

# Design and Implementation of a File Recommendation System Using Collaborative Filtering and Content-Based Recommendation for the Nextcloud Platform

Doğan Can Uçar

Frankfurt University of Applied Sciences

Faculty of Barrierefreie Systeme - Intelligente Systeme

[dogan@dogan-ucar.de](mailto:dogan@dogan-ucar.de)

[www.dogan-ucar.de](http://www.dogan-ucar.de)

# Agenda

Introduction

Requirement Analysis

State of the Art

Implementation

Evaluation

Conclusion

# Introduction (1)

## Background & Motivation

- "Artificial Intelligence" offers advantages over „conventional“ software.
- In the analog world, people would browse a set of items and chose the best.
- In the age of internet, the selection seems to be endless which makes "browsing" impossible.
- Therefore, **Recommendation Systems** came up in the early stages of the Web 2.0.

## Introduction (2)

### Background & Motivation

- Amazon.com's "**customer who bought X have also bought Y**" was one of the first recommendation systems.
- Building a recommendation system to help people sense fewer impulses, such as notifications, emails or entries in an activity feed.
- Contributing to the community by making the recommendation system open source.

## Introduction (3)

### Objectives & Scientific Context

- Recommendation systems are part of Information Retrieval particularly of Information Filtering.
- But also a part of Machine Learning since they "learn" user preferences.
- "Content-Based" recommendation systems can also be seen as part of Natural Language Processing.
- Goal: creating a Nextcloud App that works as a recommendation system providing a better overview about uploaded files.

# Introduction (4)

## about Nextcloud

- Nextcloud is an open source file hosting software with similar to Google Drive or Dropbox.
- Several (third party) apps on the Nextcloud App Store extend the core functionality.
- Founded as a fork of ownCloud in 2016 and differs from it in that it is completely open source.

# Requirement Analysis (1)

## Problem Definition & General Conditions

- Growth of files leads to a loss of overview.
- Machine Learning approach for filtering files.
- App requirements are oriented to those of Nextcloud 13.
  - PHP 7.0, since it offers a massive performance improvement.
- Development and evaluation mainly on a MacBook (Pro) and the company's Nextcloud instance.

# Requirement Analysis (2)

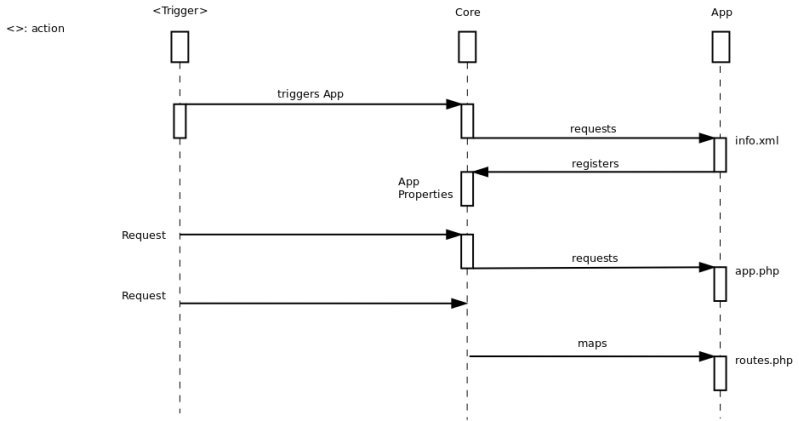
## Problem Definition & General Conditions

- Only "local" training possible due to the Nextcloud philosophy "safe home for all your data".
- Stakeholders have demanded a "Nextcloud App" as the resulting software, which requires PHP as the programming language.



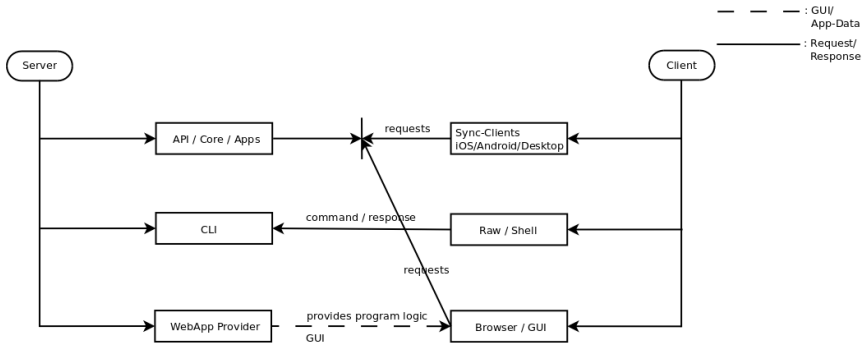
# Requirement Analysis (3)

## Software Architecture & Design



# Requirement Analysis (4)

## Software Architecture & Design



# State of the Art (1)

## Introduction

- Multiple approaches for recommendation systems available.
- Collaborative Filtering, Content-Based Recommendation and Knowledge-Based Recommendation are the most "important and widely used" techniques.
- Combining two or more techniques is called "hybridization".
- **Collaborative Filtering** and **Content-Based Recommendation** used for this thesis.

## State of the Art (2)

### Collaborative Filtering

- **Basic Idea:** finding users with common interests.
- **Basic assumption:** people with similar tastes in the past will likely have similar tastes in the future.
- **Main property:** user ratings for measuring "similarity".
- **Strengths:** media independent.
- **Weaknesses:** ineffective for a large item base with few ratings (Sparsity), users without ratings (Early Rater), users whose opinion do not match with others (Gray Sheep).

# State of the Art (3)

## Collaborative Filtering

### Similarity Calculation

$$sim(x, y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\|_2 \times \|\vec{y}\|_2} = \sum_{s \in S_{xy}} \frac{r_{x,s} r_{y,s}}{\sqrt{\sum_{s \in S_{xy}} (r_{x,s})^2} \sqrt{\sum_{s \in S_{xy}} (r_{y,s})^2}} \quad (1)$$

### Prediction

$$P_{u,i} = \frac{\sum_{k < N} s_{i,k} * r_{u,k}}{\sum_{k < N} |s_{i,k}|} \quad (2)$$

# State of the Art (4)

## Content-Based Recommendation

- Uses item "properties" (content, keywords, etc).
- User profile containing properties that describes user.
- **Strengths:** less affected by problems stated for Collaborative Filtering.
- **Weaknesses:** does not differ between "good" and "bad" content, preprocessing step required for multimedia content.

# State of the Art (5)

## Content-Based Recommendation

- **Stemming:** summarizing words that are similar in their meaning in their root forms ("computation", "computers" → compute).
- **Stopword Removal:** "naturally" with the TF-IDF measure.
- **TF-IDF:** words occurring often in a document are relevant, whereas words that occur often in the whole item base are irrelevant.
- **Degree of Match:** The overlap of user profile's keyword with those of the item.

# State of the Art (6)

## Content-Based Recommendation

### TF-IDF

$$w(t) = TF \times IDF$$

$$TF = \frac{n_i}{n} \quad (3)$$

$$IDF = \log_{10} \frac{|I|}{m}$$

### Degree of Match

$$M = \frac{|D \cap P|}{\min(|D|, |P|)} \quad (4)$$



# State of the Art (7)

## Other Filtering Techniques

- **Knowledge-Based Filtering:** based on constraints, for example "price < 150 USD".
- **Demographic Filtering:** recommendations based on demographic information like age, sex, religion, ethnicity, etc.

# State of the Art (8)

## Hybridization

Hybridization combines two or more recommendation techniques to a final recommendation.

- **Weighted Average:** weights the input and builds an average.
- **Model Using:** using machine learning to learn a recommendation model.
- **Pipelining:** The output of one technique is the input of another one.

# Implementation (1)

## Introduction

- Support from Nextcloud/ownCloud documentations and the core developers.
- Implementing a Nextcloud app provides a framework where basic conditions are defined.

## Implementation (2)

### Background Job

- **RecommenderJob** is registered as a background job.
- Represents the entry point for recommendation process.
- Extends **TimedJob** class.
- Calls **RecommenderService** class which contains the business logic.
  - iterates over all users and their files.
  - valid files are added to a list.
  - list contains instances **Item** which represents the file, its rating and content.

## Implementation (3)

### User Ratings

- Nextcloud provides a "tag as favorite" function which can be interpreted as a binary rating.
- TU Berlin has provided statistics about the "tag as favorite" functionality:
  - 1.982 from approximately 22.000 students are using the function and 2.727 items are tagged.
- in a second evaluation attempt, ratings are converted out of last modification time stamps.

# Implementation (4)

## Key classes

- two performance critical classes:
  - **ItemList**: contains all items (documents).
  - **KeywordList**: contains all keywords belonging to a document.
- implemented as a (kind of) **Set** datastructure:
  - a **Set** is defined among others as a list which contains each value only once.
  - before an item/keyword is added, it has to be ensured that it is not already available in the list.

# Implementation (5)

## Collaborative Filtering

- **CosineComputer** computes similarity of two **Item** instances injected through the constructor.
- **RatingPredictor** predicts a users rating for an item.

# Implementation (6)

## Content-Based Recommendation

- User profile keywords assembled in a timed interval.
- **UserProfileJob** is registered as a background job.
- Stopword removal using TF-IDF.
- Degree of Match: by formula presented above.



# Implementation (7)

## Hybridization

- Hybridization using Weighted Average.
- implemented as a static class method (due to its simplicity).
- Recommendation Transparency: users should know why they get items recommended.

# Evaluation (1)

## Introduction

- Two evaluation attempts: static file tags and last modification time stamps.
- Lack of test data, problems explained in State of the Art not addressed:
  - 5 randomly created users.
  - 25 news articles regarding to different categories.
- Evaluation has shown: Content-Based Recommendation leads to poor results in Nextcloud environment.

## Evaluation (2)

### Modification Time Stamps - Content-Based Recommendation

	<b>John</b>	<b>Luke</b>	<b>Robert</b>	<b>Tom</b>
<b>D1</b>	0,85	0,85	1,10	0,80
<b>D2</b>	0,90	1,00	0,85	0,85
<b>D3</b>	0,85	0,85	0,65	0,80
<b>D4</b>	1,05	0,95	0,80	0,85
<b>D5</b>	0,70	0,85	0,80	0,65

**Table:** Degree of Match Results

## Evaluation (3)

### Modification Time Stamps - Collaborative Filtering

	D1	D2	D3	D4	D5
<b>Brian</b>	1	5	4	3	2
<b>John</b>	2	4	4	3	3
<b>Luke</b>	1	3	?	2	1
<b>Robert</b>	5	?	?	2	5
<b>Tom</b>	4	?	2	?	?

Table: Ratings by Users

? = searched user ratings.

## Evaluation (4)

### Modification Time Stamps - Collaborative Filtering

	<b>D1</b>	<b>D2</b>	<b>D3</b>	<b>D4</b>	<b>D5</b>
<b>D1</b>	1	0,33	0,49	0,60	0,93
<b>D2</b>	0,33	1	0,85	0,92	0,29
<b>D3</b>	0,49	0,85	1	0,78	0,37
<b>D4</b>	0,60	0,92	0,78	1	0,57
<b>D5</b>	0,93	0,29	0,37	0,57	1

Table: Similarity Matrix

## Evaluation (5)

### Modification Time Stamps - Collaborative Filtering

	D1	D2	D3	D4	D5
<b>Brian</b>	1	5	4	3	2
<b>John</b>	2	4	4	3	3
<b>Luke</b>	1	3	1,98	2	1
<b>Robert</b>	5	2,07	2,38	2	5
<b>Tom</b>	4	1,63	2	1,98	2,87

Table: Rating Predictions per User

# Evaluation (6)

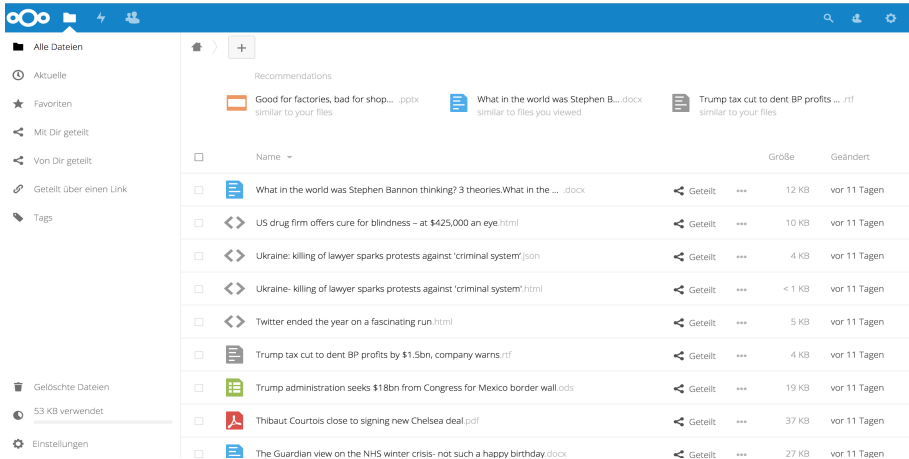
## Modification Time Stamps - Hybridization

	D1	D2	D3	D4	D5
<b>Brian</b>					
<b>John</b>					
<b>Luke</b>			1,42		
<b>Robert</b>		1,46	1,52		
<b>Tom</b>		1,63		1,42	1,76

Table: Hybridization Results

# Evaluation (7)

## Screenshots



The screenshot shows the OneDrive web interface. On the left is a navigation pane with the following items:

- Alle Dateien
- Aktuelle
- Favoriten
- Mit Dir geteilt
- Von Dir geteilt
- Geteilt über einen Link
- Tags
- Gelöschte Dateien
- 53 KB verwendet
- Einstellungen

The main area displays a list of files and folders. At the top, there are recommendations:

- Good for factories, bad for shop... .pptx similar to your files
- What in the world was Stephen B... .docx similar to files you viewed
- Trump tax cut to dent BP profits ... .rtf similar to your files

Below the recommendations is a table of files:

<input type="checkbox"/>	Name		Größe	Geändert
<input type="checkbox"/>	What in the world was Stephen Bann... .docx	Geteilt	12 KB	vor 11 Tagen
<input type="checkbox"/>	US drug firm offers cure for blindness - at \$425,000 an eye.html	Geteilt	10 KB	vor 11 Tagen
<input type="checkbox"/>	Ukraine: killing of lawyer sparks protests against 'criminal system'.json	Geteilt	4 KB	vor 11 Tagen
<input type="checkbox"/>	Ukraine- killing of lawyer sparks protests against 'criminal system'.html	Geteilt	< 1 KB	vor 11 Tagen
<input type="checkbox"/>	Twitter ended the year on a fascinating run.html	Geteilt	5 KB	vor 11 Tagen
<input type="checkbox"/>	Trump tax cut to dent BP profits by \$1.5bn, company warns.rtf	Geteilt	4 KB	vor 11 Tagen
<input type="checkbox"/>	Trump administration seeks \$18bn from Congress for Mexico border wall.ods	Geteilt	19 KB	vor 11 Tagen
<input type="checkbox"/>	Thibaut Courtois close to signing new Chelsea deal.pdf	Geteilt	37 KB	vor 11 Tagen
<input type="checkbox"/>	The Guardian view on the NHS winter crisis- not such a happy birthday.docx	Geteilt	27 KB	vor 11 Tagen



# Conclusion (1)

## Summary

- Further development of the app is planned.
- Software and thesis release on GitHub (@doganoo) and my personal website ([www.dogan-ucar.de](http://www.dogan-ucar.de)) as soon as possible.
- challenges during this thesis:
  - Programming language PHP.
  - Machine Learning as a part of a web application.
  - no (official) PHP API's for Recommendation Systems.

## Conclusion (2)

### Results

- Last modification time stamps has proven to be more suitable
  - user is not required to make manual steps (tag as favorite).
  - rating range 0 to 5 provides more accuracy.
- Effectiveness not measurable within this work, as:
  - lack of test and training data.
  - benefit for users is a subjective measure.

## Conclusion (3)

### Future Work

- Machine learning based learning models in order to define weights and thresholds.
- Using user behaviour as input for the recommendation system.
- Custom keywords for user profiles (Content-Based Recommendation).
- Defining other types of "Content" provided by Nextcloud (tags, comments, sharing, activity).

**Thank you for your  
Attention**