Movie recommendation project

The goal of this project is to apply all material learned in INF161 and sucessfully complete a data science project. Please read the description carefully!

This project is a compulsory part of the course. This project contributes 40% to the final grade. The grade will be based on good choice of methods, correctness of answers, clarity of code and thoroughness and clarity of reporting.

Requirements

You will build a recommender system to recommend movies. The system takes as its input some user data together with movie scores given by users and predicts the missing scores, i.e., the score each user would give to each movie they have not yet rated. The recommender system you build will consist of four subsystems:

- Data cleanup (40 pts):
  - Input: raw data
  - Output: clean data
  - Features: This systems takes the data from folder `raw_data` and generates a folder `clean_data` consisting of csv files formated the same way as the data from the folder `sample_data`.
- Modelling and prediction (40 pts):
  - Input: clean data
  - Output: machine learning model, expected generalisation RMSE
  - Features: This system takes the data from either `sample_data` or `clean_data` and builds a machine learning model for predicting movie scores. Model selection is an important part of this system. You should evaluate at least 2 fundamentally different modeling approaches before selecting the final model. We evaluate the performance of the system by comparing the predicted ratings with the known ratings on a validation/test data set. Specifically, the system should be evaluated with the root mean squared error (RMSE) of predictions, i.e.

    $$\sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_{ij} - y_{ij})^2}{N}},$$

    where N is the number of predictions, $\hat{y}_{ij}$ is the i-th prediction of user j and y_{ij} ist the corresponding true rating. The sum runs over the set of known ratings in the validation/test data set. The system should report the expected generalisation RMSE.
- Prediction (10 pts):
  - Input: machine learning model and new data
  - Output: prediction of movie scores
  - Features: Given new users data, including user data and ranking data formated as the `sample_data`, this system should return predictions of the rating on a scale from 1 to 5 that each user would have given to every movie if they had rated it. This includes predictions for the movies that user has already rated.

- Website (10 pts):
  - Features: The website should allow users to enter personal data and some film scores. Then the website should interact with the prediction system to return the 10 movies with the highest predicted scores that the user has not seen yet. Note that this is a HTML document that exists on your personal computer that you open with your browser and not a website hosted on the internet.

Deadlines

The project consists of three parts with three distinct deadlines. In the first part you will build and evaluate a recommendation system based on a sample of cleaned data. In the second part of the project you will get new raw data and work on cleaning up the data to get it ready to use for the recommendation system built in the first part. The raw data will contain many more users and ratings than the sample for the first part, so the cleaned data will be different from the sample data, but should use the same format. And the last part of the project consists of creating a simple website that runs your movie recommendation system.

- Deadlines:
  - Part 1: Sunday, 20.09, 23.59
  - Parts 1&2: Sunday, 11.10, 23.59
  - Complete project: Sunday, 30.10, 23.59
- Deliver here MittUIB.no/assignments

Deliverables

All three dealdines are mandatory. For the first two deadlines that include parts I and II it suffices to send a report that describes what you have done, why you have done so and what your results were. These reports will not be graded, but you will receive feedback that will improve your final project.

You may in addition to the report submit the preliminary versions of your jupyter notebooks `cleanup.ipynb` and `model.ipynb` for parts I and II respectively (see below). This is optional, but it will allow us to give you more detailed feedback on your progress.

For the final submission, please provide the following:

- A jupyter notebook `cleanup.ipynb` for data cleanup, generating a new folder `cleaned_data` with the same structure as the `sample_data` folder. The notebook acts as both report and submitted code. It should contain all the code to reproduce your work and a report of all your methodological choices and results. Please “restart and run all” before submission, so that you submit a clean version.
- A jupyter notebook `model.ipynb` for modelling, generating a machine learning model that is saved to disk. The notebook acts as both report and submitted code. It should contain all the code to reproduce your work and a report of all your methodological choices and results. Please “restart and run all” before submission, so that you submit a clean version.
- A jupyter notebook `predict.ipynb` for prediction, loading a machine learning model and user data and outputing predictions. The notebook acts as both report and submitted code. It should contain all the code to reproduce your work and a report of all your methodological choices and results. Please “restart and run all” before submission, so that you submit a clean version.
- A file `predictions.csv` with predictions for each user and each movie. The file should have columns ’FilmID’ and then one column per ’BrukerID’ with the ’BrukerID’ as column name. The first colum should contain the film ids and the remaining columns should contain predicted user ratings for that film. All data should be between 1 and 5 with no missing values.

- A zip file that contains the file `app.py` for running a local movie recommendation website and all neccesary files such that `app.py` runs with the command `python3 app.py`. The website should then run at `localhost:8080/`.

Note that each notebook and the app need to run independently.

In addition to packages from the standard library, you may use the following python packages: `xlrd`, `numpy`, `pandas`, `scipy`, `sklearn`, `matplotlib`, `seaborn`, `requests`, `plotly`, `flask`, `django`, `waitress`. If you use any other packages we will not be able to run your app and you will fail the project.

Code should be documented and tricks (e.g. to avoid division by zero, to make sure it takes finite time to run, etc.) should be reported. The rational behind all steps in the code should be clear from the report.

NOTE: This project is a learning experience. If we see that you have copied your answers from online resources, you will get 0 points. This is an application project, that means you may use any freely available library for the application tasks of the project.

Model selection is an important part of the task and will be graded accordingly. Before applying machine learning algorithms, you should always consider (and report) what results you expect. When you have successfully applied machine learning algorithms, you should always comment on how well the results match your expectations.

Data for this project has been adapted from MovieLens.

Leaderboard

There will be a leaderboard. TBD.

Suggestions

Here is a list of suggested activities for each of the parts. Following this list should help you to get a good system that fullfills all requirements. It is not a necessary that you complete each step on this list, as long as your system fullfills all requirements.

Part I

1. Take the data in folder `sample_data` and read the README. This is a clean dataset that can be used directly for a recommender system.
2. Have a look at your data and give an overview over basic statistics that you consider to be important. At least have a look at the number of entries in each data set, an overview over missing data, some measure of location and spread of the features. Justify your choices of measures used.
3. Visualize the data using appropriate visualization techniques. You should have at least 3 visualizations of features that you consider to be relevant for making good predictions (e.g., a histogram over all scores given to movies by movie genre).
4. Split your data appropriately into train, validation and test sets for proper model selection and evaluation.
5. Generate a baseline model to compare your recommender system with. Consider the simplest way you would make predictions without personalizing the predictions.
6. Use content-based filtering, collaborative filtering, and a combined approach to model movie ratings.
7. Compare these approaches to each other.
8. Pick the best model and compute its root mean squared error from the test data.

Part II

1. Take the data in folder `raw_data` and read the README. This dataset is much larger than the dataset from part I, but has not been cleaned up.
2. Read all datasets.
3. Find errors and missing values in datasets and fix those errors.
4. Prepare the dataset for analysis, such that the structure is the same as for the `sample_data`.
5. Redo the analysis from part I with the full dataset.
6. What is the best model now and what is its root mean squared error?
7. Engineer at least 2 new features to the dataset. This may be a combination of genres that you consider to be similar or dividing the timestamp of the ratings into day and night. Whatever feature you engineer, you should explain why you think it may help you make better predictions.
8. Describe how the new features changed your estimated generalization root mean squared error.

Part III

1. Adjust the `index.html` from the 'data science implementasjon' lecture to allow to insert some personal data and movie ratings.
2. Using the model developed in task II, generate a simple website where users can insert some personal data and movie ratings and receive a list of 10 movies that may interest them.
3. Create a beautiful website with nice interface and animations (only do this when you have finished all other tasks, you will not get points for this).