Alessandro Di Patria

Beer Reccomandation System

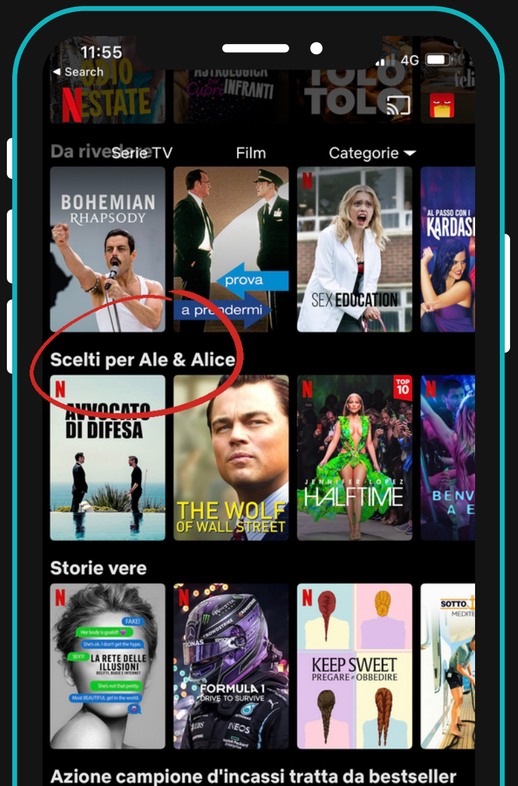
Indice

What is a reccomender system ?

Tool that make recommendations for the users to help them in their choices

Content Based Filtering

- Makes suggestions based on description of the item and a profile of the user's preferences.



Collaborative filtering

- Gives recommendations using only information about rating profiles for different users or items (behaviour of people)

Project Goals

DATA ANALISYS

Analisis of the single features, data sparcity, statistics

RECOMMENDER SYSTEM MULTI APPROACH

Build recommender system with different approach, ALS and SVD for Collaborative filtering. Cosine similarty for Content based

WEB RECCOMANDER SYSTEM

Building a web app with python and streamlit using content based model

Beer Reviews Dataset

- More then 2 millions of reviews
- 99% of sparsity
- Clean dataset, only a few null values

Relevant features

- beer_id, ratings, userID
- Different Ratings based on characteristic of a beer
- ABV (alcolic grade)
- Beer_style (IPA, Double Malt...)

Dataset Exploration

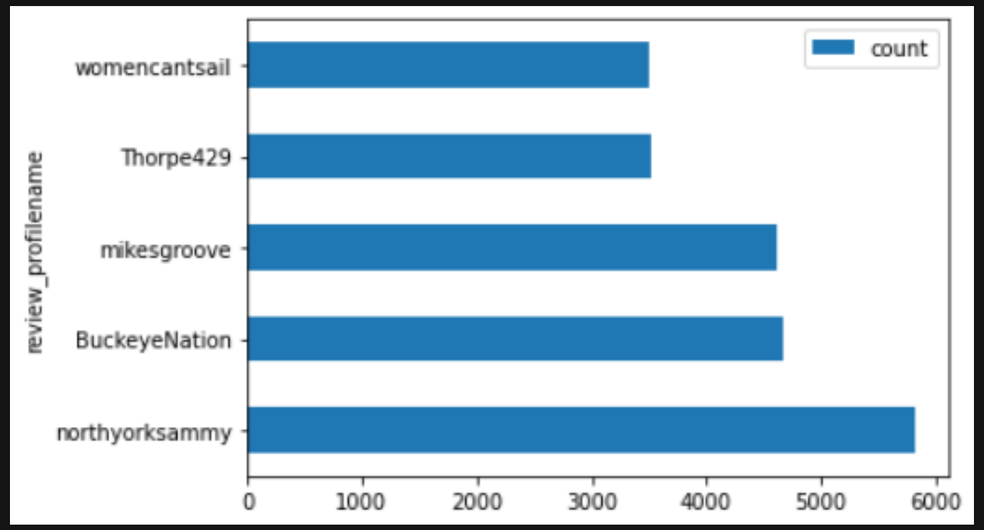
Data Relevant statistic（most popular, most strong beers...）

Correlations between overall reviews and other types of reviews

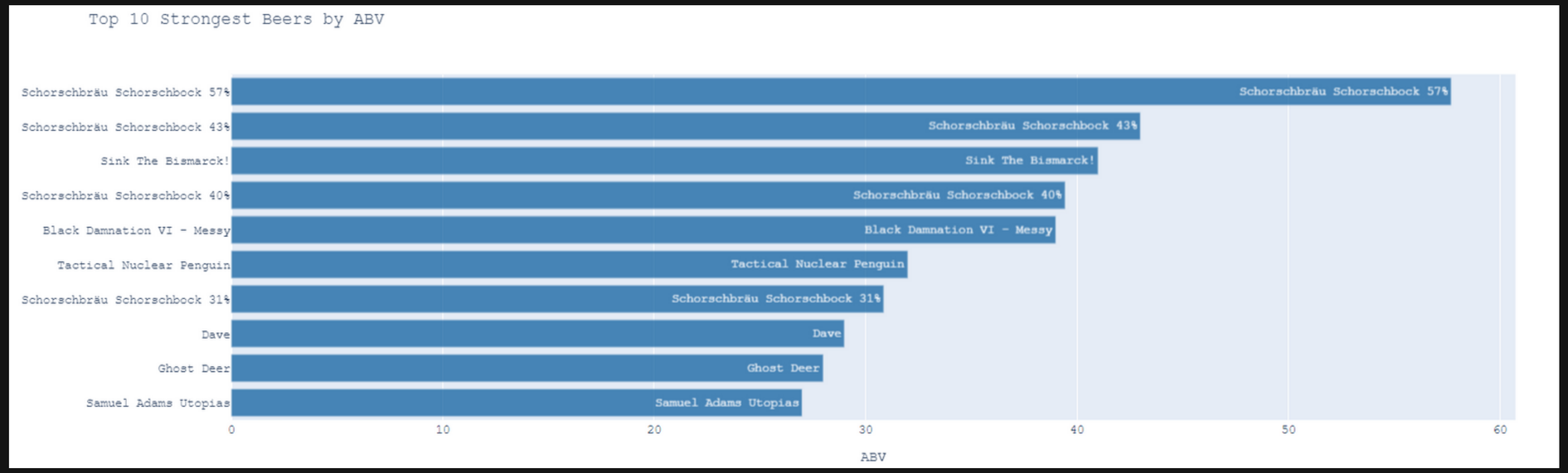
Dataset Exploration

Most popular Beers

Most Active Users



Dataset Exploration



Dataset Exploration

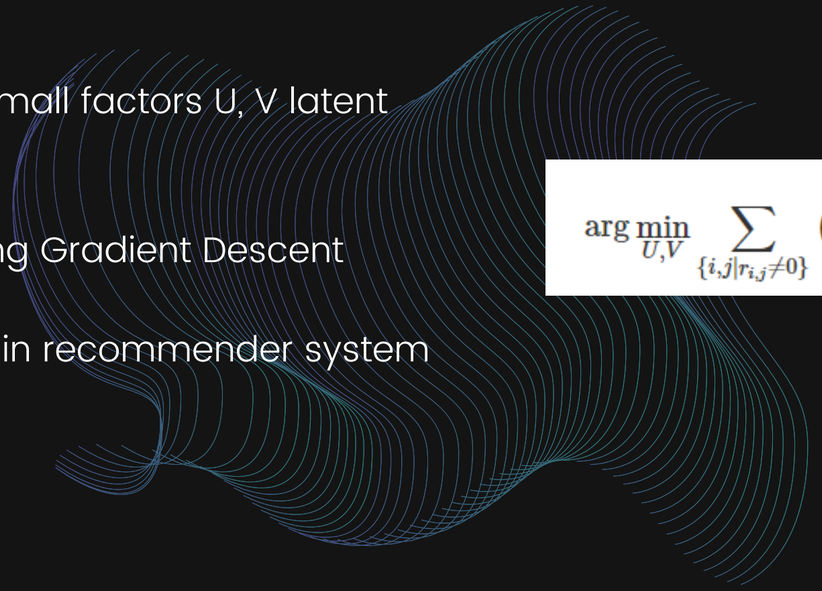
- Analysis of the taste, appereance, aroma, palate review
- I have made their correlations with correlation matrix
- Correlation matrix useful to summarize a dataset and to identify and visualize patterns in the given data.
- With 0,78 beer_taste seems the best factor

Which rating is more important to determine the overall quality of a beers ?

	review_appearance	review_aroma	review_palate	review_taste	review_overall
review_appearance	1.000000	0.561029	0.566634	0.546980	0.501732
review_aroma	0.561029	1.000000	0.616947	0.716776	0.616013
review_palate	0.566634	0.616947	1.000000	0.734135	0.701914
review_taste	0.546980	0.716776	0.734135	1.000000	0.789816
review_overall	0.501732	0.616013	0.701914	0.789816	1.000000

Alternating Least of Square

- Factorize given matrix R(ratings) in two small factors U, V latent vectors
- ALS aims to minimize loss function applyng Gradient Descent
- Resolve problems for sparse matrix used in recommender system



$$\arg\min_{U,V} \sum_{\{i,j|r_{i,j}\neq 0\}} \left(r_{i,j} - u_i^T v_j\right)^2 + \lambda \left(\sum_i n_{u_i} \|u_i\|^2 + \sum_j n_{v_j} \|v_j\|^2\right)$$

Implementation

Pre processing

Drop Null score values

1

Drop null/none values

Drop duplicates

2

Drop duplicates review

String Indexer

3

String indexer to turn review_profilename to numerical id

Split Dataset

4

Split dataset into 20% test and 80% train

Implementation

Training

- Train The model
- Hyperparameter tuning using k-cross Validation :
  - Rank 10-25
  - regParam 0,1-1
  - maxIter 10
- Use RMSE to evaluate and choose the best set of hyperparameters

Evaluation

RMSE 0,61

- In RMSE the deviations are squqred to prevent positive and negative values from cancelling each other out.
- RMSE = $\sqrt{\Sigma(\hat{y}i - yi)2 / n}$

MAE 0,47

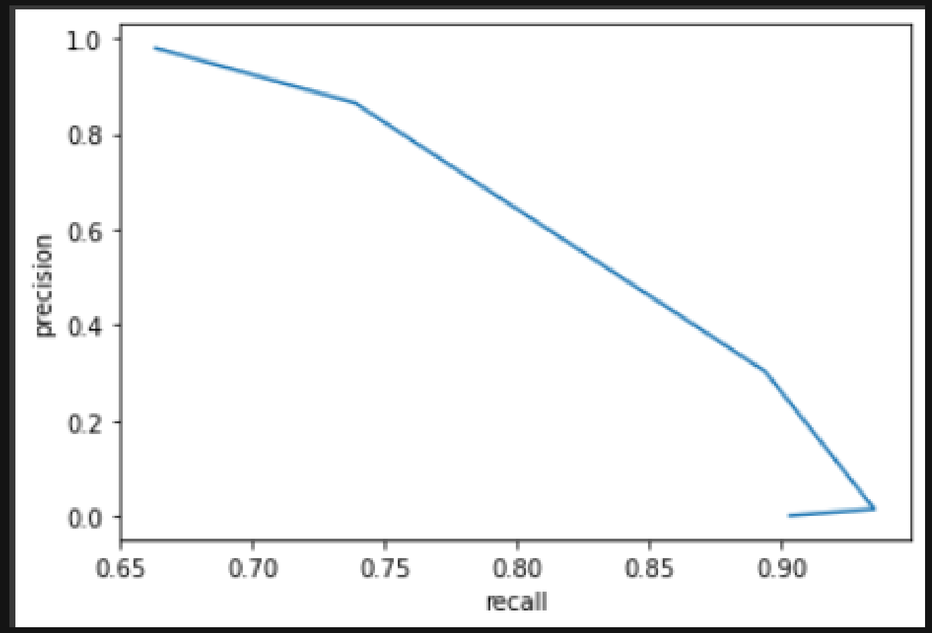
- Error = Actual observation – predicted observation
- Treating the positive and negative errors observed as absolute.
- Adding all the error and divide for the number of observations

MSE 0,38

- MSE is the average squared distance between the observed and predicted values.
- MSE = $\Sigma(\hat{y}i - yi)2 / n$

Precision & Recall

- We have classified all score as relevant or irrelevant
- Threshold of ratings = 3, while I'll change the treshold for prediction from 3-5
- If both parameters are above the threshold, true positive while if not TN and so on
- The graph shows a good result



Single Value Decomposition

- Factorize given matrix A(ratings) in UsV where U and V rapresent the items /ratings
- Singular-Value Decomposition is a matrix decomposition method for reducing a matrix to its constituent parts in order to make certain subsequent matrix calculations simpler.
- The SVD has some problems. It is not scalable and it suffer of missing values (sparcity).
- Surprise SVD Library to build the model and make predicitons

Single Value Decomposition

Training

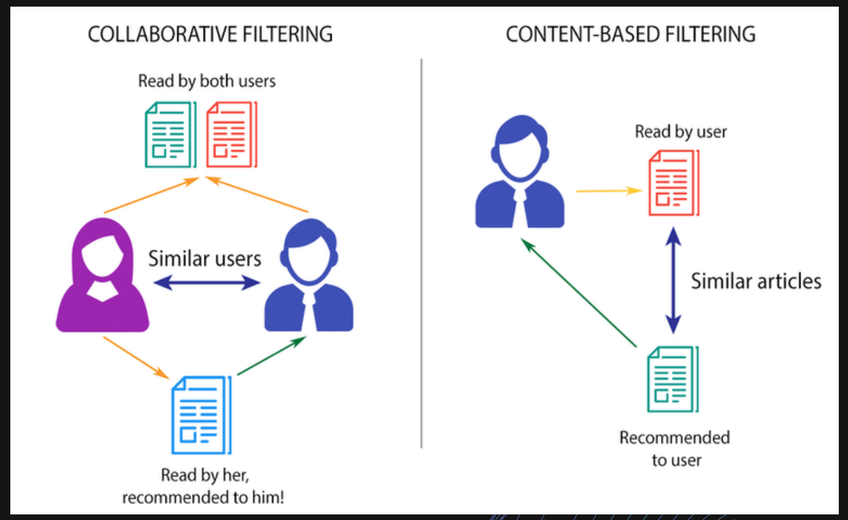
- Cross Validation (5-folds)
- different models was compered with RMSE and MAE

Evaluation

- RMSE for the best model 0,64
- MAE for the best model 0,48

Content Based Filtering

- In this approach I'll use some properties and the metadata of a particular item to suggest other items with similar characteristics.
- In this section I'll calculate pairwise cosine similarity scores for all beers based on their Beer_style and Beer_ABV
- I'm also using Natural Language processing to understand and suggest similar movies to a user's input.



Implementation

Pre processing

Drop Null score values and duplicates

1

Drop null values and delete duplicates review

Striping and merging beer_style e ABV

2

We are merging two relevant feature of the item in one single feature description to increase the accuracy of recongnition

Vectorize the words merged

3

Term-Frequency-Inverse Document that give a matrix where each column represents a word in the description vocabulary and each row represents a Beer.

Implementation

Cosine Similarity

- Cosine similarity is a mathematical computation that tells us the similarity between two vectors A and B in this case our beers.

- The main problem with using cosine similarity is the computation between each possible pair of movies. If we have 1000 beers , then we need to perform 1000000 beers. So I will decrese my dataset.

- this model don't suffer of Could Start problem because is based on item profile.

	0	1	2	3	4	5
0	1.00	0.57	0.51	0.26	0.31	0.33
1	0.57	1.00	0.54	0.25	0.31	0.43
2	0.51	0.54	1.00	0.19	0.25	0.36
3	0.26	0.25	0.19	1.00	0.50	0.38
4	0.31	0.31	0.25	0.50	1.00	0.56
5	0.33	0.43	0.36	0.38	0.56	1.00

Implementation

A possible evaluation Precision and Recall

- RMSE work only with prediction rates so i think about using precision and recall to evaluate your recommendations.
- Take one of the most favorite beers by one user.
- Take 30 suggested beers by one user and see if this user have already drunk these beers If both thinks appened True Positive, if a beer suggested is not in the list of viewed is a false positive. While if a rated beer is not in suggested is a false negative and so on....

Web App

- Coded in Python using streamlit library.
- Content based model to avoid could start problem
- Select a beer_name with the selectbox and click show to recommend a beer.