The **Dockerfile** and **Scraper.py** are included in the PyCharm project. You can either build the image from these files, or use the pre-built compressed image **myscraper.tar.gz** included in the project. To install the image on your local Docker, run the following command:

docker load -i myscraper.tar.gz  
Once the image is installed locally, you can run the scraper with:

docker run --rm -v C:/LyricsTextMining/Songs:/songs myscraper python Scraper.py --start\_page 1 --end\_page 100 --genre grunge --output 50\_100\_grunge.csv

**Where:**

* **C:/LyricsTextMining/Songs** → The location where the CSV file will be saved.
* **-- start\_page / -- end\_page** → The range of website pages you want to crawl.
* **-- genre** → Optional filter by music genre.
* **-- output** → The name of the output CSV file.

Part1\_Crawling.ipynb connects to the website, crawls the songs, and saves them into a CSV file.  
That CSV can be generated either from the Docker container or directly from Jupyter.

The CSV should then be imported into an SQL database called TextMiningHa, which contains a single table named songs. A backup of this database is also included in the PyCharm project.

Note: All functions in the Jupyter notebooks pull their data from this database. If you want the project to run on your local machine, you will need to update the database credentials in each notebook accordingly

Just change the credential SERVER=IVAN\_PC\SQLEXPRESS to match the SQL server on your local machine.

**Part2\_PreProcessing.ipynb handles preprocessing tasks, including text cleaning and feature analysis.**

**Part3.1\_Clustering&Evaluation.ipynb** – applies three clustering methods, then visualizes and evaluates the results for each clustering method.

**Part3.3.1\_SoftGridSearch.ipynb** – performs grid search optimization for the soft clustering method.

**Part3.3.2\_HardGridSearch.ipynb** – performs grid search optimization for the hard clustering method.

Part3.3.3\_LDAGridSearch.ipynb – performs grid search optimization for the LDA method.

Part4.1\_SimilaritySearch.ipynb – implements similarity search methods to compare and retrieve relevant songs.

Part4.2\_TitlevsLyrics.ipynb – analyzes the relationship between song titles and their corresponding lyrics.

Note: Each cell in all Jupyter notebooks is documented with a Markdown cell describing its functionality, and all code is commented accordingly.  
In addition, in the end of each notebook includes a Markdown cell with conclusions and final thoughts.