



# Book Recommender System

Ορφανός Σταμάτιος  
Ε17113



# Περιεχόμενα

- Recommender Systems
- Είδη Recommender System
- Δεδομένα
- Αρχιτεκτονική Recommender System
- Αρχιτεκτονική Εφαρμογής



# Recommender Systems

- Τα Recommender Systems ξεκίνησαν στα μέσα του 1990 και είναι μια υποκατηγορία του χώρου Information Filtering Systems. Τα συστήματα αυτά έχουν ως στόχο να προβλέψουν την αξιολόγηση ή την προτίμηση ενός προϊόντος για ένα χρήστη.
- Σήμερα τα Recommender Systems είναι αρκετά διαδεδομένα και χρησιμοποιούνται σε όλες τις μεγάλες ιστοσελίδες, διαφορετικών χώρων όπως αγοράς προϊόντων, video streaming, ηλεκτρονικών εφημερίδων και άλλων χώρων όπου υπάρχει η δυνατότητα εξατομίκευσης.



# Recommender Systems

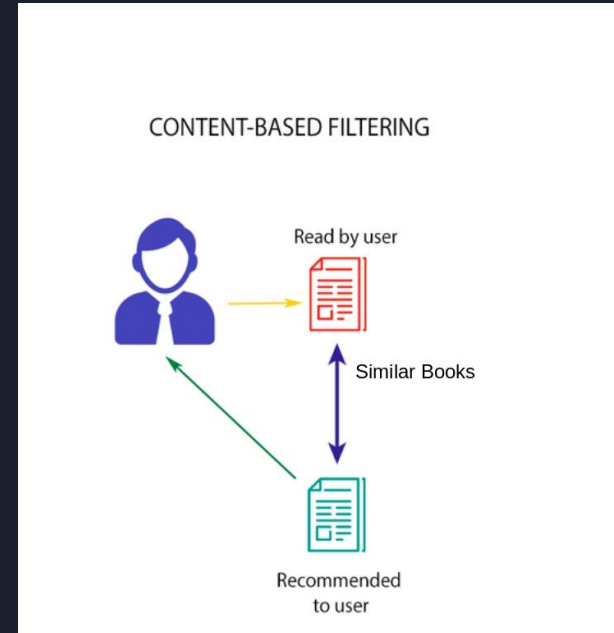
- Η πρόβλεψη για ένα προϊόν μπορεί να γίνει με διαφορετικούς τρόπους, ανάλογα με το είδος του Recommender System που χρησιμοποιούμε. Οι βασικοί τύποι recommendations είναι οι εξής :

1. Editorial - Hand Curated
2. Simple Aggregates
3. Personalised Recommendations

# Είδη Personalised Recommender Systems

## 1. Content-Based Filtering System

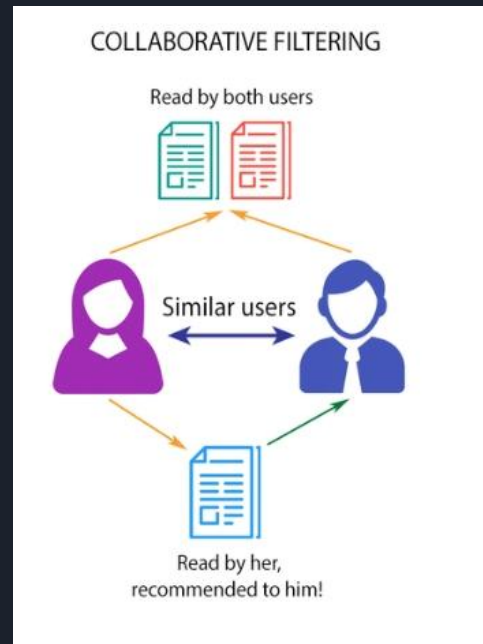
Τα Content-based filtering systems βασίζονται στην ομοιότητα των προϊόντων ή αντικειμένων προκειμένου να αναπτυχθεί το σύστημα προτάσεων. Πιο συγκεκριμένα τέτοια συστήματα χρησιμοποιούν χαρακτηριστικά των προϊόντων για να προσδιορίσουν την ομοιότητα μεταξύ δύο προϊόντων.



# Είδη Personalised Recommender Systems

## 2. Collaborative-Based Filtering System

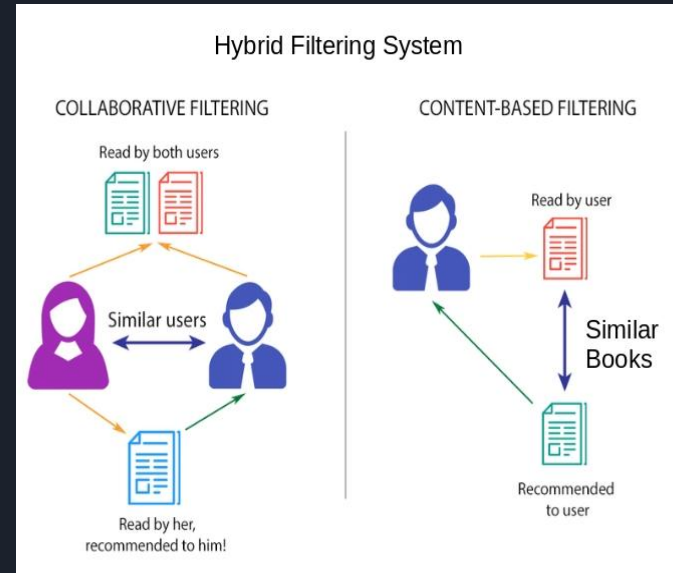
Αυτά τα συστήματα βασίζονται άμεσα στην συμπεριφορά των users, δηλαδή το σύστημα ορίζει το μοντέλο που θα χρησιμοποιήσουμε με βάση παλαιότερες επιλογές, δραστηριότητες και προτιμήσεις των users. Χρησιμοποιεί την παραπάνω γνώση για να προβλέψει τι θα αρέσει σε ένα user με βάση ένα ή περισσότερους users με παρόμοιες προτιμήσεις .



# Είδη Personalised Recommender Systems

## 3. Hybrid Filtering System

Αυτά τα συστήματα συνδυάζουν τις λειτουργίες των παραπάνω συστημάτων, έχοντας μεγαλύτερη ακρίβεια και μεγαλύτερο εύρος γνώσεων. Με αυτό τον τρόπο υπάρχει η δυνατότητα το σύστημα να παράγει περισσότερες προτάσεις για ένα user.





# Δεδομένα

Τα δεδομένα για την ανάπτυξη του μοντέλου προέρχονται από την σελίδα Kaggle και συγκεκριμένα είναι το παρακάτω data set:

<https://www.kaggle.com/zygmunt/goodbooks-10k>

Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκε το set ratings.csv, ενώ για την εφαρμογή χρησιμοποιήθηκε το set books.csv.



# Δεδομένα

Το set ratings.csv περιέχει τρεις στήλες με τα παρακάτω χαρακτηριστικά :

1. book\_id
2. user\_id
3. rating

Προφανώς για κάθε βιβλίο υπάρχει ένα σύνολο από users που έχουν δώσει ένα rating. Με αυτό το data set θα εκπαιδεύσουμε το μοντέλο όπως θα δούμε στην συνέχεια. Σε αυτό το set έχουμε 10.000 βιβλία, 53.424 users και 981.000 αξιολογήσεις.

```
In [2]: import pandas as pd

ratings = pd.read_csv('ratings.csv')
ratings.head()
```

Out[2]:

	book_id	user_id	rating
0	1	314	5
1	1	439	3
2	1	588	5
3	1	1169	4
4	1	1185	4

# Δεδομένα

Το set books.csv περιέχει 23 στήλες με χαρακτηριστικά όπως το id, η έκδοση, ο κωδικός isbn, οι συγγραφείς, η χρονιά έκδοσης και άλλα πολλά χαρακτηριστικά που περιγράφουν ένα βιβλίο.

	id	book_id	best_book_id	work_id	books_count	isbn	isbn13	authors	original_publication_year	original
0	1	2767052	2767052	2792775	272	439023483	9.780439e+12	Suzanne Collins	2008.0	The Hi G
1	2	3	3	4640799	491	439554934	9.780440e+12	J.K. Rowling, Mary GrandPré	1997.0	Harry I ar Philosop !
2	3	41865	41865	3212258	226	316015849	9.780316e+12	Stephenie Meyer	2005.0	Tv
3	4	2657	2657	3275794	487	61120081	9.780061e+12	Harper Lee	1960.0	To Mockir
4	5	4671	4671	245494	1356	743273567	9.780743e+12	F. Scott Fitzgerald	1925.0	The G

5 rows × 23 columns



# Αρχιτεκτονική Recommender System

Για την ανάπτυξη του μοντέλου χρησιμοποιήθηκε το Collaborative-Based Filtering σύστημα, καθώς έχουμε ένα σύνολο δεδομένων ιστορικού των 10.000 users. Η αρχιτεκτονική περιλαμβάνει τα εξής βήματα :

1. Αρχικά βρίσκουμε όλα τα `user_ids`, στα οποία εφαρμόζουμε Embedding έτσι ώστε να μπορούμε να πετύχουμε τόσο την αντιστοίχιση- mapping μεταξύ των users και των books, αλλά και να μπορούμε να κατηγοριοποιήσουμε τους χρήστες με βάση τις προτιμήσεις τους.
2. Ομοίως επιτελούμε τις ίδιες διαδικασίες για τα `book_ids`.
3. Αμέσως μετά χρησιμοποιούμε την κλάση Flatten της βιβλιοθήκης keras προκειμένου να διαμορφώσουμε το embedding στη σωστή μορφή έτσι ώστε να το εισάγουμε στο Neural Network.



# Αρχιτεκτονική Recommender System

Αφού έχουμε εκτελέσει τα παραπάνω βήματα, ξεκινάμε με τον ορισμό του μοντέλου όπου :

1. Ξεκινώντας με το input layer, θα προσθέσουμε τα δύο embeddings των book\_ids και των user\_ids, με την εντολή Concatenate της βιβλιοθήκης keras.
2. Στην συνέχεια στο πρώτο hidden layer θα έχουμε ένα Dense layer με 500 units με activation function την Rectified Linear Unit.
3. Στο δεύτερο hidden layer χρησιμοποιούμε την κλάση Flatten, προκειμένου να μειώσουμε τις διαστάσεις σε μια διάσταση, έτσι ώστε να μπορούμε να χρησιμοποιήσουμε τα δεδομένα στο output layer.
4. Το output layer εφαρμόζεται με ένα Dense layer, το οποίο μπορεί να μας δώσει 5 αποτελέσματα όσες και οι πιθανές βαθμολογίες για ένα βιβλίο.

# Αρχιτεκτονική Recommender System

```
(base) stamatiisorfanos@pop-os:~/git/Recommendation_System$ /usr/bin/python3 /home/stamatiisorfanos/git/Recommendation_Syst
Model: "model"
```

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 1)]	0	
input_2 (InputLayer)	[(None, 1)]	0	
embedding (Embedding)	(None, 1, 128)	1280128	input_1[0][0]
embedding_1 (Embedding)	(None, 1, 128)	6838400	input_2[0][0]
flatten (Flatten)	(None, 128)	0	embedding[0][0]
flatten_1 (Flatten)	(None, 128)	0	embedding_1[0][0]
concatenate (Concatenate)	(None, 256)	0	flatten[0][0] flatten_1[0][0]
dense (Dense)	(None, 500)	128500	concatenate[0][0]
flatten_2 (Flatten)	(None, 500)	0	dense[0][0]
dense_1 (Dense)	(None, 5)	2505	flatten_2[0][0]

```
Total params: 8,249,533
Trainable params: 8,249,533
Non-trainable params: 0
```

# Αρχιτεκτονική Recommender System

- Μετά από την εκπαίδευση του μοντέλου παρατηρούμε ότι το accuracy δεν είναι αποδεκτό. Ωστόσο αυτό το γεγονός οφείλεται κυρίως στα μειονεκτήματα του Collaborative-Filtering System, τα οποία θα αναλύσουμε στην συνέχεια.

```
Epoch 1/10
6903/6903 [=====] - 24s 3ms/step - loss: 10.3870 - accuracy: 0.2902 - val_loss: 9.9315 - val_accuracy: 0.3076
Epoch 2/10
6903/6903 [=====] - 19s 3ms/step - loss: 9.8919 - accuracy: 0.3074 - val_loss: 9.9316 - val_accuracy: 0.3085
Epoch 3/10
6903/6903 [=====] - 19s 3ms/step - loss: 9.8834 - accuracy: 0.3094 - val_loss: 9.9315 - val_accuracy: 0.3083
Epoch 4/10
6903/6903 [=====] - 19s 3ms/step - loss: 9.8854 - accuracy: 0.3082 - val_loss: 9.9316 - val_accuracy: 0.3076
Epoch 5/10
6903/6903 [=====] - 19s 3ms/step - loss: 9.8858 - accuracy: 0.3086 - val_loss: 9.9315 - val_accuracy: 0.3075
Epoch 6/10
6903/6903 [=====] - 19s 3ms/step - loss: 9.8924 - accuracy: 0.3078 - val_loss: 9.9315 - val_accuracy: 0.3081
Epoch 7/10
6903/6903 [=====] - 19s 3ms/step - loss: 9.8881 - accuracy: 0.3075 - val_loss: 9.9315 - val_accuracy: 0.3081
Epoch 8/10
6903/6903 [=====] - 19s 3ms/step - loss: 9.8922 - accuracy: 0.3080 - val_loss: 9.9316 - val_accuracy: 0.3076
Epoch 9/10
6903/6903 [=====] - 19s 3ms/step - loss: 9.8823 - accuracy: 0.3071 - val_loss: 9.9315 - val_accuracy: 0.3082
Epoch 10/10
6903/6903 [=====] - 19s 3ms/step - loss: 9.8845 - accuracy: 0.3082 - val_loss: 9.9315 - val_accuracy: 0.3079
(base) stamatiisorfanos@pop-os:~/git/Recommendation_System$
```



# Αρχιτεκτονική Recommender System

## Μειονεκτήματα :

1. Cold Start : για κάθε καινούργιο user ή item, δεν υπάρχουν αρκετά δεδομένα για να γίνει αποδοτική πρόταση και έτσι από accuracy 80-90% πέφτει πολύ χαμηλά.
2. Sparsity: δεδομένου ότι έχουμε πολλά βιβλία, δεν υπάρχουν αρκετές αξιολογήσεις για κάθε βιβλίο και έτσι δεν γίνονται recommend ορισμένα βιβλία.

Με βάση τα παραπάνω μειονεκτήματα θα μπορούσε κανείς να υποστηρίξει ότι το Collaborative-Filtering System, δεν είναι τόσο αποδοτικό και δεν χρειάζεται. Ωστόσο έχουμε ένα βασικό πλεονεκτήματα.

## Πλεονεκτήματα :

1. Το βασικότερο πλεονέκτημα είναι το γεγονός ότι δεν απαιτείται από το μοντέλο να γνωρίζει το αντικείμενο για το οποίο προβλέπει την αξιολόγηση. Έτσι μπορούμε να χρησιμοποιήσουμε το παραπάνω μοντέλο για οποιαδήποτε αντικείμενο, όπως ταινίες, μουσική κ.ά. Εφόσον η βαθμολογία παραμένει στο ίδιο διάστημα.



# Αρχιτεκτονική Εφαρμογής

Για την ανάπτυξη της εφαρμογής έχουν χρησιμοποιηθεί τεχνολογίες όπως :

1. Tensorflow Keras
2. Tensorflow Js
3. Nodejs
4. Firebase





Σας ευχαριστώ για την προσοχή σας