



UNIVERSITE ABDEL MALEK ESSAADI
FACULTE DES SCIENCES ET TECHNIQUES DE
TANGER



MEMOIRE DE PROJET DE FIN D'ETUDES

MASTER SCIENCES ET TECHNIQUES EN MOBIQUITE ET BIG DATA

Sujet :

**Conception et développement d'un site web e-commerce
basé sur les techniques de AI**

REALISE PAR :

Errzieq El Mehdi

Baki Senhaji Abdelmounim

Membres du Jury :

M. Boudhir Anouar	Président
M. Ait Kbir Mohamed	Examinateur
M. Bouhorma Mohammed	Encadrant

AVANT PROPOS

Noms et Prénoms des étudiants stagiaires :

Errzieq El Mehdi

Baki Senhaji Abdelmounim

Intitulé du sujet :

Conception et développement d'un site web e-commerce basé sur les techniques de AI

Etablissement d'accueil :

Faculté des Sciences et Techniques

Encadrant pédagogique :

M. Bouhorma Mohammed, Professeur à la FST de Tanger.

Période de projet :

Du 15 février 2020 au 10 juillet 2020

Cadre du stage :

Projet de Fin d'Etudes présenté en vue de l'obtention de Master en Sciences et Techniques.

Filière :

Master en Informatique – Mobilité & Big Data

Dédicace

Que ce travail témoigne de mes respects :

A mes parents, Meriem et Mohamed

J'ai reçu de leur de nombreuses leçons avec beaucoup d'amour et de tendresse. Rien au monde ne pourrait leur compenser pour leurs efforts, je veux bien dire Combien leur m'êtes chers. Que dieu leurs garde et leurs bénisse.

A mon frère, Amine,

A mes chères sœurs, RABAB et ZAINAB,

Pour leur soutien moral, j'espère être à la hauteur de leurs espérances. Que Dieu leur procure bonne SANTE et longue vie.

A mes Professeurs, qui m'ont guidé, illuminé mon chemin vers le savoir.

A mes amies, pour leur aides et supports DANS les moments difficiles.

A tous mes collègues, ceux que j'AIME et ceux qui m'Aiment.

Et à tous ceux qui pensent à nous et ceux qui ont contribué de près ou de loin pour que ce projet soit possible.

Errzieq El Mehdi

Dédicace

Que ce travail témoigne de mes respects :

A mes parents, Zohra et Abdelghani

En reconnaissance de leur dévouements déployés pour mon bonheur, de leur patience et de leur soutien indéfini. Je prie le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de moi.

A mes frères, ABDEREZZAK et Anass,

A ma sœur, Oumaima,

Puisse Dieu leur donner santé, bonheur, courage et surtout réussite.

A mes Professeurs, pour le soutien et l'aide permanente qu'ils n'ont cessés pas de nous apporter.

A mes amis, pour leurs aides et supports à tout moment.

A tous mes collègues, ils vont trouver ici le témoignage d'une fidélité et d'une Amitié infinie.

Et à tous ceux qui nous ont cédé leur temps et leurs connaissances pour satisfaire nos interrogations.

Baki Senhaji Abelmounim

Remerciements

Tout d'abord, nous tenons à remercier **ALLAH**, le tout Puissant de nous avoir donné la force et le courage de mener à bien ce modeste travail.

Nous tenons à remercier toutes les personnes qui ont contribué de près ou de loin le bon acheminement de ce projet de fin d'études.

Ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de notre professeur M. Bouhorma Mohammed, nous le remercions pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce projet.

Nous saisissons aussi l'occasion pour remercier également les professeurs de la Faculté des Sciences et Techniques de Tanger qui nous ont octroyé une bonne formation pour les efforts qu'ils fournissent pour nous, notre grande admiration pour leurs hautes qualités morales, humaines et professionnelles. Spécialement les professeurs du département Informatique.

Nous adressons aussi nos vifs remerciements aux membres de jury pour avoir bien voulu examiner et juger ce travail.

Un grand merci à tous ceux qui ont contribué à faire de ce stage une expérience enrichissante et valorisante.

Résumé

Ce Mémoire est le fruit de notre travail dans le cadre du projet de fin d'études, pour l'obtention de notre diplôme Master en Mobilité et Big Data. Ce projet a fait l'objet d'un stage allant du 15/02/2020 au 10/07/2020.

Notre projet porte sur l'étude, la conception et la réalisation d'une plateforme distribuée basé sur un système de recommandation des produits, de réalisation d'un chatbot qui simule une conversation avec le client tout en tentant d'apporter des réponses à ses questions basant sur l'entraînement d'une base de données de questions-réponses, et de faire des Recherche sur les produits soit par image soit par texte en utilisant les techniques de NLP, Machine Learning et Deep Learning.

Donc nous pouvons résumer en trois missions principales nous ont été confiées durant ce stage:

- Mettre en place un système de recommandation très sophistiqué manipulé par une des algorithmes de AI.
- Mettre en place une plateforme e-commerce avec les meilleures technologies qui sont les plus utilisés dans le monde professionnel ainsi que Angular9 comme frontend, Python comme backend, MongoDB comme base de données et Flask comme API.
- Intégrer les technologies que nous avons étudiés au cours de nos études notamment Chatbot, Classification des images (réseaux de neurones convolutifs) et Data Mining.

Le présent rapport donne une description des missions et fructifie l'ensemble des travaux que nous avons réalisées et les résultats obtenus, dans le cadre de stage de fin d'études du Master Mobilité et Big Data.

Mots-clés: Système de recommandation, Chatbot, NLP, Data Mining, Deep Learning, Machine Learning

Abstract

This thesis describes the different aspects of our work as part of a graduation internship, for the obtaining of our Master degree in Mobiquity and Big Data. This project was the subject of an internship from 15/02/2020 to 10/07/2020.

Our project involves the study, design and creation of a distributed platform based on a recommender system to involve the performance of the web site, the realization of a chatbot that simulates a conversation with the client while trying to provide answers to his questions based on training a database of questions-Answers, and to do research on the products either by image or by text using the techniques of NLP, Machine Learning and Deep Learning.

So, we can summarize in three main missions were entrusted to us during this internship:

- Set up a very sophisticated recommender system manipulated by one of the AI algorithms.
- Set up an e-commerce platform make up with the best technologies which are the most used in the professional world as well as Angular9 as front-end, Python as backend, MongoDB as database and Flask as API.
- Integrate the technologies we studied during our studies, notably Chatbot, Classification of images (convolutional neural networks) and also Data Mining.

This report gives a description of the missions and all the work we have carried out and the results obtained, as part of the end-of-study internship of the Master in Mobiquity and Big Data.

Key-words: Recommender system, Chatbot, NLP, Deep Learning, Data Mining, Machine Learning

ملخص

نسلط الضوء من خلال هذا البحث، على مجموعة الأعمال التي قمنا بها في إطار مشروع نهاية الدراسات لنيل شهادة الماستر، هذا المشروع كان موضوع تدريب من 15/02/2020 إلى 10/07/2020.

يتضمن مشروعنا دراسة وتصميم وإنشاء منصة موزعة تدعم نظام توصيات المنتج وتحقيق برنامج الدردشة الذي يحاكي المحادثة مع الزبائن أثناء محاولة تقديم إجابات لأسئلته بناءً على تدريب قاعدة بيانات من الأسئلة -الأجوبة ، وإجراء البحث على المنتجات إما بالصور أو عن طريق النص باستخدام تقنيات البرمجة اللغوية العصبية ، التعلم الآلي والتعلم العميق وذلك يمكننا أن نلخص في ثلاثة مهام رئيسية تم تكليفنا بها خلال هذا التدريب :

- قمنا بإعداد نظام توصيات متتطور للغاية يتم التعامل معه بواسطة إحدى خوارزميات الذكاء الاصطناعي ✓
- قمنا بإعداد منصة للتجارة الإلكترونية باستخدام أفضل التقنيات الأكثر استخداماً في العالم المهني بالإضافة إلى كواجهة أمامية Angular9 وخلفية Python وقاعدة بيانات MongoDB وواجهة لبرمجة التطبيقات Flask ✓
- دمج التقنيات التي درسناها خلال دراستنا ، لا سيما شات بوت ، تصنيف الصور ، تعدين البيانات ✓

يقدم هذا التقرير وصفاً للمهام ويبين كل العمل الذي قمنا به و النتائج التي تم الحصول عليها ، كجزء من التدريب في نهاية الدراسة للماستر

الكلمات الرئيسية : نظام التوصية، شات بوت، البرمجة اللغوية العصبية ، التعلم الآلي ، تعدين البيانات ، التعلم العميق

Table des matières

<i>Dédicace</i>	1
<i>Remerciements</i>	3
<i>Résumé</i>	4
<i>Abstract</i>	5
<i>مختصر</i>	6
<i>Table des matières</i>	9
<i>Table des figures</i>	12
<i>Listes des acronymes</i>	14
<i>Introduction générale</i>	15
<i>Chapitre 1 : Contexte général du projet</i>	17
1. Introduction	18
2. Présentation de l'organisme de travail	18
3. Présentation du projet.....	18
3.1- Contexte du projet.....	18
3.2- Problématique	18
3.3- Solutions	19
3.4- Difficultés du sujet	20
4. Conclusion.....	20
<i>Chapitre 2 : Méthodologie et Conduite de projet</i>	21
1. Introduction	22
2. Méthodologie de travail (de gestion de projet).....	22
3. Méthode Agile.....	22
3.1- Méthode adoptée	22
3.2- La méthodologie Scrum	23
4. Planning du projet	25
5. Les outils	27
5.1- Discord	27
5.2- Trello	27
5.3- Git.....	28
6. Conclusion.....	28
<i>Chapitre 3 : Conception et Modélisation UML</i>	29
1. Introduction	30
2. Les besoins fonctionnels du projet	30

3. Modèle dynamique	31
3.1- Définition UML.....	31
3.2- Diagramme des cas d'utilisation	31
3.3- Diagramme d'activités	33
3.4- Diagramme de séquences.....	35
3.5- Diagramme de classe	36
4. Conclusion.....	37
Chapitre 4 : Technologies et outils de développement.....	38
1. Introduction	39
2. Outils utilisés.....	39
2.1- L'environnement de travail.....	39
2.2- Langages utilisés	40
2.3- Librairies.....	42
3. Conclusion.....	44
Chapitre 5 : Réalisation de l'application.....	45
1. Introduction	46
2. Base de données	46
2.1- Collection de données	46
2.2- Prétraitement de données.....	47
2.3- Stockage de données	48
3. Partie Backend	48
3.1- Algorithme de recommandation.....	48
3.2- Mesure de la similarité entre les produits	53
3.3- Chatbot	56
3.4- Most Popular	59
3.5- CNN	59
4. Partie Frontend	63
4.1- Partie Création	63
4.2- Partie visiteur	64
4.3- Partie Client	65
4.4- Partie Vendeur	69
4.5- Partie Admin	73
5. Sécurité de l'application	76
5.1- L'authentification.....	76
5.2- Le contrôle d'accès	77
5.3- La confidentialité des données	77
6. Liaison backend avec Frontend	77
6.1- Flask	77
6.2- Pourquoi Flask ?	78
6.3- Méthodes.....	78
6.4- Architecture	78
6.5- Exemples	79
7. Conclusion.....	80
Conclusion et Perspectives.....	81

Annexes.....	82
1. Les Systèmes de recommandations	82
2. La mise en place d'un Système de recommandation	82
3. Types de systèmes de recommandations.....	82
4. Le choix d'algorithme	83
4.1- SVD : Singular Value Decomposition.....	83
4.2- SGD : Stochastic Gradient Descent	84
4.3- ALS : Alternating Least Squares	84
5. Partie de code	85
5.1- ALS	85
5.2- Chatbot	85
5.3- CNN	85
5.4- La métrique cosine_similarity	86
5.5- Most Popular	86
5.6- Most Active	86
5.7- Partie Flask	87
5.8- Liaison Python MongoDB.....	87
Références	88

Table des figures

Figure 1 : Illustration de fonctionnement de méthode Scrum	23
Figure 2: Diagramme de Gantt de notre projet.....	26
Figure 3 : Diagramme des cas d'utilisation.....	32
Figure 4 : diagramme d'activité de visiteur	33
Figure 5: : diagramme d'activité de client	34
Figure 6 : Diagramme d'activité de vendeur.....	34
Figure 7 : Diagramme d'activité d'admin.....	35
Figure 8 : Diagramme de séquences d'inscription	35
Figure 9 : Diagramme de séquences d'authentification.....	36
Figure 10 : Diagramme de classe	37
Figure 11: Sous Catégories de l'intelligence artificielle	46
Figure 12: Table des produits avec pandas (DataFrame)	47
Figure 13: Table des produits dans Studio3T	48
Figure 14: Erreur obtenu de ALS	49
Figure 15: la matrice de recommandation User-Item de ALS.....	52
Figure 16: L'interface You May Like dans la page home	53
Figure 17: L'interface You May Like dans la page catégorie ou sous-catégorie.....	53
Figure 18:L'interface de similaire produits	55
Figure 19: résultat de Moteur de recherche textuelle	56
Figure 20: L'architecture d'un chatbot basé sur la récupération	56
Figure 21: la base de données utilisé dans chatbot	57
Figure 22: Exemple d'une réponse de chatbot	58
Figure 23: Interface Most Popular.....	59
Figure 24: Architecture du CNN utilisé	60
Figure 25 : résultat de Moteur de recherche par image	62
Figure 26: Résultat lorsqu'un visiteur veut faire rating ou j'aime.....	65
Figure 27: Interface pour consulter la liste des produits de chaque catégorie	65
Figure 28: Interface pour Chercher des produits soit par texte ou par image.....	66
Figure 29 : Interface pour gérer le profil	66
Figure 30: Interface des produits les plus populaire	66
Figure 31 : Interface des produits similaires à celui que le client en train de voir	67
Figure 32 : Interface pour changer le type de devise des produits	67
Figure 33 : Interface d'ajouter un produit au panier, Faire rating ou Like.....	67
Figure 34: Modifier le profil.....	68
Figure 35 : Interface Chatbot.....	68
Figure 36 : Interface you may like de client.....	69
Figure 37 : Interface vendeur pour afficher liste des produits.....	69
Figure 38 : Interface vendeur pour ajouter un produit.....	70
Figure 39 : Interface vendeur pour modifier ou supprimer un produit.....	70
Figure 40: Exemple d'une statistique des produits les plus postées par catégorie	70
Figure 41: Exemple d'une statistique des produits les plus postées par sous-catégorie.....	71
Figure 42: Exemple d'une statistique des produits les plus populaire.....	71
Figure 43 : Exemple d'une statistique des catégories les plus aimés	72
Figure 44: Exemple d'une statistique des produits les moins populaire.....	72
Figure 45 : Exemple d'interface admin pour voir des statistiques sur les catégories de site web (graphe1).....	73
Figure 46: Exemple d'interface admin pour voir des statistiques sur les catégories de site web (graphe2).....	73
Figure 47 : Interface admin pour voir les utilisateurs les plus actif.....	74

Figure 48 : Interface admin pour voir les utilisateurs les moins actif	74
Figure 49 : Interface admin pour gérer les utilisateurs du site web.....	75
Figure 50 : Interface admin pour voir les réactions des utilisateurs au produits	75
Figure 51: Interface admin pour voir les produits de la base de données.....	76
Figure 52 : Interface d'authentification.....	77
Figure 53: Architecture de déroulement des données depuis la base de données jusqu'à la partie client	78
Figure 54 : Code ALS.....	85
Figure 55 : Code Chatbot	85
Figure 56 : Code CNN.....	85
Figure 57 : Code cosine_similarity.....	86
Figure 58 : Code Most popular.....	86
Figure 59 : Code Most active	86
Figure 60 : Code partie flask	87
Figure 61 : Code liaison Python MongoDB	87

Listes des acronymes

AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
ALS	Alternating Least Square
NLP	Natural Language Processing
API	Application Programming Interface
JWT	Json Web Token
HTTP	HyperText Transfer Protocol
URL	Uniform Resource Locator
REST	Representational State Transfer

Introduction générale

Au cours des dernières années, avec l'existence de Youtube, Amazon, Netflix et de nombreux autres services Web de ce type, les systèmes de recommandation ont pris de plus en plus de place dans nos vies. Du e-commerce à la publicité en ligne, les systèmes de recommandation sont aujourd'hui incontournables dans nos déplacements quotidiens en ligne.

De ce fait, Nous avons décidé d'implémenter une des systèmes de recommandation consacré pour une plateforme e-commerce en travaillons sur une base de données qui n'est pas réelle mais il a le même principe des autres bases de données réelles, nous avons basé sur différents sites pour collecter les données à savoir Kaggle, dataWorld et autres.

De manière très générale, les systèmes de recommandation sont des algorithmes visant à suggérer des éléments pertinents aux utilisateurs (dans notre projet les éléments sont des produits à acheter). Les systèmes de recommandation sont vraiment essentiels dans certaines industries car ils peuvent générer une énorme quantité de revenus lorsqu'ils sont efficaces surtout avec La croissance explosive de la quantité d'informations numériques disponibles et du nombre de visiteurs sur Internet qui crée un problème potentiel de surcharge d'informations , pour ces raisons nous avons pu d'implémenter un algorithme qui va aider l'utilisateur à réduire les coûts de transaction liés à la recherche et à la sélection des produits et à améliorer le processus décisionnel et aussi au côté de l'entreprise les système de recommandation va augmenter les revenus, car ils constituent un moyen efficace de vendre plus de produits .

En générale, Il y'a trois types de système de recommandation : Filtrage basé sur le contenu, en effet la recommandation est faite sur la base des profils d'utilisateurs en utilisant des fonctionnalités extraites du contenu des éléments que l'utilisateur a évalués dans le passé, Filtrage collaboratif qui fonctionne en créant une base de données (matrice utilisateur-élément) de préférences pour les éléments par les utilisateurs et enfin Filtrage hybride qui englobe les deux.

Nous avons choisi de travailler par Filtrage collaboratif spécifiquement l'algorithme Alternating Least Square (ALS) qui donne des meilleurs résultats sur notre type de données comparant avec autres algorithmes et avec un taux d'Erreur RMSE inférieur, donc une performance meilleure. Cette Algorithme fait le traitement sur les informations des utilisateurs et leurs interactions avec les produits (likes, ratings) et donne des prédictions sur les produits qui peut être intéressées pour le client.

Nous avons passé par trois phases pendant cette expérience : Phase de collecte d'informations, Phase d'apprentissage et la phases de prédiction / recommandation.

Ce projet se compose de six parties. Dans la première partie, nous allons présenter le contexte général du projet, dans la deuxième partie nous allons présenter méthodologie et conduite de projet, La troisième partie se penchera sur la conception et modélisation UML, Ensuite les différents outils et librairie utilisées et Enfin une dernière partie où nous allons présenter notre application.

Chapitre 1

Contexte général du projet

Ce premier chapitre est une introduction au contexte général du projet et à son objectif majeur. Il présente l'organisme de travail ainsi que le cadre général où s'inscrit le sujet du stage.

1. Introduction

Le stage de fin d'études, comme son nom l'indique, s'effectue à la fin du cursus de formation. Il sert à bien faire le lien entre les connaissances théoriques acquises et l'application dans un contexte d'entreprise.

Le sujet de stage a été proposé par notre encadrant Bouhorma Mohammed, L'un des sujets actuels : la réalisation d'un système de recommandation

2. Présentation de l'organisme de travail

Le projet système de recommandation a adopté la méthodologie Agile Scrum. Il a évolué par itération successive d'une durée constante d'une semaine. Chaque itération a fourni un livrable qui est présenté dans une séance de démonstration. Entre chaque itération, nous réunissons à distance utilisant les techniques mentionnées dans la partie (ch2-5) à afin de faire le point sur l'avancement, l'évolution de la demande et des priorités mais aussi pour réfléchir aux axes d'amélioration.

3. Présentation du projet

3.1- Contexte du projet

Les systèmes de recommandation ont une place particulièrement importante dans le marketing en ligne. Grâce à eux, des entreprises du e-commerce ont pu se différencier de leurs concurrents, faciliter la vie des clients actuels et atteindre leurs clients potentiels.

C'est pour cela notre projet est le développement d'un système de recommandation, de réalisation d'un chatbot et de faire des recherches sur les produits soit par image soit par texte en utilisant les techniques de NLP, Machine Learning et Deep Learning.

3.2- Problématique

A partir des constatations de la majorité des sites e-commerce déjà existants dans l'internet, plusieurs problématiques ont été remontées à savoir :

- La croissance explosive de la quantité d'informations numériques disponibles et du nombre de visiteurs sur Internet a créé un problème potentiel de surcharge d'informations et le client entre dans un déroulement pour choisir un produit qu'il convient et parfois il quitte la plateforme avec inquiétude à cause de cette énorme quantité de données.
- Le client peut poser des questions sur les produits pour avoir des informations et le vendeur ne peut pas être connecté 24h/24h et cela pousse le client à quitter ce magasin et donc mauvais revenus pour les vendeurs, de ce fait nous avons besoin d'un mécanisme qui va diriger et soutenir les acheteurs en ligne 24h/24h sur leurs décisions d'achat, répondre à des questions et moins la présence et d'intervention humaine.

- Parfois le client a déjà une image ou un nom du produit qui veut acheter donc l'utilisation du filtre n'est pas parfaite dans ces cas
- Puisque le système de recommandation génère des informations spécifiques pour chaque client, nous devons développer une interface web avec un langage qui prend en charge et traite ces données sans aucun décalage.
- La plupart des vendeurs veulent voir une analyse de leurs produits surtout quand il y a un grand nombre de produits afin de faire un certain changement et des modifications sur leurs produits pour attirer le maximum possible des clients.
- Le client veut toujours comparer les produits similaires pour choisir un de ces produits dépend de ses besoins.

L'objectif principal de la solution de ces problématiques est de permettre au vendeur d'augmenter les ventes et réduire les coûts avec moins d'effort et pour ça l'entreprise ou le propriétaire du plateforme doit Fournir des recommandations sur les produits en fonction des éléments susceptibles d'intéresser l'acheteur et aussi de satisfaire le vendeur pour être toujours vendent dans sa plateforme et enfin pour satisfaire le client le maximum possible pour ne pas aller acheter dans un autre plateforme.

3.3- Solutions

La solution retenue pour atteindre notre objectif et pour résoudre le problème de réalisation de notre projet se résume dans les étapes suivantes :

- Collection et prétraitement les informations de chaque client et leurs interactions avec les produits et lui afficher des prédictions sur les produits qui peut être intéressées pour lui dans chaque catégorie ou sous-catégorie.
- Réaliser un chatbot intelligent qui va répondre à toutes questions sur le contenu de site web 24h/24h et ça aide beaucoup le vendeur à tirer le maximum possible des clients.
- Réaliser un moteur de recherche par image et fait la prédition de quelle sous-catégorie appartient cette image afin de diriger le client vers la sous-catégorie chercher.
- Implémenter un site web dynamique avec l'une des nouvelles technologies qui se caractérise par l'utilisations des services, pas de recharge des pages et la souplesse pendant le développement et autres et toutes ça pour avoir un Système intégré.
- Utilisation un des plugins de Angular qui permet de faciliter l'analyse et la visualisation des données de vendeur sous forme des graphs afin de faire un certain changement et des modifications sur leurs produits pour attirer le maximum possible des clients.

- Réaliser la similarité entre les produits basée sur le Cosinus pour que le client être toujours motivé d'avoir plus de produites.

Et autres fonctionnalités notamment : Plus populaire Product, les vendeurs peut avoir des statistiques pour améliorer leur store etc.

3.4- Difficultés du sujet

La fonction de R&D (Recherche et Développement) en général nécessite de mobiliser des moyens financiers, matériels, du temps et des personnes. Pour cette raison, dans le cadre de notre stage, nous avons rencontré quelques difficultés, à savoir :

- La disponibilité des données d'apprentissage liées au domaine de marketing. En effet, les données trouvées sont toujours mal organisées et très rares et pas gratuites : C'est le problème principal lors de la réalisation d'expériences dans le domaine de l'intelligence artificielle est de trouver des données.
- Le côté matériel : l'apprentissage des modèles de filtrage collaboratif et aussi les réseau de neurones convolutionnels sur un grand base de données nécessite du matériel puissant et performant.

4. Conclusion

Ce chapitre nous a servi à mettre le projet dans son contexte et son positionnement sur le marché. Nous avons essayé de montrer aussi les solutions adoptées pour résoudre les problèmes qui répondent à nos besoins.

Le prochain chapitre est consacré à la présentation des méthodologie et Conduite de projet

Chapitre 2

Méthodologie et Conduite de projet

Dans ce chapitre, nous présentons les méthodologies et outils que nous adoptons pour mener notre projet

1. Introduction

Le choix entre une méthode et une autre, dépend de la nature du projet et de sa taille. Pour des projets de petite taille et dont le domaine est maîtrisé, par exemple, un cycle de vie en cascade s'avère largement insuffisant. Lorsqu'il s'agit d'un projet où les données ne sont pas réunies dès le départ, où les besoins sont incomplets voire floues, il faut s'orienter vers une méthode itérative ou orientées prototypes.

Tout cela, doit être géré avec des plateformes de discussion, d'hébergement et de gestion de projet.

2. Méthodologie de travail (de gestion de projet)

La finalisation du projet dans les délais de livraison est le souci majeur de chaque équipe de développement d'un logiciel. L'un des problèmes les plus fréquemment affrontés lors de la construction du logiciel est la mauvaise spécification et le changement brusque des besoins. Cela peut influencer non seulement l'équipe de développement en créant un environnement de stress, mais aussi le temps consacré pour la réalisation du projet et donc des délais de livraison dépassées. Afin d'éviter ces situations critiques, nous adoptons la méthodologie agile pour la gestion de notre projet.

3. Méthode Agile

Une méthode Agile est menée dans un esprit collaboratif et s'adapte aux approches incrémentales. Elle engendre des produits de haute qualité tout en tenant compte de l'évolution des besoins du client. Une méthode Agile assure une meilleure communication avec le client et une meilleure visibilité du produit livrable. Elle permet aussi de gérer la qualité en continu et de détecter des problèmes le plus tôt au fur et à mesure, permettant ainsi d'entreprendre des actions correctrices sans trop de pénalités dans les coûts et les délais.

Les problématiques précédemment mentionnées ont poussé les informaticiens à réinventer les méthodes de gestion de projet et de conception en introduisant ce qu'on appelle les méthodes agiles. C'est une approche incrémentale et itérative, menée dans un esprit collaboratif, avec juste ce qu'il faut de formalisme. Elle peut générer un produit de bonne qualité tout en prenant en compte l'évolution des besoins des clients.

En suivant cette approche, le logiciel est conçu dans son ensemble et peut être construit étape par étape.

3.1- Méthode adoptée

Parmi les méthodes agiles, nous citons Scrum que nous avons choisi pour les raisons suivantes :

- Scrum est la méthodologie suivie par la plupart des sociétés pour la gestion de ses projets

- Elle est entièrement développée et testée pour de courtes itérations
- La simplicité des processus
- L'augmentation de productivité
- Elle permet d'adapter le logiciel créé suivant l'évolution du projet

3.2- La méthodologie Scrum

La méthodologie de gestion de projet qui a été utilisée est la méthodologie Scrum.

Scrum permet d'avancer progressivement, d'ajuster au fur et à mesure, d'impliquer davantage le client (dans notre cas le client c'est notre encadrant) et d'échanger beaucoup plus.

La méthode Scrum permet de transformer le projet en un processus itératif et incrémental afin de structurer le développement du produit en plusieurs cycles de travail appelés « Sprint ».

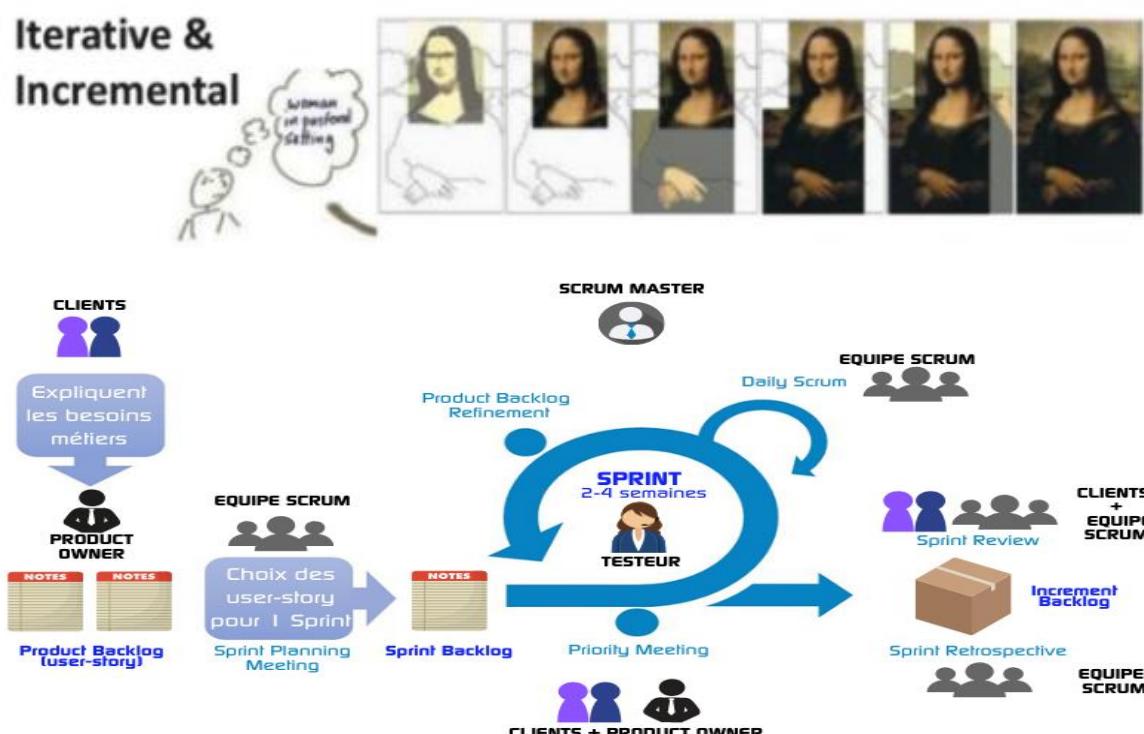


Figure 1 : Illustration de fonctionnement de méthode Scrum

3.2.1 Justification du choix :

Le choix s'est porté sur la méthode Scrum pour plusieurs raisons :

- Notre groupe étant constitué de deux personnes pour la réalisation du projet de fin d'études.
- Le besoin de faire des réunions avec l'encadrant de projet afin de valider les états d'avancement de l'équipe.

- Possibilité d'ajouter, compéter ou modifier à tout moment la liste des fonctionnalités à réaliser

Nom & prénom	Fonction	Gmail
Errzieq El Mehdi	Développer /data scientiste	errzieq@gmail.com
Baki Senhaji Abdelmounim	Développer / data scientiste	abdilmounimsh@gmail.com
M. Bouhorma Mohammed	Encadrant de projet	bouhorma@gmail.com

3.2.1 Les rôles Scrum

L'équipe Scrum est constituée d'un Product Owner, de l'équipe de développement et d'un Scrum Master. Le modèle d'équipe de Scrum est conçu pour optimiser la flexibilité, la créativité et la productivité.

Nous allons tout d'abord tenter de cerner notre équipe Scrum :

- **Product Owner (PO)** : Mohammed Bouhorma, le représentant des clients et des utilisateurs. Son rôle est d'assurer la présentation des caractéristiques et des fonctionnalités du produit à développer et approbation du produit à livrer.
- **Scrum Master (SM)** : El Mehdi Errzieq, le responsable du déroulement du processus. Le Scrum Master assure globalement la Supervision de l'avancement du projet et des activités de l'équipe. Il assure également l'organisation des réunions et la bonne application de la méthode Agile de par ce biais.
- **L'équipe de développement** : Elle comporte les personnes qui se chargent de la réalisation des histoires utilisateurs et élaboration des sprints (Abdelmounim Baki Senhaji et El Mehdi Errzieq).

3.2.2 Backlog

Le Backlog de produit correspond à une liste priorisée des besoins et des exigences du client. Les éléments du Backlog, appelé « User Stories » ou histoires d'utilisateurs, classés par priorités, ils sont formulés en une ou deux phrases décrivant de manière claire et précise la fonctionnalité désirée par le client, généralement, écrit sous la forme “En tant que X, je voudrais Y”

Le Backlog du sprint est une sélection de tâches retenues du "Backlog du produit" pour construire l'objectif du sprint.

Le Backlog présenté dans le tableau ci-dessous comprend les champs suivants :

- **ID** : C'est un nombre unique et auto-incrémenté pour chaque histoire utilisateur
- **User Story** : C'est une phrase décrivant la fonctionnalité désirée par le client.
- **Priorité** : C'est la valeur métier qui dirige la priorisation du développement des histoires utilisateurs suivant les attentes et les besoins du client (highest, high, medium, low, lowest).

- **Story point** : C'est l'effort nécessaire à la réalisation d'une histoire utilisateur.

Nous avons pu établir une étude en répartir ces stories par priorité. Les normes sur lesquelles nous nous sommes basés, sont la complexité et la priorité par rapport au client

3.2.3 La vie d'un projet Scrum

La vie d'un projet Scrum est rythmée par un ensemble de réunions clairement définies et strictement limitées dans le temps (time boxing) :

a- Planification du Sprint (*Sprint = itération*) :

Au cours de cette réunion, l'équipe de développement sélectionne les éléments prioritaires du « Product Backlog » (liste ordonnancée des exigences fonctionnelles et non fonctionnelles du projet) qu'elle pense pouvoir réaliser au cours du sprint (en accord avec le « Product Owner »).

Les "User stories" précédemment définis, sont triés par ordre de priorités et de valeurs métiers. Le but étant d'implémenter en premier ce qui a le plus de valeur. Le travail sera planifié selon des sprints que nous avons dénis. Après une réunion, on a identifié X sprints et Y releases. Dans le tableau de détail de sprint, nous présentons la planification des sprints.

b- Sprint Review ou la Démonstration du sprint :

Au cours de cette réunion qui a lieu à la fin du sprint, nous présente les fonctionnalités terminées au cours du sprint et recueille les feedbacks du Product Owner. C'est également le moment d'anticiper le périmètre des prochains sprints et d'ajuster au besoin la planification de release (nombre de sprints restants).

c- Rétrospective de Sprint :

La rétrospective qui a généralement lieu après la revue de sprint est l'occasion de s'améliorer (productivité, qualité, efficacité, conditions de travail, etc) à la lueur du "vécu" sur le sprint écoulé (principe d'amélioration continue).

d- Mêlée quotidienne ou DSM (*Daily Sprint Meeting*) :

Il s'agit d'une réunion de synchronisation qui se fait debout en 15 minutes maximum au cours de laquelle chacun répond principalement à 3 questions : « Qu'est-ce que j'ai terminé depuis la dernière mêlée ? Qu'est-ce que j'aurai terminé d'ici la prochaine mêlée ? Quels obstacles me retardent ? ».

4. Planning du projet

Avant de présenter les différents sprints qui composent notre Backlog, nous évoquons le planning de notre travail qui s'est étalé sur une période de cinq mois en utilisant le diagramme de Gantt.

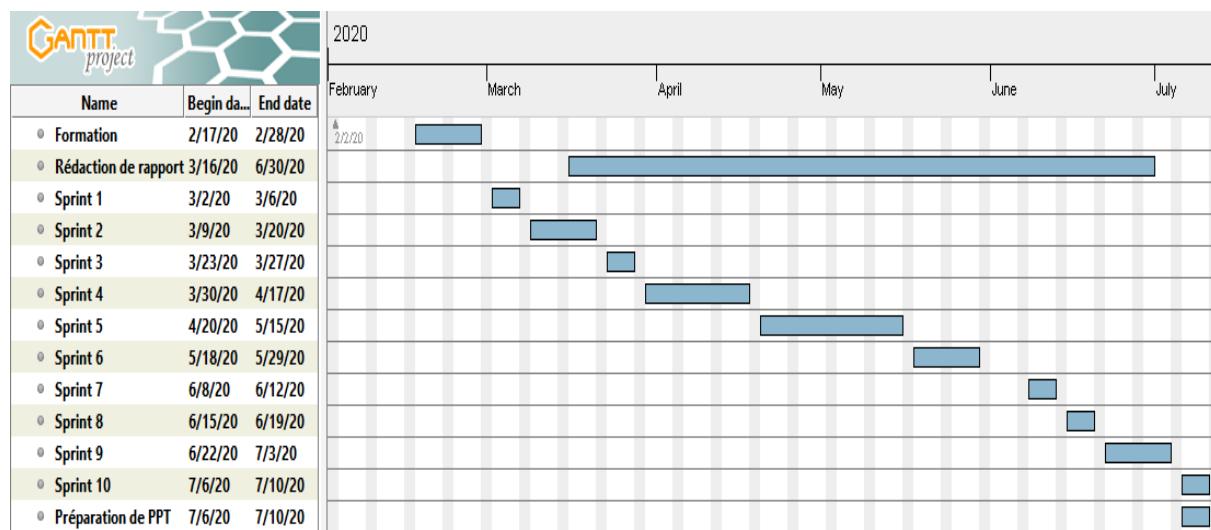


Figure 2: Diagramme de Gantt de notre projet

Dans le tableau suivant nous évoquons le détail de chaque Sprint :

No de Sprint	Les tâches	Affectation
Sprint 1	Collecter les données à partir de Kaggle Collecter les données à partir de dataWorld	A M
Sprint 2	Prétraitement des données téléchargés à partir de Kaggle Prétraitement des données téléchargés à partir de dataWorld	A M
Sprint 3	Stocker les données dans MongoDB Savoir Comment utiliser python avec MongoDB	A M
Sprint 4	Être familier sur les systèmes de recommandations Tester des algorithmes consacrés pour le système de recommandation pour construire le modèle	M/A
Sprint 5	Être familier sur Angular Créer des interfaces utilisateur (Admin, Vendeur et client) pour exploiter le modèle créé	M/A

Sprint 6	Être familier sur Flask API Créer des routes avec Flask pour passer les résultats de recommandations et autres informations à la partie web.	M/A
Sprint 7	Implémenter un chatbot avec python Lier le chatbot avec Angular	M/A
Sprint 8	Implémenter une script pour recherche par image Implémenter une script pour recherche par Texte	M/A
Sprint 9	Travailler sur la partie sécurité de l'application	M/A
Sprint 10	Faire une révision totale sur le site web et résoudre s'il y a des problèmes inattendues	M/A

M: Errzieq El Mehdi

A: Baki Senhaji Abdelmounim

5. Les outils

Nous avons décidé d'utiliser plusieurs outils pour nous aider et pour nous faciliter la vie pendant le déroulement du projet. Afin de communiquer nous utilisons l'application Discord, nous avons utilisé le site Trello, celui-ci nous permet de gérer très efficacement le principe des sprints.

5.1- Discord



Discord est un logiciel plus que simple. Il vous offre la possibilité de créer un “salon” virtuel, ce dernier permet de communiquer de deux manières différentes.

La première, avec des messages textuels, un peu comme MSN, ou Caramail pour les plus vieux d'entre nous. L'autre option proposée par Discord pour échanger avec ses amis, c'est par “vocals”, en audio, à l'oral, “en parlant” pour faire simple. Un peu comme Teamspeak, mais en gratuit.

5.2- Trello



Trello est un outil de gestion de projets et de tâches. Ses tableaux Kanban visuels sont flexibles et partageables. Trello est essentiellement un outil de collaboration où nous organisons notre projet sur des tableaux avec des listes et des fiches (tâches). Trello permet de voir facilement en un coup d'œil qui travaille sur quelque chose, quelles choses sont en cours, lesquelles sont terminées et ce que différentes personnes ont fait.

5.3- Git



Slack facilite la tâche. Git simplifie le processus de travail avec d'autres personnes et facilite la collaboration sur des projets. Les membres de l'équipe peuvent travailler sur des fichiers et fusionner facilement leurs modifications avec la branche principale du projet. Cela permet à plusieurs personnes de travailler sur les mêmes fichiers en même temps

6. Conclusion

Dans ce chapitre, nous avons présenté la démarche de découpage nécessaire pour la planification du travail. Nous avons abordé les différentes normes assurant un suivi efficace du projet. Et finalement, nous avons appliqué cette décomposition qui nous a permis de déduire les sprints du projet.

Le prochain chapitre est consacré à la conception et modélisation UML.

Chapitre 3

Conception et Modélisation UML

Ce chapitre permet de détailler, de définir, et d'exprimer des besoins fonctionnels pour atteindre nos objectives.

1. Introduction

Une méthode de conception est une démarche générale reflétant une philosophie de présentation et de suivi du système. Elle propose des outils spécifiques permettant un suivi efficace de l'information relative au système. Et notre choix se porte sur le langage UML (Unified Modeling Language) qui facilite l'interactivité avec la base de données à l'aide des diagrammes. Suite à cette constatation, nous développons ce chapitre sur deux parties. Le premier se consacre à la description des divers éléments constituant le cahier de charge à savoir : les besoins fonctionnels de l'application. Le deuxième est une présentation détaillée d'un model dynamique du projet.

2. Les besoins fonctionnels du projet

L'analyse fonctionnelle est une démarche qui « consiste à rechercher et à caractériser les fonctions offertes par un produit pour satisfaire les besoins de son utilisateur ». Par conséquent, il demeure nécessaire de définir les points essentiels de notre travail, ainsi que ses limites. Tout cela sera donc regroupé dans le cahier des charges en fonction de l'évolution des besoins.

L'objectif de cette étape est généralement de décrire le déroulement des opérations depuis le lancement de l'application jusqu'à sa clôture.

Les spécifications qui suivent représentent les fonctionnalités générales du projet :

Visiteurs (internautes) :

- Créer un compte
- Consulter les produits
- Observer les produits les plus populaires

Clients (utilisateurs inscrits) :

Après avoir créé un compte, les utilisateurs inscrits bénéficient de toutes les options citées ci-dessus en plus de :

- Gestion de profil
- Voter des produits
- Aimer les produits
- Ajouter les produits dans le panier
- Observer les produits qui pourraient m'intéresser

Admin (responsable de l'application) :

- Observer le contenu de la base de données (produits, utilisateurs inscrits et leurs métas de données) d'une façon claire, simple et mieux organisée avec la permission de supprimer
- Visualisations des Dashboard et des statistiques sur les produits et les utilisateurs

Vendeurs :

- Observer les informations sur leurs produits d'une façon claire et simple
- Visualisations des Dashboard et des statistiques sur leurs produits
- Ajouter/modifier/supprimer leurs produits de la base de données

3. Modèle dynamique

3.1- Définition UML

L’UML (Unified Modeling Language) est un langage de modélisation orientée objet, elle est développée dans le but de définir la notion standard pour la modélisation des applications construites à l'aide des objets. Elle est utilisée pour spécifier un logiciel ou pour le concevoir, le modèle décrit les classes et les cas d'utilisation vus de l'utilisateur final du logiciel. Le modèle produit par une conception orientée objet est en général une extension du modèle issu de la spécification, il l'enrichit de classe dites techniques qui n'intéressent pas l'utilisateur final du logiciel mais seulement ses concepteurs.

3.2- Diagramme des cas d'utilisation

Un cas d'utilisation (Use Cases) est un diagramme qui modélise des interactions fonctionnelles entre le système informatique à développer et les acteurs interagissant avec le système.

Permettent de définir les besoins et les fonctionnalités du système, ainsi que les cas les plus importants du système :

- **Acteur** : rôle joué par un utilisateur humain ou un autre système qui interagit directement avec le système étudié. Un acteur participe à au moins un cas d'utilisation.
- **Cas d'utilisation (use case)** : ensemble de séquences d'actions réalisées par le système produisant un résultat observable intéressant pour un acteur particulier. Collection de scénarios reliés par un objectif utilisateur commun.
- **Association** : utilisée dans ce type de diagramme pour relier les acteurs et les cas d'utilisation par une relation qui signifie simplement « participe à ».
- **Inclusion** : le cas d'utilisation de base en incorpore explicitement un autre, de façon obligatoire, à un endroit spécifié dans ses enchaînements.
- **Extension** : le cas d'utilisation de base en incorpore implicitement un autre, de façon optionnelle, à un endroit spécifié indirectement dans celui qui procède à l'extension
- **Généralisation** : les cas d'utilisation descendants héritent de la description de leur parent commun. Chacun d'entre eux peut néanmoins comprendre des relations spécifiques supplémentaires avec d'autres acteurs ou cas d'utilisation.

3.2.1. Identification des acteurs

Pour établir le diagramme de cas d'utilisation nous allons commencer tout d'abord par identifier les différents acteurs qui interagissent avec l'application et ainsi que les acteurs qui vont communiquer avec le système, ‘être humain, un autre système ou un robot’.

Les acteurs sont représentés par un stick man sous-titré par le nom de l'acteur. On distingue 2 catégories d'acteurs :

- **Les acteurs principaux** (ex : visiteur, client, ...etc.).
- **Les acteurs secondaires** (ex : opérateur de maintenance, system, ...etc.).

3.2.2. Liste des acteurs

Dans notre application nous avons défini trois acteurs principaux qui sont :

Administrateur : il a la possibilité de visualisations de Dashboard et statistiques sur les produits de la base des données. De plus il a le droit Observer les tables des clients, produits et métas données et aussi de supprimer les produits, les comptes des clients et vendeurs.

Vendeur : il a le droit d'ajouter, modifier et supprimer les produits. De plus il a le droit de visualiser de Dashboard et statistiques sur leurs produits de la base des données.

Visiteur : Consulter l'application, découvrir les produits avec leurs informations.

Client : Les fonctionnalités consacrée pour le visiteur en plus d'ajouter les produits dans le panier, aimer et voter les produits.

Opérateur de maintenance : Gérer les problèmes de site web.

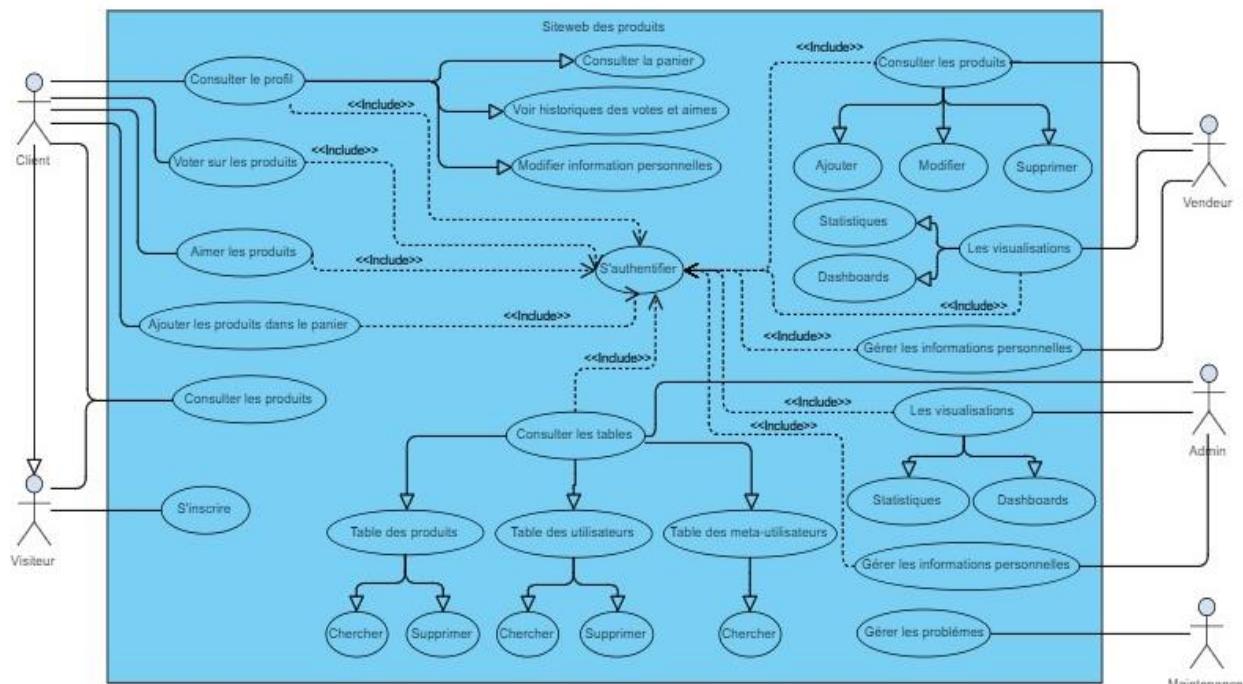


Figure 3 : Diagramme des cas d'utilisation

3.3- Diagramme d'activités

Le diagramme d'activités représente les activités que réalisent un ou plusieurs objets. Il peut correspondre à la description en détail d'une activité du diagramme d'états transitions, à la description d'une méthode. Il peut également décrire l'activité d'un système ou d'un sous-système en assignant les responsabilités à chaque acteur. Le diagramme d'activités constitue aussi un bon choix pour décrire un cas d'utilisation.

Nous avons opté à utiliser le diagramme d'activité pour modéliser les traitements adaptés à la modélisation des flots de données, et de la représentation graphique de déroulement d'un processus graphique, des figures ci-dessous illustre les digrammes d'activité.

3.3.1. Visiteur

Ce diagramme représente les différentes activités réalisées par le visiteur pendant sa navigation sur le site web

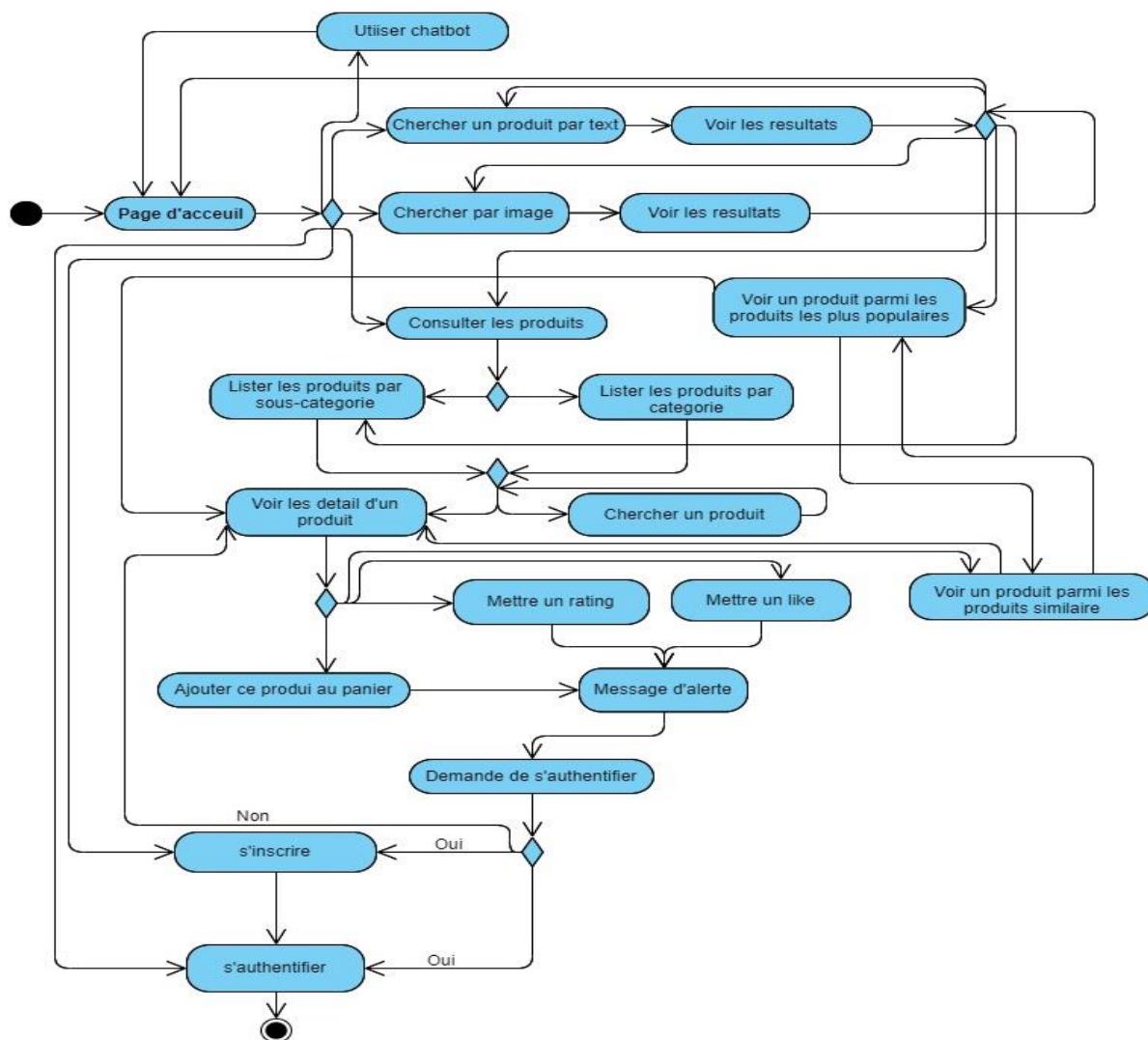


Figure 4 : Diagramme d'activité de visiteur

3.3.2. Client

Ce diagramme représente les différentes activités réalisées par le client pendant sa navigation sur le site web

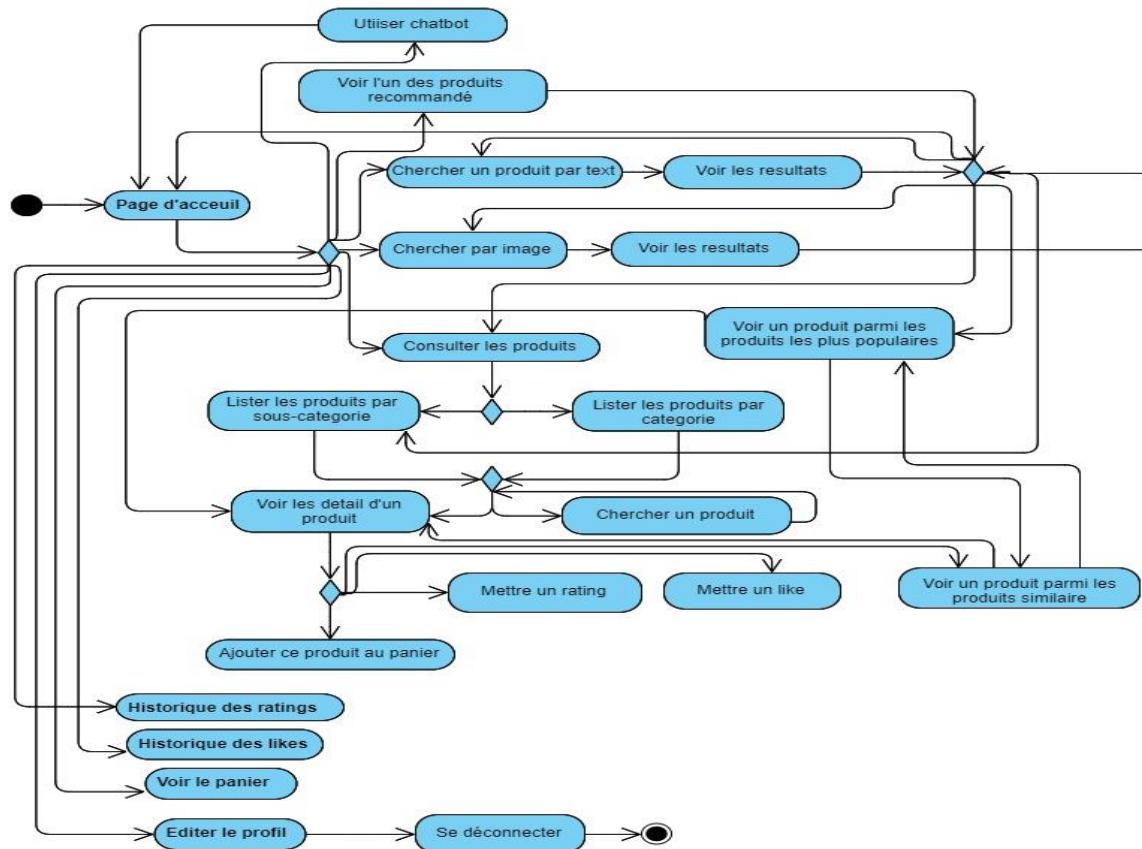


Figure 5: Diagramme d'activité de client

3.3.3. Vendeur

Ce diagramme représente les différentes activités réalisées par le vendeur pendant sa navigation sur le site web

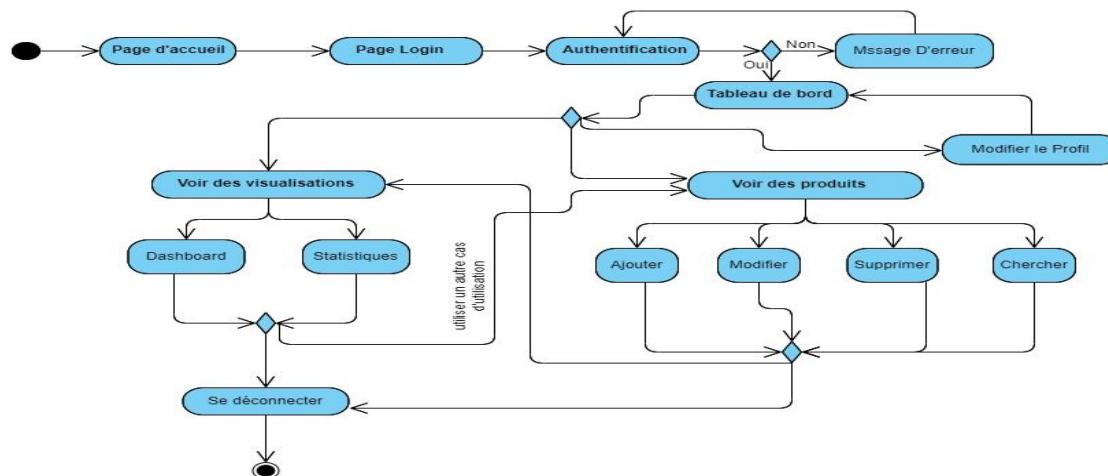


Figure 6 : Diagramme d'activité de vendeur

3.3.4. Admin

Ce diagramme représente les différentes activités réalisées par le admin pendant sa navigation sur le site web

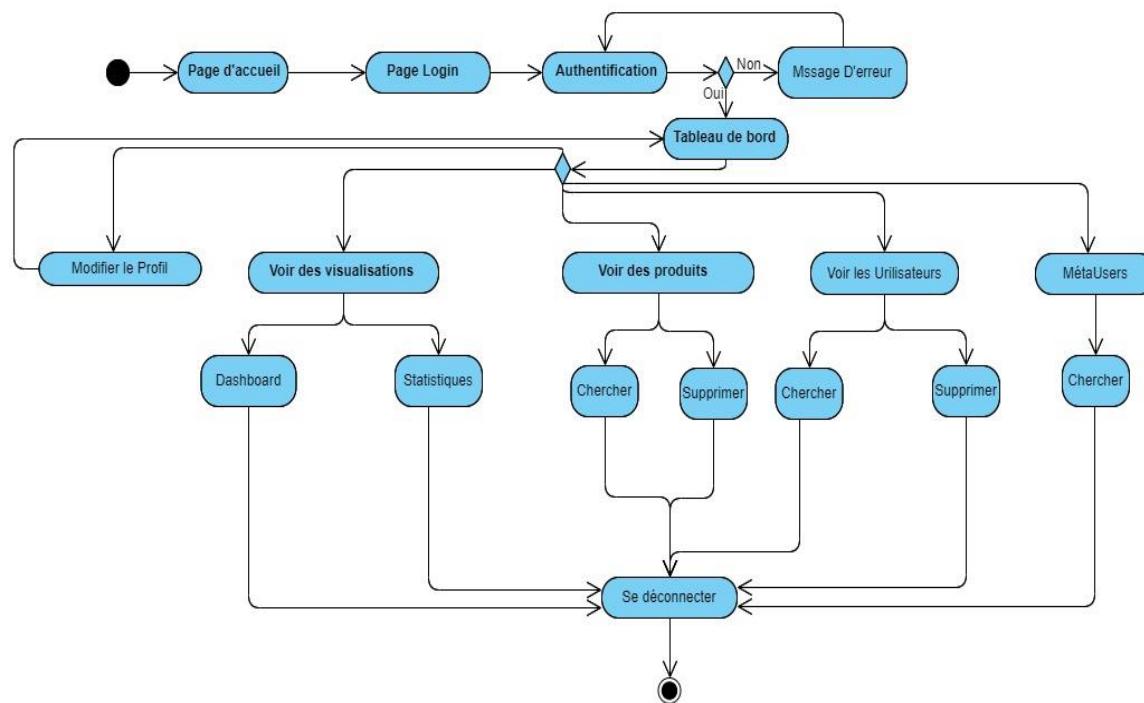


Figure 7 : Diagramme d'activité d'admin

3.4- Diagramme de séquences

Le diagramme de séquence décrit les interactions entre un groupe d'objets en montrant, de façon séquentielle, les envois de message qui interviennent entre les objets. Le diagramme peut également montrer les flux de données échangées lors des envois de message.

3.4.1. Inscription

Ce diagramme représente les différentes activités réalisées par le visiteur durant l'inscription au site, il doit d'abord saisir toutes les informations demandées, en suite le système va les enregistrer dans la base de données

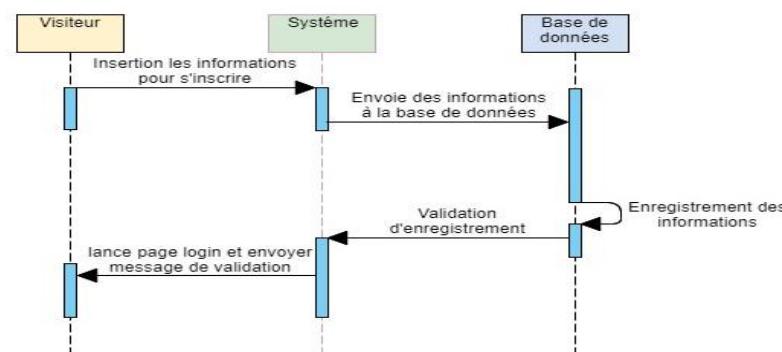


Figure 8 : Diagramme de séquences d'inscription

3.4.2. Authentification

Le diagramme exposé dans la figure ci-dessous, décrit les scénarios possibles lors d'une opération d'authentification. En effet après l'opération d'enregistrement l'utilisateur est prêt pour s'authentifier. Le système à son tour affichera une interface contenant des champs à remplir, l'utilisateur saisit leur login et leur mot de passe et valide. Le système va vérifier l'existence de login et de mot de passe qu'il lui correspondant dans la base. Le résultat retenu est soit l'acceptation de l'opération d'accès, soit le refus pour refaire le saisie.

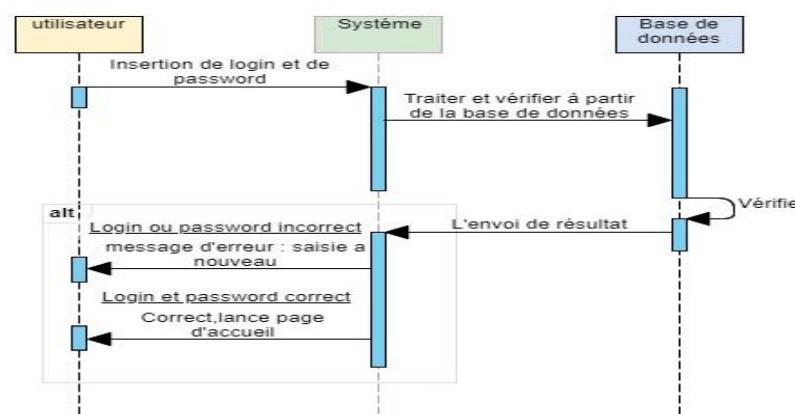


Figure 9 : Diagramme de séquences d'authentification

3.5- Diagramme de classe

Le diagramme de classe de conception représente bien la structure statique du code, par le biais des attributs et des relations entre classes, c'est un diagramme principal qui est la vue de plus haut niveau avec l'ensemble des classes de l'application, Il montre les briques de base statiques : classes, associations, interfaces, attributs, opérations, généralisations, etc.

3.5.1. Dépendances fonctionnelles

- Un produit peut être consulté par un ou plusieurs clients, et un client peut consulter un ou plus produits
- Un produit peut être ajouté au panier par un ou plusieurs clients, et un client peut ajouter au panier un ou plusieurs produits
- Un produit peut être aimé par un ou plusieurs clients, et un client peut aimer un ou plusieurs produits
- Un produit peut être voté par un ou plusieurs clients, et client peut voter un ou plusieurs produits
- Un produit peut être ajouté, modifier ou supprimer par un vendeur, et un vendeur peut ajouter modifier ou supprimer un ou plusieurs produits

- Un admin peut gérer un client, vendeur ou plusieurs clients, vendeurs, et les clients et les vendeurs avoir un seul admin

3.5.2. Diagramme de classe

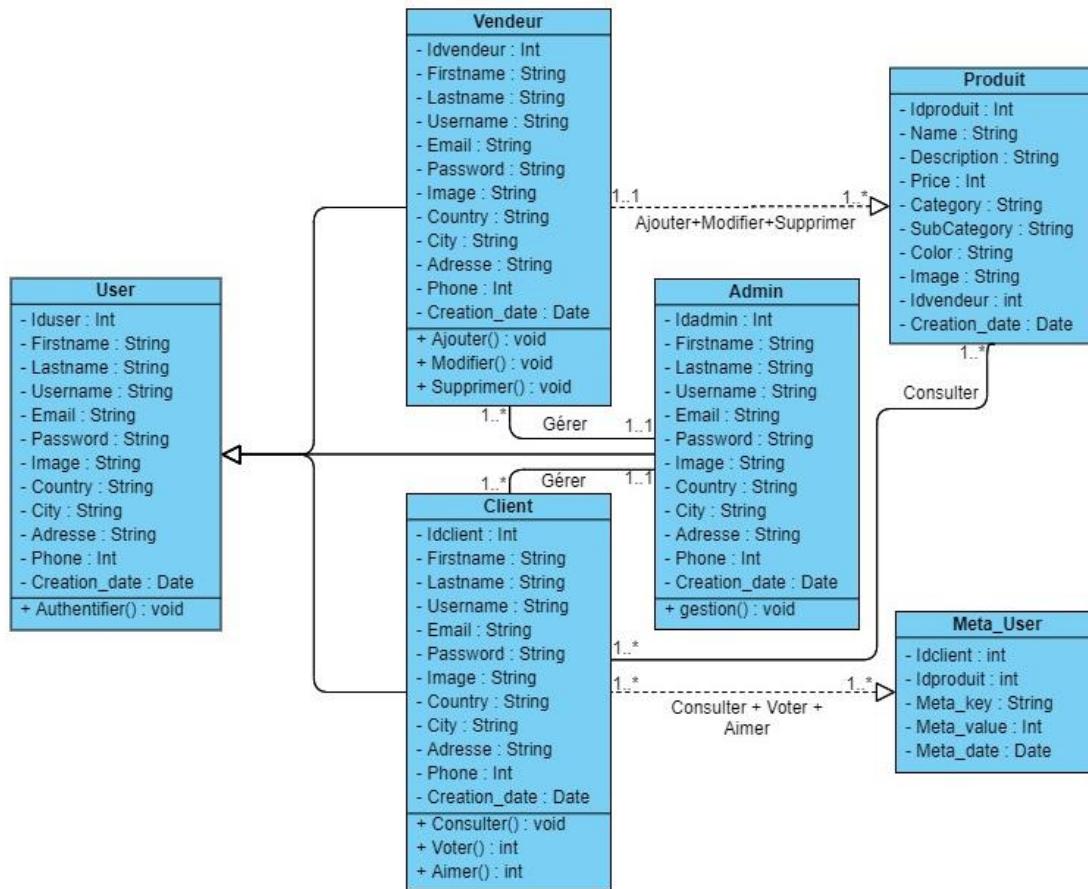


Figure 10 : Diagramme de classe

4. Conclusion

Ce chapitre est une description complète du comportement du système à développer. En premier lieu nous avons spécifié et formulé les principales fonctionnalités que doit offrir le système à concevoir, ensuite nous avons réalisé une modélisation à travers les diagrammes qui mettent en exergue le séquencement des opérations et des relations éventuelles entre les services.

Dans le chapitre suivant on va voir les technologies et outils de développement.

Chapitre 4

Technologies et outils de développement

Ce chapitre présente les différentes technologies et les outils que nous avons utilisé pour réaliser notre travail.

1. Introduction

Dans ce chapitre, nous présentons les technologies que nous avons utilisé et les préparations des environnements de travail. Nous allons présenter aussi la mise en œuvre du projet.

2. Outils utilisés

Dans cette partie, nous apportons les réponses à la question "Avec quels moyens le projet a-t-il été réalisé ?"

De ce fait, nous parlerons :

- De l'environnement de travail
- Du langage de programmation utilisé ainsi que des outils tierces qui lui sont inhérents.

2.1- L'environnement de travail

a- *Anaconda*



Anaconda Distribution est une solution Open Source, c'est le moyen le plus simple d'exercer la science des données et le Machine Learning avec Python/R sous Linux, Windows et Mac OS. Avec plus de 11 millions d'utilisateurs à travers le monde, il s'agit du standard de l'industrie pour le développement, les test et l'apprentissage sur une seule machine, permettant aux Data scientifiques de :

- Téléchargez rapidement plus de 1 500 packages Python / R de science des données.
- Gérer les bibliothèques, les dépendances et les environnements avec Conda
- Développer et entraîner des modèles Machine Learning et Deep Learning avec Scikit-learn, TensorFlow et Theano
- Analyser les données avec évolutivité et performance avec Dask, NumPy, pandas et Numpy
- Visualisez les résultats avec Matplotlib, Bokeh, Datashader et Holoviews

b- *Spyder*



Spyder est un environnement scientifique puissant écrit en Python, pour Python, et conçu par et pour les scientifiques, les ingénieurs et les analystes de données. Il offre une combinaison unique des fonctionnalités avancées d'édition, d'analyse, de débogage et de profilage d'un outil de développement complet avec l'exploration de données, l'exécution interactive, l'inspection approfondie et les belles capacités de visualisation d'un package scientifique.

c- Visual Studio



L'environnement *de développement intégré* Visual Studio est une zone de lancement créative que vous pouvez utiliser pour modifier, déboguer et créer du code, puis publier une application. Un environnement de développement intégré (IDE) est un programme riche en fonctionnalités qui peut être utilisé pour de nombreux aspects du développement logiciel.

Au-delà de l'éditeur et du débogueur standard fournis par la plupart des IDE, Visual Studio inclut des compilateurs, des outils de complétion de code, des concepteurs graphiques et de nombreuses autres fonctionnalités pour faciliter le processus de développement logiciel.

d- Robo 3T



robo 3T est un partenaire technologique MongoDB.

C'est une interface graphique et un IDE pour les développeurs et les ingénieurs de données qui travaillent avec MongoDB. Les fonctionnalités de gestion des données telles que l'édition sur place et les connexions de base de données faciles sont associées à la génération de code de requête polyglotte, au shell avancé avec auto-complétions, à l'importation / exportation SQL.

2.2- Langages utilisés

a- Python



Python est un langage de script de haut niveau, structuré et open source. Il est multi-paradigme et multi-usage.

Développé à l'origine par Guido van Rossum en 1989, il est, comme la plupart des applications et outils open source, maintenu par une équipe de développeurs un peu partout dans le monde.

Conçu pour être orienté objet, il n'en dispose pas moins d'outils permettant de se livrer à la programmation fonctionnelle ou impérative ; c'est d'ailleurs une des raisons qui lui vaut son appellation de « langage agile ».

Parmi les autres raisons, citons la rapidité de développement (qualité propre aux langages interprétés), la grande quantité de modules fournis dans la distribution de base ainsi que le nombre d'interfaces disponibles avec des bibliothèques écrites en C, C++ ou Fortran. Il est également apprécié pour la clarté de sa syntaxe.

Pour plusieurs raisons, les data scientistes ont choisi python comme meilleurs langage utilisé en data science, parmi ces raisons nous trouvons :

- Python est facile à utiliser
- Python est polyvalent
- Python est meilleur pour la construction d'outils d'analyse
- Visualisation des données avec Python
- La communauté Python s'agrandit
- Python est meilleur pour l'apprentissage en profondeur

b- Flask API



Flask API est un remplacement direct de Flask qui fournit une implémentation d'API navigables. Il vous donne des réponses négociées correctement et une analyse intelligente des demandes. Il fournit des fonctionnalités pour la création d'applications Web, y compris la gestion des requêtes HTTP et des modèles de templates.

c- Angular



Angular est un Framework open source écrit en JavaScript qui permet la création d'applications Web et plus particulièrement de ce qu'on appelle des « Single Page Applications » en utilisant HTML et TypeScript : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action.

Le Framework est basé sur une architecture du type MVC et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités. Un type d'architecture qui a largement fait ses preuves et qui permet une forte maintenabilité et une amélioration du travail collaboratif.

d- MongoDB



mongoDB

MongoDB est une base de données NoSQL orientée objet, simple, dynamique et évolutive. Il est basé sur le modèle de magasin de documents NoSQL. Les objets de données sont stockés en tant que documents séparés dans une collection - au lieu de stocker les données dans les colonnes et les lignes d'une base de données relationnelle traditionnelle. La motivation du langage MongoDB est d'implémenter un magasin de données qui offre des performances élevées, une haute disponibilité et une mise à l'échelle automatique. MongoDB est extrêmement simple à installer et à implémenter. MongoDB utilise des documents JSON ou BSON pour stocker des données

2.3- Librairies

a- NumPy



NumPy est le paquet fondamental du calcul scientifique avec Python. Il contient entre autres :

- Un puissant objet tableau N-dimensionnel
- Fonctions sophistiquées (diffusion)
- Outils d'intégration de code C / C ++ et Fortran
- Algèbre linéaire utile, transformée de Fourier et capacités de nombres aléatoires

Plus ses utilisations scientifiques évidentes, NumPy peut également être utilisé comme un conteneur multidimensionnel efficace de données génériques. Des types de données arbitraires peuvent être définis. Cela permet à NumPy de s'intégrer de manière transparente et rapide à une grande variété de bases de données.

NumPy est sous licence BSD, ce qui permet sa réutilisation avec peu de restrictions.

b- Pandas



Pandas est une bibliothèque sous licence BSD2. Elle est écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles.

Les principales structures de données sont :

- Les séries (pour stocker des données selon une dimension - grandeur en fonction d'un index)
- Les dataframes (pour stocker des données selon 2 dimensions - lignes et colonnes)
- Les Panels (pour représenter des données selon 3 dimensions)
- Les Panels4D ou les dataframes avec des index hiérarchiques aussi nommés Multi Index (pour représenter des données selon plus de 3 dimensions - hypercube)

Ses principales fonctionnalités sont :

- L'objet dataframe pour manipuler des données aisément et efficacement avec des index pouvant être des chaînes de caractères
- Des outils pour lire et écrire des données structurées en mémoire depuis et vers différents formats : fichiers CSV, fichiers textuels, fichier du tableur Microsoft Excel, base de

données SQL ou le format rapide et permettant de gérer de gros volume de données nommé HDF5

- Alignement intelligent des données et gestion des données manquantes (NaN = not a number), alignement des données basé sur des étiquettes (chaines de caractères) et tri selon divers critères de données totalement désordonnées
- Redimensionnement et table pivot ou pivot table en anglais (aussi nommé tableau croisé dynamique)
- Fusion et jointure de large volume de données
- Analyse de séries temporelles

c- *Keras*



La bibliothèque Keras permet d'interagir avec les algorithmes de réseaux de neurones profonds et de machine learning, notamment Tensorflow, Theano, Microsoft Cognitive Toolkit ou PlaidML.

Conçue pour permettre une expérimentation rapide avec les réseaux de neurones profonds, elle se concentre sur son ergonomie, sa modularité et ses capacités d'extension. Elle a été développée dans le cadre du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System). Elle a été initialement écrite par François Chollet.

Keras a été créé pour être convivial, modulaire, facile à étendre et fonctionner avec Python. L'API a été « conçue pour les êtres humains, pas pour les machines », et « suit les meilleures pratiques pour réduire la charge cognitive ».

Les couches neuronales, les fonctions de coût, les optimiseurs, les schémas d'initialisation, les fonctions d'activation et les schémas de régularisation sont tous des modules autonomes que vous pouvez combiner pour créer de nouveaux modèles. De nouveaux modules sont simples à ajouter, comme de nouvelles classes et fonctions. Les modèles sont définis dans du code Python, pas des fichiers de configuration de modèle séparés.

Les principales raisons d'utiliser Keras proviennent de ses principes directeurs, principalement celui d'être convivial. Au-delà de la facilité d'apprentissage et de la construction de modèles, Keras offre les avantages d'une large adoption, la prise en charge d'un large éventail d'options de déploiement en production, l'intégration avec au moins cinq moteurs principaux (TensorFlow, CNTK, Theano, MXNet et PlaidML), et un support solide pour plusieurs GPU et une formation distribuée. De plus, Keras est soutenu par Google, Microsoft, Amazon, Apple, Nvidia, Uber et autres

d- Scikit-learn



Scikit-learn est une bibliothèque libre Python destinée à l'apprentissage automatique. Elle est développée par de nombreux contributeurs notamment dans le monde académique par des instituts français d'enseignement supérieur et de recherche comme Inria et Télécom ParisTech

Il comporte plusieurs méthodes mathématiques pour :

- La classification automatique
- La régression
- La réduction de dimensions
- Le clustering
- La sélection de modèle
- Les pré-traitements incluant la normalisation

Elle est conçue pour s'harmoniser avec d'autres bibliothèques libres Python, notamment NumPy et SciPy.

e- Pymongo



PyMongo est une distribution Python contenant des outils pour travailler avec MongoDB, et est la méthode recommandée pour travailler avec MongoDB à partir de Python. Cette documentation tente d'expliquer tout ce que vous devez savoir pour utiliser PyMongo

La bibliothèque PyMongo permet l'interaction avec la base de données MongoDB via Python; il s'agit d'un pilote Python natif pour MongoDB.

3. Conclusion

Durant ce chapitre nous avons présenté les Technologies et outils de développement, Le prochain chapitre est consacré à la présentation du travail fait pour la mise en œuvre du projet.

Chapitre 5

Réalisation de l'application

Ce Chapitre présente la réalisation de notre application commençons par la base de données utilisé, puis la présentation des algorithmes utilisés dans le backend, la présentation de frontend, Enfin le déploiement de modèle dans un rest API en utilisant le framework « Flask »

1. Introduction

Après avoir terminé l'étude et la conception et les outils et les technologies utilisées dans notre application, nous allons présenter dans ce chapitre différentes processus que nous avons suivi pour réaliser une tache de notre projet ainsi que le résultat du travail effectué. Nous clôturons ce chapitre par quelques captures d'écran démontrant les fonctionnalités de notre application.

L'IA se compose de plusieurs sous catégories en termes de techniques et de domaines d'application (voir figure 11), nous nous concentrons dans la suite de ce rapport sur les sous-catégories : Machine Learning, vision et Traitement de la Langue Naturel (NLP).



Figure 11: Sous Catégories de l'intelligence artificielle

2. Base de données

2.1- Collection de données

2.1.1 Définition

La collecte des données réfère à tout processus dont l'objectif est d'acquérir ou de faciliter l'acquisition des données. On procède à cette collecte en demandant et en obtenant des données pertinentes auprès de personnes ou d'organismes au moyen de procédés adéquats.

2.1.2 Sources des données

a) Kaggle



Kaggle est une plate-forme de concours de modélisation prédictive et d'analyse dans laquelle les entreprises et les chercheurs publient des données et les statisticiens et les mineurs de données s'affrontent pour produire les meilleurs modèles de prédiction et de description des données.

b) Data.World



Data.world englobe la plus grande communauté de données collaborative au monde, qui est gratuite et ouverte au public. C'est là où les gens découvrent les données, partagent l'analyse et s'associent sur tout.

2.2- Prétraitement de données

Nous avons donc collecté des données et les stocker dans des dataframes par des différentes caractéristiques et après on fait le traitement et le nettoyage par les outils de pandas pour faire une sorte des données les plus propres possibles, notre résultat était la suivante :

- 25000 produits, 8 caractéristiques

produitId	Categories	Product_img	Product_name	Product_price	baseColor	description	gender	subCategory
0	Accessories	http://assets.mynta...	Parx Men Brown Belt	389	Green	Belts	Men	belts
1	Accessories	http://assets.mynta...	Estelle Gold Earrings	225	Gold	Earrings	Women	jewellery
2	Accessories	http://assets.mynta...	Pepe Jeans Men Casual S...	247	Black	Sunglasses	Men	eyewear
3	Accessories	http://assets.mynta...	Lucera Women Silver Ring	523	Silver	Ring	Women	jewellery
4	Accessories	http://assets.mynta...	Lucera Women Silver Penda...	279	Silver	Pendant	Women	jewellery
5	Accessories	http://assets.mynta...	Ivory Tag Women Dazzli...	2729	Gold	Necklace and Chains	Women	jewellery
6	Accessories	http://assets.mynta...	Estelle Women Gold Earrings	359	Gold	Earrings	Women	jewellery
7	Accessories	http://assets.mynta...	Adrika Blue Necklace	299	Blue	Necklace and Chains	Women	jewellery
8	Accessories	http://assets.mynta...	United Colors of Benetton ...	759	Lavender	Handbags	Women	bags
9	Accessories	http://assets.mynta...	Estelle Women Gold Bracelet	804	Gold	Bracelet	Women	jewellery
10	Accessories	http://assets.mynta...	Miki Pearl Women Ivory ...	680	White	Jewellery Set	Women	jewellery
11	Accessories	http://assets.mynta...	Proline Men Navy & White...	140	Navy Blue	Caps	Men	headwear
12	Accessories	http://assets.mynta...	Ivory Tag Women Clutte...	296	Gold	Necklace and Chains	Women	jewellery
13	Accessories	http://assets.mynta...	Pitaraa Silver Beade...	679	Silver	Necklace and Chains	Women	jewellery
14	Accessories	http://assets.mynta...	Baggit Women Orange Tinki...	745	Orange	Handbags	Women	bags
15	Accessories	http://assets.mynta...	Lino Perros Women Leathe...	976	Brown	Handbags	Women	bags
16	Accessories	http://assets.mynta...	Femella Bronze Neckl...	162	Bronze	Necklace and Chains	Women	jewellery
17	Accessories	http://assets.mynta...	Lucera Women Silver Earri...	235	Silver	Earrings	Women	jewellery
18	Accessories	http://assets.mynta...	Femella Silver Knot ...	2956	Silver	Necklace and Chains	Women	jewellery
19	Accessories	http://assets.mynta...	Basics Men Red Cap	247	Red	Caps	Men	headwear
20	Accessories	http://assets.mynta...	Femella Blue Necklace	957	Blue	Necklace and Chains	Women	jewellery

Figure 12: Table des produits avec pandas (DataFrame)

On a donc uniquement exporté Les informations importantes pour l'affichage dans un fichier json grâce aux outils de pandas. Ensuite nous pouvons stocker les résultats dans notre base de données.

2.3- Stockage de données

Le stockage de données est l'ensemble des méthodes et technologies permettant d'entreposer et de conserver les informations numériques.

Pour le stockage de données nous avons utilisé Studio3T pour importer les données sous forme de fichier json dans notre base de données MongoDB

- Exemple de Table Product :

_id	productid	Categories	Product_img	Product_name	Product_price	baseColor	description	gender	subCategory	vendeurid
5e74a9b5...	28	Accessories	http://assets.my...	Lucera Wom...	389	Silver	Pendant	Women	jewellery	381
5e74a9b5...	29	Accessories	http://assets.my...	ToniQ Wom...	487	Multi	Necklace and...	Women	jewellery	381
5e74a9b5...	30	Accessories	http://assets.my...	Lencia Sterli...	139	Black	Ring	Women	jewellery	381
5e74a9b5...	31	Accessories	http://assets.my...	Lencia Sterli...	295	Silver	Pendant	Women	jewellery	381
5e74a9b5...	32	Accessories	http://assets.my...	Royal Diade...	957	Gold	Earrings	Women	jewellery	381
5e74a9b5...	33	Accessories	http://assets.my...	Lucera Wom...	500	Silver	Pendant	Women	jewellery	381
5e74a9b5...	34	Accessories	http://assets.my...	Idee Men Bla...	195	Black	Sunglasses	Men	eyewear	381
5e74a9b5...	35	Accessories	http://assets.my...	Speedo Men...	450	Black	Sunglasses	Men	eyewear	381
5e74a9b5...	36	Accessories	http://assets.my...	United Color...	467	Black	Sunglasses	Men	eyewear	381
5e74a9b5...	37	Accessories	http://assets.my...	Ray-Ban Me...	190	Metallic	Sunglasses	Men	eyewear	381
5e74a9b5...	38	Accessories	http://assets.my...	Estelle Wom...	363	Gold	Pendant	Women	jewellery	381
5e74a9b5...	39	Accessories	http://assets.my...	Royal Diade...	279	Gold	Bangle	Women	jewellery	381
5e4a9b5...	40	Accessories	http://assets.my...	Fabinda Wo...	454	Silver	Earrings	Women	jewellery	381
5e74a9b5...	41	Accessories	http://assets.my...	Royal Diade...	1548	Green	Earrings	Women	jewellery	381
5e74a9b5...	42	Accessories	http://assets.my...	Ivory Tag Wo...	5580	Gold	Necklace and...	Women	jewellery	381
5e74a9b5...	43	Accessories	http://assets.my...	Oakley Men ...	277	Black	Sunglasses	Men	eyewear	381

Figure 13: Table des produits dans Studio3T

3. Partie Backend

3.1- Algorithme de recommandation

3.1.1 Définition

Un système de recommandation est une sorte de forme spécifique de filtrage de l'information visant à présenter les éléments susceptibles d'intéresser l'utilisateur.

Généralement, un système de recommandation permet de comparer le profil d'un utilisateur à certaines caractéristiques de référence, et cherche à prédire l'« avis » que donnerait un utilisateur. Et si en parle de système de recommandation dans l'environnement social c'est automatiquement on parle d'approche de filtrage collaboratif

3.1.2 filtrage collaboratif :

Cette recommandation se base sur les événements qui lient les utilisateurs aux items et qui présente d'une manière implicite ou explicite son goût.

Nous distinguons deux types de Collaborative Filtering :

- **Memory-based :** les modèles Memory based sont facile à implémenter et produisent une qualité de prévision raisonnable. Ce type de modèles n'est pas scalable, c-à-d n'est pas pratique dans un problème d'une grande base de données vu qu'il calcule à chaque fois la corrélation entre tous les utilisateurs et les items. Ainsi, il ne résout pas le problème de cold start, lorsqu'on commence avec un nouvel utilisateur/item dont on n'a pas assez d'information.
- **Model-Based :** c'est une méthode d'apprentissage non supervisé de décomposition et de réduction de dimensionnalité pour les variables cachées. Le but de la matrice de factorisation est d'apprendre les préférences cachées des utilisateurs et les attributs cachés des items depuis les ratings connus dans notre jeu de données, pour enfin prédire les ratings inconnus en multipliant les matrices de variables latentes des utilisateurs et des items. Il existe plusieurs techniques de réduction de dimensionnalité dans l'implémentation des systèmes de recommandations.

Dans notre projet, nous avons utilisé : ALS : Alternating Least Squares, car sa valeur de l'erreur est la meilleure parmi toutes les autres méthodes testées (SGD, SVD).

```
[Epoch 1/2] train error: 1.046265, test error: 1.373295
[Epoch 2/2] train error: 0.765880, test error: 1.243846
Algorithm converged
```

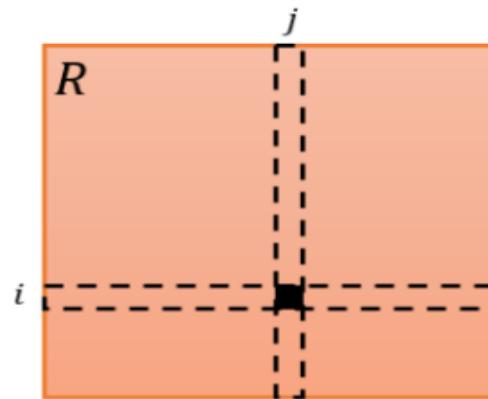
Figure 14: Erreur obtenu de ALS

Cet algorithme est le meilleur de tous les autres algorithmes. Dans la deuxième itération on a trouvé une erreur de train qui est égale à 0.765880 et une erreur de test qui est égale à 1.243846 Comme c'est l'algorithme le plus rapide et le plus efficace, On a décidé de le généralisé sur tout le jeu de données.

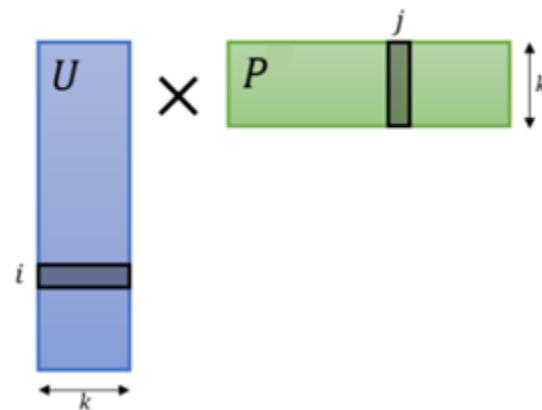
3.1.3 Recommandation Collaborative Filtering (ALS) :

Après avoir choisi ALS comme le meilleur modèle, nous avons présenté les étapes principales de ce modèle en détails :

- Comme le montre l'image ci-dessous, supposons que nous avons une matrice de ratings originale R de taille $M \times N$, où M est le nombre d'utilisateurs et N est le nombre de produits.



- Cette matrice est assez clairsemée, car la plupart des utilisateurs n'interagissent qu'avec quelques éléments. Nous pouvons factoriser cette matrice en deux matrices distinctes plus petites : une avec des dimensions $M \times K$ qui seront nos vecteurs de fonctionnalités utilisateur latents pour chaque utilisateur U et une seconde avec des dimensions $K \times N$, qui contiendra nos vecteurs de fonctionnalité de produits latent pour chaque élément P .



- Initialisez le matrice P en attribuant le rating moyenne pour chaque produit comme première ligne, et de petits nombres aléatoires pour les entrées restantes.
- Afin de résoudre pour U et P , nous avons appliqué ALS pour approcher la factorisation plus précisément. Pour ce faire, nous pouvons Fixer P et résoudre pour U en minimisant la fonction objective (RMSE). Ensuite, nous pouvons revenir en arrière et résoudre pour P en utilisant notre solution pour U qui est dans ce cas fixer en minimisant la fonction objective de la même manière. Nous utilisant les formules suivantes :

$$\begin{aligned} &\Rightarrow (M_{I_i} M_{I_i}^T + \lambda n_{u_i} E) \mathbf{u}_i = M_{I_i} R^T(i, I_i), \quad \forall i \\ &\Rightarrow \mathbf{u}_i = A_i^{-1} V_i, \quad \forall i \end{aligned}$$

Dans : $A_i = M_{I_i} M_{I_i}^T + \lambda n_{u_i} E$, $V_i = M_{I_i} R^T(i, I_i)$, et E est l'identité $K \times K$ matrice, M_{I_i} désigne la sous-matrice de P où les colonnes $j \in I_i$ sont sélectionnées, et $R(i, I_i)$ est le vecteur ligne où on prend les colonnes $j \in I_i$ de la i -ème ligne de R .

De même, lorsque P est mis à jour, nous pouvons calculer les m_j individuels via une solution de moindres carrés linéaires régularisés, en utilisant les vecteurs de fonctionnalité des utilisateurs qui ont évalué produit j , et leurs ratings, comme suit :

$$\Rightarrow (\mathbf{U}_{I_j} \mathbf{U}_{I_j}^T + \lambda n_{m_j} E) \mathbf{m}_j = \mathbf{U}_{I_j} \mathbf{R}(I_j, j), \quad \forall j,$$

$$\Rightarrow \mathbf{m}_j = A_j^{-1} \mathbf{V}_j, \quad \forall j,$$

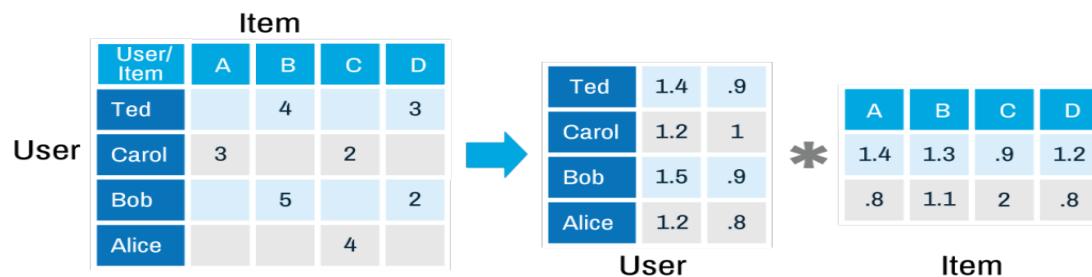
Dans : $\mathbf{A}_j = \mathbf{U}_{I_j} \mathbf{U}_{I_j}^T + \lambda n_{m_j} E$ et $\mathbf{V}_j = \mathbf{U}_{I_j} \mathbf{R}(I_j, j)$. et E est l'identité $K \times K$ matrice, \mathbf{U}_{I_j} désigne la sous-matrice de \mathbf{U} où la colonne $i \in I_j$ sont sélectionnés, et $\mathbf{R}(I_j, j)$ est le vecteur colonne où on prend les lignes $i \in I_j$ de la j -ème colonne de \mathbf{R} .

- Continuez à itérer dans les deux sens comme ceci jusqu'à ce que nous obtenions une convergence qui se rapproche R du mieux que nous pouvons. En fait, le critère d'arrêt que nous utilisons est basé sur le RMSE observé ($RMSE < 1$)

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (X_{obs,t} - X_{model,t})^2}{n}}$$

[Epoch 1/2] train error: 1.046265
[Epoch 2/2] train error: 0.765880
Algorithm converged

Avec : $X_{obs,t}$ est la matrice original , $X_{model,t}$ est la matrice après prédiction et n est la longueur de matrice original



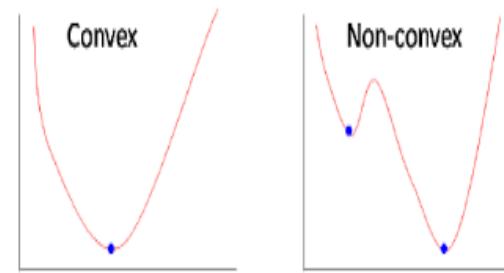
- L'itération s'arrête et nous utilisons les U , P obtenus pour faire des prédictions finales sur l'ensemble de données de test pour avoir une matrice R de ratings totalement remplie. ($R =$ Produit scalaire de U Transposé et P)

	U1	U2	U3	U4
I1	3.55	2.91	3.68	3.85
I2	3.25	2.66	3.37	3.08
I3	3.7	3.42	4	3.85
I4	3.32	2.86	3.6	3.49

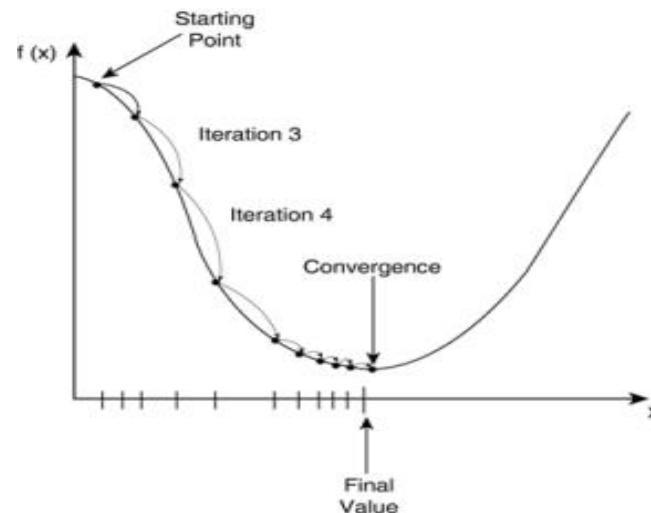
R

Remarque :

Plus en détail, pourquoi nous alternons, parce que nous transformons le problème d'un problème non convexe en problème convexe. L'optimisation simultanée de U et V est non convexe et difficile à résoudre.



Si nous fixons U ou V, nous devenons convexes, ce qui est facile à résoudre.



Dans notre cas nous avons généré la matrice de recommandation User-Item sur tout notre jeu de données : 380 Users-25000 Produits.

Et voici une illustration qui montre d'une part la matrice après l'application de l'algorithme ALS et d'autre part l'efficacité de ce module à travers les cellules sélectionnées.

	2849	2850	2851	2852	2853		2849	2850	2851	2852	2853
0	0	0	0	0	2		0	1	3	3	3
1	0	0	0	0	0		1	2	3	2	2
2	0	0	0	0	0		2	1	3	2	2
3	0	0	0	0	0		3	1	2	3	2
4	0	0	0	0	0		4	2	3	3	1
5	0	0	0	0	0		5	1	3	3	2
6	0	0	0	0	0		6	1	3	3	3
7	0	0	0	0	0		7	1	3	3	1
8	1	0	0	0	0	Résultat	8	1	3	3	2
9	0	0	1	0	0		9	1	3	3	2
10	0	0	0	0	0		10	1	3	3	2
11	0	0	0	0	0		11	1	3	2	2

Figure 15: la matrice de recommandation User-Item de ALS

Nous voulons à travers ces résultats de retourner les plus recommander produits pour n'importe client.

3.1.4 Exemple d'utilisation :

Dans notre application nous affichons pour chaque client le résultat des produits les plus recommander pour lui.

You May Like

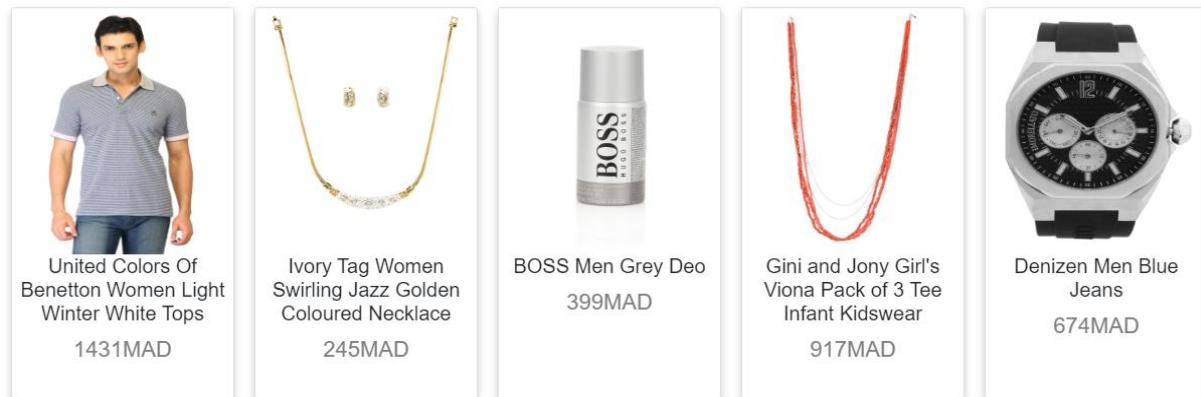


Figure 16: L'interface You May Like dans la page home

Si le client click sur une catégorie ou bien une sous-catégorie, on va l'afficher les produits plus recommander juste pour cette catégorie ou sous-catégorie.

You May Like

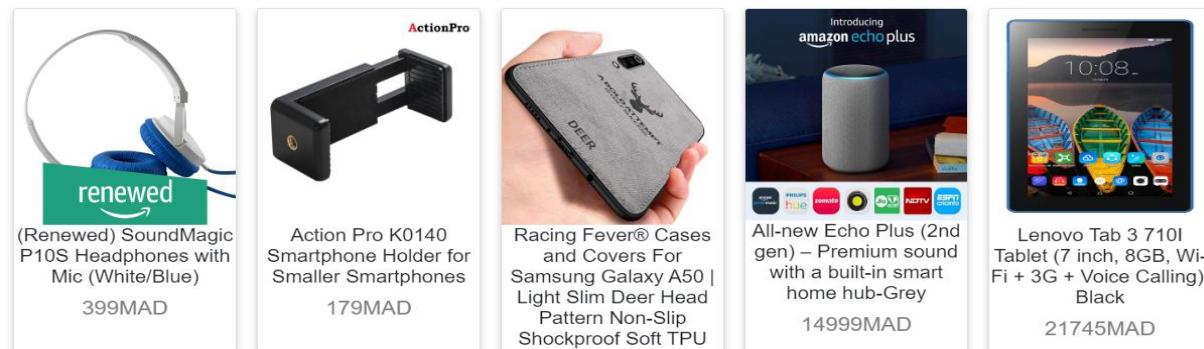


Figure 17: L'interface You May Like dans la page catégorie ou sous-catégorie

3.2- Mesure de la similarité entre les produits

3.2.1 Définition

L'idée ici est de créer une matrice carrée ($l \times l$: avec $l =$ nombre de produits) qui contient la distance entre une paire de produit afin d'en déduire les produits les plus proche entre eux, et donc les plus similaires :

item						
		1				
item	1		*		*	
	*					
			*			
				*		
					*	
						1

Pour réaliser cette matrice nous avons besoin du nombre de mots des mots dans chaque document, on a utilisé la fonction CountVectorizer () du package sklearn de Python. Et La sortie de ceci vient comme sparse_matrix.

On peut convertir Sparse Matrix en Pandas Dataframe pour voir les fréquences des mots.

Maintenant qu'on a construit la matrice que l'on souhaitait, il est temps de choisir une métrique pour calculer la similarité entre deux items.

Il existe plusieurs métriques pour mesurer la similarité entre deux items, dans notre cas on a choisi cosine_similarity. Car elle est avantageuse, même si les deux documents ou deux vecteurs similaires sont éloignés l'un de l'autre par la distance euclidienne par exemple, il est probable qu'ils puissent toujours être orientés plus près l'un de l'autre.

Plus l'angle est petit, plus la similitude cosinus est élevée.

3.2.2 La métrique cosine_similarity

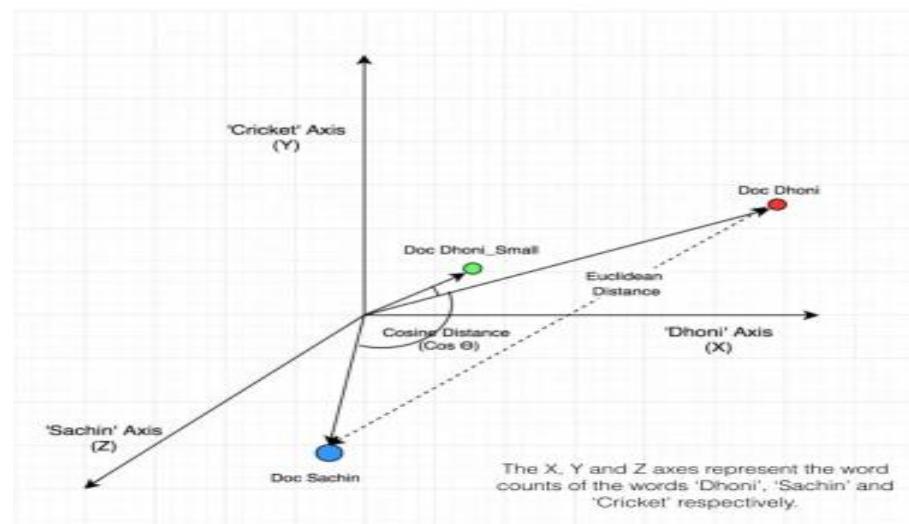
cosine_similarity entre deux vecteurs (ou deux documents sur l'espace vectoriel) est une mesure qui calcule le cosinus de l'angle entre eux. Mathématiquement, il mesure le cosinus de l'angle entre deux vecteurs projetés dans un espace multidimensionnel. Dans ce contexte, les deux vecteurs dont nous parlons sont des tableaux contenant le nombre de mots de deux documents. Ce que nous devons faire pour construire l'équation de cosine_similarity est de résoudre l'équation suivante :

$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \|\vec{b}\| \cos \theta$$

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|}$$

Ensuite, utilisez cosine_similarity() pour obtenir la sortie finale.

Projection of Documents in 3D Space



Cette Méthode de cosine_similarity nous l'avons utilisé sur deux cas dans notre projet :

a) Similaire Produits

Traitons le cas où on veut mesurer la similarité des items en fonction des variables suivants : "nom", "description", "Catégorie", "Sous-catégorie", "Avis".

Après l'application de la métrique cosine_similarity, on obtient finalement notre matrice qui mesure la similarité entre toutes paires d'item :

1	0.612372	0.158114	0.25
0.612372	1	0.129099	0.204124
0.158114	0.129099	1	0.158114
0.25	0.204124	0.158114	1

Exemple d'utilisation :



Figure 18:L'interface de similaire produits

b) Moteur de recherche textuelle

Traitons le cas où on veut mesurer la similarité des items en fonction de la variable suivants : "nom"

Après l'application de la métrique cosine_similarity, on obtient finalement notre matrice qui mesure la similarité entre toutes paires d'item :

0.257179	0.284194	0.203234	0.281075	0.211807	0.257901
----------	----------	----------	----------	----------	----------

Exemple d'utilisation :

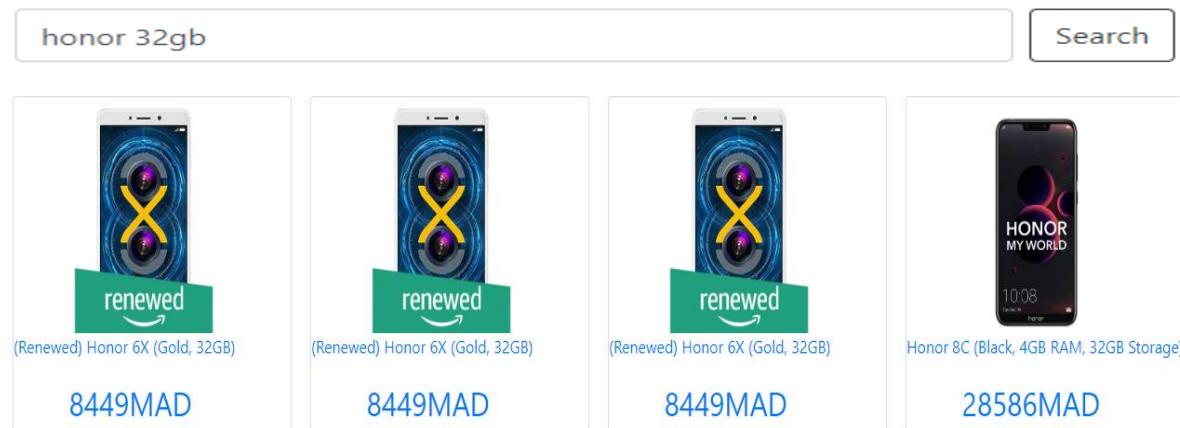


Figure 19: résultat de Moteur de recherche textuelle

3.3- Chatbot

3.3.1 Définition et explication

Un chatbot est un logiciel intelligent capable de communiquer et d'exécuter des actions similaires à un être humain. Les chatbots sont beaucoup utilisés dans l'interaction client, le marketing sur les sites de réseaux sociaux et la messagerie instantanée du client. Il existe deux types de base de modèles de chatbot : Modèle basés sur la récupération et Modèle basés sur la génération.

Dans notre projet, nous allons créer un chatbot basé sur la récupération. En effet le deuxième chatbot nécessite une très grande quantité des données avec l'utilisation des super ordinateurs.

3.3.2 Architecture

Un chatbot basé sur la récupération utilise des modèles d'entrée et des réponses prédefinis. Il utilise ensuite un certain type d'approche heuristique pour sélectionner la réponse appropriée, et voici un schéma qui explique le mécanisme de cette chatbot :

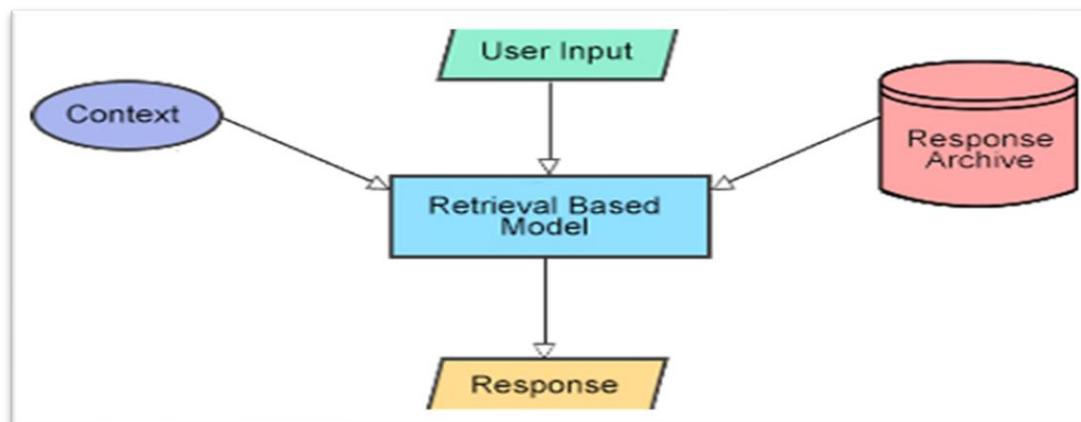


Figure 20: L'architecture d'un chatbot basé sur la récupération

3.3.3 Structure

a) La base de données

L'ensemble de données que nous avons utilisé est «PFE.json». Il s'agit d'un fichier JSON qui contient les modèles que nous devons rechercher et les réponses que nous voulons renvoyer à l'utilisateur de plus nous avons utilisé l'anglais et le français en raison de la langue de l'utilisateur, et voici une illustration de notre base de données json :

```
{
  "tag": "genre_Personnal_Care",
  "patterns": ["Quelle sont les genre , types ou subcatégories de soins personnels que vendez vous?"],
  "responses": ["les genres des soins personnels sont : Parfum, lèvres, yeux, cheveux, maquillage, bain et corps, soins de la peau et ongles"],
  "context": []
},
{
  "tag": "kind_Personnal_Care",
  "patterns": ["What kinds, types or subcategories of personal care do you sell?"],
  "responses": ["The kinds of Personnal Care are : Fragrance,Lips,Eyes,Hair,Makeup,Bath and Body,Skin Care and Nails"],
  "context": []
},
{
  "tag": "genre_Sporting_Goods",
  "patterns": ["Quelle sont les genre , types ou subcatégories de articles de sport que vendez vous?"],
  "responses": ["Les genres des articles de sport sont : Sacs, couvre-chefs, bouteille d'eau, lunettes, montres et équipement de sport"],
  "context": []
},
{
  "tag": "kind_Sporting_Goods",
  "patterns": ["What kinds, types or subcategories of Sporting Goods that you sell?"],
  "responses": ["The kinds of Sporting Goods are : Bags,Headwear,Water Bottle,Eyewear,Watches and Sport equipement"],
  "context": []
}
},
```

Figure 21: la base de données utilisé dans chatbot

b) Structure des fichiers

PFE.json : Le fichier de données qui a des modèles et des réponses prédéfinis.

PFEtrain_chatbot.py : Dans ce fichier Python, nous avons écrit un script pour construire le modèle, prétraiter et vectoriser la question de requête puis prédire la phrase de réponse afin de former notre chatbot.

PFEchatbot_words.pkl : Il s'agit d'un fichier pickle dans lequel nous stockons les mots objet Python qui contient une liste de notre vocabulaire.

PFEchatbot_classes.pkl : Le fichier pickle des classes contient la liste des catégories.

PFEmodel_chatbot.h5 : Il s'agit du modèle entraîné qui contient des informations sur le modèle.

3.3.4 étapes

Nous allons construire un chatbot en utilisant des techniques d'apprentissage approfondi. Le chatbot sera formé sur l'ensemble de données qui contient les catégories (intentions), le modèle et les réponses. Nous utilisons un réseau neuronal artificiel (ANN) pour classer à quelle catégorie appartient le message de l'utilisateur, puis nous donnerons une réponse aléatoire à partir de la liste des réponses.

En ce qui concerne les étapes que nous avons pris pour construire ce chatbot sont les suivants :

- Import et charger le fichier de données json
- Prétraitement des données : effectuer divers prétraitements sur les données avant de créer un modèle d'apprentissage
 - La tokenisation est la première et la plus simple chose que vous puissiez faire sur les données texte. La tokenisation est le processus de décomposition de tout le texte en petites parties comme des mots. Et ajoutons chaque mot dans la liste de mots.
 - La lemmatisation est le processus de conversion d'un mot dans sa forme d'origine, puis de création d'un fichier pickle pour stocker les objets Python que nous utiliserons lors de la prédiction. Et supprimer les mots en double de la liste.
- Créer les données d'entraînement : nous allons créer les données d'entraînement dans lesquelles nous fournirons l'entrée et la sortie. Notre entrée sera le modèle (question) et la sortie sera la classe à laquelle appartient notre modèle d'entrée. Mais l'ordinateur ne comprend pas le texte, nous allons donc convertir le texte en chiffres.
- Construire le modèle : nous allons construire un réseau neuronal profond qui a 3 couches. Nous utilisons pour cela l'API séquentielle Keras.
- Prédire la réponse :
 - Le modèle nous indiquera que la classe à laquelle il appartient, nous allons donc implémenter certaines fonctions qui identifieront la classe.
 - Coder une interface utilisateur graphique. Nous prendrons le message d'entrée de l'utilisateur puis utiliserons les fonctions d'assistance que nous avons créées pour obtenir la réponse du bot et l'afficher sur l'interface graphique

3.3.5 Illustration de notre chatbot



Figure 22: Exemple d'une réponse de chatbot

3.4- Most Popular

3.4.1 Définition :

Les normes des produits les plus populaires varient d'un site web à l'autre, ça dépend de la façon de penser d'administrateur. Pour nous les normes sont les suivantes : nombres d'aimes par les produits, moyenne de ratings par le produits/nombres de personnes qui donne ses ratings (pour éviter qu'on puisse trouver des produits avec des bonnes ratings mais juste peux de personne qui le note, ce n'est pas équivalent au produit qui ont les mêmes ratings mais plusieurs personnes le note)

3.4.2 Exemple d'utilisation

Afin de donner une idée à l'utilisateur sur les produits les plus populaire dans notre site web, Nous avons exploré une partie de notre site web pour une liste des produits les plus populaire par les normes qui sont déjà cité dans la partie précédente.

Most Popular

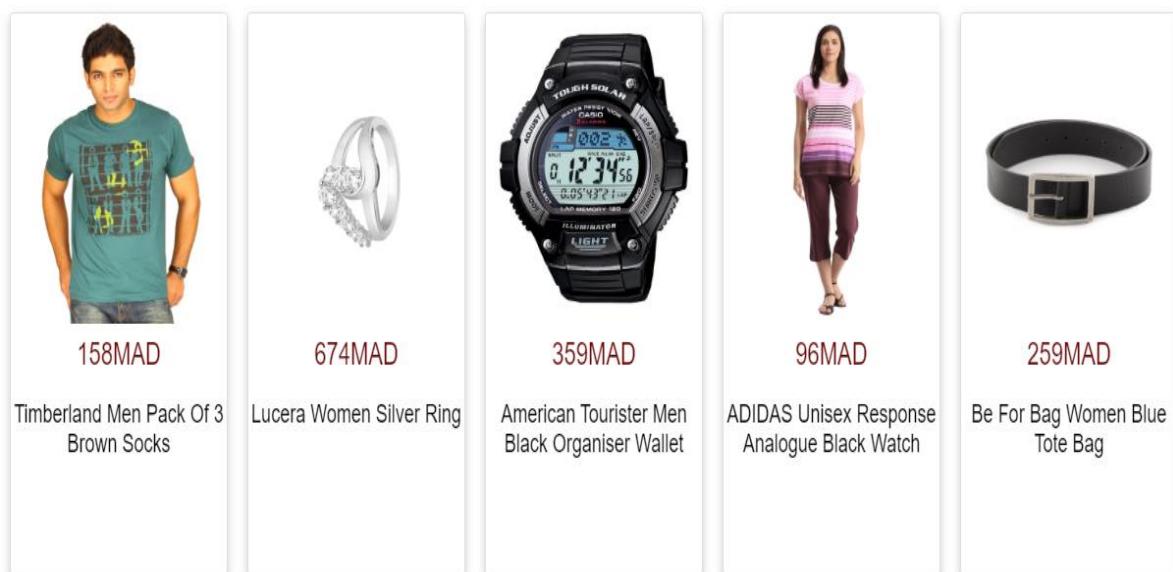


Figure 23: Interface Most Popular

3.5- CNN

3.5.1 Définition

Le réseau neuronal convolutif, également connu sous le nom de convnets ou CNN, est une méthode bien connue dans les applications de vision par ordinateur. Ce type d'architecture est dominant pour reconnaître les objets d'une image et la classifier. Un réseau neuronal convolutif n'est pas très difficile à comprendre. Une image d'entrée est traitée pendant la phase de convolution et ensuite attribuée une étiquette.

3.5.2 Architecture

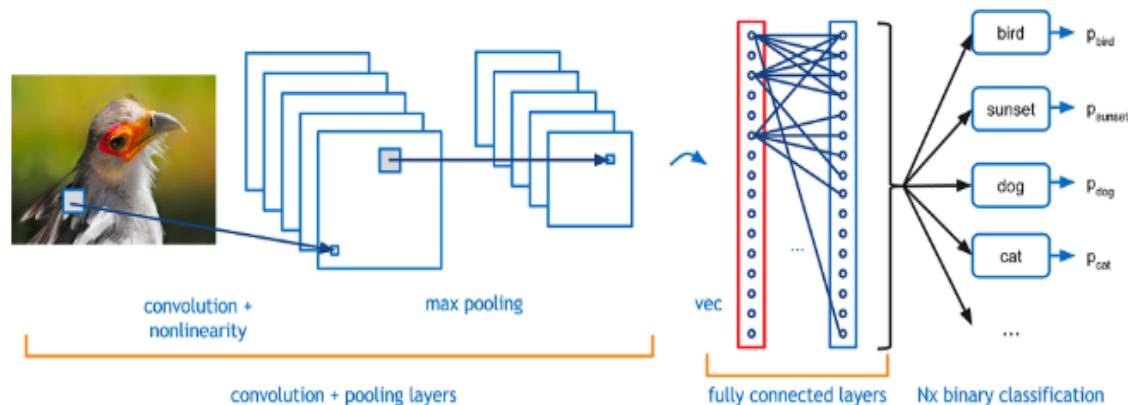
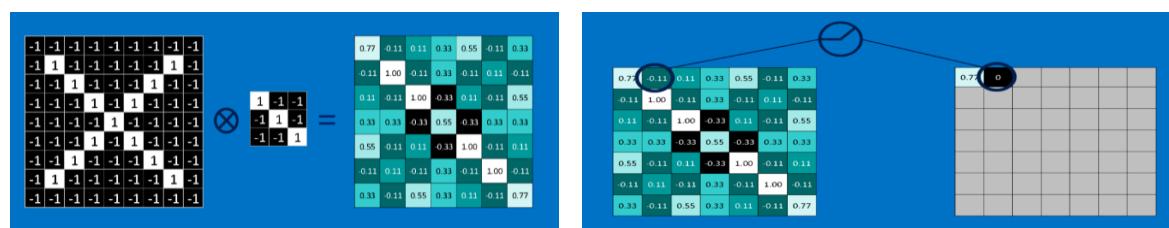


Figure 24: Architecture du CNN utilisé

3.5.3 Les étapes

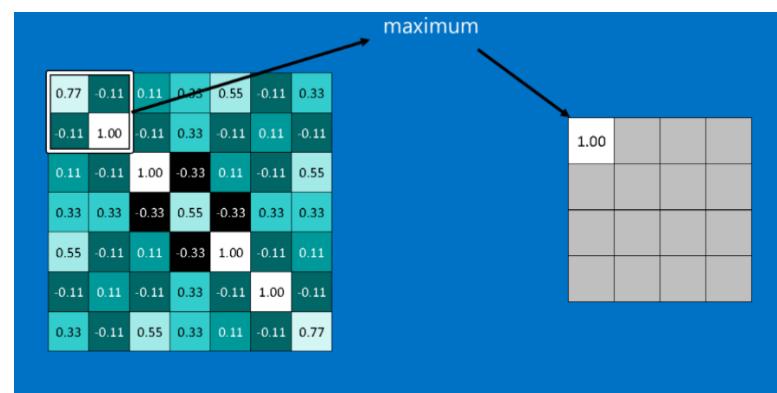
Le réseau de neurones à convolution compile différentes couches avant de faire une prédiction. Un réseau de neurones a :

- Importer l'ensemble des données : chaque sous-catégorie dans un dossier contient centaine des images.
- Une couche convolutionnelle et Fonction d'activation Relu : (conv2d) Construit une couche convolutionnelle bidimensionnelle avec le nombre de filtres, la taille du noyau de filtre, le remplissage et la fonction d'activation.
 - Une couche convolutionnelle Mathématiquement est la somme du produit élément par élément de deux matrices. Prenons deux matrices, X et Y. Si nous « convoluons l'image X à l'aide du filtre Y », cette opération produira la matrice Z. Elle applique différents filtres sur une sous-région de l'image pour extraire des caractéristiques de l'image d'entrée. La convolution aide au flou, à la netteté, à la détection des bords, à la réduction du bruit ou à d'autres opérations qui peuvent aider la machine à apprendre les caractéristiques spécifiques d'une image.
 - Fonction d'activation Relu Mathématiquement supprime les valeurs négatives d'une carte d'activation en les mettant à zéro. Elle augmenter de la non-linéarité dans le réseau. En effet, les images elles-mêmes sont très non linéaires.



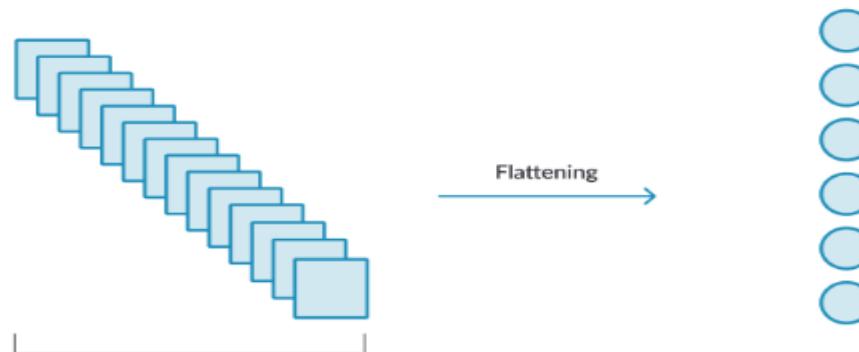
- Couche de mise en commun (MaxPooling2d) : Construit une couche de regroupement bidimensionnelle à l'aide de l'algorithme de regroupement maximal.

Couche de mise en commun Mathématiquement consiste à saisir la valeur maximale à chaque endroit de l'image. Cela supprime 75% des informations qui ne sont pas de fonctionnalité. En prenant la valeur maximale des pixels, nous tenons compte de la distorsion. Si l'entité tourne un peu vers la gauche ou la droite ou autre, l'entité regroupée sera la même. Elle réduit la dimensionnalité de l'image sans perdre des caractéristiques ou des motifs importants. Couche de mise en commun permet de détecter des objets dans une image et permet également de contrôler le surajustement (overfitting).



Remarques : Toutes ces couches extraient des informations essentielles des images.

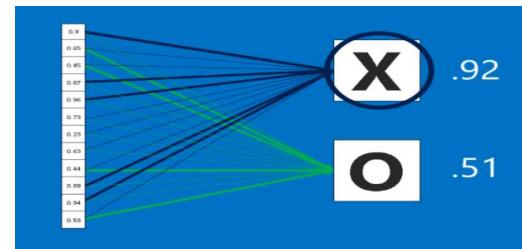
- Aplanissement (Flatten) : Transforme une matrice bidimensionnelle de caractéristiques en un vecteur qui peut être introduit dans un classificateur de réseau neuronal entièrement connecté.
- Il s'agit d'une étape assez simple. Cela permet à ces informations de devenir la couche d'entrée d'un réseau neuronal artificiel pour un traitement ultérieur.



- Couche fortement connectée (Dense) : Construit une couche dense avec les couches et unités cachées

La couche entièrement connectée est un Perceptron multicouche traditionnel permet au

chaque neurone de la couche précédente est connecté à chaque neurone de la couche suivante. La couche entièrement connectée est également connue sous le nom de couche dense, dans laquelle les résultats des couches convolutives sont alimentés à travers une ou plusieurs couches neuronales pour générer une prédiction. Elle utilise une fonction d'activation softmax, qui génère une valeur de probabilité de 0 à 1 pour chacune des étiquettes de classification que le modèle tente de prédire.



Après la création de model, on stock le résultat dans un fichier hdf5 et donc maintenant n'importe quelle image on peut la classifier dans quelle sous-catégorie appartient.

3.5.4 Exemple d'utilisation :

Nous avons utilisé CNN dans le moteur de recherche par image dans notre application

- On importe une image :



- Donc avec python on fait la prédiction de quelle sous-catégorie appartient cette image et il va envoyer le résultat par flask, et à partir de ce résultat en va le diriger vers la sous-catégorie chercher.

Product	Color	RAM	Storage	Price (MAD)
Honor 8X	Blue	4GB	64GB	28767
Honor 8C	Black	4GB	64GB	28115
Honor 8X	Blue	6GB	64GB	29130
Honor Play	Navy Blue	6GB	64GB	29021

Figure 25 : résultat de Moteur de recherche par image

4. Partie Frontend

4.1- Partie Création

4.1.1 Crédit des composants :

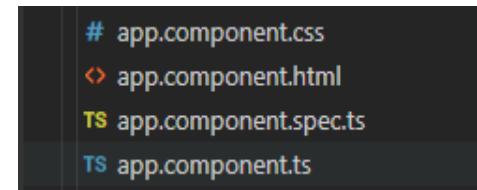
Les composants sont les composantes de base d'une application Angular : une application est une arborescence de plusieurs components.

Les Components reposent sur un certain nombre de règles :

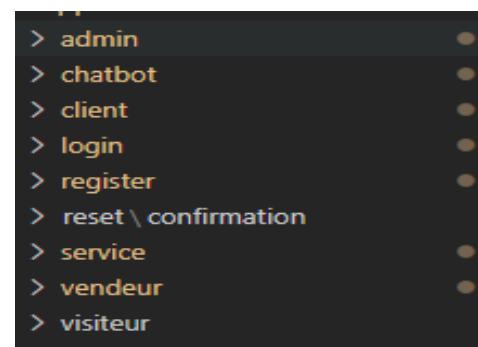
- Un component ne gère que sa vue et ses données. Il n'a pas à modifier les données ou le DOM qui est en dehors de son scope.
- Chaque component a son propre cycle de vie.
- Les components communiquent entre eux via les évènements, il n'y a pas d'interactions directes.

Tout d'abord, notre AppComponent est notre component principal : tous les autres components de notre application seront emboîtés celui-ci.

Ce component est inclus automatiquement avec la création d'un projet angular

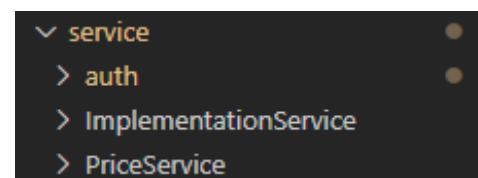


Dans notre projet, nous avons 8 components principaux et chacun construit avec d'autres components



4.1.2 Crédit des services :

Pour notre projet on a utilisé 3 services :



- **Service d'authentification et Guard :**

Contient un fichier pour s'authentifier (fonction de déconnecter, effacer un utilisateur d'un stockage local pour définir un autre utilisateur et une fonction pour obtenir le profil d'un utilisateur connecté) et un fichier qui manipule le system Guard pour sécuriser les routes avec la fonction canActivate.

- **Service de changer la devise :**

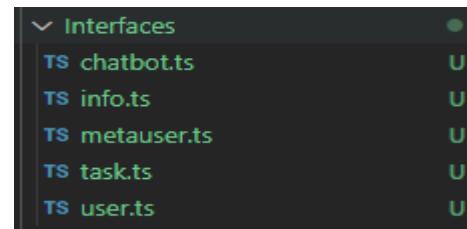
C'est un service qui manipule le changement de type de devise (MAD, Dollar ou Euro).

- **Service pour se connecter à l'API :**

C'est un service qui a toutes les routes que nous avons utilisé pour se connecter au Flask API.

4.1.3 Création des Interfaces :

C'est la structure des données manipulées. Ils sont utilisés pour réaliser l'abstraction et l'héritage multiple.



4.1.4 Création des routes :

Nous avons utilisé les routes pour naviguer d'une page à une autre ainsi l'utilisation des guards pour chaque route selon le besoin :

```
{
  path: 'api/:categorie/:subCategorie', canActivate: [AuthGuard], component: PortfolioImagesComponent },
  { path: 'api/:categorie', canActivate: [AuthGuard], component: PortfolioImagesComponent },
  { path: 'product/:id', canActivate: [AuthGuard], component: DetailProductComponent },
  { path: 'api/all/:categorie', canActivate: [AuthGuard], component: PortfolioImagesComponent },
  { path: 'api/all/:categorie/:subCategorie', canActivate: [AuthGuard], component: PortfolioImagesComponent },
  {path: '',redirectTo: "/app-login",pathMatch: 'full'},
```

4.2- Partie visiteur

4.1.1. Définition

Puisque le démarrage à froid est un problème très connu dans les systèmes de recommandation et plus précisément , il concerne le problème selon lequel le système ne peut tirer aucune conclusion pour les utilisateurs ou les éléments sur lesquels il n'a pas encore recueilli suffisamment d'informations , le visiteur est un cas de ce problème , de ce fait nous avons décidé de donner la possibilité au visiteur de voir la liste des produits mais il n'a pas la

possibilité de faire un rating ou bien une J'aime jusqu'à authentifier à fin de voir un compte dans la base donnée pour le modèle créer peut lui donner des recommandations sinon il peut voir juste les produits les plus populaire et les similaires de chaque catégories des produits .

4.2.2. Les cas d'utilisation

Comme nous avons dit avant que le visiteur s'authentifier il n'a pas la possibilité de faire un j'aime ou un rating et donc il n'a pas une liste d'historique et un panier et autres, voici le résultat quand il veut faire un rating ou bien like :

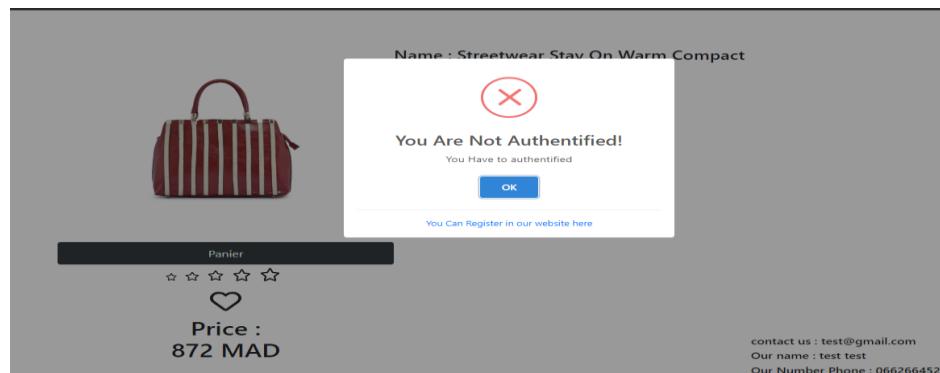


Figure 26: Résultat lorsqu'un visiteur veut faire rating ou j'aime

Juste après l'authentification il devient comme un client normal que on va voir toute à l'heure.

4.3- Partie Client

4.2.1 Définition

Un client est une personne qui utilise les services proposés par le site web d'une part et consommer les produits postés par le vendeur d'autre part, il a plusieurs possibilités qu'il puisse faire et que nous verrons plus tard.

4.2.2 Les cas d'utilisation

Notre but est de Fournir un service au client d'aller d'un point A à un point B en évitant le maximum d'obstacles. Et voici brièvement les possibilités qu'un client peut faire :

- Consulter la liste des produits de chaque catégorie

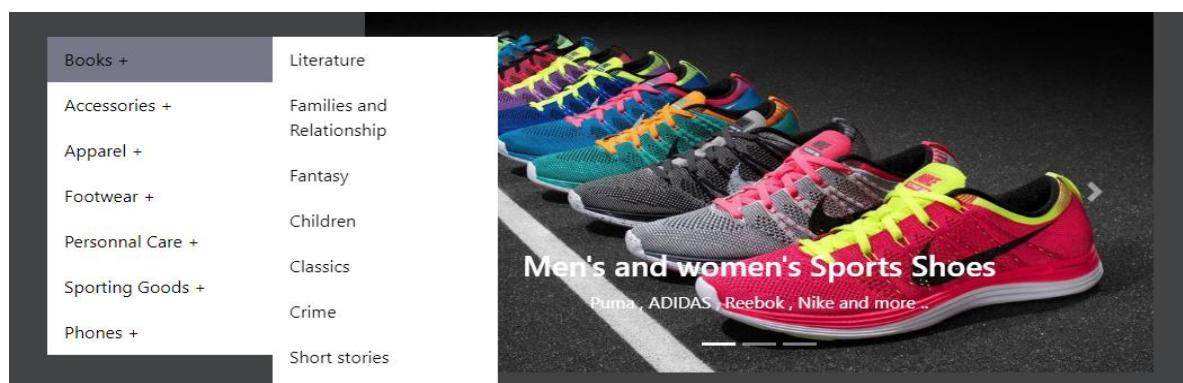


Figure 27: Interface pour consulter la liste des produits de chaque catégorie

b) Chercher des produits soit par texte ou par image

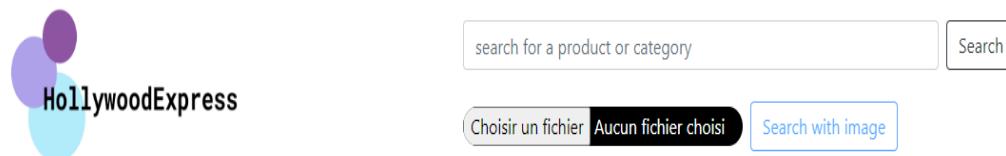


Figure 28: Interface pour Chercher des produits soit par texte ou par image

c) Gérer le profil.

Figure 29 : Interface pour gérer le profil

d) Voir les produits les plus populaire

Figure 30: Interface des produits les plus populaire

- e) Voir des produits similaires à celui qu'il est en train de voir

Similiar Products



Figure 31 : Interface des produits similaires à celui que le client en train de voir

- f) Changer le type de devise des produits

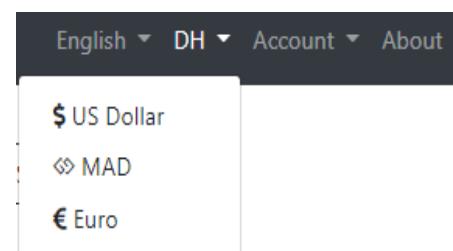


Figure 32 : Interface pour changer le type de devise des produits

- g) Ajouter un produit au panier et faire rating ou j'aime



Figure 33 : Interface d'ajouter un produit au panier, Faire rating ou Like

h) Modifier son compte

Figure 34: Modifier le profil

i) Utiliser le Chatbot

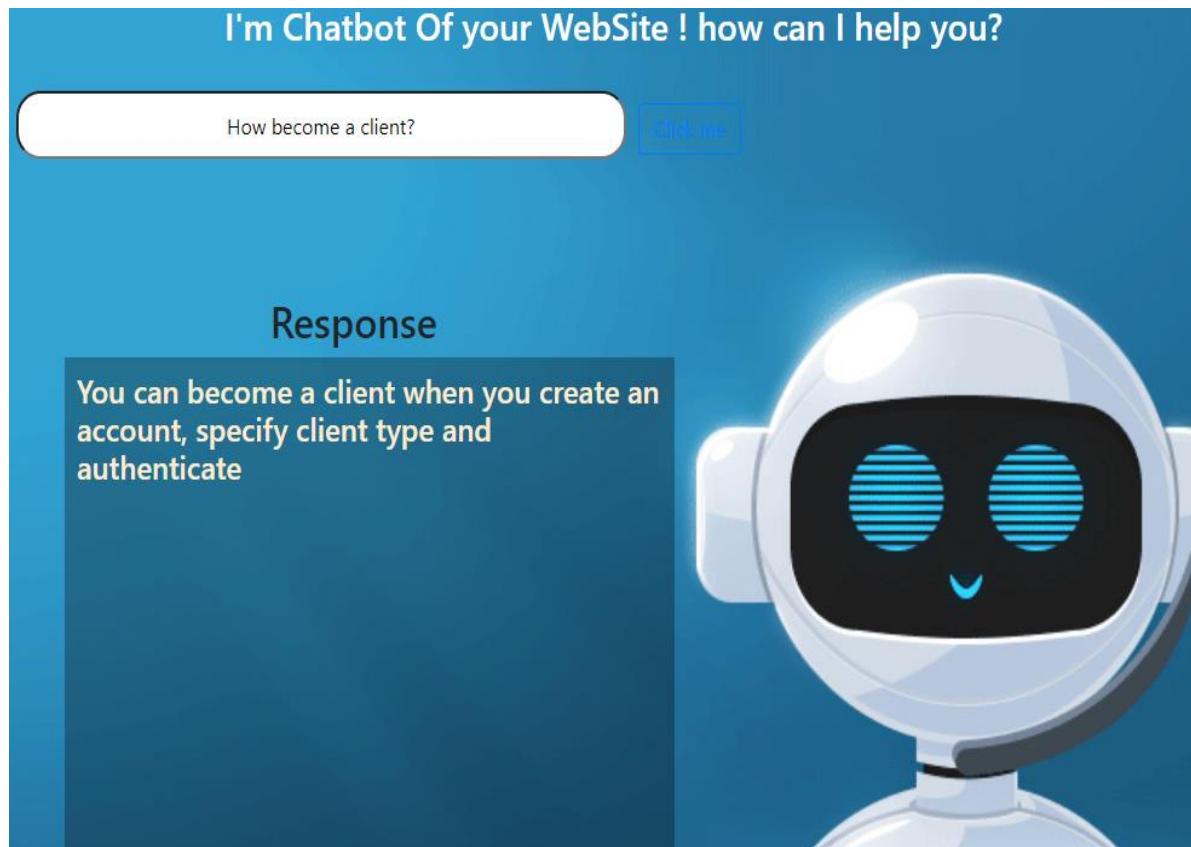


Figure 35 : Interface Chatbot

j) Le client avoir des recommandation (May Like)

You May Like

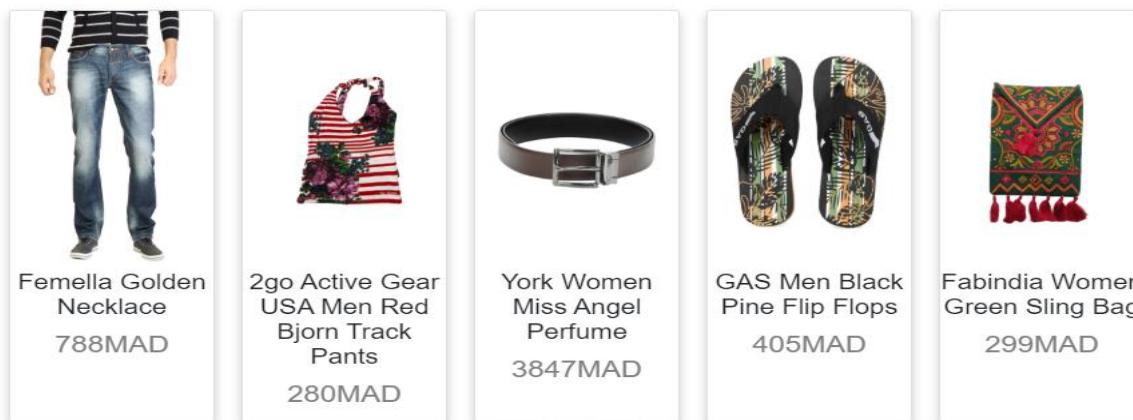


Figure 36 : Interface you may like de client

4.4- Partie Vendeur

4.3.1 Définition

La principale différence entre un vendeur et un client est qu'un vendeur fait de la vente par contre le client fait de l'achat et on va voir dans les cas d'utilisations les différentes possibilités qu'un vendeur peut faire.

4.3.2 Les cas d'utilisation

a) Voir sa propre liste de produits

All Products							
Name product	Category	subCategory	Description	Product price	gender	baseColor	Product_img
Parx Men Brown Belt	Accessories	belts	Belts	389MAD	Men	Green	http://assets.myntassets.com/v1/images/style/properties/1
Estelle Gold Earrings	Accessories	jewellery	Earrings	225MAD	Women	Gold	http://assets.myntassets.com/v1/images/style/properties/Watch_089a863fcc2aa3989c09a367fa96e6b9_images.jpg
Pepe Jeans Men Casual Sunglasses	Accessories	eyewear	Sunglasses	247MAD	Men	Black	http://assets.myntassets.com/v1/images/style/properties/Earrings_dc31d728c321bc83bb84c3c05ef535ed_images.jp
Lucera Women Silver Ring	Accessories	jewellery	Ring	523MAD	Women	Silver	http://assets.myntassets.com/v1/images/style/properties/
Lucera Women Silver Pendant with Chain	Accessories	jewellery	Pendant	279MAD	Women	Silver	http://assets.myntassets.com/assets/images/7217/2016/8/Soft-Black-Shoe-4671471349017623-1.jpg

Figure 37 : Interface vendeur pour afficher liste des produits

b) Ajouter un nouveau produit dans la boutique en ligne

Add Product

product	description	▼	product price
▼	gender	baseColor	<input type="button" value="Choisir un fichier"/> Aucun fichier choisi
SUBMIT			

Figure 38 : Interface vendeur pour ajouter un produit

c) Modifier en cliquant sur la ligne ou supprimer un produit

gender	baseColor	Product_img	Delete
Men	Green	http://assets.myntassets.com/v1/images/style/properties/920fa68e4629d0f4168e4589d394c2c3_images.jpg	<input type="button" value="Delete"/>
Women	Gold	http://assets.myntassets.com/v1/images/style/properties/Lucera-Silver-Earrings_dc31d728c321bc83bb84c3c05ef535ed_images.jpg	<input type="button" value="Delete"/>
Men	Black	http://assets.myntassets.com/v1/images/style/properties/Lucera-Silver-Earrings_dc31d728c321bc83bb84c3c05ef535ed_images.jpg	<input type="button" value="Delete"/>
Women	Silver	http://assets.myntassets.com/v1/images/style/properties/1dd6778eb68e49d4f7f046f9780cc586_images.jpg	<input type="button" value="Delete"/>
Women	Silver	http://assets.myntassets.com/assets/images/7217/2016/8/16/11471349017844-Rockport-Men-Carnforth-Soft-Black-Shoe-4671471349017623-1.jpg	<input type="button" value="Delete"/>
Women	Gold	http://assets.myntassets.com/v1/images/style/properties/Allen-Solly-Unisex-Unisex-Plastic-SG-White-Sunglasses_a80e0d810edb7d3dc6d9b04962ff7971_images.jpg	<input type="button" value="Delete"/>
Women	Gold	http://assets.myntassets.com/v1/images/style/properties/Estelle-Earrings_7271511bf974f34b0c0e80431617e034_images.jpg	<input type="button" value="Delete"/>

Figure 39 : Interface vendeur pour modifier ou supprimer un produit

d) Voir des statistiques sur leur produits, voici deux exemples

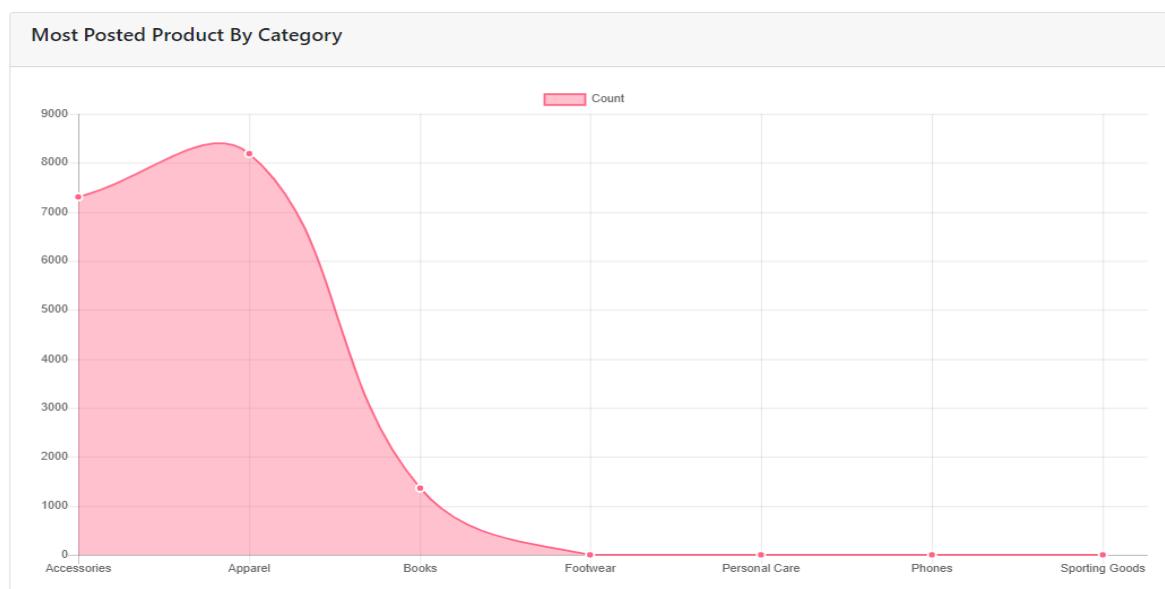


Figure 40: Exemple d'une statistique des produits les plus postées par catégorie

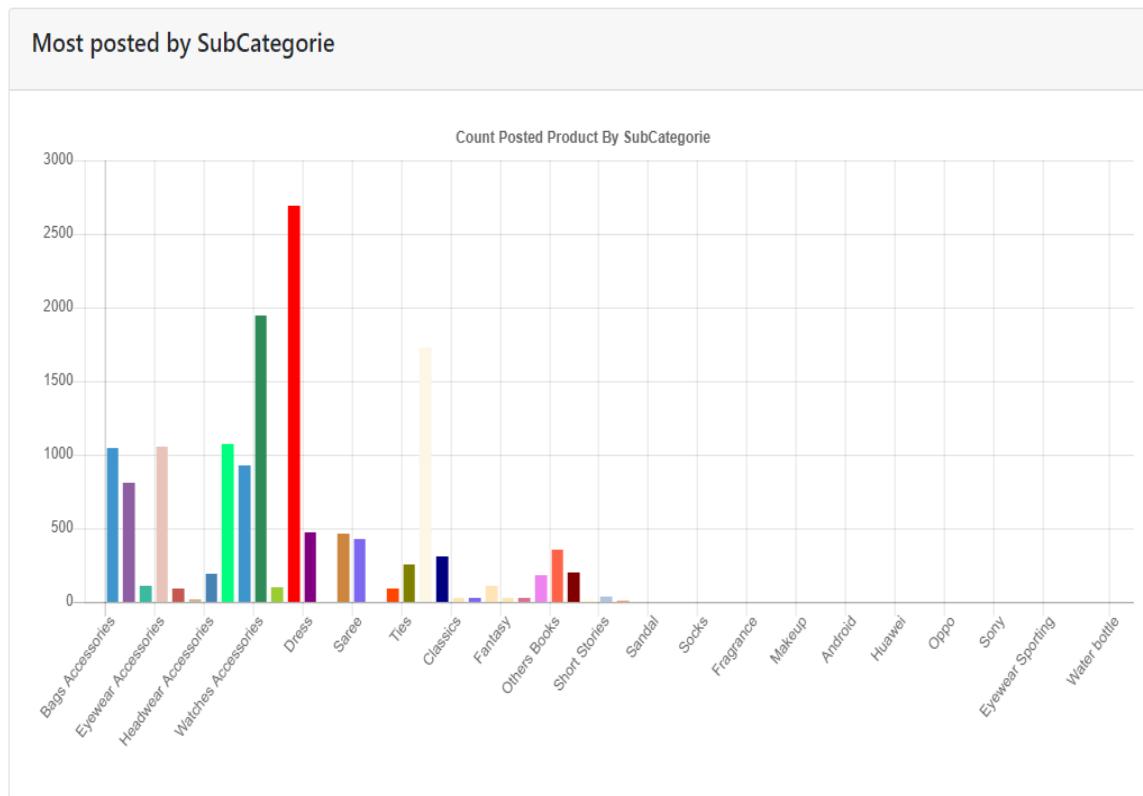


Figure 41: Exemple d'une statistique des produits les plus postées par sous-catégorie

- e) Voir des statistiques qui est en relation avec le client et avec les autres vendeurs afin d'améliorer sa boutique

popular Category

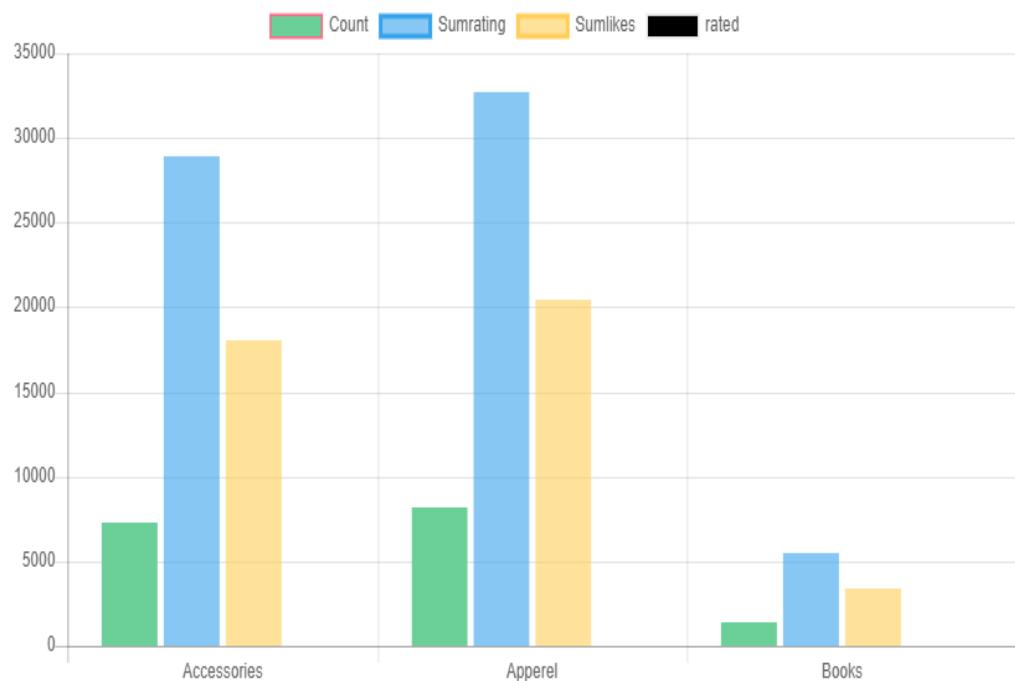


Figure 42: Exemple d'une statistique des produits les plus populaire

most liked category

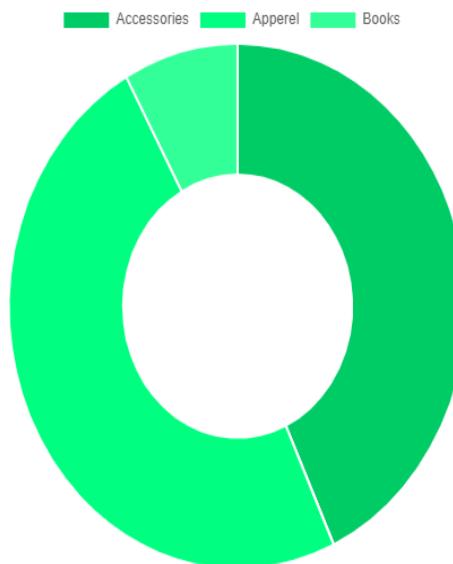


Figure 43 : Exemple d'une statistique des catégories les plus aimés

f) Produits les moins populaires

Least Popular Products

Name product	Count	Like	Mean	Delete
Mismatch	0	0	0	<button>Delete</button>
The Golden Valkyrie	0	0	0	<button>Delete</button>
WARREN BUFFETTS MANAGEMENT SECRETS	0	0	0	<button>Delete</button>
The Arrow	0	0	0	<button>Delete</button>
How To Woo A Reluctant Lady	0	0	0	<button>Delete</button>
Demon From The Dark	0	0	0	<button>Delete</button>
Blue Horizon	0	0	0	<button>Delete</button>
The Jacqueline Wilson Treasury	0	0	0	<button>Delete</button>
Damned	0	0	0	<button>Delete</button>
Love, Dishonor, Marry, Die, Cherish, Perish	0	0	0	<button>Delete</button>

Figure 44: Exemple d'une statistique des produits les moins populaires

4.5- Partie Admin

4.4.1 Définition

Un webmaster ou admin est une personne responsable d'un site web, de sa maintenance et de sa sécurité. Il peut Supprimer un vendeur ou un client s'il voit que c'est dangereux sur le site et voici les cas d'utilisations de l'admin.

4.4.2 Les cas d'utilisation

- a) Voir des statistiques sur les produits de site web, exemple

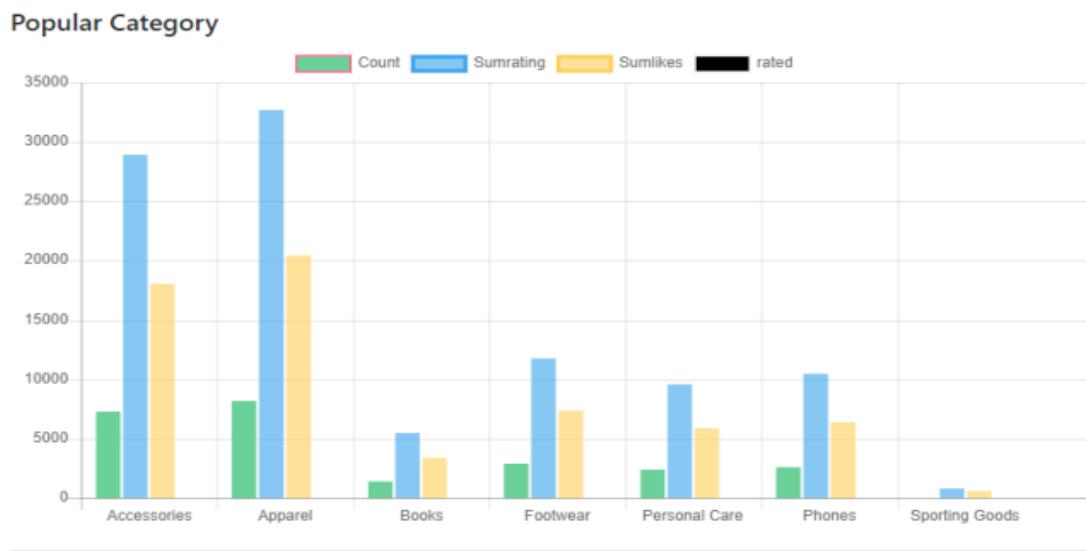


Figure 45 : Exemple d'interface admin pour voir des statistiques sur les catégories de site web (graphe1)

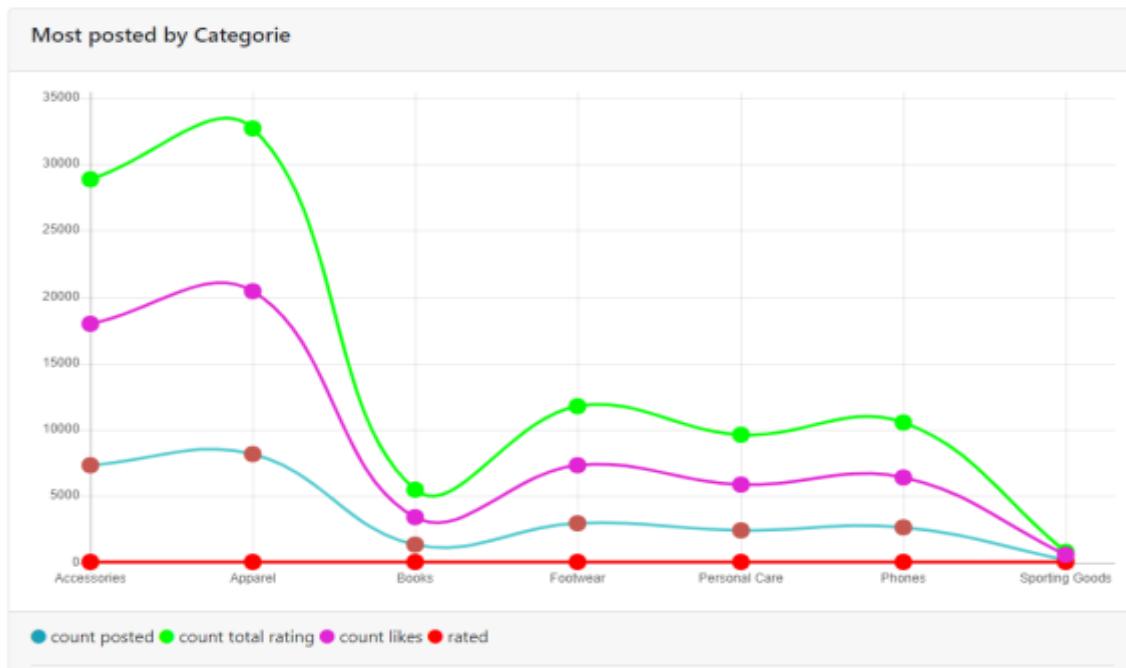


Figure 46: Exemple d'interface admin pour voir des statistiques sur les catégories de site web (graphe2)

b) Voir des statistiques qui est en relation avec les clients

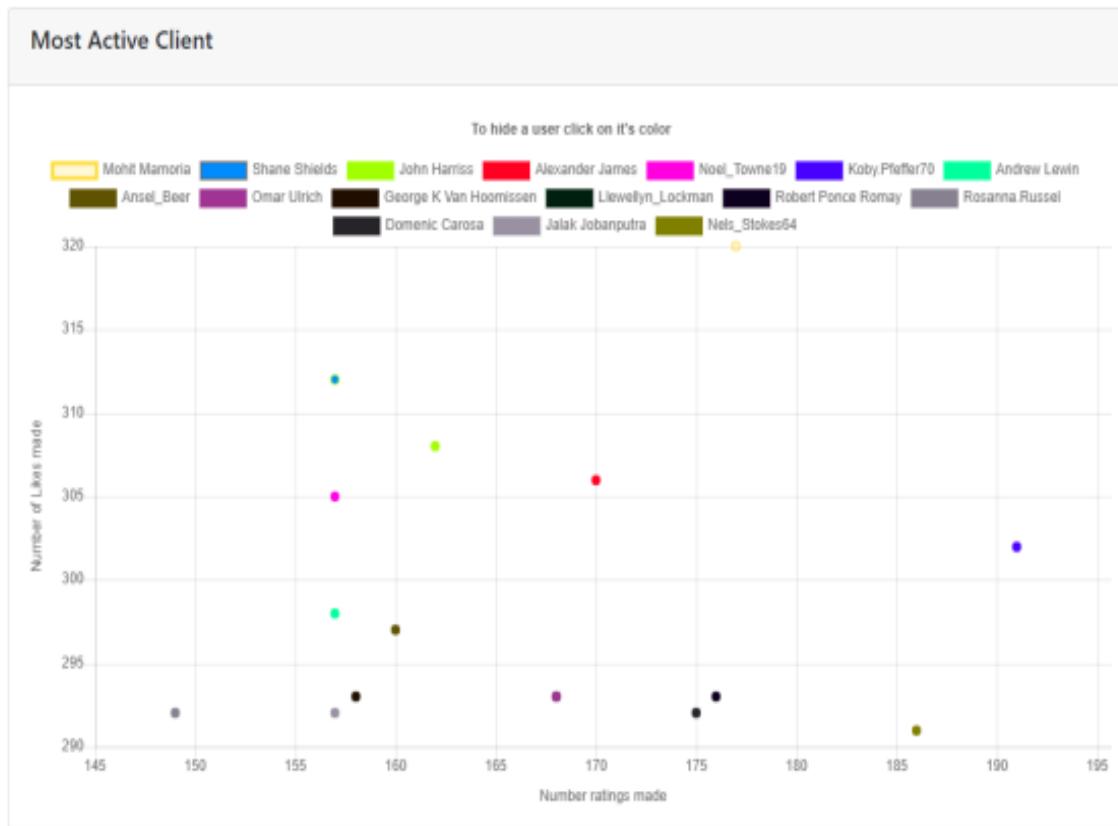


Figure 47 : Interface admin pour voir les utilisateurs les plus actif

Least Active Users

Username	Count	Like	Mean	Delete
Shamsher Khan Beng	234	173	3.688034188	<button>Delete</button>
Richard Galvin	234	159	3.6581196581	<button>Delete</button>
America.Fisher91	234	173	3.6153846154	<button>Delete</button>
Hans de Back	234	154	3.6153846154	<button>Delete</button>
Jeremy Boynton	234	166	3.547008547	<button>Delete</button>
Richard Muirhead	233	164	3.7424892704	<button>Delete</button>
Jayne_Dietrich	233	163	3.5965665236	<button>Delete</button>
Gust87	232	144	3.6551724138	<button>Delete</button>

Figure 48 : Interface admin pour voir les utilisateurs les moins actif

c) Gérer les utilisateurs du site web

All Users								
firstname	lastname	address	num_phone	email	city	country	Delete	
Adam	Draper	55 East 3rd Avenue	2128804543	info@boost.vc	San Mateo	Morocco	<button>Delete</button>	
Adam	Valkin	20 University Road, 4th Floor	2129886862	gcinfo@generalcatalyst.com	Cambridge	United States	<button>Delete</button>	
Adam	Grimsley	152 W. 57th Street, Floor 10	2124815581	info@primefactor.capital	London	United Kingdom	<button>Delete</button>	
Adi	Shelag	Berkovitz 4, 18 Floor	2124064125	info@singulariteam.com	Tel Aviv	Israel	<button>Delete</button>	
Adrian	Bialonczyk	1 Broadway, 14th Floor	2128173332	adrian@algodepth.com	Cambridge	United States	<button>Delete</button>	
Alan	Martinez	Lisboa 300 Torreon	2121077902	Adrian@gmail.com	Coahuila	Mexico	<button>Delete</button>	
Alex	Mittal	237 Kearney Street, #424	2120719675	press@fundersclub.com	San Francisco	United States	<button>Delete</button>	
Alex	Lazovsky	2916 Ramona Street	2129732068	info@scale-up.vc	Palo Alto	United States	<button>Delete</button>	
Alexander	James	65 Hopton Street, Flat 301 Hopton Street	2122776748	info@distributedalpha.com	London	United Kingdom	<button>Delete</button>	
Alexei	Chekhlov	152 W. 57th Street, Floor 10	2124318841	info@systematicalpha.com	New York	United States	<button>Delete</button>	

« Previous 1 2 3 4 5 ... 39 Next »

Figure 49 : Interface admin pour gérer les utilisateurs du site web

d) Voir les réactions des utilisateurs au produits

All Metausers				
username	Product name	rating	Like	
Nellie_Franecki	Gini and Jony Kids Girls Solid Red Skirts	-	1	
Oleg Ivanov	Lino Perros Women Grey Sling Bag	-	1	
Martin Lalonde	The Book of William: How Shakespeare's First Folio Conquered the World	-	1	
Boris Wertz	Belmonte Men Bright Assorted Steel Cufflinks	-	1	
Westley_Murray83	Nyk Women Grey Vespa Handbag	-	1	
Novella_Reynolds6	ADIDAS Men Black 3/4 Length Pants	-	1	
Gracie_Ratke	FCUK Black & White Boxers	-	1	
Blake Richardson	Streetwear Makeup Remover	-	1	
Brian Keane	Nokia 2.1 Blue-Silver	-	1	
P. Bart Stephens	Wyoming Bride	-	1	

« Previous 1 2 3 4 5 ... 16,173 Next »

Figure 50 : Interface admin pour voir les réactions des utilisateurs au produits

e) Voir les produits de la base de données

All Products							
Name product	Category	subCategory	Description	Product price	gender	baseColor	Product_img
Parx Men Brown Belt	Accessories	belts	Belts	389MAD	Men	Green	http://assets.myntassets.com/v1/images/style/properties/1/Belts_dc31d728c321bc83bb84c3c05ef535ed_images.jpg
Estelle Gold Earrings	Accessories	jewellery	Earrings	225MAD	Women	Gold	http://assets.myntassets.com/v1/images/style/properties/Watch_089a863fcc2aa3989c09a367fa96e6b9_images.jpg
Pepe Jeans Men Casual Sunglasses	Accessories	eyewear	Sunglasses	247MAD	Men	Black	http://assets.myntassets.com/v1/images/style/properties/Earrings_dc31d728c321bc83bb84c3c05ef535ed_images.jpg
Lucera Women Silver Ring	Accessories	jewellery	Ring	523MAD	Women	Silver	http://assets.myntassets.com/v1/images/style/properties/Soft-Black-Shoe-4671471349017623-1.jpg
Lucera Women Silver Pendant with Chain	Accessories	jewellery	Pendant	279MAD	Women	Silver	http://assets.myntassets.com/assets/images/7217/2016/8/Soft-Black-Shoe-4671471349017623-1.jpg
Ivory Tag Women Dazzling Agates Turquoise Blue and Yellow Necklace	Accessories	jewellery	Necklace and Chains	2729MAD	Women	Gold	http://assets.myntassets.com/v1/images/style/properties/Sunglasses_a80e0d810edb7d3dc6d9b04962ff7971_image.jpg

Figure 51: Interface admin pour voir les produits de la base de données

5. Sécurité de l'application

La sécurité, c'est l'ensemble des moyens (techniques, organisationnels, humains, légaux) pour minimiser une application ou un système contre les menaces.

La sécurisation d'une application ou d'un système s'attache à 3 aspects fondamentaux :

5.1- L'authentification

Nous avons fait un system d'authentification qui sépare chaque utilisateur à l'autre ensuite on applique le mécanisme de Route Guard qui accorde ou de supprime l'accès à certaines parties de la navigation.

Email address

Password

Remember me

Sign in

New around here? [Sign up](#)

Forgot password?

Figure 52 : Interface d'authentification

5.2- Le contrôle d'accès

En effet Une fois l'utilisateur authentifié, il souhaite accéder à des fonctionnalités offertes par l'application. Au préalable, On contrôle s'il a le droit d'y accéder

5.3- La confidentialité des données

Nous avons assuré La confidentialité des données pour assurer lors d'échange de données sensibles (mot de passe, les produits de chaque vendeur, le panier de chaque client etc.) Il s'agit de garantir que des données acquises illégalement soient inutilisables.

6. Liaison backend avec Frontend

Puisque le frontend fait référence au côté client ou à la conception Web et Le backend fait référence au côté serveur donc nous avons besoins d'un mécanisme qui va lier les deux concepts ou bien de conduire les résultats de backend au frontend, cette intermédiaire est Flask API.

6.1- Flask

Flask est l'un des Frameworks Python à la croissance la plus rapide, et des sites Web populaires, y compris Netflix, Pinterest et LinkedIn, ont intégré Flask dans leurs piles de développement.

Les API REST sont basées sur HTTP, qui signifie Hypertext Transfer Protocol. C'est ce qui est au cœur du web. C'est un protocole qui définit la communication entre les différentes parties du web. L'échange est basé sur des requêtes client et serveur. Un client lance une requête HTTP, et le serveur renvoie une réponse.

Les API REST basées sur HTTP sont définies par :

- Un URI de base, comme `http://api.example.com/collection/` ;
- Des méthodes HTTP standards (par ex. : GET, POST, PUT, PATCH et DELETE) ;

6.2- Pourquoi Flask ?

Flask présente plusieurs avantages majeurs :

- Python est populaire et largement utilisé, donc quiconque connaît Python peut développer pour Flask.
- C'est léger et minimaliste.
- Construit avec la sécurité à l'esprit.
- Excellente documentation avec beaucoup d'exemples de code clairs et fonctionnels.

6.3- Méthodes

Méthode	Rôle
GET	Cette méthode existe pour récupérer des données d'une ressource
POST	Utilisée pour envoyer des données dans une requête et souvent pour l'ajouter à la ressource précisée dans la partie URI
PUT	Cette méthode permet de remplacer ou d'ajouter une ressource sur le serveur. L'URI fourni est celui de la ressource en question.
DELETE	Utilisée pour supprimer la ressource donnée dans l'URI.

6.4- Architecture

Voici une architecture qui explique le déroulement des données depuis la base de données jusqu'à la partie client :

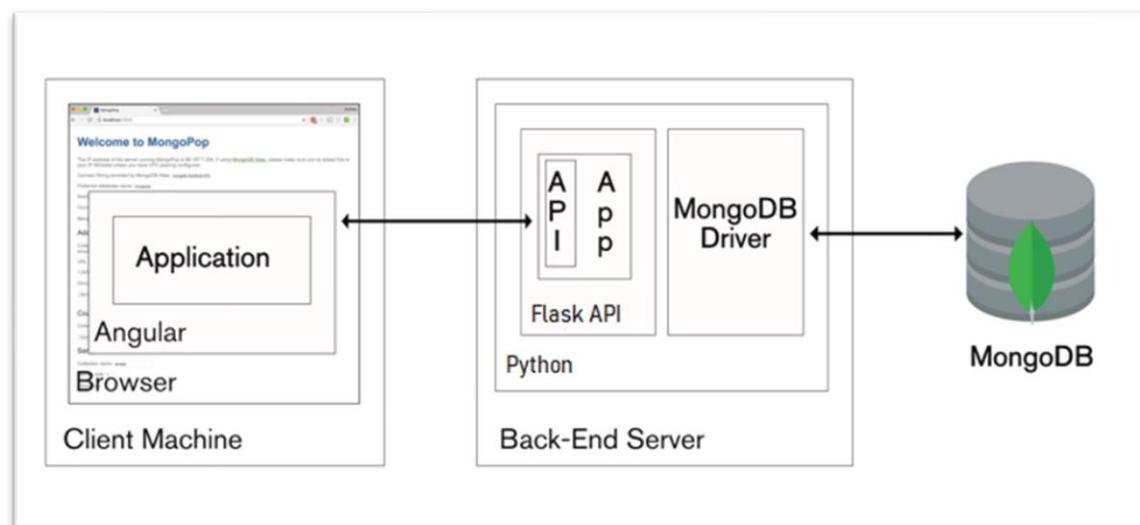


Figure 53: Architecture de déroulement des données depuis la base de données jusqu'à la partie client

6.5- Exemples

Voici un exemple de la façon dont Flask peut permettre aux utilisateurs de récupérer les produits à partir d'un serveur à l'aide de la méthode HTTP GET :

```
@app.route('/api/all', methods=['GET'])
def get_all():
    tasks = mongo.db.products
    result = []
    for field in tasks.find():
        result.append({'productId': str(field['productId']), 'Product_name': str(field['Product_name'])})
    return jsonify(result)
```

On lancer notre api pour être prêt à utiliser, pour le consommer et avoir des prédictions :

```
* Serving Flask app "Flask" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a
  production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Et voici le résultat :

```
[{"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/920fa8e4629d0f4168e4589d394c2c3_images.jpg", "Product_name": "Parx Men Brown Belt", "Product_price": "389", "baseColor": "Green", "description": "Belts", "gender": "Men", "productId": "0", "subCategory": "belts", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/titan-Men-White-Watch_089a863fc2aa989c09a367fa966b9_images.jpg", "Product_name": "Estelle Gold Earrings", "Product_price": "225", "baseColor": "Gold", "description": "Earrings", "gender": "Women", "productId": "1", "subCategory": "jewellery", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/Lucera-Silver-Earrings_dc31d728c321bc83bb84c3c05ef535ed_images.jpg", "Product_name": "Pepe Jeans Men Casual Sunglasses", "Product_price": "247", "baseColor": "Black", "description": "Sunglasses", "gender": "Men", "productId": "2", "subCategory": "eyewear", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/1dd6778eb68e49daff0f9780cc586_images.jpg", "Product_name": "Lucera Women Silver Ring", "Product_price": "523", "baseColor": "Silver", "description": "Ring", "gender": "Women", "productId": "3", "subCategory": "jewellery", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/assets/images/7217/2016/8/16/11471349017844-Rockport-Men-Carnforth-Soft-Black-Shoe-4671471349017623-1.jpg", "Product_name": "Lucera Women Silver Pendant with Chain", "Product_price": "279", "baseColor": "Silver", "description": "Pendant", "gender": "Women", "productId": "4", "subCategory": "jewellery", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/Alain-Solly-Unisex-Plastic-5G-White-Sunglasses_a80e0d810edbd7d3dc6d9b04962ff7971_images.jpg", "Product_name": "Ivory Tag Women Dazzling Agates Turquoise Blue and Yellow Necklace", "Product_price": "2729", "baseColor": "Gold", "description": "Necklace and Chains", "gender": "Women", "productId": "5", "subCategory": "jewellery", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/Estelle-Earrings_2715110ff974f4b0e080431617e034_images.jpg", "Product_name": "Est Women Gold Earrings", "Product_price": "359", "baseColor": "Gold", "description": "Earrings", "gender": "Women", "productId": "6", "subCategory": "jewellery", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/62704a4b179c73a36f77650682caa549_images.jpg", "Product_name": "Adrika Blue Necklace", "Product_price": "299", "baseColor": "Blue", "description": "Necklace and Chains", "gender": "Women", "productId": "7", "subCategory": "jewellery", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/Provogue-Men-Provogue-SHOE-TAN_BLACK-Tan-Casual-Shoes_cb9800aa5ba6aad04087550fb5ab9395_images.jpg", "Product_name": "United Colors of Benetton Women Solid Lavender Handbags", "Product_price": "559", "baseColor": "Lavender", "description": "Handbags", "gender": "Women", "productId": "8", "subCategory": "bags", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/46d1a2e4fcf37556207fa972a3ce4183_images.jpg", "Product_name": "Estelle Women Gold Bracelet", "Product_price": "804", "baseColor": "Gold", "description": "Bracelet", "gender": "Women", "productId": "9", "subCategory": "jewellery", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/fffdbdc3be29725a99855835d4638bdff1_images.jpg", "Product_name": "Miki Pearl Women Ivo Jewellery Set", "Product_price": "680", "baseColor": "White", "description": "Jewellery Set", "gender": "Women", "productId": "10", "subCategory": "jewellery", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/95b209f05bfbd7e41761e84e245305dd_images.jpg", "Product_name": "Proline Men Navy & W Patterned Beanie", "Product_price": "140", "baseColor": "Navy Blue", "description": "Caps", "gender": "Men", "productId": "11", "subCategory": "headwear", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/a7bf7bae55214a42c97bf22f13b6bd2_images.jpg", "Product_name": "Ivory Tag Women Clut Aqua Crema and Green Necklace", "Product_price": "296", "baseColor": "Gold", "description": "Necklace and Chains", "gender": "Women", "productId": "12", "subCategory": "jewellery", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/332c65ac6b1f7d32be80bb360af8d24d_images.jpg", "Product_name": "Pitaraa Silver Beaded Sheet Necklace", "Product_price": "679", "baseColor": "Silver", "description": "Necklace and Chains", "gender": "Women", "productId": "13", "subCategory": "jewellery", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/bf275da6995a42ddc1a2d7793bed69_images.jpg", "Product_name": "Baggit Women Orange Tower Sling Bag", "Product_price": "745", "baseColor": "Orange", "description": "Handbags", "gender": "Women", "productId": "14", "subCategory": "bags", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/4000cfb4266bbc8a4951e59818e95d1_images.jpg", "Product_name": "Lino Perros Women Leatherette Brown Handbag", "Product_price": "976", "baseColor": "Brown", "description": "Handbags", "gender": "Women", "productId": "15", "subCategory": "bags", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/1e035sea9cad8c0f6722dc05d46e1359_images.jpg", "Product_name": "Femelle Bronze Necklace", "Product_price": "162", "baseColor": "Bronze", "description": "Necklace and Chains", "gender": "Women", "productId": "16", "subCategory": "jewellery", "vendeurId": "381"}, {"Categories": "Accessories", "Product_img": "http://assets.myntassets.com/v1/images/style/properties/Alayna-Women-Black-Kurta_145b3d2b39cba6b21f3238a780c6688_images.jpg", "Product_name": "Lucera Women Silver"}]
```

Et voici comment on exploite cette api dans le côté Frontend :

```
getTasks(): Observable<Task[]> {
  return this.http.get<Task[]>('http://127.0.0.1:5000/api/all')
```

7. Conclusion

Nous avons exploré les différents aspects théoriques des éléments principaux de notre projet, à savoir le machine Learning, le Deep Learning, les Techniques de traitement du langage naturel.

Durant ce chapitre, Nous avons montré les étapes de notre travail passant par l'exploration et la préparation des données, l'apprentissage et l'évaluation des résultat

Conclusion et Perspectives

Le présent mémoire présente d'une manière explicite le déroulement de notre stage qui s'inscrit dans notre Projet Fin d'études du master en sciences et Techniques Mobilité et Big Data.

Ce stage, nous a été bénéfique sur plusieurs plans :

- Sur le plan Techniques, nous avons pu consolider notre formation notamment côté Data Science. Nous avons pu nous développer dans le Machine Learning et Deep Learning.
- Sur le plan professionnel, nous avons appris à travailler en équipe avec des contraintes et des consignes à respecter notamment le travail agile.
- Sur un plan personnel, ce stage nous a fait identifier certaines caractéristiques du milieu dans lequel nous allons faire carrière.

Nous tenons à préciser que le stage se poursuit au-delà de notre soutenance car nous souhaitons arriver au bout des objectifs précités.

Comme perspectives de travaux futurs, nous aspirons :

- Construire un système de recommandation se base aussi sur les commentaires et autres informations de clients.
- Initialiser le travail sur la construction d'une application mobile de système de recommandation purement Deep Learning

Annexes

1. Les Systèmes de recommandations

Un système de recommandation est une sorte de forme spécifique de filtrage de l'information visant à présenter les éléments susceptibles d'intéresser l'utilisateur.

2. La mise en place d'un Système de recommandation

- **Collecter les données** : ce sont des données sur l'utilisateur et ses réactions qui se présente généralement sous forme d'une note sur 5 (rating), ou bien des événements binaires tels que les "like" et "follow" et parfois on s'intéresse aussi sur les commentaires.
- **Création du modèle** : Créer un modèle Deep Learning qui va implémenter ces données collectées via de algorithmes spécifiés
- **Extraire des recommandations** : L'extraction se fait à partir des modèles et en appliquant quelques traitements pour trouver les recommandations pour un utilisateur, les utilisateurs ou les items les plus similaires.

3. Types de systèmes de recommandations

Selon les besoins et selon les données, on utilise souvent l'un de ces approches :

- **La recommandation non personnalisée** : D'habitude, nous utilisons ce type de recommandation lorsque nous n'avons pas assez d'informations sur un utilisateur (lors de son inscription).

Une telle recommandation est indépendante de l'utilisateur et se présente souvent sous forme d'une de ces affirmations :

- Le bar le plus aimé.
- Le restaurant le plus visité.
- La liste la plus suivie.

- **Le Collaboratif Filtering** : Cette recommandation se base sur les événements qui lient les utilisateurs aux items et qui présente d'une manière implicite ou explicite son goût. Ces recommandations sont souvent présentées sous ces affirmations :

- Les utilisateurs qui sont similaires à vous, ont aimé aussi.
- Les utilisateurs qui ont aimé ça, ont aimé ça aussi.

- **Le Content Based** : Il s'agit de recommander les items en se basant sur les caractéristiques et les propriétés qui identifient cet item d'une part, et sur les caractéristiques qui intéressent le plus notre utilisateur.

➤ **La recommandation hybride :** Cette recommandation utilise les Trois derniers types de recommandation.

4. Le choix d'algorithme

Dans notre projet collaboratif Filtering est le meilleur type qui répond à nous besoin, il se compose de deux types : Memory-based (Basé sur la mémoire) et Model-based (Basé sur un modèle).

- Les systèmes de recommandation basés sur la mémoire ne sont pas utilisables dans les grandes bases de données, Car Ils utilisent la base de données entière chaque fois qu'il fait une prédiction, il doit donc être en mémoire, c'est très, très lent.
- Les systèmes basés sur le modèle sont également susceptibles d'être plus rapides par rapport aux systèmes basés sur la mémoire, et les algorithmes basés sur un modèle sont beaucoup plus petits que l'ensemble de données réel, de sorte que même pour de très grands ensembles de données, le modèle finit par être suffisamment petit pour être utilisé efficacement.

Donc nous avons choisi dans notre projet d'utiliser les systèmes basés sur le modèle.

Les systèmes de recommandation basés sur modèle contient 3 algorithmes les plus connues : SVD, SGD, ALS.

Pour choisir le meilleur modèle pour travailler avec, nous avons faire un test sur la mesure de performance avec RMSE.

4.1- SVD : Singular Value Decomposition

Cette technique, comme toutes les autres, consiste à réduire la dimensionnalité de la matrice User-Item calculée précédemment. Posons R la matrice User-Item de taille m x n (m : nombre de utilisateurs, n : nombre d'items) et k: la dimension de l'espace des caractères latents. L'équation générale de SVD est donnée par : $R=USV^T$ avec :

La matrice U des caractères latents pour les utilisateurs : de taille m*k

La matrice V des caractères latents pour les items : de taille n*k

La matrice diagonale de taille k x k avec des valeurs réelles non-négatives sur la diagonale

On peut faire la prédiction en appliquant la multiplication des 3 matrices.

Mesure de performance :

Nous avons calculé la performance avec RMSE entre la matrice prédictive et la matrice du test

```
# Calcul de performance avec RMSE entre la matrice estimée et la matrice du test
print ('RMSE: ' + str(rmse(x, test_data_matrix)))
```

RMSE: 1.4699791103650217

4.2- SGD : Stochastic Gradient Descent

Quand on utilise le filtrage collaboratif pour SGD, on veut estimer 2 matrices P et Q :

La matrice P des caractères latents pour les utilisateurs : de taille $m*k$ (m : nombre d'utilisateurs, k : dimension de l'espace des caractères cachés)

La matrice Q des caractères latents pour les items : de taille $n*k$ (n : nombre d'items, k : dimension de l'espace des caractères cachés)

Après l'estimation de P et Q, on peut alors prédire les ratings inconnus en multipliant les matrices P et la transposée de Q.

Mesure de performance :

Nous avons calculé la performance avec RMSE

```
print ('RMSE : ' + str(np.mean(test_errors)))
```

```
RMSE : 1.4815661569016596
```

4.3- ALS : Alternating Least Squares

Dans cette partie, notre objectif avec ALS est d'estimer les matrices P et Q, tout comme SGD décrit dans la section précédente.

Après l'estimation de P et Q, on peut alors prédire les ratings inconnus en multipliant la transposée de la matrice P par la matrice Q.

Comme les anciennes méthodes Model Based traitée, nous avons calculé la RMSE pour le modèle ALS.

Comme pour SVD, ALS admet un nombre d'étapes défini à travers lequel la RMSE diminue jusqu'à converger.

Dans le cas de notre modèle, bien que l'exécution de ALS prenne plus de temps que SGD, nous nous sommes limités à 2 étapes de l'algorithme.

Mesure de performance :

Nous avons calculé la performance avec RMSE

```
[Epoch 1/2] train error: 1.046265, test error: 1.373295
[Epoch 2/2] train error: 0.765880, test error: 1.243846
```

Remarque

Le meilleur modèle est celui qui a la plus petite valeur pour RMSE

Nous remarquons que cette valeur de l'erreur pour ALS est la meilleure parmi toutes les autres méthodes testées. Nous pouvons conclure que le meilleur modèle Collaborative Filtering est ALS.

5. Partie de code

5.1- ALS

```
# Repeat until convergence
for epoch in range(n_epochs):
    # Fix P and estimate U
    for i, Ii in enumerate(I):
        nui = np.count_nonzero(Ii) # Number of items user i has rated

        # Least squares solution
        Ai = np.dot(P, np.dot(np.diag(Ii), P.T)) + lmbda * nui * E
        Vi = np.dot(P, np.dot(np.diag(Ii), data_matrix[i].T))
        U[:,i] = np.linalg.solve(Ai,Vi)

    # Fix U and estimate P
    for j, Ij in enumerate(I.T):
        nmj = np.count_nonzero(Ij) # Number of users that rated item j

        # Least squares solution
        Aj = np.dot(U, np.dot(np.diag(Ij), U.T)) + lmbda * nmj * E
        Vj = np.dot(U, np.dot(np.diag(Ij), data_matrix[:,j]))
        P[:,j] = np.linalg.solve(Aj,Vj)

    train_rmse = rmse2(I,data_matrix,P,U)
    train_errors.append(train_rmse)
def prediction(U,P):
    return np.dot(U.T,P)
model_matrix=prediction(U,P)
```

Figure 54 : Code ALS

5.2- Chatbot

```
model = Sequential() # model de keras faire organiser les couches
model.add(Dense(128, input_shape=(len(train_x[0]),), activation='relu')) # relu pour rendre l'
model.add(Dropout(0.5)) # une couche d'abandon ignore un ensemble de neurones (au hasard) , c
model.add(Dense(64, activation='relu')) # Une couche dense est une couche de réseau neuronal
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax')) # softmax pour faire prédiction de 0

# Compiler model. Stochastic gradient descent avec Nesterov accéléré le gradient et donne de
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True) # utilisée pour l'optimisation d'
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy']) # S'il s'
#training(en donne le input) et sauvegarder le model
hist = model.fit(np.array(train_x), np.array(train_y), epochs=2150, batch_size=30, verbose=1)
model.save('C:/Users/errza/Tests/PFEchatbot_model.h5', hist)
```

Figure 55 : Code Chatbot

5.3- CNN

```
model = Sequential()

model.add(Conv2D(filters=4, kernel_size=2, padding='same',
                 activation='relu', input_shape=(100, 100, 3)))
model.add(MaxPooling2D(pool_size=2))
model.add(Dropout(0.1))

model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.4))

model.add(Dense(7, activation='softmax'))

model.summary()

model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accuracy'])

epochs = 50

checkpointer = ModelCheckpoint(filepath='C:/Users/errza/Tests/dataset/weights.best.image_classifier.hdf5',
                               verbose=1, save_best_only=True)

model.fit(train_tensors, train_targets, validation_data=(valid_tensors, valid_targets),
          epochs=epochs, batch_size=64, callbacks=[checkpointer], verbose=1)
```

Figure 56 : Code CNN

5.4- La métrique cosine_similarity

```

for index, row in df.iterrows():
    words = ''
    for col in columns:
        words += ' '.join(row[col]) + ' '
    df['Bag_of_words'][index] = words

df1 = df[['Product_name', 'Bag_of_words']]

df1['Bag_of_words'] = df1['Bag_of_words'].astype(str)

count = CountVectorizer()
count_matrix = count.fit_transform(df1['Bag_of_words'])
cosine_sim = cosine_similarity(count_matrix, count_matrix)

Dataframe1 = pd.DataFrame(data=cosine_sim[1:,1:])

for i in range(len(products_df)-1):
    for j in range(len(products_df)-1):
        if int(cosine_sim2[i][j])==1 and int(cosine_sim5[i][j])==1:
            cosine_sim[i][j] = ((cosine_sim1[i][j] * 4) + cosine_sim3[i][j] + cosine_sim4[i][j])/6

Dataframe = pd.DataFrame(data=cosine_sim[:, :])

```

Figure 57 : Code cosine_similarity

5.5- Most Popular

```

#merge two dataset create df1
df1 = pd.merge(dfrating, count, how='right', on=['produitId'])
#merge two dataset create df7
df7 = pd.merge(df1, mean, how='right', on=['produitId'])
#merge two dataset create df6
df6 = pd.merge(df7, countlikes, how='right', on=['produitId'])

#sorted products
df2 = df6.sort_values(by=['Mean', 'like', 'Count'], ascending=False)
#20 Most Popular Products
df8 = df9.head(20)
#20 least Popular Products
df10 = df9.tail(20)

```

Figure 58 : Code Most popular

5.6- Most Active

```

#compute the count and mean value as group by the users
count = dfrating.groupby("userId", as_index=False).count()
mean = dfrating.groupby("userId", as_index=False).mean()
countlikes = dflike.groupby("userId", as_index=False).count()

#sorted users
df2 = df6.sort_values(by=['Count', 'like', 'Mean'], ascending=False)
#20 Most Active Users
df8 = df9.head(20)
#20 least Active Users
df10 = df9.tail(20)

```

Figure 59 : Code Most active

5.7- Partie Flask

```

@app.route('/api/tasks/<id>', methods=['GET'])
def get_all_tasksuser(id):
    id = int(id)
    Mostpopularuser = Popular.loc[Popular['vendeurId'] == id]
    Mostpopularuser = Mostpopularuser.head(10)
    rv = Mostpopularuser.to_json(orient='records')
    return rv

@app.route('/api/leasttasks', methods=['GET'])
def get_Least_popular_products():
    rv = Leastpopular.to_json(orient='records')
    return rv

@app.route('/api/leasttasks/<id>', methods=['GET'])
def get_Least_popular_productsuser(id):
    id = int(id)
    Leastpopularuser = Popular.loc[Popular['vendeurId'] == id]
    Leastpopularuser = Leastpopularuser.tail(10)
    rv = Leastpopularuser.to_json(orient='records')
    return rv

@app.route('/api/activeusers', methods=['GET'])
def get_active_users():
    rv = MostActive.to_json(orient='records')
    return rv

@app.route('/api/leastactiveusers', methods=['GET'])
def get_least_active_users():
    rv = LeastActive.to_json(orient='records')
    return rv

```

Figure 60 : Code partie flask

5.8- Liaison Python MongoDB

```

from flask_pymongo import PyMongo
from flask_cors import CORS

app.config['MONGO_DBNAME'] = 'PFE'
app.config['MONGO_URI'] = 'mongodb://localhost:27017/PFE'

mongo = PyMongo(app)
CORS(app)

tasks = mongo.db.products

```

Figure 61 : Code liaison Python MongoDB

Références

- Kevin Liao, (17 Nov 2018). Prototyping a Recommender System Step by Step Part 2: Alternating Least Square (ALS) Matrix Factorization in Collaborative Filtering. Available: <https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-2-alternating-least-square-als-matrix-4a76c58714a1>
- Christian S. Perone, (Sept 2013). Learning: Cosine Similarity for Vector Space Models. Available: <http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii>
- Jaime Zornoza, (18 juil 2019). Deep Learning for NLP: Creating a Chatbot with Keras. Available: <https://towardsdatascience.com/deep-learning-for-nlp-creating-a-chatbot-with-keras-da5ca051e051>
- Anne Bonner, (2 Fév 2019). The Complete Beginner’s Guide to Deep Learning: Convolutional Neural Networks and Image Classification. Available: <https://towardsdatascience.com/wtf-is-image-classification-8e78a8235acb>
- Vimal, (Jan 2017). Amazon Reviews: Unlocked Mobile Phones. Available: <https://www.kaggle.com/PromptCloudHQ/amazon-reviews-unlocked-mobile-phones>.
- Nicholle nsdatafiniti, (2017). reviews, products, amazon, kindle, consumer, sentiment. Available: <https://data.world/datafiniti/consumer-reviews-of-amazon-products>
- Francis B, (28 Nov 2019). Sécurité de l’application. Available: <https://openclassrooms.com/fr/courses/1761931-securisez-vos-applications/5702500-identifiez-les-6-aspects-de-la-securite-d-une-application>
- Matt Makai, (2019). Flask. Available: <https://www.fullstackpython.com/flask.html>
- Elsa Negre, (20 Sept 2018). Les systèmes de recommandation. Available: <https://interstices.info/les-systemes-de-recommandation-categorisation>
- Selva Prabhakaran, (2019). Cosine Similarity – Understanding the math and how it works. Available: <https://www.machinelearningplus.com/nlp/cosine-similarity>
- Yunhong Zhou, Dennis Wilkinson, Robert Schreiber et Rong Pan, (2018). Large-scale Parallel Collaborative Filtering for the Netflix Prize. Available: <https://endymecy.gitbooks.io/spark-ml-source-analysis/content/%E6%8E%A8%E8%8D%90/papers/Large-scale%20Parallel%20Collaborative%20Filtering%20the%20Netflix%20Prize.pdf>
- Abdelghani Azri, (Déc 2019). Matrix factorization techniques : SGD vs ALS. Available: https://www.researchgate.net/publication/338170657_Matrix_factorization_techniques_SGD_vs_ALS