

Université Mohammed V de Rabat  
Département Informatique

---

# Système De Recommendation Fashion Sous Django

---

Realisé par:

Noureddine Ech-chouky  
Med Amine Bouzmoul  
Abdelkabir Aarab

Supervisé Par:

Pr. El Qadi Mohamed

Filière :

Ingénierie Numériques En Intelligence Artificielle Et Data Science

29 Décembre, 2021

## Déclaration

I, Nouredine Ech-chouky, Med Amine Bouzmoul, Abdelkabar Aarab du Département d'informatique de École D'arts Et Métiers Rabat, confirmer qu'il s'agit de notre propre travail et que les figures, tableaux, équations, extraits de code, œuvres d'art et illustrations du présent rapport sont originaux et n'ont pas été tirés de l'œuvre d'une autre personne, sauf lorsque les œuvres d'autrui ont été explicitement reconnues, citées et référencées. Je comprends que si on ne le fait pas, cela sera considéré comme un cas de plagiat. Le plagiat est une forme d'inconduite académique et sera pénalisé en conséquence.

Nous consentons à ce qu'une copie de mon rapport soit communiquée aux futurs étudiants à titre d'exemple.

Nous consentons à ce que mon travail soit mis à la disposition du public qui s'intéressent à l'enseignement, à l'apprentissage et à la recherche.

Nouredine Ech-chouky, Med Amine Bouzmoul, Abdelkabar Aarab  
December 30, 2021

## abstrait

Les systèmes de recommandation font partie des concepts les plus utilisés, en termes d'application de *Machine Learning* et *Deep Learning*.

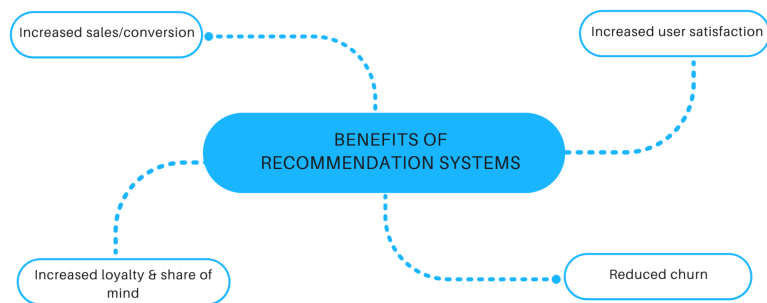


Figure 1: Les Bénéfices Des Systèmes De Recommandation

Dans ce rapport de projet, on a décrit la procédure de développement d'un *fashion recommendation system* en utilisant *Django* et *nlp : machine learning* seulement.

# Contents

1	Introduction	1
1.1	Contexte . . . . .	1
1.2	Déclaration de problème . . . . .	1
1.3	Les Objectives . . . . .	2
1.4	Approche sur la solution . . . . .	2
2	Les Types Des Systemes De Recommendation	3
2.1	Système de recommandation collaboratif : . . . . .	3
2.2	Système de recommandation basé sur le contenu : . . . . .	4
2.3	Système de recommandation basé sur la démographie : . . . . .	5
2.4	Système de recommandation basé sur les utilités : . . . . .	5
2.5	Système de recommandation basé sur les connaissances : . . . . .	6
3	Data Preprocessing	8
3.1	Réduction De Dimension : . . . . .	8
3.2	Suppression des valeurs nulles : . . . . .	9
3.3	Retirer les produits avec une brève description : . . . . .	9
3.4	Suppression des produits avec double titre par tri : . . . . .	10
3.5	Suppression des produits avec double titre: . . . . .	11
4	Les Modèles Machine Learning	13
4.1	<b>CountVectorizer :</b> . . . . .	13
4.2	<b>Bag-of-Words :</b> . . . . .	14
4.3	<b>Cosine_Similarity :</b> . . . . .	14
4.4	<b>pairwise_distances :</b> . . . . .	15
5	Structure Du Site Web	16
5.1	Templates : . . . . .	16
5.2	Models : . . . . .	17
5.3	Les fonctionnalités : . . . . .	18
5.4	User/Guest Checkout : . . . . .	20
6	Site Web Et Processus De Recommendation	21
7	Conclusion	24

# List of Figures

1	Les Bénéfices Des Systèmes De Recommandation . . . . .	ii
2.1	Système de recommandation collaboratif . . . . .	4
2.2	Système de recommandation basé sur le contenu . . . . .	4
2.3	Système de recommandation basé sur la démographie . . . . .	5
2.4	Système de recommandation basé sur les utilités . . . . .	6
2.5	Système de recommandation basé sur les connaissances . . . . .	7
4.1	Forme de la matrice des caractéristiques : Nombre de documents x Longueur du vocabulaire . . . . .	14
4.2	cosine similarity en 2-D . . . . .	14
5.1	Templates . . . . .	16
5.2	Models . . . . .	17
5.3	Home Page . . . . .	18
5.4	Payment Integration . . . . .	19
5.5	Page d'Administration . . . . .	19
5.6	Les produits . . . . .	20
6.1	Recommendation Example N° 1 . . . . .	21
6.2	Recommendation Example N° 2 . . . . .	22
6.3	Recommendation Example N° 3 . . . . .	22
6.4	Recommendation Example N° 4 . . . . .	23
6.5	Recommendation Example N° 5 . . . . .	23

# List of Abbreviations

NLP	Natural language processing
-----	-----------------------------

# Chapter 1

## Introduction

### 1.1 Contexte

**Les systèmes de recommandation** sont une sous-classe de l'*apprentissage automatique* qui traite généralement du classement ou de l'évaluation des produits/utilisateurs. Peu défini, un système de recommandation est un système qui prédit les notes qu'un utilisateur pourrait donner à un élément spécifique. Ces prédictions seront ensuite classées et retournées à l'utilisateur.

Ils sont utilisés par diverses grandes sociétés de noms comme *Google, Instagram, Spotify, Amazon, Reddit, Netflix, etc.* souvent pour augmenter l'engagement avec les utilisateurs et la plate-forme. Par exemple, *Spotify* vous recommanderait des chansons similaires à celles que vous avez écoutées ou aimées à plusieurs reprises afin que vous puissiez continuer à utiliser leur plateforme pour écouter de la musique. Amazon utilise des recommandations pour suggérer des produits à divers utilisateurs en fonction des données qu'ils ont recueillies pour cet utilisateur.

*Les systèmes de recommandation* sont souvent considérés comme une boîte noire, le modèle créé par ces grandes entreprises n'est pas très facile à interpréter. Les résultats qui sont générés sont souvent des recommandations pour l'utilisateur pour les choses qu'il a besoin / veut, mais ne sont pas conscients qu'ils ont besoin / veulent jusqu'à ce qu'ils aient été recommandés à eux.

### 1.2 Déclaration de problème

*Le monde du marketing* évolue. De nombreuses entreprises ferment et sont remplacées par des magasins en ligne, des services d'adhésion, etc. Alors que *le mélange de la collecte de produits attire le client à l'achat en ligne*, beaucoup de plateformes de commerce électronique ne parviennent pas à vendre à travers leurs marchandises. Cela se produit souvent en raison de *la mauvaise expérience de navigation* des utilisateurs. Les clients après avoir parcouru le site Web pendant des heures ne peuvent pas trouver l'élément qu'ils recherchent. Cela doit changer et *les acheteurs doivent avoir des choix en fonction de leurs préférences* afin de créer un environnement approprié pour les clients.

### 1.3 Les Objectives

On veut réaliser un système de recommandation fashion, sur un site web E-Commerce qui va regrouper plusieurs techniques de recommandation.

### 1.4 Approche sur la solution

Pour Construire notre site web, on va utiliser des modèles machine learning :

- *Modèle Basé sur CountVctrizer*
- *Modèle Bag-of-Words*
- *Modèle Basé sur Cosine\_Similarity*
- *Modèle Basé sur pairwise\_distances*



## Chapter 2

# Les Types Des Systemes De Recommendation

Examinons cinq types fondamentaux de systèmes de recommandation. [TheBluePi \(2021\)](#)

Il convient de noter que le système de recommandation ne se limite pas à l'utilisation d'un type particulier et peut combiner un certain nombre de types différents en fonction des exigences de l'exploitation commerciale spécifique.

### 2.1 Système de recommandation collaboratif :

Le *feedback* de l'utilisateur est la base pour d'autres suggestions dans ce type de système de recommandation. Voici comment ça marche :

- Tout d'abord, le système agrège les résultats de l'utilisateur - divers types d'historique de recherche, les cotes, les commentaires et les recommandations de produits ou de morceaux de contenu dans un grand ensemble de données. Ensuite, il compare les résultats de différents utilisateurs pour des produits spécifiques, trouve des éléments communs et calcule les correspondances entre différents éléments de contenu.
- Ce type de système de recommandation est commun dans les places de marché *e-commerce*. Par exemple, une variante de l'algorithme du système de recommandation collaborative est actuellement utilisée sur *Amazon*.
- Dans l'ensemble, le système collaboratif est un moyen relativement simple de faire des suggestions pertinentes aux clients. D'autre part, c'est un bon moyen de comprendre quels produits sont préférés par les utilisateurs et dans quelle mesure par rapport aux autres produits.



Figure 2.1: Système de recommandation collaboratif

## 2.2 Système de recommandation basé sur le contenu :

La recommandation basée sur le contenu va dans la direction opposée des systèmes de collaboration. Au lieu de se concentrer sur le comportement des utilisateurs, la recommandation basée sur le contenu est construite autour de l'inventaire des articles (produits, contenu) et de la comparaison d'attribution. Voici comment ça marche :

- Dans ce cas, si l'utilisateur recherche des ordinateurs *IBM "Think"*, le système suggérera probablement des ordinateurs portables de **taille et de spécifications techniques similaires**.
- **Les mots clés** qui décrivent les éléments jettent les bases pour les suggestions, et chaque produit a généralement plus d'un mot clé pour rendre l'appariement plus facile et plus précis. Ces mots clés, associés à l'activité de l'utilisateur, constituent le champ d'application de la recommandation de produit.
- Ce type de moteur de recommandation est largement utilisé dans les magasins de niche *e-commerce* (*Discogs* et *Artsy* utilisent cette approche) et aussi sur les sites Web d'agrégation de contenu avec des sélections étendues de contenu spécifique à contourner (*comme Mashable et The Next Web*).

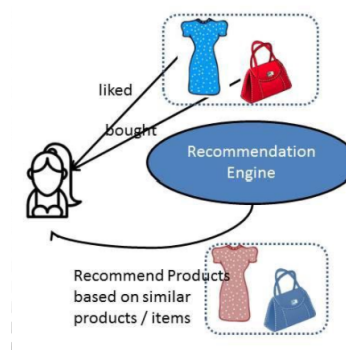


Figure 2.2: Système de recommandation basé sur le contenu

### 2.3 Système de recommandation basé sur la démographie :

Le système basé sur la démographie fournit des suggestions basées sur les caractéristiques des segments d'auditoire particuliers. Voici comment ça marche :

- Ce type de système de recommandation prend les données d'utilisateurs disponibles (*âge, sexe, lieu, etc.*), les classe en segments d'audience spécifiques, puis les met dans une plus grande image pour combler les lacunes dans les données.
- Les suggestions démographiques sont largement utilisées sur les sites Web d'agrégation de contenu et sur le marché général du commerce électronique. Habituellement, ce type de recommandation fournit une opération de fond dans le cas où il n'y a pas d'autres informations disponibles.
- Le système basé sur la démographie est l'un des types les plus simples de systèmes de recommandation qui exigent un ensemble limité de données pour fournir des suggestions générales. En tant que tel, il est moins dépendant des données des utilisateurs. Cependant, pour qu'il fonctionne, ce système nécessite une étude de marché complète comme base.

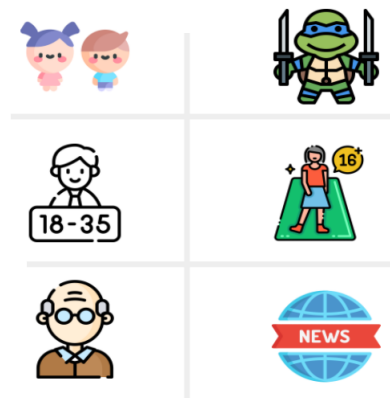


Figure 2.3: Système de recommandation basé sur la démographie

### 2.4 Système de recommandation basé sur les utilités :

Le système basé sur les utilités est celle qui tente de calculer l'utilité du produit particulier en fonction des préférences exprimées par les utilisateurs. Il est plus difficile de s'ajuster que les autres en raison de nombreux éléments supplémentaires dans l'équation. Pour calculer l'utilité du produit, vous devez :

- la corrélation de la requête de recherche
- comparaison avec les produits similaires considérés
- Disponibilité du produit
- classement du fournisseur et autres éléments pertinents

- Les recommandations basé sur les utilités exigent une large quantité d'informations sur les utilisateurs disponibles pour fournir des suggestions attrayantes. En tant que tel, il est utilisé pour suggérer des produits de niche sur les marchés de commerce électronique polyvalents tels qu'**Amazon** et aussi sur les magasins de niche pour le matériel et d'autres produits.

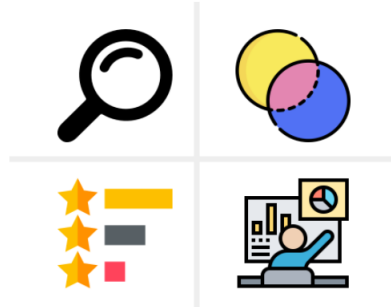


Figure 2.4: Système de recommandation basé sur les utilités

## 2.5 Système de recommandation basé sur les connaissances :

Le système basé sur la connaissance a approfondi le comportement des utilisateurs pour calculer les suggestions à partir des interactions enregistrées et des besoins et préférences présumés. Contrairement aux autres types de systèmes de recommandation, la connaissance calcule l'appariement et les possibilités et tente de prédire les plus puissants. Pour ce faire, le système :

- prend les renseignements disponibles sur les utilisateurs;
- le traite à travers une combinaison d'algorithmes d'apprentissage automatique prédictifs et normatifs.
- Évalue comment un produit spécifique répond aux préférences de l'utilisateur.
- Le résultat est une suggestion plus précise avec un potentiel plus substantiel d'obtenir une conversion.

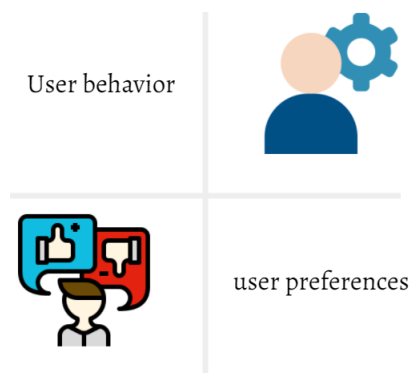


Figure 2.5: Système de recommandation basé sur les connaissances

## Chapter 3

# Data Preprocessing

### 3.1 Réduction De Dimension :

Initialement notre Data Set à 19 *features*, comme suit :

List Des Features
-------------------

sku
asin
product_type_name
formatted_price
author
color
brand
publisher
availability
reviews
large_image_url
availability_type
small_image_url
editorial_review
title
model
medium_image_url
manufacturer
editorial_reivew

```
1 data = data.loc[~data['formatted_price'].isnull()]
2 print('Number of data points After eliminating price=NULL :', data.shape
  [0])
3
4 data =data.loc[~data['color'].isnull()]
5 print('Number of data points After eliminating color=NULL :', data.shape
  [0])
6
7 # Output :
8 Number of data points After eliminating price=NULL : 28395
9 Number of data points After eliminating color=NULL : 28385
```

Pour avoir seulement 7 features qui nous intéressent :

List Des Features
asin
product_type_name
formatted_price
color
brand
title
<u>medium_image_url</u>

### 3.2 Suppression des valeurs nulles :

Dans cette partie, on a supprimé les données avec un price ou color = nan, parce que ce sont deux champs importants dans les étapes suivantes :

```

1 data = data.loc[~data['formatted_price'].isnull()]
2 print('Number of data points After eliminating price=NULL :', data.shape
  [0])
3
4 data = data.loc[~data['color'].isnull()]
5 print('Number of data points After eliminating color=NULL :', data.shape
  [0])
6
7
8 # Output :
9 Number of data points After eliminating price=NULL : 28395
10 Number of data points After eliminating color=NULL : 28385

```

### 3.3 Retirer les produits avec une brève description :

Aussi dans cette partie, on va supprimer les produits avec un short description parce que encore une fois, la description est un facteur important dans les traitements nlp qu'on va faire par la suite.

```

1 # Remove All products with very few words in title
2 data_sorted = data[data['title'].apply(lambda x: len(x.split())>4)]
3 print("After removal of products with short description:", data_sorted.
  shape[0])
4
5 # Output :
6 After removal of products with short description: 27949

```

### 3.4 Suppression des produits avec double titre par tri :

Dans cette partie, on va supprimer les duplication dans le titre, Par exemple :

Titles 1:

16. woman's place is in the house and the senate shirts for Womens XXL White
17. woman's place is in the house and the senate shirts for Womens M Grey

Title 2:

25. tokidoki The Queen of Diamonds Women's Shirt X-Large
26. tokidoki The Queen of Diamonds Women's Shirt Small
27. tokidoki The Queen of Diamonds Women's Shirt Large

Title 3:

61. psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Head Shirt for woman Neon Wolf t-shirt
62. psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Head Shirt for woman Neon Wolf t-shirt
63. psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Head Shirt for woman Neon Wolf t-shirt
64. psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Head Shirt for woman Neon Wolf t-shirt

```

1 print('The number of entries with duplicate title is %d'%sum(data.
    duplicated('title')))
2 # Sort the whole data based on title (alphabetical order of title)
3 data_sorted.sort_values('title',inplace=True, ascending=False)
4
5 indices = []
6 for i,row in data_sorted.iterrows():
7     indices.append(i)
8 import itertools
9 stage1_dedupe_asins = []
10 i = 0
11 j = 0
12 num_data_points = data_sorted.shape[0]
13 while i < num_data_points and j < num_data_points:
14     previous_i = i
15     # store the list of words of ith string in a, ex: a = ['tokidoki', '
The', 'Queen', 'of', 'Diamonds', 'Women's', 'Shirt', 'X-Large']
16     a = data['title'].loc[indices[i]].split()
17     # search for the similar products sequentially
18     j = i+1
19     while j < num_data_points:
20         # store the list of words of jth string in b, ex: b = ['tokidoki',
'The', 'Queen', 'of', 'Diamonds', 'Women's', 'Shirt', 'Small']
21         b = data['title'].loc[indices[j]].split()
22         # store the maximum length of two strings
23         length = max(len(a), len(b))
24         # count is used to store the number of words that are matched in
both strings
25         count = 0
26         # itertools.zip_longest(a,b): will map the corresponding words in
both strings, it will appened None in case of unequal strings
27         # example: a =['a', 'b', 'c', 'd']
28         # b = ['a', 'b', 'd']

```



```

29     # itertools.zip_longest(a,b): will give [('a','a'), ('b','b'), ('c
    ', 'd'), ('d', None)]
30     for k in itertools.zip_longest(a,b):
31         if (k[0] == k[1]):
32             count += 1
33     # if the number of words in which both strings differ are > 2 , we
    are considering it as those two apperals are different
34     # if the number of words in which both strings differ are < 2 , we
    are considering it as those two apperals are same, hence we are
    ignoring them
35     if (length - count) > 2: # number of words in which both sentences
    differ
36     # if both strings are differ by more than 2 words we include
    the 1st string index
37     stage1_dedupe_asins.append(data_sorted['asin'].loc[indices[i
    ]])
38     # if the comaprision between is between num_data_points,
    num_data_points-1 strings and they differ in more than 2 words we
    include both
39     if j == num_data_points-1: stage1_dedupe_asins.append(
    data_sorted['asin'].loc[indices[j]])
40     # start searching for similar apperals corresponds 2nd string
41     i = j
42     break
43     else:
44         j += 1
45     if previous_i == i:
46         break
47
48 data = data.loc[data['asin'].isin(stage1_dedupe_asins)]
49 print('Number of data points now is: ', data.shape[0])
50
51 # Output :
52 The number of entries with duplicate title is 2325
53 Number of data points now is: 17593

```

### 3.5 Suppression des produits avec double titre:

Dans la partie précédente, nous avons trié des données entières dans l'ordre alphabétique des titres. Ensuite, nous avons supprimé des titres qui sont adjacents et très similaire titre

Mais il y a certains produits dont les titres ne sont pas adjacents mais très similaires, Par exemple :

Titles-1

86261. UltraClub Women's Classic Wrinkle-Free Long Sleeve Oxford Shirt, Pink, XX-Large

115042. UltraClub Ladies Classic Wrinkle-Free Long-Sleeve Oxford Light Blue XXL

Titles-2

75004. EVALY Women's Cool University Of UTAH 3/4 Sleeve Raglan Tee

109225. EVALY Women's Unique University Of UTAH 3/4 Sleeve Raglan Tees

120832. EVALY Women's New University Of UTAH 3/4-Sleeve Raglan Tshirt

```

1 indices = []
2 for i,row in data.iterrows():
3     indices.append(i)
4
5 stage2_dedupe_asins = []
6 while len(indices)!=0:
7     i = indices.pop()
8     stage2_dedupe_asins.append(data['asin'].loc[i])
9     # consider the first apperal's title
10    a = data['title'].loc[i].split()
11    # store the list of words of ith string in a, ex: a = ['tokidoki', '
The', 'Queen', 'of', 'Diamonds', 'Women's', 'Shirt', 'X-Large']
12    for j in indices:
13        b = data['title'].loc[j].split()
14        # store the list of words of jth string in b, ex: b = ['tokidoki',
'The', 'Queen', 'of', 'Diamonds', 'Women's', 'Shirt', 'X-Large']
15        length = max(len(a),len(b))
16        # count is used to store the number of words that are matched in
both strings
17        count = 0
18        # itertools.zip_longest(a,b): will map the corresponding words in
both strings, it will appened None in case of unequal strings
19        # example: a =['a', 'b', 'c', 'd']
20        # b = ['a', 'b', 'd']
21        # itertools.zip_longest(a,b): will give [('a','a'), ('b','b'), ('c
','d'), ('d', None)]
22        for k in itertools.zip_longest(a,b):
23            if (k[0]==k[1]):
24                count += 1
25        # if the number of words in which both strings differ are < 3 , we
are considering it as those two apperals are same, hence we are
ignoring them
26        if (length - count) < 3:
27            indices.remove(j)
28
29 data = data.loc[data['asin'].isin(stage2_dedupe_asins)]
30 print('Number of data points after stage two of dedupe: ',data.shape[0])
31 data.to_pickle('pickels/16k_apperal_data')

```

## Chapter 4

# Les Modèles Machine Learning

Dans cette section, on va lister tous les modèles qu'on a utilisé dans notre projet. J'expliquerai 4 stratégies différentes pour la classification multiclassées de texte :

### 4.1 CountVectorizer :

[towardsdatascience \(2021\)](#) CountVectorizer tokenises (tokenisation signifie décomposer une phrase ou un paragraphe ou n'importe quel texte en mots) le texte avec effectuer le prétraitement très basique comme enlever les marques de ponctuation, convertir tous les mots en minuscules, etc. Vous pouvez importer, ce modèle sous sklearn avec :

```
1 from sklearn.feature_extraction.text import CountVectorizer
```

Le vocabulaire des mots connus est formé qui est également utilisé pour l'encodage du texte invisible plus tard.

Un vecteur codé est retourné avec une longueur du vocabulaire entier et un nombre entier pour le nombre de fois que chaque mot est apparu dans le document. Prenons un exemple pour voir comment cela fonctionne.

Considérez la phrase suivante :

*"All the countries of the world are rich"* Ce texte sera donc représenté comme suit :

all	the	countries	of	world	are	rich
1	2	1	1	1	1	1

Index	0	1	2	3	5	6	7
Doc	1	2	1	1	1	1	1

Des tableaux ci-dessus nous pouvons voir la représentation matricielle clairesmée de CountVectorizer des mots. Le tableau A est la façon dont vous y pensez visuellement tandis que le tableau B est la façon dont il est représenté dans la pratique.

La ligne de la matrice ci-dessus représente le document et les colonnes contiennent tous les mots uniques avec leur fréquence. Dans le cas où un mot ne s'est pas produit, alors il est attribué zéro corrélation au document dans une rangée.

Pour plus d'info sur ce modèle vous pouvez consulter : [La documentation de sklearn](#)

## 4.2 Bag-of-Words :

[geeksforgeeks \(2021a\)](#) Vous pouvez importer, ce modèle sous sklearn avec :

```
1 from sklearn.feature_extraction.text import TfidfVectorizer
```

Le modèle Bag-of-Words: il construit un vocabulaire à partir d'un corpus de documents et compte combien de fois les mots apparaissent dans chaque document. Autrement dit, chaque mot du vocabulaire devient une caractéristique et un document est représenté par un vecteur ayant la même longueur de vocabulaire (un Bag of words). Par exemple, prenons 3 phrases et représentons-les avec cette approche :

	I	like	this	article	medium	data
I like this article	1	1	1	1	0	0
I like medium	1	1	0	0	1	0
I like data	1	1	0	0	0	1

Figure 4.1: Forme de la matrice des caractéristiques : Nombre de documents  
x Longueur du vocabulaire

Notez que les vecteurs ici ont la longueur 6 au lieu de 3 en raison de l'élément 0 supplémentaire à la place des mots qui ne sont pas présent dans notre phrase.

Pour plus d'info sur ce modèle vous pouvez consulter : [La documentation de sklearn](#)

## 4.3 Cosine\_Similarity :

[geeksforgeeks \(2021b\)](#) Vous pouvez importer, ce modèle sous sklearn avec :

```
1 from sklearn.metrics.pairwise import cosine_similarity
```

La similitude cosinienne est l'une des meilleures façons de juger ou de mesurer la similitude entre les documents. Indépendamment de la taille, Cet outil de mesure de similitude fonctionne très bien. Nous pouvons également mettre en œuvre sans sklearn module. Mais ce sera une tâche plus fastidieuse. Sklearn simplifie cela. J'espère que cet article, doit avoir approuvé la mise en œuvre. Cependant, si vous avez trouvé, l'une des lacunes de l'information. S'il vous plaît, laissez-nous savoir. Vous pouvez également commenter comme commentaire ci-dessous.

L'idée de *Cosine\_Similarity* c'est qu'il convertit les phrases a des vecteurs et mesurer la **cos**.

voici la figure en 2D :

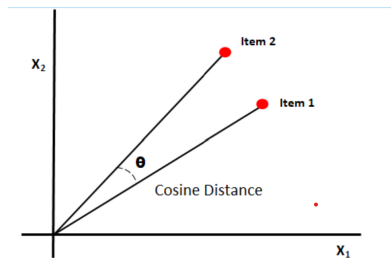


Figure 4.2: cosine similarity en 2-D

Pour plus d'info sur ce modèle vous pouvez consulter : [La documentation de sklearn](#)

## 4.4 `pairwise_distances` :

[geeksforgeeks \(2021c\)](#) Vous pouvez importer, ce modèle sous sklearn avec :

```
1 from sklearn.metrics import pairwise_distances
```

Simplement, ce modèle repose sur les calculs des distances :  
sous *sklearn* ils peuvent être :

- *cityblock*
- *cosine*
- *euclidean*
- *l1* ou *l2*
- *manhattan*

**Notez :**

$$\text{Cosine\_distance} = 1 - \text{cosine\_similarity}$$

Pour plus d'info sur ce modèle vous pouvez consulter : [La documentation de \*sklearn\*](#)

## Chapter 5

# Structure Du Site Web

Dans cette section, on va essayer de découvrir la structure générale de notre site web :

### 5.1 Templates :

Ce projet se concentrera sur 4 modèles principaux, *store.html*, *cart.html*, *view.html* et *checkout.html*.

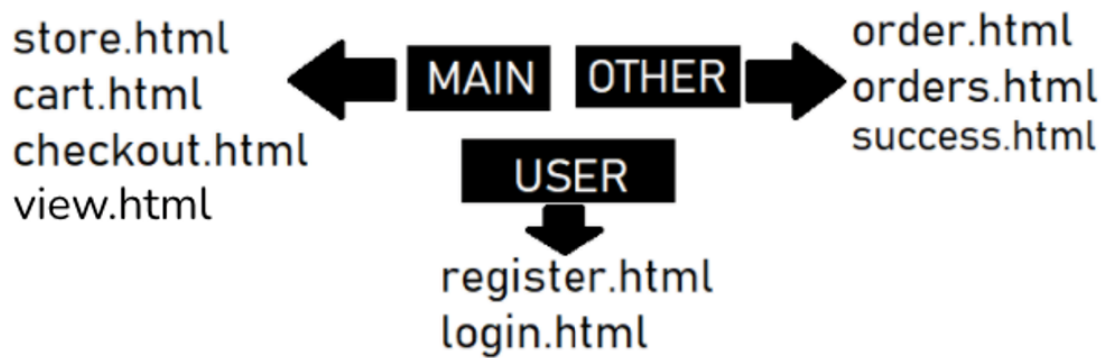


Figure 5.1: Templates

## 5.2 Models :

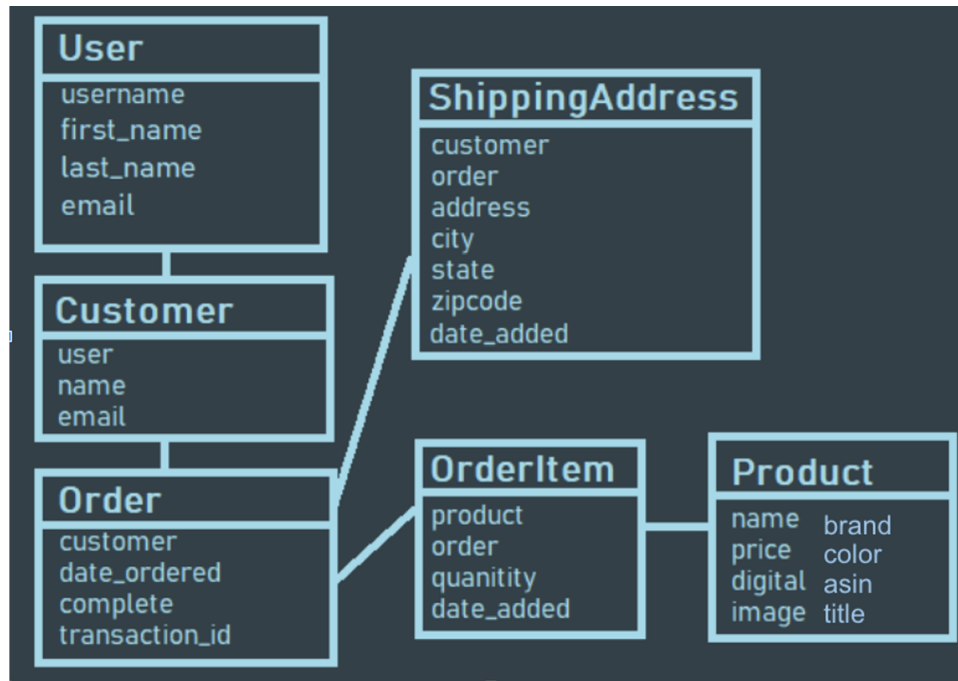


Figure 5.2: Models

Ce projet se composera de 6 modèles:

### 1. **USER :**

Modèle utilisateur Django intégré, instance créée pour chaque client qui s'inscrit.

### 2. **CUSTOMER :**

En plus d'un modèle d'utilisateur, chaque client contiendra un modèle de client qui entretient une relation un à un avec chaque utilisateur. (OneToOneFied)

### 3. **PRODUCT :**

Le modèle de produit représente les types de produits que nous avons en magasin.

### 4. **ORDER :**

Le modèle de commande représentera une transaction passée ou en attente. Le modèle contiendra des informations telles que l'ID de la transaction, les données complétées et le statut de la commande. Ce modèle sera un modèle enfant ou client mais un parent pour les articles de commande.

## 5. ORDERITEM :

Un article de commande est un article avec une commande. Par exemple, un panier peut être composé de plusieurs articles mais fait tous partie d'une seule commande. Par conséquent, le modèle OrderItem sera un enfant du modèle PRODUCT ET du modèle ORDER.

## 6. SHIPPING :

Toutes les commandes n'auront pas besoin d'informations d'expédition. Pour les commandes contenant des produits physiques qui doivent être expédiés, nous devons créer une instance du modèle d'expédition pour savoir où envoyer la commande. L'expédition sera simplement un enfant du modèle de commande si nécessaire.

### 5.3 Les fonctionnalités :

- Home Page :

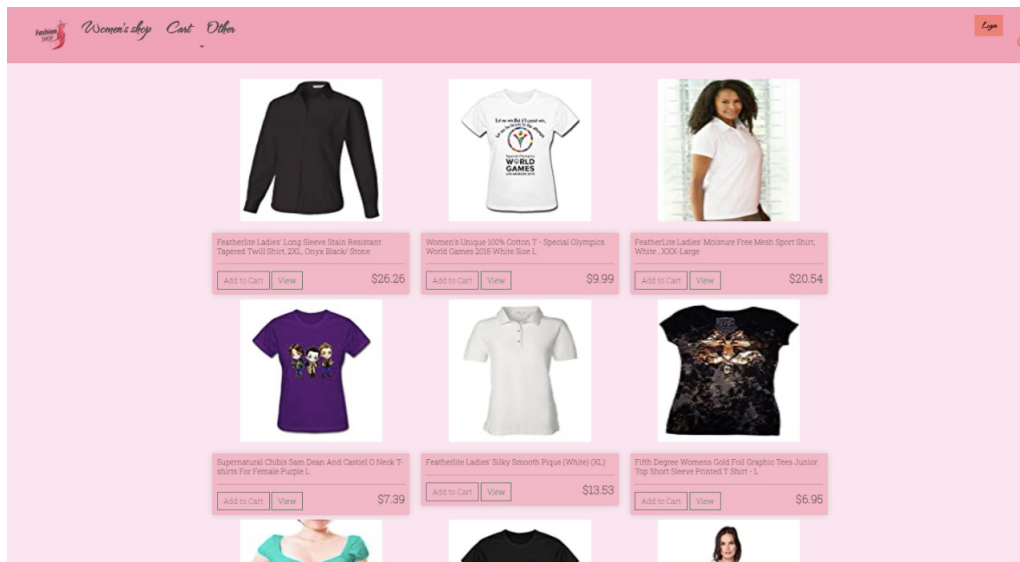


Figure 5.3: Home Page



### • Payment Integration:

The screenshot displays a checkout interface. On the left, a 'Shipping Information' form contains fields for Address, City, State, and Zip code, with a 'Continue' button below. On the right, an 'Order Summary' section includes a 'Back to Cart' button and a table of items. The table lists five items with their respective prices and quantities. Below the table, it shows 'Items: 10' and a 'Total: \$197.08'.

Item Image	Item Description	Price	Quantity
	Women's Unique 100% Cotton T - Special Olympics World Games 2018 White Size L	\$9.99	x5
	FeatherLite Ladies' Long Sleeve Stain Resistant Tapered Twill Shirt, XXL, Gorys Black/ Stone	\$26.26	x1
	FeatherLite Ladies' Moisture Free Mesh Sport Shirt, White, XXX-Large	\$20.54	x2
	Supernatural Chubis Sam Dean And Castiel O Neck T-shirts For Female Purple L	\$7.39	x1
	Feel The Piece Sam's Dip Dye Top One Size in Navy	\$72.40	x1

Items: 10  
Total: \$197.08

Figure 5.4: Payment Integration

### • Page d'Administration:

The screenshot shows the Django administration interface. The top navigation bar includes the 'Django administration' logo and a welcome message for 'AMANARD1' with links to 'VIEW SITE', 'CHANGE PASSWORD', and 'LOG OUT'. The main content area is titled 'Site administration' and is divided into two sections: 'AUTHENTICATION AND AUTHORIZATION' and 'STORE'. The 'AUTHENTICATION AND AUTHORIZATION' section lists 'Groups' and 'Users' with 'Add' and 'Change' links. The 'STORE' section lists 'Customers', 'Order items', 'Orders', 'Products', and 'Shipping address', each with 'Add' and 'Change' links. On the right, a 'Recent actions' sidebar shows a list of actions, including one by 'amanar' (Customer).

Figure 5.5: Page d'Administration

- Les produits:

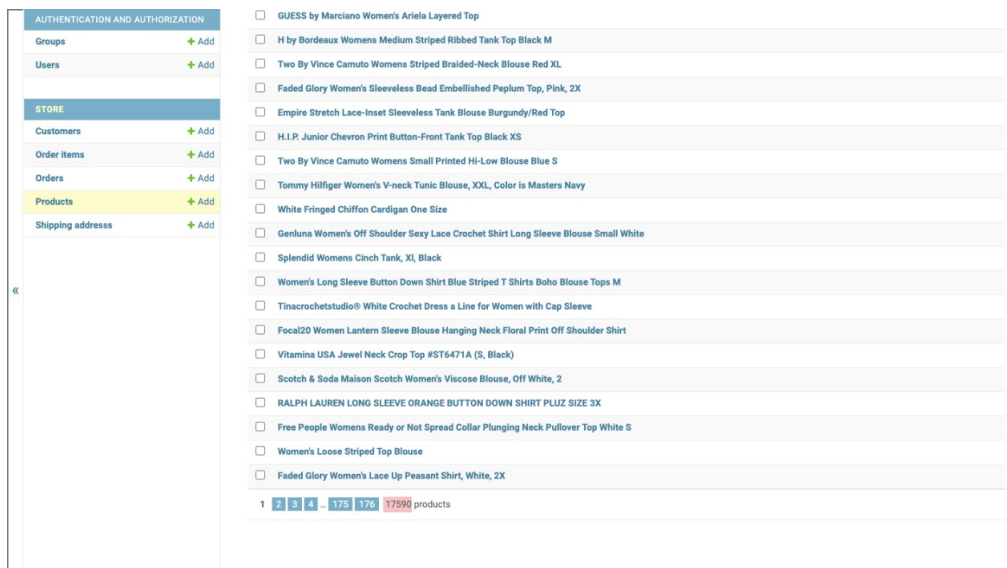


Figure 5.6: Les produits

## 5.4 User/Guest Checkout :

ce site Web de commerce donner à l'utilisateur la possibilité de passer à la caisse sans créer de compte. **Authenticated User Vs Guest Checkout**

- Authenticated User Process

- Ajouter un article au panier → Modifier la commande → Commander
- Afficher les commandes en attente et précédentes + Détails de la commande

- Guest Checkout Process

- Ajouter un article au panier → Modifier la commande → Commander
- Créer un compte pour voir la commande

## Chapter 6

# Site Web Et Processus De Recommendation

- **Recommendation :**

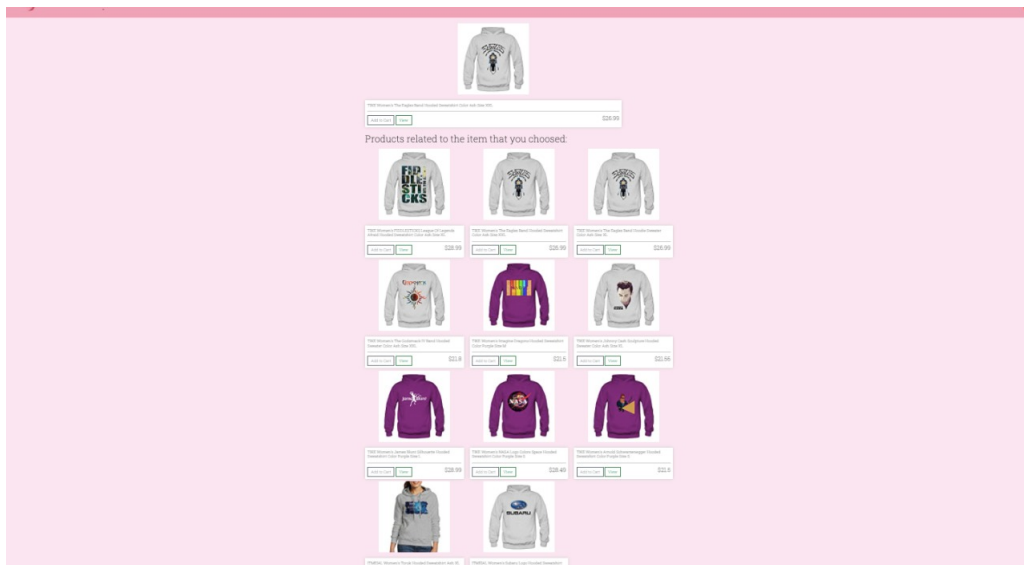


Figure 6.1: Recommendation Example N° 1

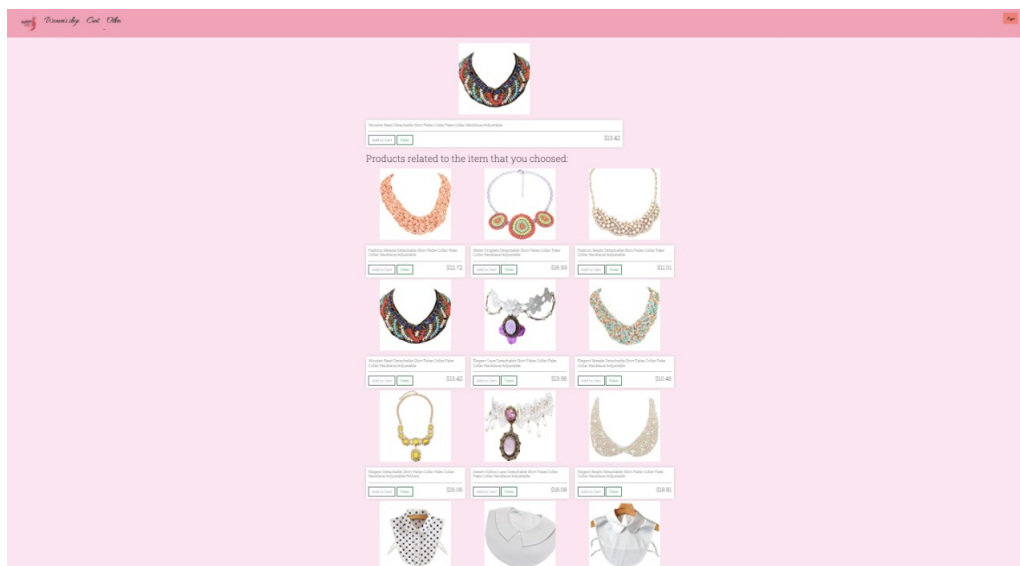


Figure 6.2: Recommendation Example N° 2

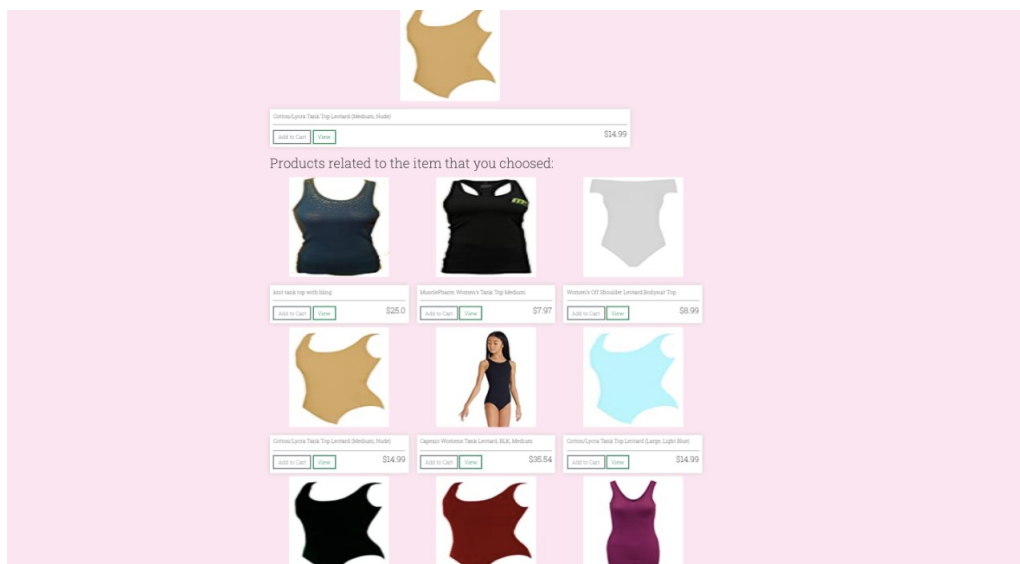


Figure 6.3: Recommendation Example N° 3

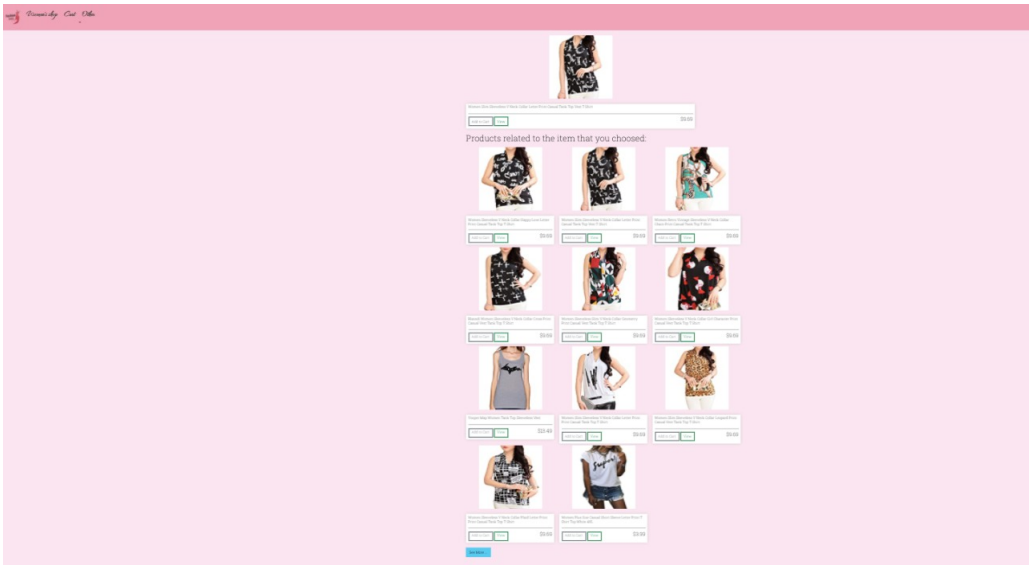


Figure 6.4: Recommendation Example N° 4

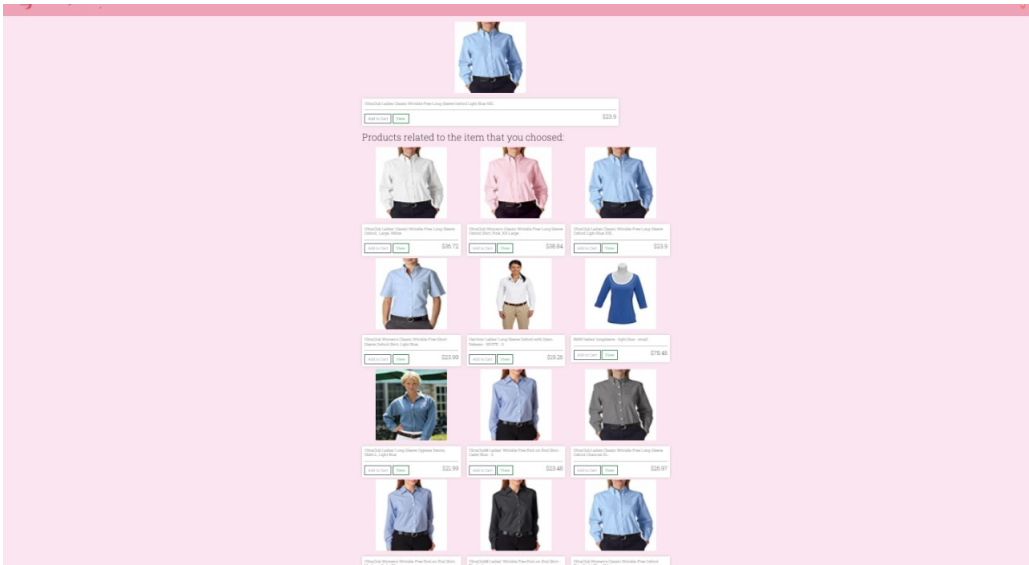


Figure 6.5: Recommendation Example N° 5

## Chapter 7

# Conclusion

Sur Internet, où le nombre de choix augmente, il faut un filtre pour redessiner l'information en fonction de son intérêt et de son utilité. Elle est devenue essentielle pour les utilisateurs afin que son absence entraîne une baisse significative de la qualité du service ainsi qu'une réduction de la satisfaction des utilisateurs. Ces systèmes sont maintenant une partie importante du magasin, des nouvelles, des médias sociaux, des films et de la musique, des livres.

Et notre projet est un site web démonstratif des sur les concepts des systèmes de recommandation en utilisons Django et Nlp, que notre souhait d'être utile pour vous pour démarrer votre propre site Web ou avoir une idée sur les systèmes de recommandation actuelles, comme **Spotify**, **Amazon**, ...

# References

geeksforgeeks (2021a), ‘Bag-of-words’. (accessed Décembre 26, 2021).

URL: <https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/>

geeksforgeeks (2021b), ‘Cosine similarity’. (accessed Décembre 6, 2021).

URL: <https://www.geeksforgeeks.org/cosine-similarity/>

geeksforgeeks (2021c), ‘pairwise distances’. (accessed Décembre 16, 2021).

URL: [https://www.programcreek.com/python/example/100423/sklearn.metrics.pairwise.pairwise\\_distances](https://www.programcreek.com/python/example/100423/sklearn.metrics.pairwise.pairwise_distances)

TheBluePi (2021), ‘Five types of recommender systems and their benefits’. (accessed Aout 26, 2021).

URL: <https://www.bluepiit.com/blog/classifying-recommender-systems/>

towardsdatascience (2021), ‘countvectorizer’. (accessed Décembre 1, 2021).

URL: <https://towardsdatascience.com/basics-of-countvectorizer-e26677900f9c>