Insérer Vider Supprimer	1 136	InnoDB	utf8mb4_general_ci	128,0 kio	-
<input type="checkbox"/> predicton123	★ Parcourir Structure Rechercher Insérer Vider Supprimer	148	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> predictionnbv	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> preventes	★ Parcourir Structure Rechercher Insérer Vider Supprimer	704	InnoDB	utf8mb4_general_ci	112,0 kio	-
<input type="checkbox"/> sales_values	★ Parcourir Structure Rechercher Insérer Vider Supprimer	9 829	InnoDB	utf8_general_ci	3,5 Mio	-
12 tables	Somme	~66 510	InnoDB	utf8mb4_general_ci	13,8 Mio	0 o

Figure 3.2: Tables sélectionnées pour le projet

- **Les tables utilisées dans la Release 2 :** Dans cette parties nous avons utilisé 5 tables qui se résume dans les figures suivantes.

- La table grid_listen3_stored_calls : utilisée pour ressortir le score de l'agent ainsi que sa note.

id	qualification_valeur	qualification_total	qualification_score	selected_scale_class	selected_scale_name	agent_id	agent_supervisor_id	superviseur d'agent lors de	
								trein	dat
1	150	170	88.24%	green	Bien	30	3	17	202
2	0	180	0%	NULL	NULL	14	3	17	202
3	0	180	0%	red	NC	14	3	17	202
4	145	180	80.56%	yellow	Moyen	31	50	50	202
5	160	180	88.89%	green	Bien	31	50	50	202
6	155	180	86.11%	NULL	NULL	10	3	17	202
7	0	180	0%	red	NC	61	50	50	202
<hr/>									
date_call	operation_id	is_live_id	enregistrement_id	call_qualification_id	phone	language_id	duration_call	grid_listen_id	commentaire
2020-01-03	1	1	5	650136009	1	11.35		2	NULL
2020-01-03	1	1	3	614131028	1	5.50		2	NULL
2020-01-03	1	1	2	385725361	1	27.35		2	NULL
2020-01-03	1	1	4	607156285	1	5.44		2	NULL
2020-01-06	1	1	2	677847143	1	9.25		2	NULL
2020-01-06	1	1	2	767019062	1	13.25		2	NULL
2020-01-10	1	1	1	685656765	1	13.56		2	NULL
<hr/>									

Figure 3.3: La table grid_listen3_stored_calls

- La table sales_values : utilisée pour extraire le nombre de ventes réalisées par chaque agent dans un mois défini.

id	ref	operation_id	groupe_id	agent_id	supervisor_id	valideur_id	current_etat	mark_id	offre_id	option_id	valeur_vente	date_current_etat_c	date	
													date	date
1	1577955634	1	1	30	3	30	3	2	4	NULL	1	2020-01-06 10:08:09	2020-01-06 10:08:09	
2	15779557512	1	1	37	3	21	3	2	4	NULL	1	2020-01-02 10:55:20	2020-01-02 10:55:20	

Figure 3.4: La table sales_values

- La table users : Nous avons utilisé cette table pour ajouter le nom et le prénom de chaque agent.

+ Options										
	Id	recrut_id	first_name	last_name	role_id	poste_id	name	niveau_id	email	supervisor_id
<input type="checkbox"/> Éditer Copier Supprimer	1	NULL	Said	SIDI	1	1	Sidi	2	said@1waycom.com	NULL
<input type="checkbox"/> Éditer Copier Supprimer	3	NULL	Sofien	LABIDI	4	9	NULL	2	sofien@1waycom.com	36
<input type="checkbox"/> Éditer Copier Supprimer	4	NULL	Abir	CHERIF	2	1		3	cherifabir1991@gmail.com	9
<input type="checkbox"/> Éditer Copier Supprimer	5	NULL	Amira	SULDI	2	1		2	amirasulidigamaoul@gmail.com	11
<input type="checkbox"/> Éditer Copier Supprimer	6	NULL	Amira	ZAGHDOUDI	2	1		3	amirazaghoudi28@gmail.com	50
<input type="checkbox"/> Éditer Copier Supprimer	7	NULL	Khoulouid	LAARBI	2	1		3	arbi.khoulouid@yahoo.fr	50
<input type="checkbox"/> Éditer Copier Supprimer	8	NULL	Salma	JERBI	2	1	NULL	3	salma@1waycom.com	9
<input type="checkbox"/> Éditer Copier Supprimer	9	NULL	Walid	CHAHBI	3	18		3	chahbi1way@gmail.com	3
<input type="checkbox"/> Éditer Copier Supprimer	10	NULL	Chaïma	AIT LACHGAR	2	1		3	chaimaitlachgar13@gmail.com	9
<input type="checkbox"/> Éditer Copier Supprimer	11	NULL	Dorra	BOUCHIBA	3	4		2	dorra@1waycom.com	3

Figure 3.5: La table users

- La table holiday : utilisé pour extraire les jours fériés, qui seront utilisés dans le calcul du nombre d'heures travaillées par l'agent.

id	name	date_debut	date_fin	operation_id	created_at	updated_at
12	Fête du travail	2019-05-01	2019-05-01	1	2019-06-28 11:47:23	2019-06-28 11:47:23
13	Pentecôte	2019-06-10	2019-06-10	1	2019-06-28 11:48:00	2019-06-28 11:48:00
14	Fête nationale	2019-07-14	2019-07-14	1	2019-06-28 11:49:42	2019-06-28 11:49:42
15	Assomption	2019-08-15	2019-08-15	1	2019-06-28 11:53:13	2019-06-28 11:53:13
16	TOUSSAINT	2019-11-01	2019-11-01	1	2019-06-28 11:53:51	2019-06-28 11:53:51
17	L'ARMISTICE	2019-11-11	2019-11-11	1	2019-06-28 11:54:31	2019-06-28 11:54:31
18	NOËL	2019-12-25	2019-12-25	1	2019-06-28 11:55:07	2019-06-28 11:55:07
19	Jour de l'an	2020-01-01	2020-01-01	1	2019-06-28 11:55:43	2019-06-28 11:55:43
20	Fête du travail	2020-05-01	2020-05-01	1	2020-07-13 12:18:01	2020-07-13 12:18:01
21	Fête de la victoire de 1945	2020-05-08	2020-05-08	1	2020-07-13 12:19:22	2020-07-13 12:19:22
22	Fête Nationale	2020-07-14	2020-07-14	1	2020-07-13 12:19:59	2020-07-13 12:19:59

Figure 3.6: La table holiday

- La table absences : Utilisé pour ressortir le nombre d'heures travaillées pour chaque agent. Le nombre d'heures travaillées pouvant, d'après les superviseurs de 1WayCom influencer la productivité des agents.

id	user_id	created_by	groupe_id	operation_id	type 1 absent, 2 retard, 3 congé, 4 sortie	nombre_heures	heure_sortie	heure_retard	heure_supplementaire	date	type_conge	is_justified 0 non, 1 justifié
2	114	114	9	1	1	8	0	0	0	2020-10-01	NULL	
3	146	114	9	1	1	8	0	0	0	2020-10-01	NULL	
4	112	114	9	1	1	8	0	0	0	2020-10-01	NULL	
5	166	114	9	1	2	0	0	0	0	2020-10-01	NULL	
6	183	114	9	1	1	8	0	0	0	2020-10-01	NULL	
7	186	114	9	1	1	8	0	0	0	2020-10-01	NULL	
8	207	114	9	1	1	8	0	0	0	2020-10-01	NULL	
9	50	50	4	1	1	8	0	0	0	2020-10-01	NULL	
10	15	50	4	1	2	0	0	0	0	2020-10-01	NULL	
11	109	50	4	1	1	8	0	0	0	2020-10-01	NULL	
...
type_conge	is_justified 0 non, 1 justifié	notice_agent	raison	created_at		updated_at		cloture_id				
NULL	1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1				
NULL	1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1				
NULL	1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1				
NULL	1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1				
NULL	1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1				
NULL	1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1				
NULL	1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1				
NULL	1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1				
NULL	1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1				
NULL	1	0	NULL	2020-10-01 18:59:23		2020-10-01 18:59:23		1				
NULL	1	0	NULL	2020-10-02 10:28:16		2020-10-02 10:28:16		2				
NULL	1	0	NULL	2020-10-02 10:28:16		2020-10-02 10:28:16		2				
NULL	1	0	NULL	2020-10-02 10:28:16		2020-10-02 10:28:16		2				
NULL	1	0	NULL	2020-10-02 10:28:16		2020-10-02 10:28:16		2				
...	2020-10-02 10:28:16		2020-10-02 10:28:16		2				

Figure 3.7: La table absences

3.2 Compréhension des données

Ce processus a été expliqué en détail dans la première Release.

3.2.1 Données quantitatives continues

Les valeurs continues sont des valeurs qui s'expriment dans un intervalle de nombres continus infinis. Nous avons dans notre jeu de données deux variables quantitatives continues : -La variable qualification_score exprime le score obtenu par l'agent lors d'une écoute. -La variable nbr_heures exprime le nombre d'heures travaillées dans le mois où se fait l'écoute.

3.2.2 Les données quantitatives discrètes

Les valeurs discrètes sont des valeurs qui s'expriment dans un intervalle de nombres discrets finis. Dans notre jeu de données, nous avons cinq valeurs quantitatives discrètes : -Les variables agent_id, supervisor_id et niveau_id expriment respectivement l'identificateur de l'agent, l'identificateur du superviseur de l'agent et le niveau de l'agent lors du recrutement (débutant, intermédiaire, expert). -La variable nbr_ventes exprime le nombre de vente cumulées dans le mois correspondant à l'écoute réalisée. -La variable nbr_ventes_prec exprime le nombre de ventes effectuées dans le mois

qui précède celui où se fait l'écoute.

3.2.3 Les données qualitatives nominales

Les valeurs nominales sont des valeurs qualitatives exprimant le nom d'une catégorie : nom, sexe, métier, voiture, etc. Dans notre jeu de données, nous avons deux valeurs qualitatives nominales :

- Les variables agent_name et agent_lastname.

3.2.4 Les données qualitatives ordinaires

Les valeurs ordinaires sont des valeurs qualitatives qui sont naturellement ordonnées et qui peuvent être traduites par une valeur numérique, comme le rang par exemple : élevé, moyen, bas. Dans notre jeu de données, nous avons une seule valeur qualitative ordinaire : - La variable notation, représente la mention Bien, Moyen ou NC.

3.2.5 Les données qualitatives temporelles

Les données temporelles sont des valeurs numériques qui suivent l'évolution du temps. Nous avons dans notre jeu de données une seule variable temporelle : -La variable date_call exprime la date de l'écoute en question.

3.2.6 Représentation des données

La représentation des données permet de modéliser les données sous une forme mathématique compréhensible.

- **Représentation des variables prédictives :** Les variables prédictives sont appelées variables indépendantes et sont largement connues dans la Data Science sous le nom anglais « Features ». Dans une Data Set, les variables prédictives $X = x_1, x_2, \dots, x_n$ peuvent être modélisées dans une matrice d'ordre $m \times n$ où m est la taille du training set et n est le nombre des caractéristiques de chaque instance. Dans notre étude, $n=5$ et $m=2589$.

$$\begin{bmatrix} x(1,1) & x(1, \dots) & \dots & x(1,5) \\ \vdots & \vdots & & \vdots \\ x(2589,1) & x(2589, \dots) & & x(2589,5) \end{bmatrix}$$

Figure 3.8: Matrice représentative des variables prédictives

- **Représentation de la variable cible :** La variable cible est la variable que nous souhaitons prédire à partir des variables prédictives. Elle est aussi appelée en anglais « Target variable ». Dans notre cas, nous allons prédire le nombre de ventes pour chaque agent. Nous sommes donc en présence d'une régression multiple où chaque instance de $X = x_1, x_2, \dots, x_5$ peut avoir un seul output Y . Notre variable cible est donc représentée par une matrice $m \times 1$ ou $m = \text{Taille de la Data Set} (2589)$.

$$\begin{bmatrix} y(1) \\ \vdots \\ y(2589) \end{bmatrix}$$

Figure 3.9: Matrice représentative de la variable cible

3.2.7 Exploration des données

Dans l'exploration des données, le Data Scientiste essaie d'extraire les connaissances statistiques qui pourraient l'aider dans ses tâches ultérieures. Cette procédure est expliquée en détail dans le premier Release.

3.2.8 Synthèse

Dans le premier tableau, nous allons affecter les features que nous allons utiliser dans notre modèle selon la nature de la variable et son caractère prédictif ou cible. Nous donnerons également les valeurs statistiques des variables prédictives et de la variable cible dans le tableau suivant.

Tableau 3.1: Matrice représentative de la variable cible.

La variable	Quantitative continue	Quantitative discrète	Qualitative nominale	Qualitative ordinale	Temporelle	Préditive	Cible
qualification_score	✗					✗	
notation				✗		✗	
agent_id		✗					
supervisor_id		✗					
date_call					✗		
nbr_ventes		✗					✗
nbr_heures	✗					✗	
niveau_id		✗				✗	
agent_name			✗				
agent_lastname			✗				
nbr_ventes_prec		✗				✗	

Tableau 3.2: Mesures statistiques de base.

Variable	Moyenne	Mode	Médiane	Q1	Q3
qualification_score	71.4012671	0	86.11	75.930000	90.092500
notation	<u>x</u>	10	10	5	10
nbr_ventes	16	18	16	9	23
nbr_heures	141.63	160	149.25	129.75	161.00
niveau_id	<u>x</u>	1	2	1	3
nbr_ventes_prec	11	6	12	7	16

3.3 Agrégation des données

Dans ce livrable, nous ne disposons pas d'une table déjà prête sur laquelle nous pouvons faire tourner nos algorithmes. Il nous fallait exploiter la base de données de 1wayCom et apporter de nombreuses modifications à travers des requêtes SQL et des scripts en Python pour aboutir à une table, qui par la suite sera convertie en un fichier csv exploitable par le Machine Learning.

3.3.1 Les modifications apportées sur les données :

En appliquant les modifications nécessaires sur les données, nous avons commencé par la table grid_listen3_stored_calls représentée par la figure 3.10 :

id	qualification_valeur	qualification_total	qualification_score	selected_scale_class	selected_scale_name	agent_id	agent_supervisor_id Superviseur d'agent lors de l'attribution	supervisor_id	date	
1	150	170	88.24%	green	Bien	30	3	17	202	
2	0	180	0%	NULL	NULL	14	3	17	202	
3	0	180	0%	red	NC	14	3	17	202	
4	145	180	80.56%	yellow	Moyen	31	50	50	202	
5	160	180	88.89%	green	Bien	31	50	50	202	
6	155	180	86.11%	NULL	NULL	10	3	17	202	
7	0	180	0%	red	NC	61	50	50	202	
date_call	operation_id	is_live_id Enregistrement live	call_qualification_id	phone	language_id	duration_call	grid_listen_id	commentaire	need_briefing	briefing_agent
2020-01-03	1	1	5	650138009	1	11.35	2	NULL	1	NULL
2020-01-03	1	1	3	614131028	1	5.50	2	NULL	2	suite au mythos vais appliquer !
2020-01-03	1	1	2	385725361	1	27.35	2	NULL	2	NULL
2020-01-03	1	1	4	607156285	1	5.44	2	NULL	1	NULL
2020-01-06	1	1	2	677847143	1	9.25	2	NULL	1	NULL
2020-01-06	1	1	2	767019062	1	13.25	2	NULL	1	NULL
2020-01-10	1	1	1	685656765	1	13.56	2	NULL	2	NULL

Figure 3.10: la table grid_listen3_stored_calls

- Suppression de toutes les colonnes inutiles pour notre apprentissage automatique :

```
ALTER TABLE grid_listen3_stored_calls DROP column agent_supervisor_id, DROP
COLUMN operation_id, drop column is_live_id , DROP COLUMN commentaire, DRO
P COLUMN language_id , DROP COLUMN phone, DROP COLUMN briefing_agent_commen
taire ,DROP COLUMN briefing_supervisor_commentaire, DROP COLUMN briefing_by
,drop COLUMN valid_by_agent, DROP COLUMN valid_by_supervisor , DROP column
date_valid_by_agent, DROP COLUMN date_valid_by_supervisor, DROP COLUMN aud
io_path , drop COLUMN audio_name , DROP COLUMN site_id , DROP COLUMN fourni
sseur, DROP COLUMN updated_by , drop COLUMN updated_at , DROP COLUMN create
d_at, DROP COLUMN deleted_at , DROP COLUMN date_briefing
```

Figure 3.11: Code de suppression de toutes les colonnes inutiles

- Ajout et mise à jour de la colonne niveau _id :

```
1 UPDATE predventes SET niveau_id=(SELECT niveau_id| FROM users WHERE users.id=predventes.agent_id)
```

Figure 3.12: Code d'ajout et mise à jour de la colonne niveau

- Ajout de la colonne agent_name :

```
1 UPDATE predventes SET agent_name=(SELECT first_name FROM users WHERE users.id=predventes.agent_id)
```

Figure 3.13: la table grid_listen3_stored_calls

- Ajout de la colonne agent_lastname :

```
1 UPDATE predventes SET agent_lastname=(SELECT first_lastname FROM users WHERE users.id=predventes.agent_id)
```

Figure 3.14: Code d'ajout de la colonne agent_lastname

- Ajout et mise à jour de la colonne nbr_ventes à partir de la table sales_values :

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Fri Apr  2 15:16:06 2021
4
5 @author: RAMSA
6 """
7
8 import pymysql
9 connection = pymysql.connect(host="localhost",
10                               user="root",
11                               passwd="",
12                               database="pfe3")
13
14 cursor=connection.cursor()
15
16 sql1="""SELECT agent_id FROM sales_values"""
17 cursor.execute(sql1)
18 id =cursor.fetchall()
19 for x in id:
20     for y in range(1,13):
21         sql2="SELECT COUNT(*) FROM sales_values WHERE agent_id=%s and MONTH(created_at)=%s and YEAR(created_at)=2020 "
22         cursor.execute(sql2,(x,y))
23         nbr=cursor.fetchall()
24         #print(nbr)
25         sql="UPDATE ecoutes SET nbr_ventes=%s WHERE agent_id=%s and MONTH(date_call)=%s and year(date_call)=2020"
26         cursor.execute(sql,(nbr[0][0],x,y))
27         sql2="SELECT COUNT(*) FROM sales_values WHERE agent_id=%s and MONTH(created_at)=%s and YEAR(created_at)=2021 "
28         cursor.execute(sql2,(x,y))
29         nbr=cursor.fetchall()
30         #print(nbr)
31         sql="UPDATE ecoutes SET nbr_ventes=%s WHERE agent_id=%s and MONTH(date_call)=%s and year(date_call)=2021"
32         cursor.execute(sql,(nbr[0][0],x,y))
33         connection.commit()
34
35 connection.commit()
36
37

```

Figure 3.15: Code d'ajout et mise à jour de la colonne nbr_vente

- Ajout et mise à jour de la colonne nbr_ventes_prec à partir de la table sales_values :

```

18
19
20 sql1="""SELECT id FROM users where role_id=2 """
21 cursor.execute(sql1)
22 id =cursor.fetchall()
23 for x in id:
24     for y in range(1,13):
25         sql2="SELECT SUM(nombre_heures) FROM absences WHERE user_id=%s and MONTH(date)=%s and YEAR(date)=2020 "
26         cursor.execute(sql2,(x,y))
27         nbr=cursor.fetchall()
28         sql="UPDATE ecoutes SET nbr_heures=%s WHERE agent_id=%s and MONTH(date_call)=%s and year(date_call)=2020"
29         cursor.execute(sql,(nbr,x,y))
30         sql2="SELECT SUM(nombre_heures) FROM absences WHERE user_id=%s and MONTH(date)=%s and YEAR(date)=2021"
31         cursor.execute(sql2,(x,y))
32         nbr=cursor.fetchall()
33         sql="UPDATE ecoutes SET nbr_heures=%s WHERE agent_id=%s and MONTH(date_call)=%s and year(date_call)=2021"
34         cursor.execute(sql,(nbr,x,y))
35         connection.commit()
36         connection.commit()
37

```

Figure 3.16: Code d'ajout et mise à jour de la colonne nbr_ventes_prec

- Ajout et mise à jour de la colonne nbr_heures à partir de la table absences :

```

19
20
21 sql1="""SELECT id FROM users where role_id=2 """
22 cursor.execute(sql1)
23 id =cursor.fetchall()
24 for x in id:
25     for y in range(1,13):
26         sql2="SELECT SUM(nombre_heures) FROM absences WHERE user_id=%s and MONTH(date)=%s and YEAR(date)=2020 "
27         cursor.execute(sql2,(x,y))
28         nbr=cursor.fetchall()
29         sql="UPDATE ecoutes SET nbr_heures=%s WHERE agent_id=%s and MONTH(date_call)=%s and year(date_call)=2020"
30         cursor.execute(sql,(nbr,x,y))
31         sql2="SELECT SUM(nombre_heures) FROM absences WHERE user_id=%s and MONTH(date)=%s and YEAR(date)=2021"
32         cursor.execute(sql2,(x,y))
33         nbr=cursor.fetchall()
34         sql="UPDATE ecoutes SET nbr_heures=%s WHERE agent_id=%s and MONTH(date_call)=%s and year(date_call)=2021"
35         cursor.execute(sql,(nbr,x,y))
36         connection.commit()
37         connection.commit()

```

Figure 3.17: Code d'ajout et mise à jour de la colonne nbr_heures

3.3.2 Table résultante

Nous avons effectué plusieurs manipulations pour finalement obtenir la table écoutes qui suit :

id	agent_id	agent_name	agent_lastname	supervisor_id	qualification_valeur	qualification_total	qualification_score
1470	186	Jihene	BOURougaa	114	170	180	94.44
1471	112	Aymen	JABALLAH	17	165	180	91.67
1472	112	Aymen	JABALLAH	17	165	180	91.67
1473	149	Anwa	YOUNSI	154	155	180	86.11
1474	146	Myriam	STILI	114	0	180	0.00
1475	52	Cyrine	RIAHI	17	145	180	80.56
1476	24	Bassma	MESSAI	17	160	180	88.89
1477	107	Cyrine	BELHIBA	17	160	180	88.89
1478	144	Safa	SAID	17	150	180	83.33
nbr_ventes	nbr_ventes_prec	notation	nbr_heures	niveau_id	need_briefing	duration_call	date_call
18	17	10	161	1	1	28	2020-10-01
30	26	10	200	1	1	39	2020-10-02
30	26	10	200	1	1	48	2020-10-02
9	26	10	161	2	2	20	2020-10-02
23	39	0	151	3	2	15	2020-10-01
10	14	5	130	3	1	25	2020-10-01
15	2	10	176	2	1	26	2020-10-01
8	5	10	141	3	1	11	2020-10-02
32	19	5	168	2	1	36	2020-10-02

Figure 3.18: La table écoutes résultante

3.4 Nettoyage des données

Dans cette section nous allons nettoyer notre Data Set des valeurs nulles, des lignes dupliquées, des types de colonnes incompatibles à l'apprentissage automatique, etc.

Nous obtiendrons par la suite un fichier csv prêt à être exploité par nos algorithmes.

Nous avons encore une fois choisi la librairie Pandas, largement connue comme étant très efficace et facile pour la manipulation, l'analyse et le nettoyage des données.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2589 entries, 0 to 2591
Data columns (total 13 columns):
id                  2589 non-null int64
qualification_valeur 2589 non-null int64
qualification_total  2589 non-null int64
qualification_score  2589 non-null object
selected_scale_name  2256 non-null object
agent_id             2589 non-null int64
supervisor_id        2589 non-null int64
date_call            2589 non-null object
call_qualification_id 2589 non-null int64
duration_call        2589 non-null object
grid_listen_id       2589 non-null int64
need_briefing        2589 non-null int64
date_briefing         785 non-null object
dtypes: int64(8), object(5)
memory usage: 232.6+ KB
```

Figure 3.19: Notre Data Set avant le nettoyage des données

3.4.1 Nettoyage des lignes dupliquées

N La fonction drop_duplicates() Avant la suppression des lignes dupliquées notre table avait 2589 lignes. D'après la figure 3.20 nous constatons que le nombre de lignes est resté le même ce qui

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2589 entries, 0 to 2591
Data columns (total 13 columns):
id                  2589 non-null int64
qualification_valeur 2589 non-null int64
qualification_total  2589 non-null int64
qualification_score  2589 non-null object
selected_scale_name  2256 non-null object
agent_id             2589 non-null int64
supervisor_id        2589 non-null int64
date_call            2589 non-null object
call_qualification_id 2589 non-null int64
duration_call        2589 non-null object
grid_listen_id       2589 non-null int64
need_briefing        2589 non-null int64
date_briefing         785 non-null object
dtypes: int64(8), object(5)
memory usage: 232.6+ KB
```

Figure 3.20: Notre Data Set après le nettoyage des lignes dupliquées

induit qu'il y'a pas de duplication des lignes.

3.4.2 Nettoyage des valeurs nulles

Dans cette étape nous pouvons soit supprimer les valeurs nulles soit les remplacer par :

- **Le mode** : la valeur qui apparaît le plus dans la colonne concernée
- **La médiane** : La valeur au milieu de la colonne concernée.
- **La moyenne** : La moyenne des valeurs de la colonne concernée.

Ces méthodes ne résolvent pas correctement notre problématique et ont erroné les modèles d'apprentissages automatique dans la suite, puisque les valeurs ne sont jamais aléatoires et sont toujours liées avec les autres colonnes. Nous avons donc préféré de supprimer les lignes contenant des valeurs nulles. Ce qui a fait passer notre table de 2589 lignes à 600. La table après la suppression des valeurs nulles avec la fonction dropna() de la librairie pandas :

```
wdir='C:/Users/RAHMA/.spyder-py3'
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Data columns (total 20 columns):
id              600 non-null int64
qualification_valeur 600 non-null int64
qualification_total 600 non-null int64
qualification_score 600 non-null float64
notation          600 non-null int64
agent_id          600 non-null int64
supervisor_id     600 non-null int64
date_call         600 non-null object
call_qualification_id 600 non-null int64
duration_call     600 non-null int64
grid_listen_id    600 non-null int64
need_briefing     600 non-null int64
date_briefing     600 non-null object
nbr_ventes        600 non-null int64
nbr_heures        600 non-null float64
niveau_id         600 non-null int64
defis             600 non-null float64
agent_name        600 non-null object
agent_lastname    600 non-null object
nbr_ventes_prec   600 non-null int64
dtypes: float64(3), int64(13), object(4)
memory usage: 84.4+ KB
None
```

Figure 3.21: Notre Data Set après le nettoyage des valeurs nulles

3.4.3 Nettoyage du format des colonnes

Nous avons transformé quelques colonnes pour qu'elles puissent être exploitables par les algorithmes d'apprentissage automatique :

- La colonne qualification_score avait le type varchar => son nouveau type est float.
- La colonne notation était de type string => son nouveau type est int : valeur « Bien » est remplacée par 10.
- La valeur « Moyen » est remplacée par 5.
- La valeur « NC » est remplacée par 0.

3.4.4 Fichier CSV résultant

Notre Data Set est maintenant prête à être proprement exploitée

A	B	C	D	E	F	G	H	I	J	K	L
id	qualification_score	notation	agent_id	supervisor_id	date_call	nbe_ventes	nbe_heures	niveau_id	agent_name	agent_lastname	nbr_ventes_prec
2	1470	94.44	10	186	114	10/1/2020	18	161	1 Jihene	BOUROUGAA	17
3	1471	91.67	10	112	17	10/2/2020	30	200	1 Aymen	JABALLAH	26
4	1472	91.67	10	112	17	10/2/2020	30	200	1 Aymen	JABALLAH	26
5	1473	86.11	10	149	154	10/2/2020	9	161	2 Arwa	YOUNSI	7
6	1474	0	0	146	114	10/1/2020	23	151	3 Myriam	STILI	39
7	1475	80.56	5	52	17	10/1/2020	10	130	3 Cyrine	RIABI	14
8	1476	88.89	10	24	17	10/1/2020	15	176	2 Bassma	MESSAI	2
9	1477	88.89	10	107	17	10/2/2020	8	141	3 Cyrine	BELHIBA	6
10	1478	83.33	5	144	17	10/2/2020	32	168	2 Safa	SAID	19
11	1479	91.67	10	32	11	10/2/2020	15	200	2 Sana	KHDIMALLAH	13
12	1480	0	0	206	17	10/2/2020	3	36.5	1 Zakaria	TROUDI	2
13	1481	91.67	10	24	17	10/2/2020	15	176	2 Bassma	MESSAI	2
14	1482	85.29	10	207	17	10/2/2020	4	98	1 Yasmine	TOUATI	3
15	1485	83.33	5	165	17	10/2/2020	11	131.9	2 Aymen	ACHOURI	9
16	1486	86.11	10	109	50	10/2/2020	11	184	1 Chiraz	BOUZIDI	10
17	1487	91.67	10	132	50	10/5/2020	36	194	1 Hadja Aseta	COULIBALY	30
18	1490	85.29	10	60	17	10/5/2020	5	44.5	2 Mohamed Ali	MEDIOUNI	7
19	1491	72.22	5	199	11	10/5/2020	1	43.5	3 Mohamed	ZEKRI	0
20	1492	82.35	5	99	17	10/5/2020	5	150	3 Lynda	RAHALI	1
21	1493	79.41	5	148	17	10/5/2020	24	149	2 Jihene	JNAOUI	23
22	1496	80.56	5	186	17	10/5/2020	18	161	1 Jihene	BOUROUGAA	17
23	1497	88.89	10	14	17	10/5/2020	12	200	3 Fedia	BEN MOXODED	19
24	1498	83.33	5	12	17	10/5/2020	23	198	3 Emma	KADDACHI	26
25	1500	0	0	191	17	10/6/2020	19	184	2 Neyla	FERCHICHI	9
26	1502	91.67	10	200	17	10/7/2020	17	200	2 Henda	GAMMOUDI	6
27	1503	82.35	5	32	17	10/6/2020	15	200	2 Sana	KHDIMALLAH	13
28	1504	91.67	10	110	17	10/6/2020	11	192	3 Daniella Francine	BAHA	16
29	1505	88.89	10	166	114	10/6/2020	24	192	2 Imen	MAKHLOUF	13

Figure 3.22: Fichier CSV de notre Data Set

3.4.5 La visualisation des données

La visualisation des données permet de mieux comprendre les données, puisqu'elle les résume d'une manière graphique compréhensible par quasiment tout le monde.

3.4.6 Visualisation de la fréquence des nombres de ventes

L'axe des x est le nombre de vente et l'axe des y est la fréquence dudit nombre de vente dans la Dataset. Ces graphes montrent que la production augmente d'un mois à l'autre. Le mode des ventes du mois X étant 18, et celui du mois X-1 étant 6.

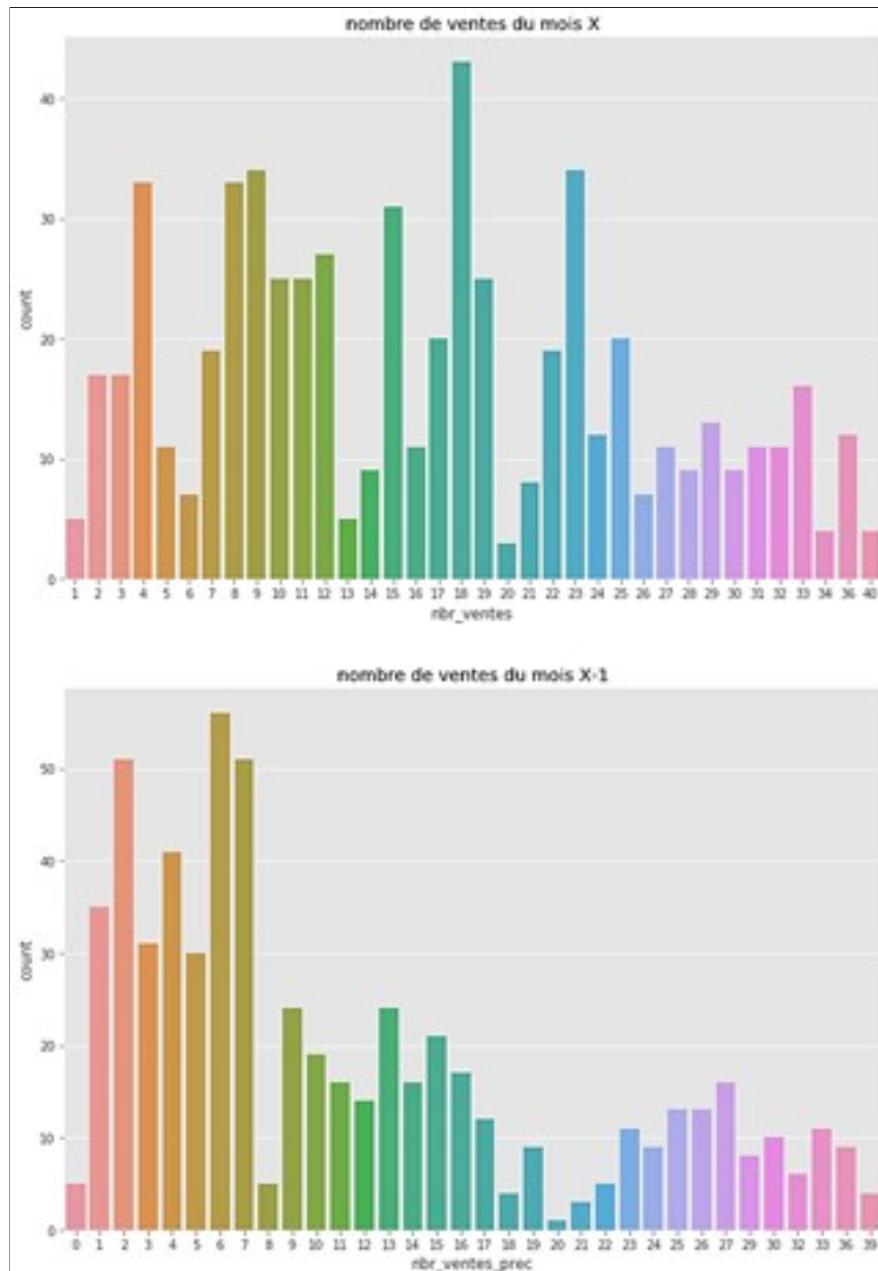


Figure 3.23: Graphe de fréquences des ventes sur deux mois consécutifs

3.4.7 Visualisation de la distribution des scores et des notes :

Dans le premier graphe, l'axe des x est le score de qualification de l'agent et l'axe des y est la densité de ce score dans la Dataset. Dans le deuxième graphe, l'axe des x est la note de l'agent et l'axe des y est la densité de cette note dans la Dataset. Ces graphes montrent que le score et la note n'ont pas la même distribution, bien qu'ils devraient. Ceci est dû à la sévérité de la note NC (Non Conforme) lorsqu'elle est attribuée à un agent dans une catégorie déterminée de la feuille d'écoute, qui fait chuter le score d'un agent à 0, même si la qualité globale de l'appel est bonne, méritant la note 10.

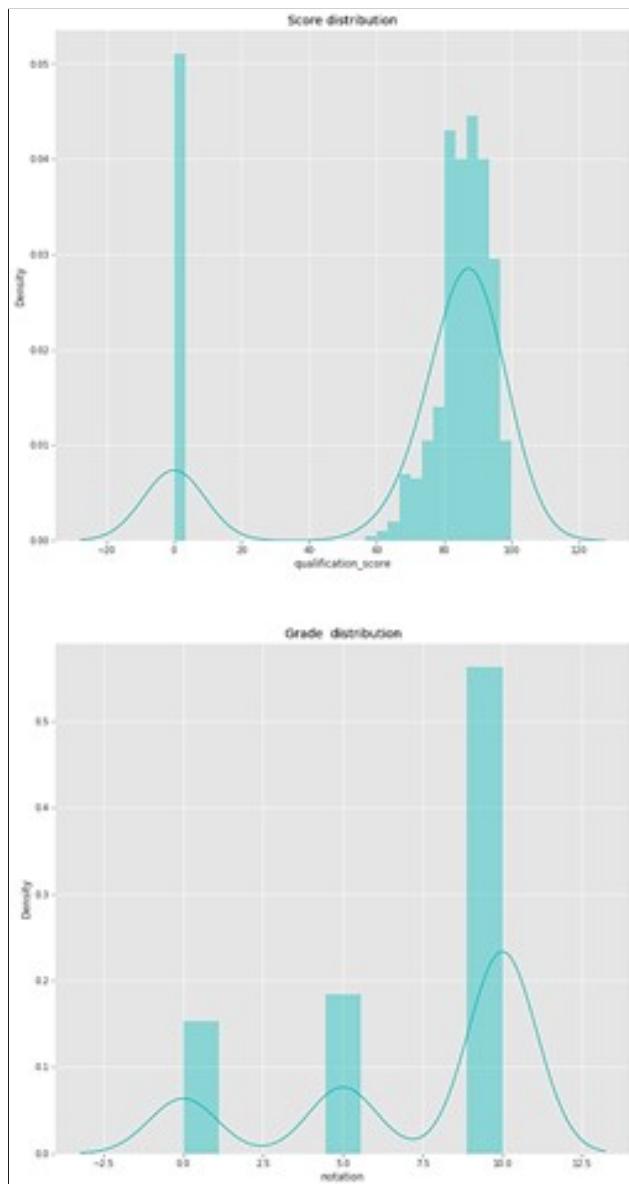


Figure 3.24: Visualisation de la distribution des scores et des notes

3.4.8 Visualisation de la corrélation entre les variables

Le graphe 3.25 montre une bonne corrélation entre le nombre des heures pour lesquelles un agent travaille pendant un mois et le nombre de ventes qu'il réalise au cours de ce mois-là

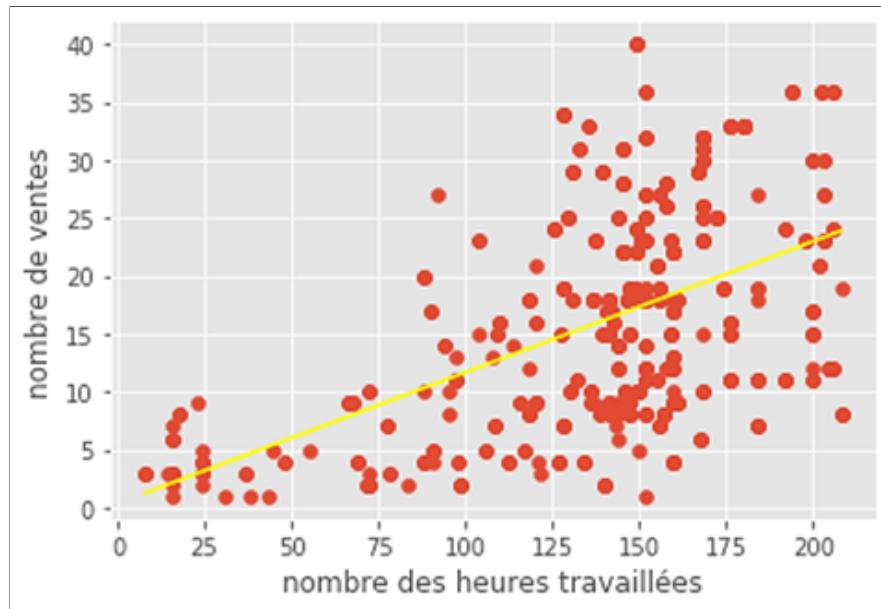


Figure 3.25: Graphe pour montrer la corrélation entre le nombre de ventes et le nombre d'heures travaillées

Le graphe 3.26 avec les axes(1 : niveau débutant / 2 : niveau intermédiaire /3 : niveau expert) montre qu'il n'y a pas de grande différence de productivité reliée au niveau d'un agent déterminé. Par contre, nous pouvons voir que les agents experts présentent quand même une différence par rapport aux agents intermédiaires et débutants, étant les seuls à atteindre une quarantaine de ventes par mois.

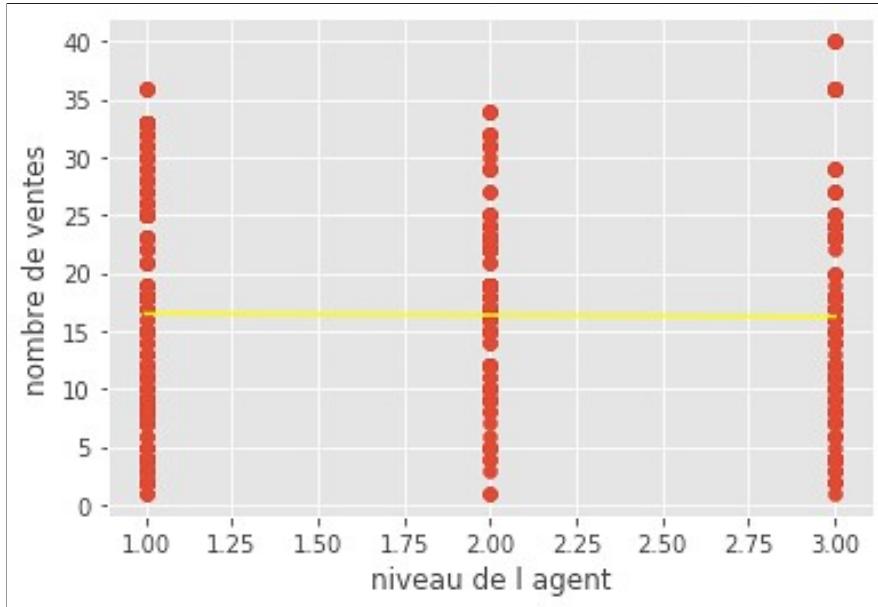


Figure 3.26: Graphe de corrélation entre le niveau de l'agent et sa productivité

Le graphe 3.27 montre que la productivité d'un agent en termes de nombre de ventes réalisées n'est pas directement liée au score qu'il obtient dans ses feuilles d'écoutes. Par contre, les agents atteignant les quarante ventes par mois n'ont jamais obtenu le score 0.

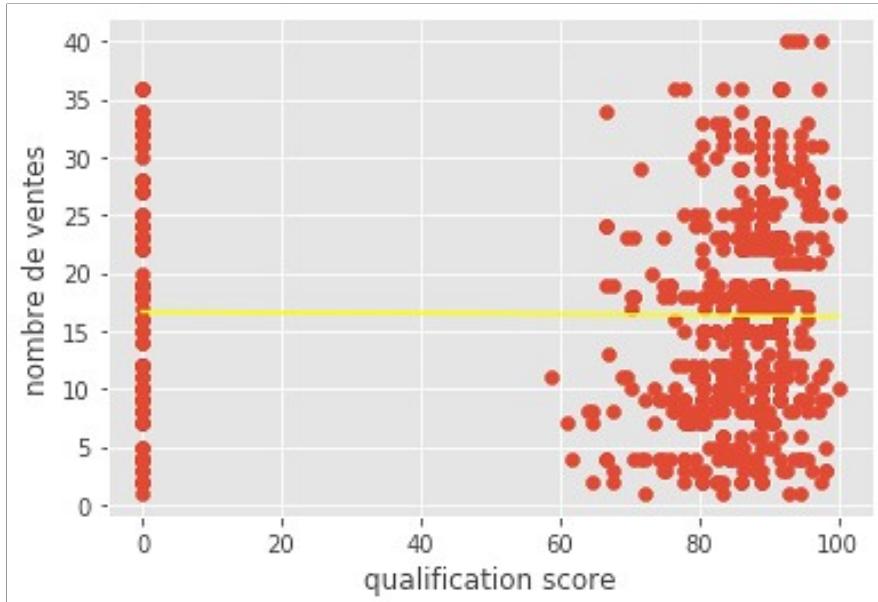


Figure 3.27: Graphe de corrélation entre le score d'un agent et sa productivité

3.4.9 La matrice de corrélation de la Data Set

La matrice de corrélation de notre Data Set montre que les variables prédictives les plus corrélées avec notre variable cible sont : nbr_ventes_prec avec une corrélation de 0.61 et nbr_heures

avec une corrélation de 0.5 Le coefficient de corrélation linéaire r est donc considéré faible à modéré. Cela suggère que nos données ne sont pas linéaires et exclut les algorithmes de régression linéaire (lasso, ridge, etc.) dans l'étape de modélisation.

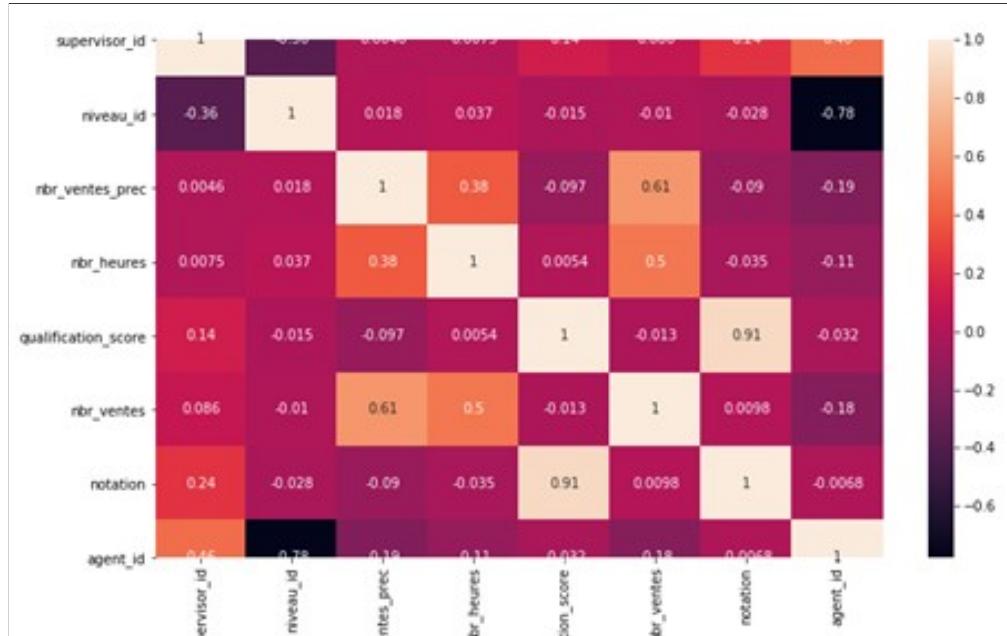


Figure 3.28: Matrice de corrélation de la Data Set

SPRINT2 : Modélisation et évaluation

Dans ce sprint, nous avons étudié les méthodes et les algorithmes qui nous permettent de modéliser nos données en prenant compte de leurs particularités. Nous avons ensuite évalué les modèles utilisés à travers une étude comparative détaillée

3.5 La modélisation

Pour résoudre un problème de régression avec une Data Set de taille petite, il est conseillé d'utiliser des algorithmes et des techniques définies. Dans cette section, nous nous sommes intéressés à la modélisation de nos algorithmes selon ces particularités.

3.5.1 Choix de la technique de prédiction adéquate

Afin de choisir la technique de prédiction adéquate à son problème, il faut se poser les questions suivantes :

- Les performances de calcul sont-elles un problème ?

Si oui, il est préférable de :

- Réduire la dimensionnalité.
- Utiliser des algorithmes peu coûteux.
- Sélectionner seulement les attributs nécessaires à la prédiction.
- Choisir des algorithmes appelés « Lazy Learners » comme KNN.

• **Quel est le type de ma variable cible ?**

Le type de la variable cible est presque définitif dans le choix de la technique de prévision pour un problème d'apprentissage automatique. En effet, il est largement reconnu que :

- Quand la variable à prédire est continue : Il s'agit d'un problème de Régression, qui est notre cas.
- Quand la variable à prédire est catégoriale (nominale) : Il s'agit d'un problème de Classification.
- Quand la variable à prédire est ordinaire : Il s'agit d'un problème de Classification classée.
- Pas de variable à prédire, le but est de trouver une structure dans les données : Il d'agit d'un problème de Clustering ou de Projection.

• **Est-ce que les données sont linéairement séparables ?**

A partir de la matrice de corrélation de notre jeu de données, nous avons réalisé que notre problème n'était pas linéaire, dû au coefficient de corrélation faible.

• **Quelle est la taille des données ?**

Certaines Data Sets sont très larges et ne peuvent pas être stockées dans la mémoire de l'ordinateur. Dans ces cas-là, il faut utiliser :

- L'apprentissage hors noyau.
- Les systèmes distribués.

La taille de notre Data Set est $m*n=600*5$ avec m le nombre de lignes et n le nombre de variables prédictives C'est une petite taille de données qui peut être normalement stockée dans la mémoire de l'ordinateur. La petite taille de notre Data Set implique l'utilisation de modèles d'apprentissage automatique pouvant fonctionner convenablement malgré cette contrainte. D'autant plus, nous devons faire un Tuning adéquat des Hyperparamètres des algorithmes pour contourner ce problème.

3.5.2 Le Tuning des hyperparamètres

Comme nous l'avons mentionné ci-dessus, le Tuning est très important dans une Data Set considérée petite. Pour comprendre son fonctionnement, il est indispensable de connaître la différence entre les paramètres et les hyperparamètres d'un algorithme, et de savoir choisir la stratégie de Tuning la plus adaptée au problème. Toutes ses notions ont été expliquées dans la première Release. Nous avons choisi le Grid search pour le Tuning de nos algorithmes parce qu'elle est la méthode la plus adéquate en termes de taille de données.

3.5.3 La régression

Dans les modèles de régression, nous prédisons une variable cible à partir de certaines variables prédictives. Cela veut dire que nous utilisons la régression pour trouver une corrélation entre une variable dépendante (cible) et une variable indépendante (prédictive). Dans les modèles de régression, l'output est une valeur quantitative continue ou bien discrète. Nous sommes en présence d'une régression multiple puisque nous avons une seule variable cible et plusieurs variables prédictives. Pour notre problème de régression, nous avons utilisé 4 modèles d'apprentissage automatique :

- XGBoost
- KNN
- Random Forest
- SVR

Nous avons déjà expliqué les trois premiers algorithmes dans la première Release. Nous allons donc expliquer l'algorithme SVR.

3.5.4 SVR (Support Vector Regression)

Les personnes impliquées dans le domaine du Machine Learning ont sûrement entendu parler du SVM ou Support Vector Machine. Cet algorithme-là est utilisé pour des problèmes de classification. SVR est un peu différent de SVM, puisque, comme son nom l'indique, SVR est utilisé pour la régression. Dans les algorithmes de régression simples, nous essayons de minimiser la fréquence de l'erreur. Dans SVR, nous essayons d'ajuster l'erreur dans un certain seuil [13]. La ligne bleue : Hyperplan. La ligne rouge : ligne de limite (seuil). La figure montre que les points verts (les erreurs) ne dépassent pas la limite des lignes rouges. Notre meilleure ligne d'ajustement est la ligne bleue

Hyperplan qui contient le maximum de points de données.

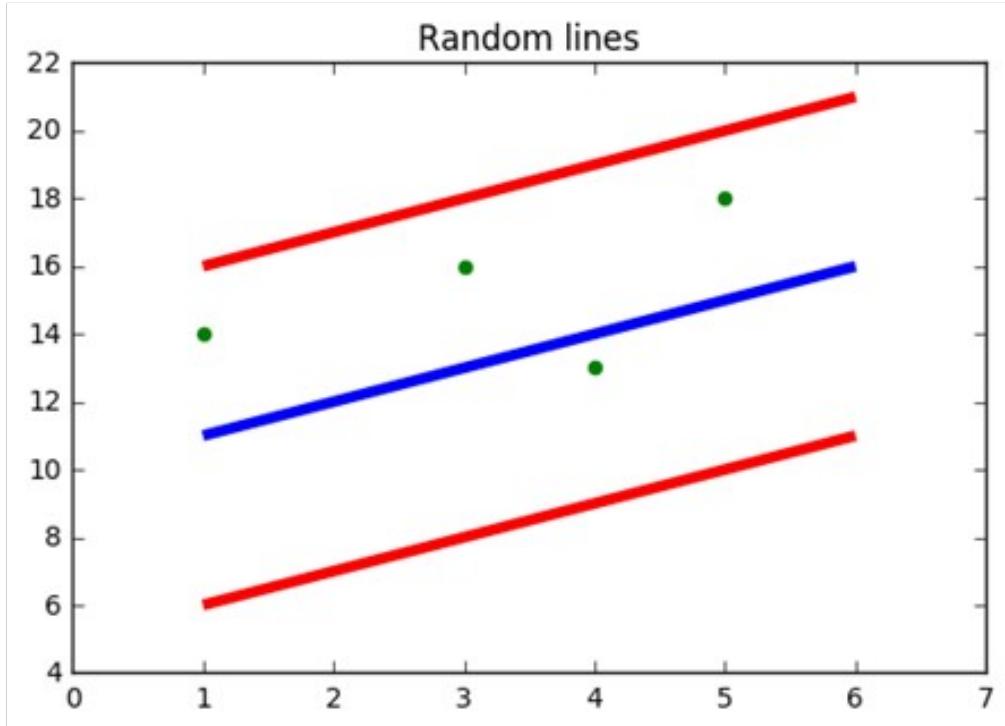


Figure 3.29: Fonctionnement de SVR

3.5.5 Explication du choix des modèles utilisés

Lorsque nous sommes en présence d'une Dataset de petite taille, il y a un risque que nos modèles ne soient pas très performants, ou fassent du Surapprentissage (le terme anglais est Overfitting). Le surapprentissage : ce phénomène se produit lorsque le modèle apprend trop les particularités des données fournies, et fait beaucoup d'erreurs sur le Test set. Il existe plusieurs techniques pour éviter ces problèmes, dont les plus importantes sont expliquées ci-dessous :

- **Première solution : utilisation des modèles simples :** Lorsque notre data set est de petite taille, il faut utiliser des algorithmes simples, représentatifs et qui ne provoquent pas un Overfitting des données. Pour un problème de régression, il est conseillé de choisir l'algorithme de régression linéaire (ou meilleur encore, celui de Ridge ou Lasso). Comme nous l'avons expliqué précédemment, notre problème n'est pas linéaire, dû au coefficient de corrélation « faible à modéré » avec la variable cible. Nous ne pouvons donc utiliser ni le modèle de régression linéaire, ni Ridge, ni Lasso (qui sont également linéaires).
- **Deuxième solution : utilisation des modèles basés sur les arbres de décisions :** Une autre alternative est d'implémenter les algorithmes basés sur les arbres de décisions,

qui sont les meilleurs pour éviter l'Overfitting. Nous avons donc utilisé XGBoost (méthode ensembliste basée sur les arbres de décisions et le gradient descent Boost) et Random Forest (forêts aléatoires d'arbres de décisions).

- **Troisième solution : essayer plusieurs modèles :** Comme la comparaison entre plusieurs algorithmes est recommandée pour ce type de problème, nous avons ajouté les modèles SVR et KNN, en tâchant de bien régulariser les algorithmes (dans le tuning des hyperparamètres).

3.5.6 Synthèse

Nous allons résumer l'étape de modélisation dans le tableau suivant, détaillant chaque algorithme utilisé avec les Hyperparamètres ajustés (Tuning).

Tableau 3.3: Les modèles utilisés avec leurs paramètres ajustés.

Modèle	Hyperparamètres
SVR	kernel = rbf C = 100
Random Forest	n_estimators= 400 max_features= auto max_depth= 800
XGBoost	max_depth = 5 n_estimators = 800 min_child_weight = 3
KNN	n_neighbors= 4 weights = distance algorithm = auto

Nous passons alors à l'explication du tableau 3.3

- **SVR :** Hyperparamètres déjà expliqués.

- **Random Forest :**

- n_estimators : le nombre d'arbres.
- max_features : le nombre d'attributs à considérer lors de la division dans un nœud.
- max_depth : la profondeur maximale d'un arbre.

- **XGBoost :**

- max_depth : la profondeur maximale d'un arbre.
- n_estimators : le nombre d'arbres.
- min_child_weight : le poids minimum (ou le nombre d'échantillons si tous les échantillons ont un poids de 1) requis pour créer un nouveau nœud dans l'arborescence.

- **KNN :**

- n_neighbors : nombre de voisins à utiliser.
- weights : fonction de poids utilisée dans la prédiction.
- algorithm : algorithme utilisé pour calculer les plus proches voisins.

3.6 L'évaluation

L'évaluation d'un problème de régression est très différente que celle d'un problème de classification. Les mesures de performances pour les problèmes de régression sont orientées vers le calcul d'erreurs.

3.6.1 Mean Squared error

Error ou MSE est l'erreur quadratique moyenne dans un modèle d'apprentissage automatique. C'est une mesure de performance très populaire pour ce type d'algorithmes [14]. La MSE est calculée comme la moyenne des différences au carré entre les valeurs prédictes et les valeurs réelles dans un ensemble de données. La fonction de MSE :

$$MSE = \frac{1}{n} * \sum_i^n (y_i - yhat_i)^2$$

y_i : la i ème valeur réelle de la data set

$yhat_i$: la i ème valeur prédictive de la data set

Figure 3.30: Calcul de la mesure de performance MSE

3.6.2 Root mean squared error

Root Mean Squared Error ou RMSE est une extension de MSE. Il est important de noter que la racine carrée de l'erreur est calculée, ce qui signifie que les unités du RMSE sont les mêmes que les unités d'origine de la variable cible prédictive. La fonction de RMSE :

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} * \sum_i^n (y_i - yhat_i)^2}$$

Figure 3.31: Calcul de la mesure de performance RMSE

3.6.3 Mean absolute error

MAE est une métrique populaire car, comme RMSE, les unités du score d'erreur correspondent aux unités de la valeur cible prédictive. Contrairement au RMSE, les changements de MAE sont linéaires et donc intuitifs. Autrement dit, MSE et RMSE punissent les erreurs importantes plus que les erreurs plus petites, gonflant ou amplifiant le score d'erreur moyen. Cela est dû au carré de la valeur d'erreur. Le MAE ne donne pas de poids aux différents types d'erreurs et au contraire, les scores augmentent linéairement avec les augmentations de l'erreur. La fonction de MAE :

$$MAE = \frac{1}{N} * \sum_i^n |(y_i - yhat_i)|$$

Figure 3.32: Calcul de la mesure de performance MAE

3.6.4 Evaluation des algorithmes utilisés

Contrairement aux problèmes de classification, la performance des modèles de régression ne se mesure ni avec la précision, ni le rappel, ni la proportion des prédictions vraies, etc. La régression a ses propres métriques de performance, dont notamment le MSE, le RMSE et le MAE expliquées ci-dessus. Pour comparer nos algorithmes, nous avons choisi le RMSE, parce que son unité est la même que celle de la variable cible, ce qui le rend encore plus représentatif.

Tableau 3.4: Etude comparative des algorithmes.

Algorithme	MSE	RMSE	MAE
KNN	27.01	5.19	3.35
SVR	37.24	6.10	4.69
Random Forest	14.36	3.79	2.50
XGBoost	8.22	2.86	1.75

D'après le tableau et le graphe précédents, nous allons retenir le modèle XGBoost qui a le RMSE le plus bas (2.86).

3.6.5 Visualisation des résultats de prédictions pour l'algorithme retenu XGBoost

Nous allons visualiser les résultats de l'algorithme retenu en vue d'observer davantage son comportement.

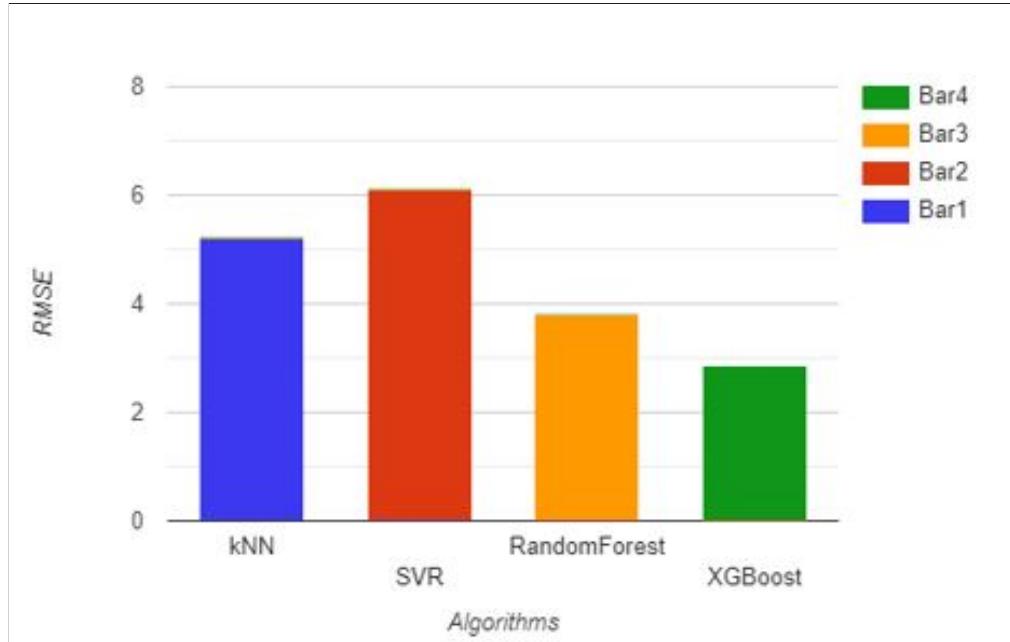


Figure 3.33: Graphe de comparaison entre les RMSE des algorithmes

- **Ajustement des prédictions avec les observations :**

- L'axe des x est le nombre de ventes et l'axe des y est le nombre de ventes prédits ;
- Les nombres de ventes sont triés dans un ordre ascendant ;
- Idéalement, la différence entre le nombre de ventes prédit et le nombre de vente observé serait nulle.

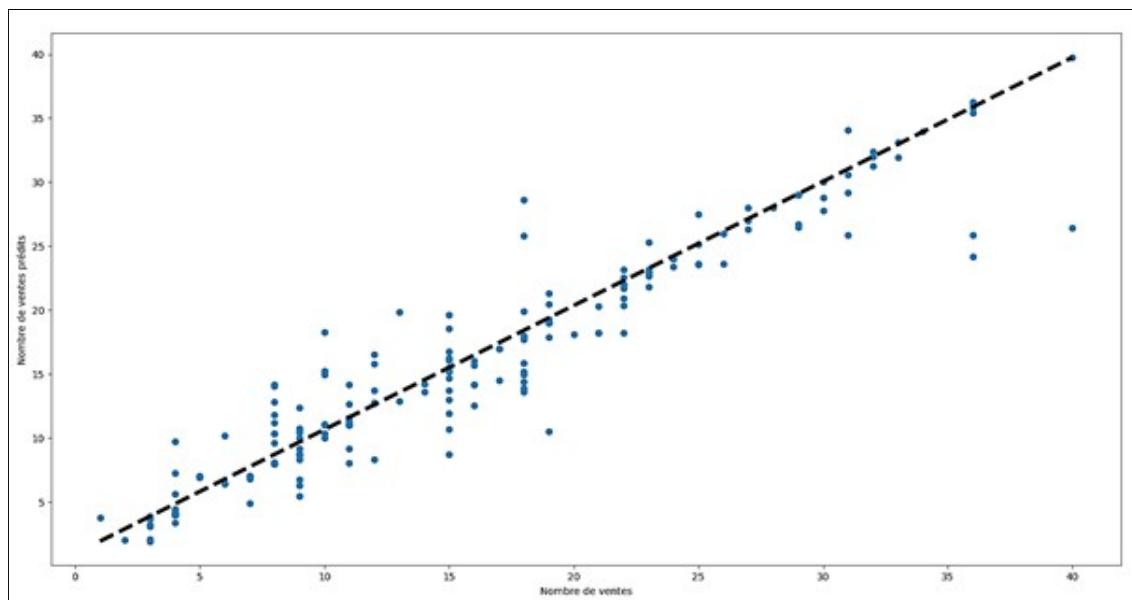


Figure 3.34: Le nombre de ventes VS le nombre de ventes prédit

Cela explique la courbe linéaire de pente positive qui décrit une différence nulle idéale. Toutes

les prédictions sont bien ajustées avec les observations, sauf pour quelques prédictions marginales.

- **Les résidus de la régression :** Les résidus de la régression, appelés « Residuals », représentent la différence entre la valeur observée et la valeur prédite. L'analyse des résidus joue un rôle important dans la validation du modèle d'apprentissage automatique.

Le graphe 3.35 montre que la densité des résidus est très importante aux alentours d'une différence nulle entre les observations et les prédictions. Nous pouvons aussi voir que la densité diminue au fur et à mesure que $y_{train} - y_{train_pred}$ augmente. La densité des résidus dans l'algorithme XGBoost est favorable, nous permettant de valider ce modèle. La raison pour laquelle les résultats sont aussi avantageux est que XGBoost est une méthode ensembliste qui associe la puissance des arbres de décisions avec l'optimisation apportée par l'algorithme Gradient Descent Boost. Sans oublier le tuning des hyperparamètres qui a évidemment amélioré les résultats de l'évaluation.

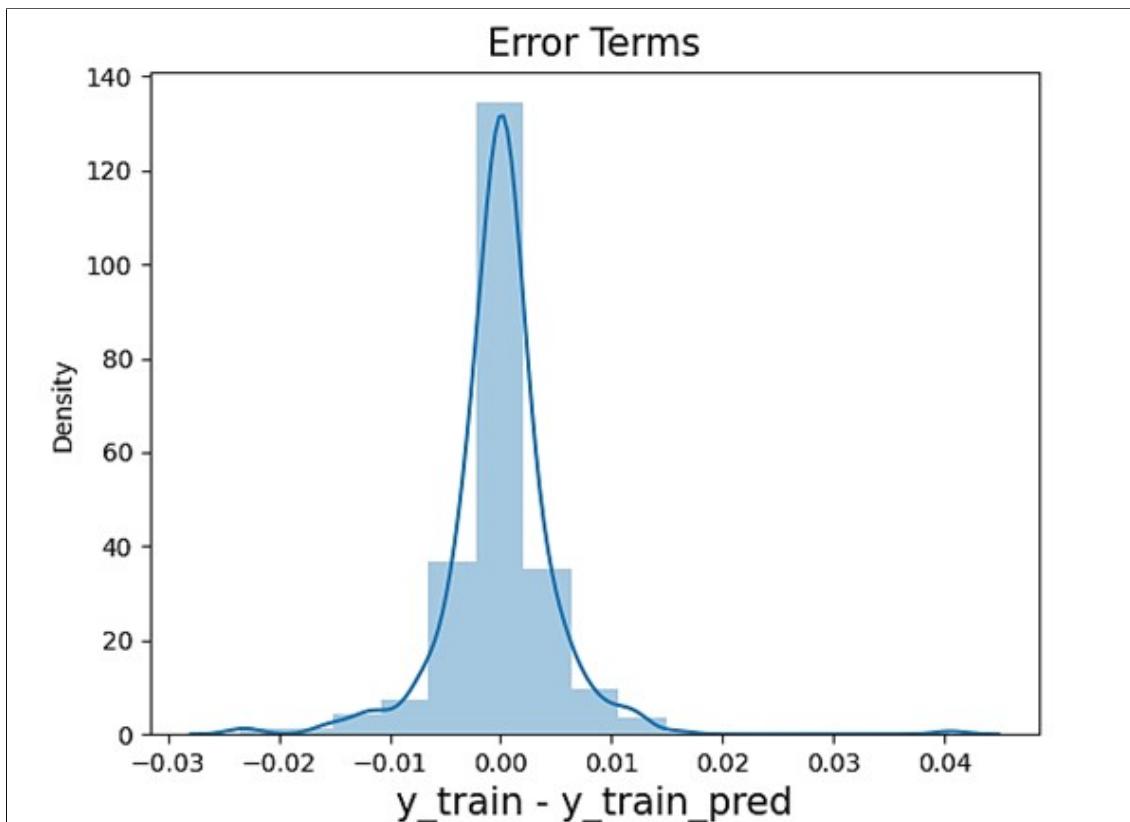


Figure 3.35: La densité des résidus de XGBoost

Ce graphe montre que la densité des résidus est très importante aux alentours d'une différence nulle entre les observations et les prédictions. Nous pouvons aussi voir que la densité diminue au fur et à mesure que $y_{train} - y_{train_pred}$ augmente.

La densité des résidus dans l'algorithme XGBoost est favorable, nous permettant de

valider ce modèle.

La raison pour laquelle les résultats sont aussi avantageux est que XGBoost est une méthode ensembliste qui associe la puissance des arbres de décisions avec l'optimisation apportée par l'algorithme Gradient Descent Boost. Sans oublier le tuning des hyperparamètres qui a évidemment amélioré les résultats de l'évaluation.

SPRINT3 : Déploiement

Dans ce chapitre, nous allons déployer les modèles d'apprentissage automatique que nous avons réalisés dans les chapitres précédents. Notre projet étant composé de deux livrables majeurs, nous avons construits deux releases. Le premier livrable, un système de détection automatique de l'offre de formation adéquate à chaque agent, sera déployé sous la forme d'un planning de formations mis à jour à la fin de chaque mois. Le deuxième livrable est le système de prédiction du nombre de ventes qu'un agent déterminé va éventuellement réaliser, déployé sous la forme d'un formulaire qui sera rempli par le superviseur pour faire les prédictions nécessaires sur ledit agent. D'autant plus, le déploiement comportera aussi des tableaux de bord où les superviseurs pourront avoir une vision globale sur l'évolution de la production et des formations.

3.7 Analyse des besoins

Avant de se lancer dans le déploiement de notre projet, il est indispensable que nous en ayons bien compris le besoin. Beaucoup de projets dérivent voire échouent à cause d'une définition des besoins approximative ou partielle. Nous devons donc recueillir avec précision les attentes de parties prenantes sur le système délivré. Notre démarche était d'enquêter auprès des superviseurs et des administrateurs pour identifier le déploiement dont ils avaient besoins. Etant déjà familiers avec les systèmes d'information, ils primaient l'efficacité, mais surtout la solution ergonomique et facile.

3.7.1 Identification des acteurs de l'application

Les acteurs sont les personnes qui interagiront avec notre système. Puisque la solution conçue va être utilisée pour le suivi des formations et de la productivité des agents de 1WayCom, nos acteurs sont exclusivement les superviseurs.

Les superviseurs :

Les superviseurs sont les supérieurs hiérarchiques directs des agents téléopérateurs. Ils sont chargés de faire des écoutes régulières sur les agents de leurs groupes, d'évaluer leur productivité et leur qualité. Cela veut dire que notre solution les concerne directement

3.7.2 Identification des besoins fonctionnels et non fonctionnels

Les besoins fonctionnels expriment les attentes des utilisateurs finaux du système conçu. Dans notre cas, il s'agit de la réalisation d'un système qui permettra aux utilisateurs de comprendre les résultats de notre apprentissage automatique et de faire des prédictions par eux-mêmes.

- **Les besoins fonctionnels du superviseur :**

- Déetecter les offres de formations.
- Faire la prédiction du nombre de vente pour un agent.
- Consulter le suivi des formations.
- Consulter le suivi de la production.

Les besoins non fonctionnels doivent être considérés dans tous les projets. Pour certains projets, ces besoins demanderont un travail important. Pour le nôtre, un simple contrôle sera suffisant, puisque notre solution sera implantée dans l'ERP de l'entreprise, qui respecte déjà tous les besoins non fonctionnels. Comme nous l'avons déjà mentionné, les parties prenantes privilégient avant tout une solution efficace et facile.

- **Identification des besoins non fonctionnels :**

- **Sécurité** : Le CRUD, la connexion, la création de compte, etc. sont déjà assurés et sécurisés dans l'ERP existant de 1WayCom. Nous allons seulement nous préoccuper de la sécurité du formulaire de prédiction que nous allons créer.
- **Performance** : La performance implique essentiellement le temps de chargement et le temps de traitement dans une application. Notre solution doit être rapide.
- **Disponibilité** : La disponibilité décrit l'aptitude d'un système à être disponible lorsqu'il est interpellé par son utilisateur. .
- **Ergonomie** : L'ergonomie offre à l'utilisateur les meilleures conditions de travail en terme d'adaptabilité, d'efficacité et de facilité d'utilisation.
- **Fiabilité** : Le système construit doit être fiable. Pour notre cas, cela se rapporte essentiellement à la fiabilité des modèles construits dans l'apprentissage automatique.

3.7.3 Diagramme des cas d'utilisation

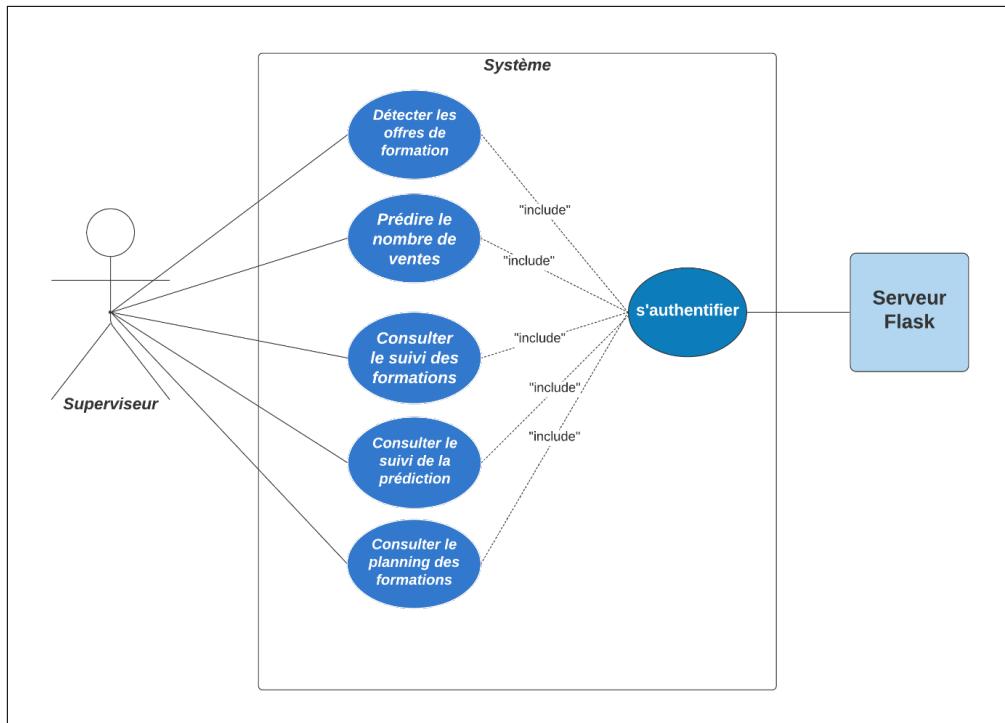


Figure 3.36: Diagramme des cas d'utilisation

3.7.4 Cas d'utilisation « DéTECTER les offres de formations »

- Description textuelle :

Tableau 3.5: Description textuelle du cas d'utilisation "DéTECTER les offres de formations".

Cas d'utilisation	DéTECTER les offres de formation
Acteur	Superviseur
Précondition	Le superviseur s'est authentifié
Post-condition	Mise à jour du planning des formations
Scénario nominal	1-Le superviseur demande d'accéder à l'interface "planning des formations" 2-Le système affiche l'interface demandée 3-Le superviseur appuie sur le bouton « détecter les formations » 4- Le système demande l'exécution de l'algorithme. 5-Le modèle d'apprentissage automatique fait son calcul et retourne le résultat au système. 6-Le système met à jour le planning des formations
Scénario alternatif	5-a : le modèle d'apprentissage automatique ne détecte pas de nouvelles formations. 5-b : afficher une message au superviseur.

- Diagramme de séquence système :

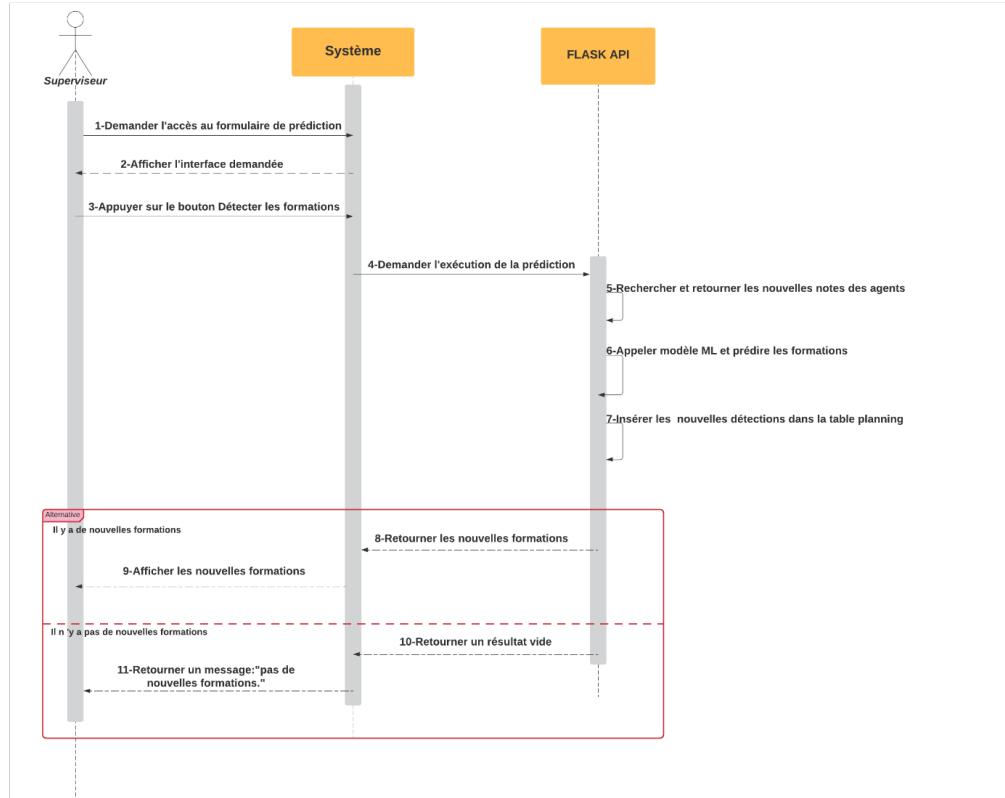


Figure 3.37: Diagramme de séquence du cas d'utilisation "Déetecter les offres de formations"

3.7.5 Cas d'utilisation « Prédire le nombre de ventes par agent »

- Description textuelle :

Tableau 3.6: Description textuelle du cas d'utilisation "Prédire le nombre de ventes par agent.

Cas d'utilisation	Prédire le nombre de ventes par agent
Acteur	Superviseur
Précondition	Le superviseur s'est authentifié
Post-condition	Affichage du nombre de ventes prédit
Scénario nominal	<p>1-Le superviseur demande d'accéder à l'interface "Prédiction du nombre de ventes "</p> <p>2-Le système affiche l'interface demandée</p> <p>3-Le superviseur choisit un agent et sélectionne le mois sur lequel il désire faire la prédiction</p> <p>4-Le système vérifie les données.</p> <p>5-Le système remplit automatiquement les champs nombre d'heures travaillées, nombre de ventes précédent, score de qualification et notation à partir de la base de données.</p> <p>6-Le superviseur peut changer les données remplies automatiquement ou appuyer sur le bouton prédire.</p> <p>7-Le système envoie les données au module d'apprentissage automatique de prédiction et demande l'exécution de l'algorithme.</p> <p>8-Le module d'apprentissage automatique fait son calcul et retourne le résultat au système.</p> <p>9-Le système affiche sur l'interface le nombre de ventes prédit.</p>
Scénario alternatif	<p>-4a : L'agent en question est récemment recruté et n'a pas de nombre de ventes précédent.</p> <p>-4a1 : Le système affiche un message d'erreur en pop-up.</p> <p>-4a2 : La séquence nominale retourne à la troisième étape.</p> <p>-5a : Le superviseur entre des données invalides quand il change les champs remplis automatiquement.</p> <p>-5a1 : Le système affiche un message d'erreur en pop-up.</p> <p>-5a2 : La séquence nominale retourne à l'étape 3.</p>

- Diagramme de séquence système :

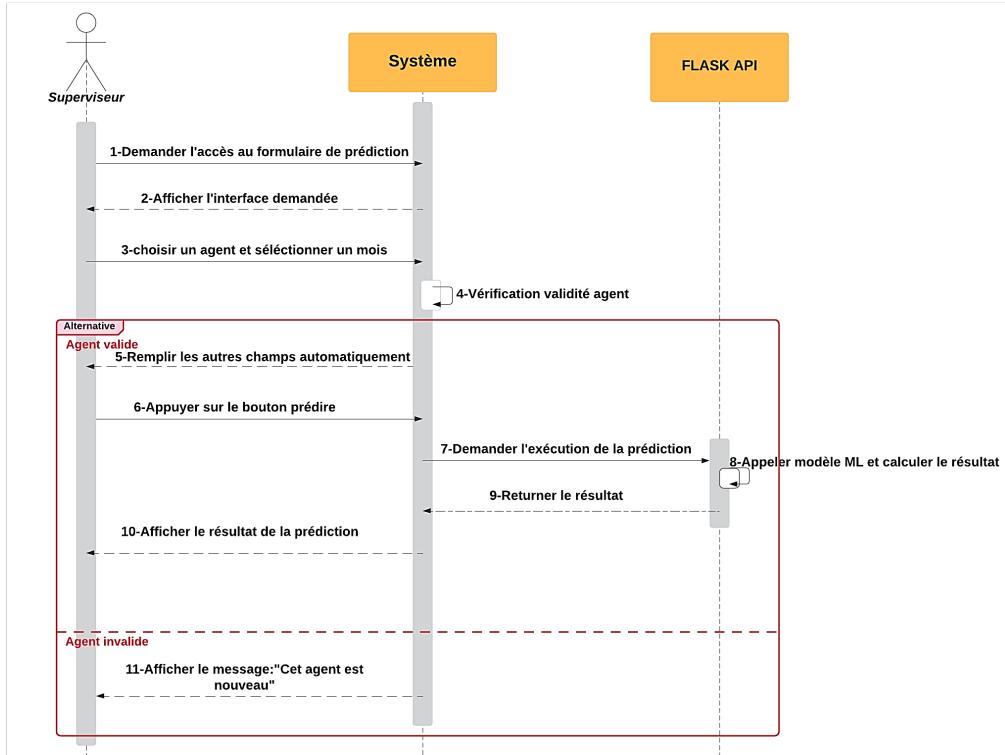


Figure 3.38: Diagramme de séquence système du cas d'utilisation "Prédire le nombre de ventes par agent"

3.7.6 Cas d'utilisation « Consulter le suivi des formations »

- Description textuelle :

Tableau 3.7: Description textuelle du cas d'utilisation 'Consulter le suivi des formations'.

Cas d'utilisation	Consulter le suivi des formations
Acteur	Superviseur
Précondition	Le superviseur s'est authentifié.
Post-condition	Afficher le tableau de bord des formations.
Scénario nominal	1-L'administrateur demande d'accéder à l'interface "Suivi formations". 2-Le système affiche l'interface demandée. 3-L'administrateur consulte le tableau de bord.
Scénario alternatif	

- Diagramme de séquence système :

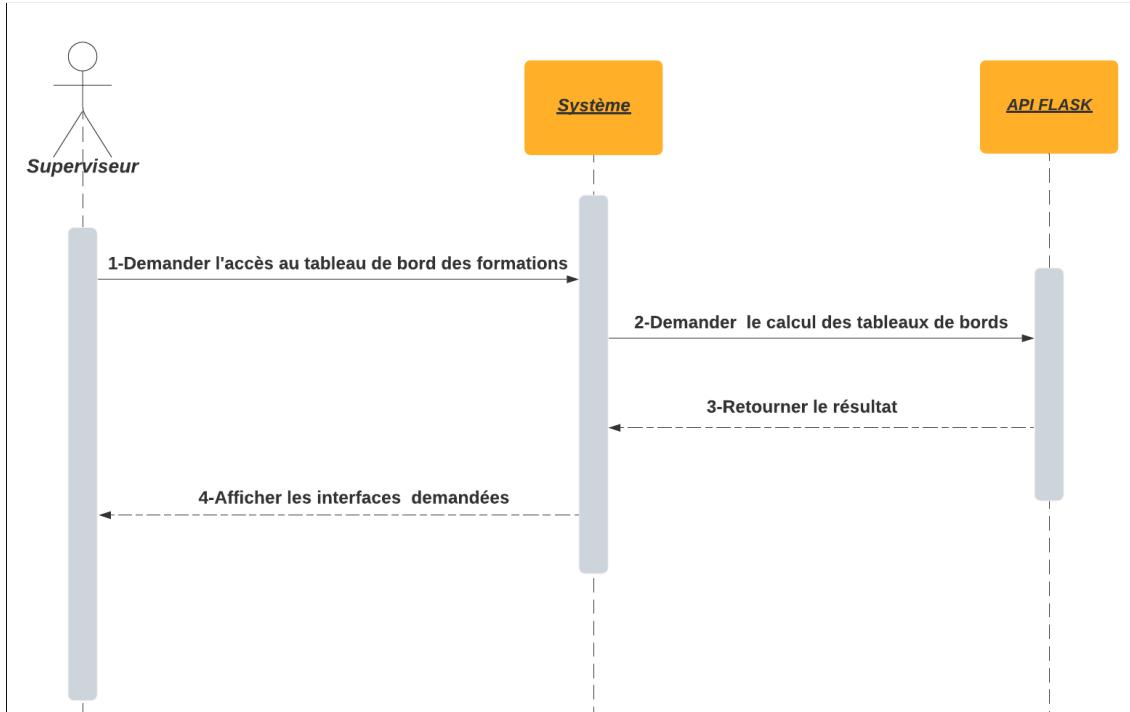


Figure 3.39: Diagramme de séquence système du cas d'utilisation "Consulter le suivi des formations"

3.7.7 Cas d'utilisation « Consulter le suivi de la production »

- Description textuelle :

Tableau 3.8: Description textuelle du cas d'utilisation "Consulter le suivi de la production".

Cas d'utilisation	Consulter le tableau de bord de la production
Acteur	Superviseur
Précondition	Le superviseur s'est authentifié
Post-condition	Affichage du tableau de bord de la production
Scénario nominal	1-L'administrateur demande d'accéder à l'interface "Suivi production" 2-Le système affiche l'interface demandée 3-L'administrateur consulte le planning des formations de son groupe
Scénario alternatif	

- Diagramme de séquence système :

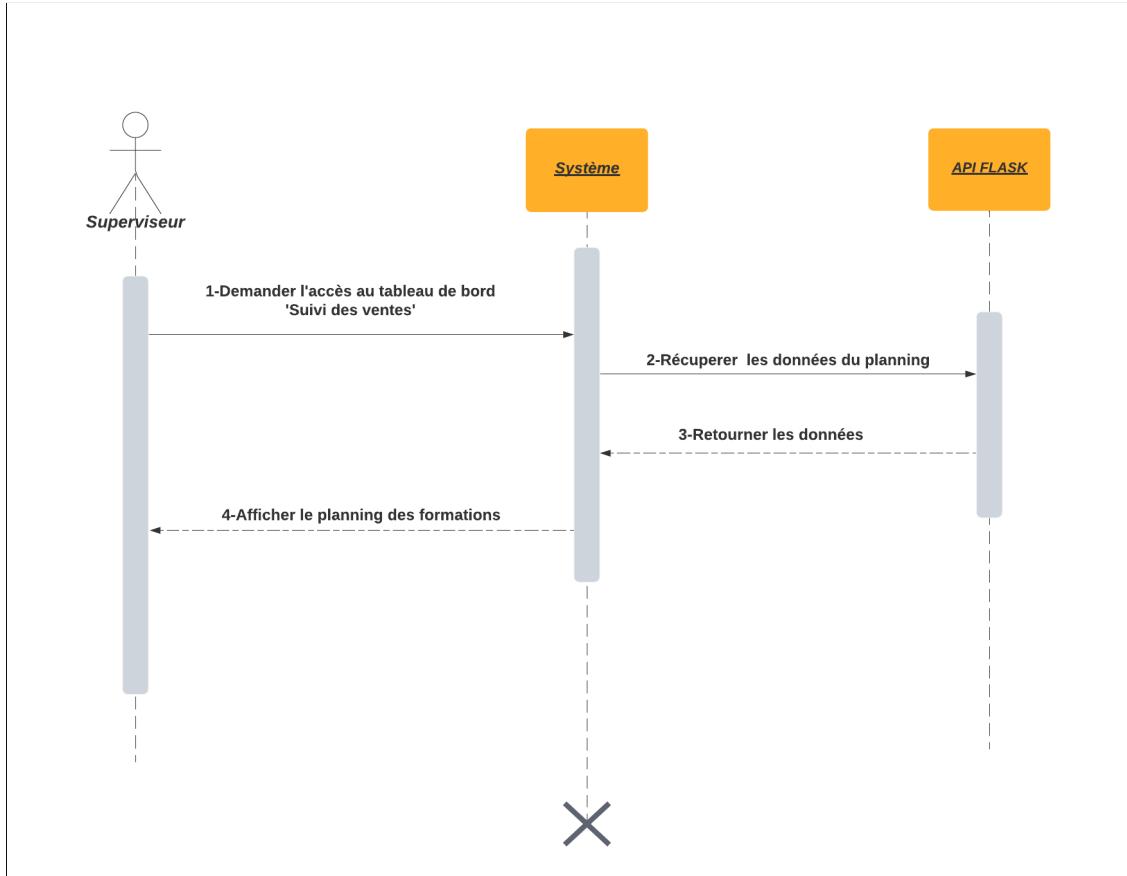


Figure 3.40: Diagramme de séquence du cas d'utilisation "Consulter le suivi de la production"

3.7.8 Cas d'utilisation « Consulter le planning des formations »

- Description textuelle :

Tableau 3.9: Description textuelle du planning des formations.

Cas d'utilisation	Consulter le planning des formations
Acteur	Superviseur
Précondition	Le superviseur s'est authentifié.
Post-condition	Afficher le planning des formations
Scénario nominal	1-L'administrateur demande d'accéder à l'interface "Planning des formations". 2-Le système affiche l'interface demandée. 3-L'administrateur consulte le planning des formations.
Scénario alternatif	

- Diagramme de séquence système :

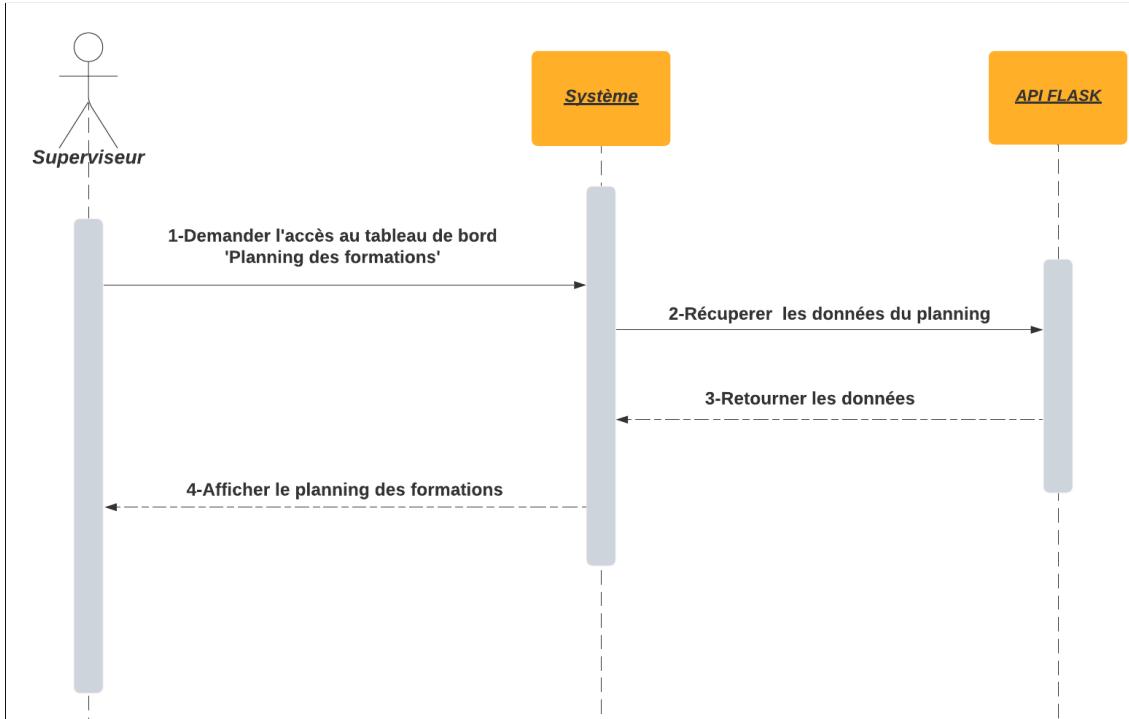


Figure 3.41: Diagramme de séquence du cas d'utilisation "Consulter le planning des formations"

3.8 Conception

La conception permet de synthétiser l'étape de l'analyse des besoins pour se préparer à la phase de déploiement.

3.8.1 Diagramme de classes participantes

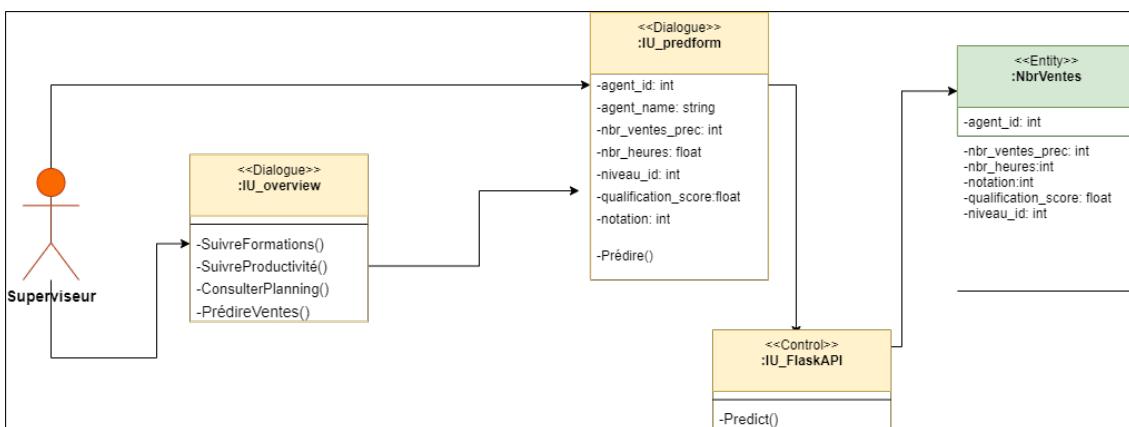


Figure 3.42: Diagramme de classes participante predire le nombre de ventes

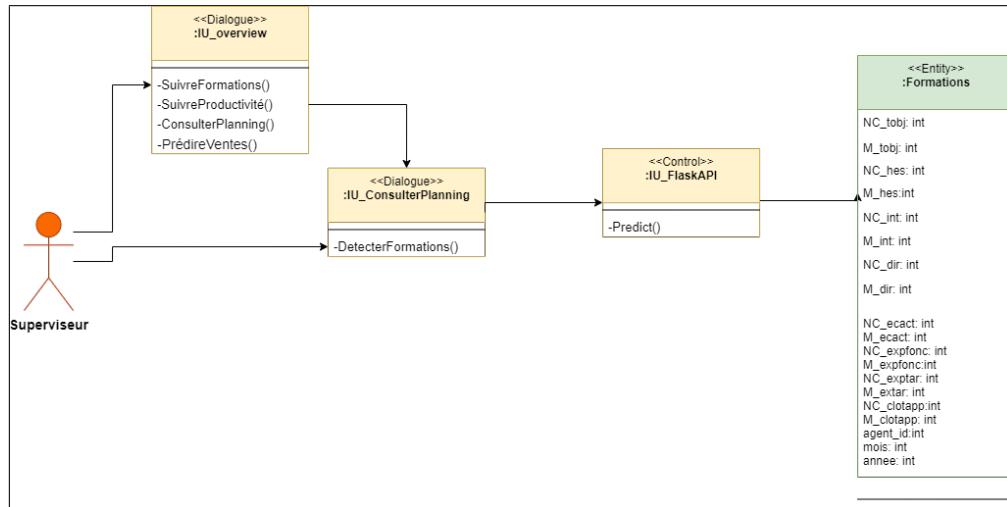


Figure 3.43: Diagramme des classes participantes du cas d'utilisation détecter la formation

3.8.2 Diagramme de séquence détaillé

- Cas d'utilisation « prédire le nombre de ventes » :

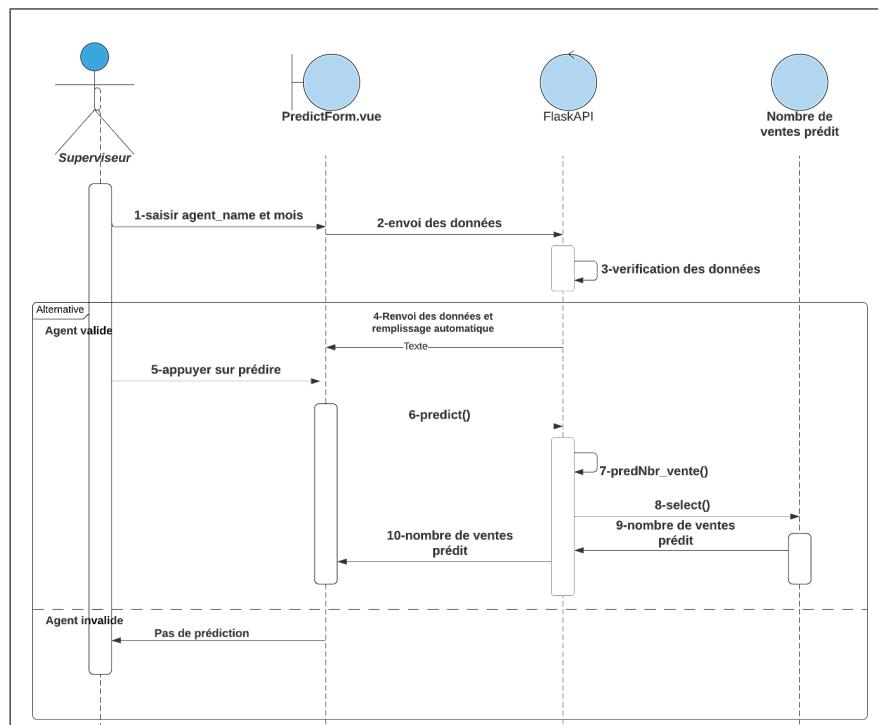


Figure 3.44: diagramme de séquence détaillé

3.8.3 Diagramme de classes

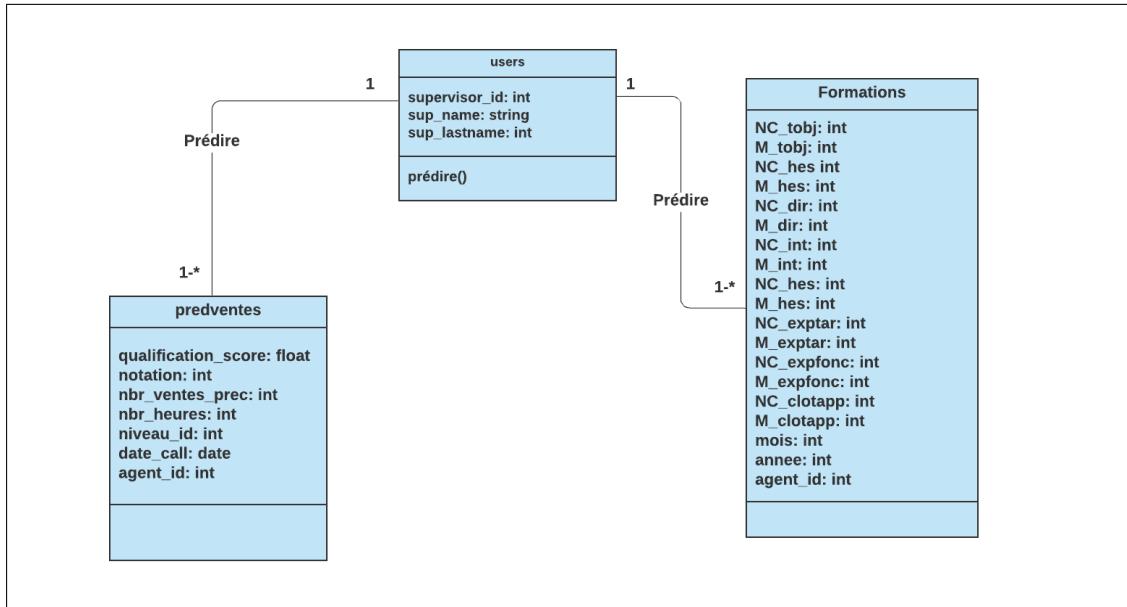


Figure 3.45: Diagramme de classes

3.9 Déploiement

Pour la phase de déploiement nous allons tout d'abord présenter la base de données utilisée, ensuite les interfaces développées.

3.9.1 Base de données

Nous avons utilisé les mêmes tables formations et prednbr de la base de données présentée dans la première et deuxième release.

3.9.2 Les interfaces

- Tableau de bord du suivi de la production :

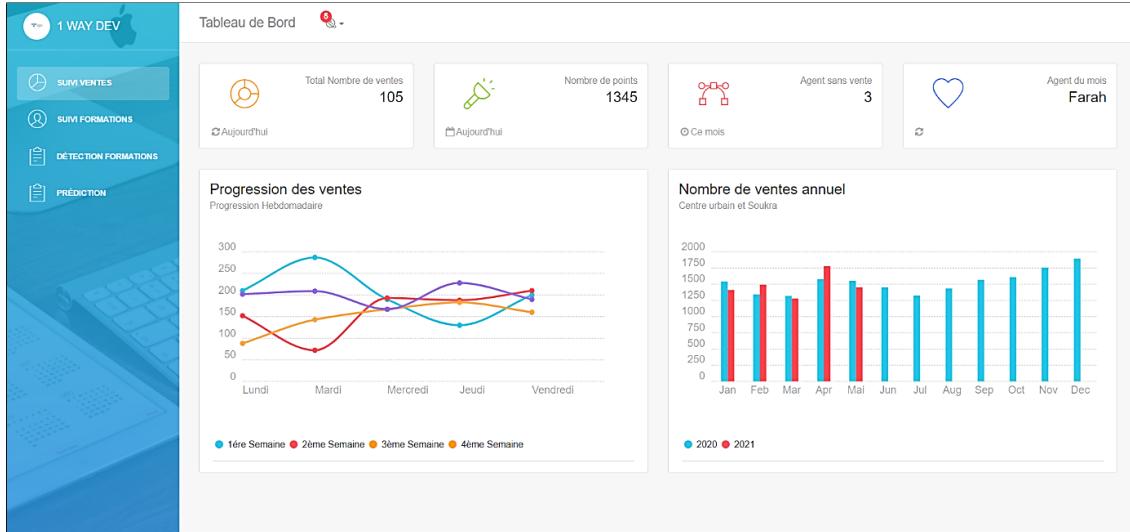


Figure 3.46: Interface suivi production

- Tableau de bord du suivi des formations :

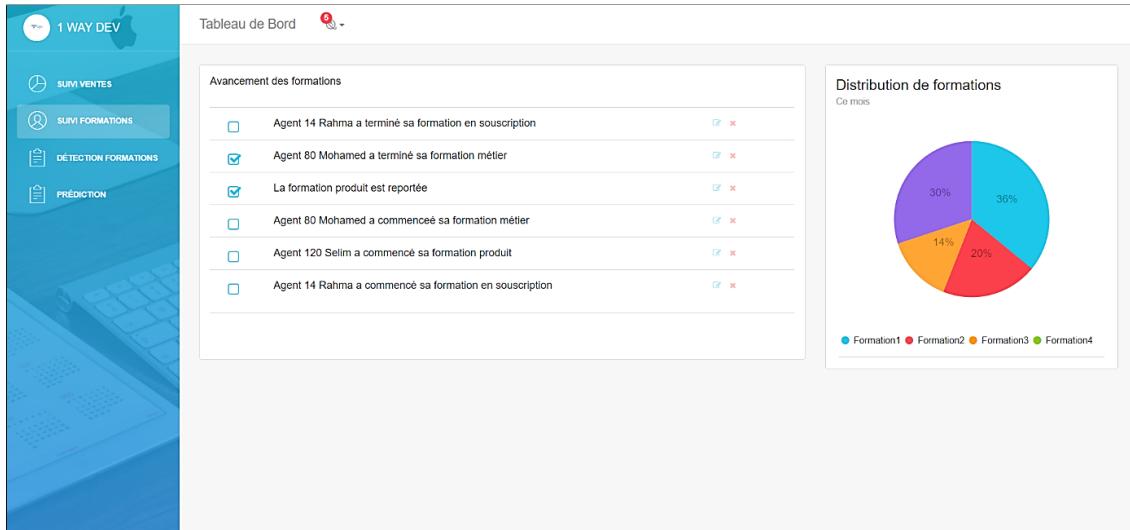


Figure 3.47: Interface suivi formations

- Interface de détection des offres de formation :

The screenshot shows a dashboard titled 'Tableau de Bord'. On the left sidebar, there are links for 'SUIV VENTES', 'SUIVI FORMATIONS', 'DÉTECTION FORMATIONS' (which is highlighted in blue), and 'PRÉDICTION'. The main content area is titled 'Liste des formations détectées pour ce mois!'. It includes a sub-instruction: 'Formations1 -> Métier / Formations2 -> Produit / Formations3 -> Souscription / Formations4 -> Traitement des objections'. Below this is a button labeled 'Détecter les nouvelles formations'. A table lists 7 training entries:

#	NOM AGENT (ID)	FORMATION1	FORMATION2	FORMATION3	FORMATION4	DATE DÉBUT	DATE FIN	Débuter	Finir
0	Marwen TRIKI (211)	Oui	-	-	Oui	Fri, 01 Jan 2021 00:00:00 GMT	Thu, 07 Jan 2021 00:00:00 GMT	<button>Débuter</button>	<button>Finir</button>
1	Safa SAID (144)	-	-	-	-	Fri, 01 Jan 2021 00:00:00 GMT	Thu, 07 Jan 2021 00:00:00 GMT	<button>Débuter</button>	<button>Finir</button>
2	Sabri AOUADHI (215)	-	-	-	-	Mon, 04 Jan 2021 00:00:00 GMT	Tue, 12 Jan 2021 00:00:00 GMT	<button>Débuter</button>	<button>Finir</button>
3	Nouha SMEDHI (30)	Oui	-	-	-	Sun, 03 Jan 2021 00:00:00 GMT	Tue, 12 Jan 2021 00:00:00 GMT	<button>Débuter</button>	<button>Finir</button>
4	Hana SAIDI (18)	Oui	-	-	Oui	Wed, 06 Jan 2021 00:00:00 GMT	Tue, 12 Jan 2021 00:00:00 GMT	<button>Débuter</button>	<button>Finir</button>
5	Nazim SAADI (235)	-	-	-	-	Sat, 09 Jan 2021 00:00:00 GMT	Fri, 15 Jan 2021 00:00:00 GMT	<button>Débuter</button>	<button>Finir</button>
6	Cyrine RIAHI (52)	-	Oui	Oui	Oui	Tue, 12 Jan 2021 00:00:00 GMT	Fri, 15 Jan 2021 00:00:00 GMT	<button>Débuter</button>	<button>Finir</button>
7	Mayssa MEJRI (219)	Oui	-	-	-	Wed, 13 Jan 2021 00:00:00 GMT	Tue, 19 Jan 2021 00:00:00 GMT	<button>Débuter</button>	<button>Finir</button>

Figure 3.48: Interface de détection des offres de formation du mois

The screenshot shows the same dashboard layout as Figure 3.48. However, a modal dialog box is centered over the content area. The dialog has a yellow exclamation mark icon at the top. The text inside reads: 'Pas de formations détectées pour ce mois' (No trainings detected for this month). At the bottom of the dialog is a blue 'OK' button.

Figure 3.49: Aucun nouveau besoin en formation dans le mois de détection

- Formulaire de prédiction du nombre de ventes :

Prédire Nombre de ventes

Hichem MANSOURI	6 Juin	NOMBRE D'HEURES TRAVAILLÉES	NOMBRE DE VENTES PRÉCÉDENT
AGENT ID	20	250	26
NIVEAU AGENT	1	NOTATION	SCORE DE QUALIFICATION
	5		90

Predire

Right panel shows user profile for Hichem Mansouri (Agent ID: 20, Expert level, LinkedIn icon).

Figure 3.50: Interface de prédiction du nombre de ventes

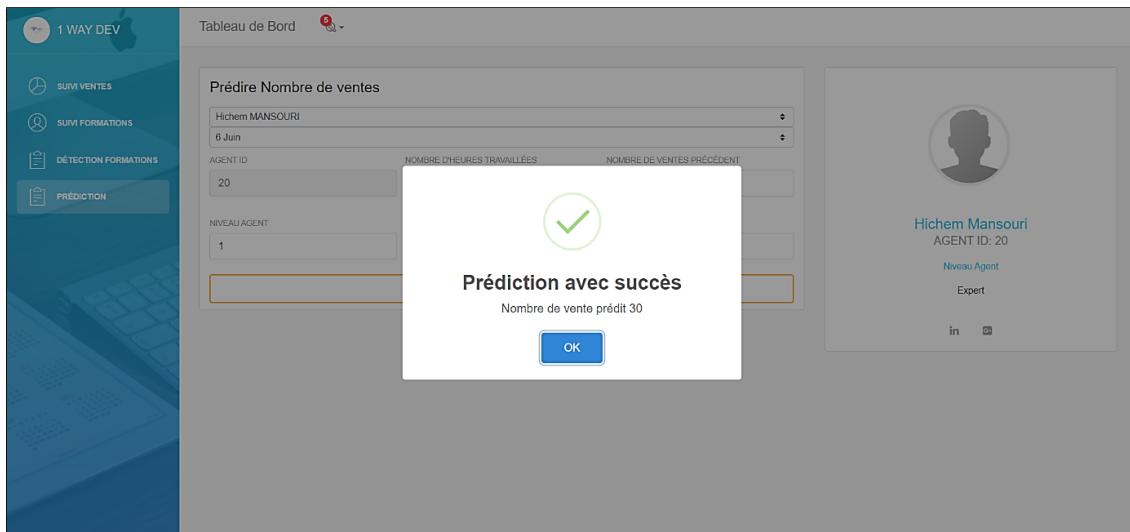


Figure 3.51: Affichage du nombre de ventes prédit

3.10 Phase de clôture

Dans la phase de clôture, nous allons présenter l'architecture physique utilisée ainsi que les outils de développement et de Machine Learning.

3.10.1 Architecture physique adoptée et intégration du Machine Learning

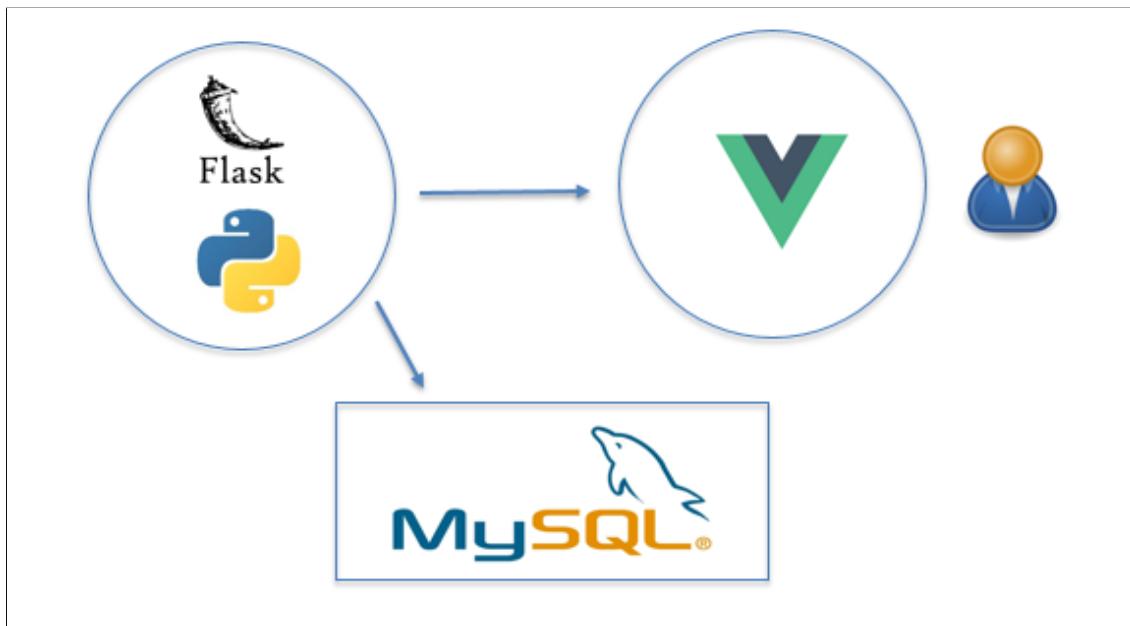


Figure 3.52: Architecture adoptée

Nous avons un modèle d'apprentissage automatique que nous voulons utiliser à partir de l'application web de l'entreprise, sachant que notre modèle est écrit en Python. Pour cela, nous devons créer une API pour notre modèle, avec laquelle l'utilisateur web peut interagir. Notre API générera donc des prédictions à partir du modèle d'apprentissage automatique que nous avons créé. Pour cela, nous avons tout d'abord construit et validé notre modèle. Ensuite, nous avons enregistré notre modèle à l'aide de la librairie Python pickle dans la fonction dump(). Pour construire notre API, nous avons utilisé le micro-framework Flask. Nous avons importé la librairie pickle et nous avons chargé notre modèle précédemment enregistré dans dump() avec la fonction load(). Ensuite, nous avons créé la fonction predict qui fait la prédiction et qui représente la ressource de notre API. Après cela notre modèle devient accessible à partir du web (pour le moment en réseau local).

3.10.2 Outils utilisés dans le déploiement

- **MySQL :**

MySQL est un système de gestion de bases de données relationnelles. Il permet de stocker les données dans des tables, et permet la manipulation à travers le langage de requêtes SQL. MySQL peut être intégré en python à travers le module pymysql.



Figure 3.53: Logo de MySQL

- **Python :**

Python est un langage de programmation orienté objet très populaire. Il est actuellement le langage le plus utilisé dans le monde, grâce à sa simplicité et sa facilité de maintenance. Python est un langage de programmation multifonctions et peut être utilisé dans beaucoup de domaines.

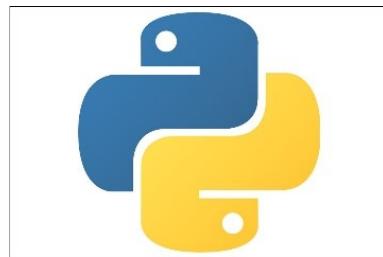


Figure 3.54: Logo de Python :

- **Flask :**

Flask est un micro Framework qui est généralement utilisé pour créer des applications web faciles à partir de Python. Le caractère léger et facile de Flask lui fait gagner beaucoup de popularité. Il est très utilisé par les data scientifiques pour déployer leur apprentissage automatique dans des pages web facilement créées grâce aux bibliothèques implémentées, en évitant de passer par des étapes complexes.

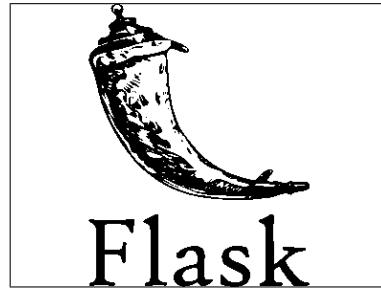


Figure 3.55: Logo de Flask

- **Le Framework Vue JS**

Vue js est un Framework open source en langage JavaScript, très utilisé pour la création d'application web complexes et monopages.



Figure 3.56: Logo de VueJS

- **Atom**

Atom est un éditeur de texte libre et très populaire, construit en HTML, CSS, JavaScript et Node Js.

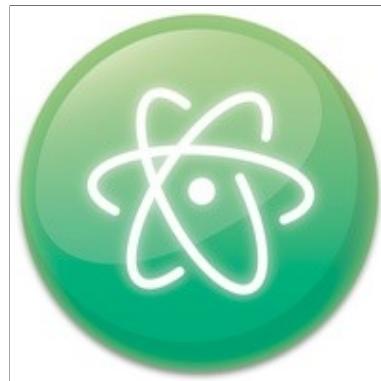


Figure 3.57: Logo d' Atom

3.10.3 Outils utilisés dans le machine learning

- **Anaconda**

Anaconda est une distribution libre et open source des langages de programmation Python et R appliquée au développement d'applications dédiées à la science des données et à l'apprentissage automatique, qui vise à simplifier la gestion des paquets et de déploiement.



Figure 3.58: Logo d'Anaconda

- **Spyder**

Un environnement de développement pour Python. Libre et multiplateforme, il intègre de nombreuses bibliothèques d'usage scientifique.

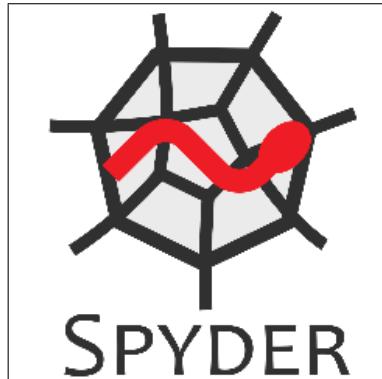


Figure 3.59: Logo de Spyder

Conclusion

Dans ce chapitre, nous avons construit le deuxième livrable de notre projet qui est le module d'apprentissage automatique permettant de prédire le nombre de ventes qu'un agent va réaliser dans un mois donné. Ensuite, nous avons procédé à la conception et au déploiement de nouvelles interfaces qui seront intégrées dans l'ERP de l'entreprise, et qui permettent aux superviseurs d'exploiter les modèles d'apprentissages automatique que nous avons conçus. **test**

Conclusion générale

Pour couronner trois années de formation, nous voulions trouver un sujet de projet de fin d'études qui nous instruise sur les technologies à la pointe de la modernité. La recherche d'un stage a été très difficile dans la mesure où elle s'orientait totalement dans cette direction. Repérer ce projet en Data Science répondait à notre objectif et présentait pour nous un réel défi. Nous n'avions pas toutes les connaissances nécessaires et il nous fallait un travail d'étude exhaustif et surtout autodidacte pour maîtriser le domaine. Lorsque nous avons entamé notre travail, nous étions confrontés à maintes reprises à des notions délicates, et nous nous sommes rendus compte que notre projet avait des difficultés supplémentaires, de par la taille des données, l'obligation de travailler avec deux techniques de prédictions différentes, la Data Set déséquilibrée, etc. Nous nous sommes efforcés de traiter toutes les particularités de ces problèmes et de construire un travail rigoureux, malgré les contraintes de temps et les entraves qui se succédaient. Ce projet nous a permis de réaliser que la persistance, la volonté et la confiance en soi représentent le moteur de la réussite. Ce travail ouvre de larges perspectives d'amélioration. Comme première perspective, notre modèle d'apprentissage automatique de classification doit être alimenté par des données que l'entreprise est entraînée à collecter dans une base de données historique. Cela va ajouter de nouvelles variables prédictives à l'algorithme retenu XGBoost, pour qu'il s'adapte encore plus aux besoins en formation. Par ailleurs, nous pouvons construire un nouveau modèle qui remplacera la procédure de recrutement traditionnelle de 1WayCom, avec des algorithmes qui classifieront les CV envoyés aux ressources humaines selon les compétences requises. Cela donnera lieu à un recrutement optimisé et dépourvu de toutes ségrégations. Une autre perspective est de prédire le chiffre d'affaire annuel de l'entreprise, en se basant sur les patterns tracés dans des unités temporelles, grâce à la technique de prédiction appelée Time Series, le terme anglais pour les séries chronologiques. Cela est très intéressant pour les dirigeants de l'organisme qui pourront se situer par rapport à l'évolution de l'entreprise et qui seront motivés par la réalisation de ces attentes devenues scientifiques grâce aux algorithmes, voire même de les surpasser.

Bibliographie

- [1] (Raphaël). « Données quantitatives continues. » [Accès le 21 Avril 2021], adresse : <https://blocnotes.iergo.fr/breve/categorielles-quantitative-discrete-ou-continue/#:~:text=Une%20variable%20quantitative%20peut%20%C3%AAtre,valeurs%2C%20formant%20un%20ensemble%20continu>.
- [2] (Raphaël). « Données qualitatives nominales. » [Accès le Avril 2021], adresse : <https://blocnotes.iergo.fr/breve/nominales-ordinales-intervalles-et-ratios/>.
- [3] (Khan Academy). « La Moyenne. » [Accès le 25 Avril 2021], adresse : <https://fr.khanacademy.org/math/be-2eme-secondaire2/x291d358f50a246d9:traitement-de-donnees-1/x291d358f50a246d9:determiner-un-effectif-un-mode-une-frequence-la-moyenne-arithmetique-le-tende-dun-ensemble-de-donnees-discretes/a/mean-median-and-mode-review>.
- [4] (scientific sentence). « Quartile. » [Accès le 29 Avril 2021], adresse : <https://scientificsentence.net/Equations/Maths2/statistiques/index.php?key=yes&Integer=mediane>.
- [5] (plotly). « Matplotlib. » [Accès le 5 Mai 2021], adresse : <https://plotly.com/>.
- [6] (Altair). « Altair. » [Accès le 5 Mai 2021], adresse : <https://altair-viz.github.io/>.
- [7] (Data camp). « Paramètre. » [Accès le 25 Avril 2021], adresse : https://www.datacamp.com/community/tutorials/parameter-optimization-machine-learning-models?utm_source=adwords_ppc&utm_campaignid=1455363063&utm_adgroupid=65083631748&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adpostion=&utm_creative=3326.
- [8] (monkeylearn). « SVM. » [Accès le 4 Mai 2021], adresse : <https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/#:~:text=How%20Does%20SVM%20Work%3F,%20The%20basics%20of%20&text=A%20support%20vector%20machine%20takes,to%20the%20other%20as%20red...>
- [9] (medium). « KNN. » [Accès le 4 Mai 2021], adresse : <https://medium.com/capital-one-tech/k-nearest-neighbors-knn-algorithm-for-machine-learning-e883219c8f26>.
- [10] (opengenus). « Gaussian Naive Bayes. » [Accès le 3 Mai 2021], adresse : <https://iq.opengenus.org/gaussian-naive-bayes/>.

Bibliographie

- [11] (sciencedirect). « Accuracy. » [Accès le 4 Mai 2021], adresse : <https://www.sciencedirect.com/topics/engineering/classification-accuracy#:~:text=Classification\%20accuracy\%20is\%20simply\%20the,of\%20the\%20cross\%2Dvalidation\%20idea>.
- [12] (towards data science). « Precision. » [Accès le 5 Mai 2021], adresse : <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>.
- [13] (medium). « SVR. » [Accès le 7 Mai 2021], adresse : <https://medium.com/coinmonks/support-vector-regression-or-svr/>.
- [14] (machine learning mastery). « Mean Squared Error. » [Accès le 7 Mai 2021], adresse : <https://machinelearningmastery.com/regression-metrics-for-machine-learning/>.

Annexes

Techniques de l'analyse prédictive

L'analyse prédictive est une branche de la data science qui est essentiellement utilisée pour prédire des résultats ou des évènements futurs.

- **Apprentissage supervisé :**

C'est une série de fonctions qui relient les variables entrantes aux variables sortantes en se basant sur des exemples d'inputs et de leurs outputs. Il existe plusieurs types d'apprentissage supervisé.

- **Problèmes de Régression :**

Dans les modèles de régression, nous prédisons une variable cible à partir de certaines variables prédictives. Cela veut dire que nous utilisons la régression pour trouver une corrélation entre une variable dépendante (cible) et une variable indépendante (prédictive). Dans les modèles de régression, l'output est une valeur quantitative continue ou bien discrète. Les modèles de régression les plus populaires sont :

- **La régression linéaire :** C'est un algorithme qui vise à trouver une ligne qui viendra diviser les données. Ses extensions sont la régression multiple et la régression polynomiale.
- **Les arbres de décision :** Les arbres de décision construisent le problème de régression sous la structure d'un arbre. Ils décomposent la Data Set et des sous-ensembles de données de plus en plus petits tout en développant en même temps un arbre de décision associé. Le résultat final est un arbre avec des nœuds de décision et des nœuds feuilles [10].
- **Les forêts aléatoires :** Les forêts aléatoires utilisent une méthode ensembliste de régression. Ils opèrent en construisant plusieurs arbres de décisions durant l'apprentissage automatique, et leur résultat est la moyenne des classes comme prédiction de tous les arbres [11].
- **Les réseaux de neurones :** C'est un modèle d'apprentissage très populaire, multicouches, et qui est inspiré par le cerveau humain. C'est une série d'algorithmes qui essaient de reconnaître les relations sous-jacentes dans un ensemble de données grâce à un processus qui imite le fonctionnement du cerveau humain.

- **Problèmes de Classification**

Dans les modèles de classification, on prédit une variable cible à partir de certaines variables prédictives. Cela veut dire que nous utilisons la régression pour trouver une corrélation entre une variable dépendante (cible) et une variable indépendante (prédictive). Dans les modèles de classification, l'output est une valeur catégoriale, comme OUI ou NON, Réussite ou échec, etc. Les modèles les plus populaires pour la classification sont :

- **La régression logistique** : La régression logistique ou « Logistic Regression » est un modèle d'apprentissage automatique similaire à la régression linéaire, sauf que le résultat est une valeur binomiale. Le résultat est l'impact de chaque variable sur le rapport des chances de l'évènement d'intérêt observé.
- **Support Vector Machine** : C'est une technique d'apprentissage automatique supervisé qui a l'objectif de trouver un hyperlien dans un espace de n dimensions qui classifie distinctivement les points de données.
- **Naïves Bayes** : C'est un classificateur qui agit comme un modèle d'apprentissage automatique probabiliste pour les tâches de classification basé sur le théorème de Bayes.
- **Les arbres de décision, Les forêts aléatoires et les réseaux de neurones** : Ces modèles suivent la même logique expliquée précédemment pour la régression, la seule différence est que l'output est une variable catégoriale.
- **Apprentissage non supervisé** : Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé est utilisé pour tirer des inférences, trouver des modèles et une structure dans les données d'entrée sans référence à un résultat étiqueté.
- **Clustering** : Le Clustering implique le regroupement des points de données, il est largement utilisé dans la segmentation des clients, la détection des fraudes et la classification des documents. Les techniques de regroupement incluent : K-Means, les techniques Hiérarchiques, les techniques de regroupement avec Mean Shift, les techniques de regroupement Density-based, etc.