## Expert Systems

Expert systems where one of the earliest branches of artificial intelligence developed back in the 70s. The motivation was (is) to capture the thought process of human experts and mimic their decision-making ability. Expert systems are still in use to this day by various industries with varying sentiment towards their return on investment towards the company (Simin Samadi Soufi et. al 2013) .

An expert system is made of facts that allow inference through a set of rules. Facts define the knowledge body encompassed by the specific domain covered. The rules allow conditional and fuzzy logic to filter which facts apply or not. The rules can be prioritized (usually priority is presented by order) to allow more generic rules not to override specific rules.

Different sources differ on the full definition of an expert system, the consensus is that expert systems are made of the following four components (KMS 2003):

**A Knowledge Base**: This is the 'database' of knowledge encompassed by the expert system. It can contain the facts, the rules and other important information to be able to infer decisions.

**An Inference Engine**: The execution part that allows for the manipulation of the information provided in the knowledge base to extract decisions.

**A User Interface**: A portal that allows humans to interact with the expert system. This could be a web interface, a command line or a rich client. It depends usually on the company implementation.

**A Knowledge acquisition system**: Allows the expert system to stay up to date with new knowledge. This might be through manual updates or automated knowledge extraction.

Furthermore, some sources expand further to include:

**An Explanation Facility**: This allows the user to be able to analyze the output; what led to this decision. This leads to better agreement and allows auditing facilities to review any decisions made.

**A Knowledge refinement system**: Like a feedback look, this system is responsible to analyze the results and improve the system itself based on outcome. This usually comes from human expert feedback or automated key performance indicators. (Meseguer, P. and Verdaguer, A)

### Expert Systems in Business

In a paper titled "An Analysis of Expert Systems for Business Decision Making at Different Levels and in Different Roles" the author examines the use of expert systems in business. The paper describes a large number of different studies published between 1984 and 2002 and tallies which levels and roles where exposed to an expert system. The study concludes that in general expert systems where utilized at managerial levels with little or no application in executive levels. Furthermore the areas of business in which expert systems are mostly utilized are finance, accounting, marketing and production.

## Adopting an Expert System

Expert systems are complicated and to adopt such system is not trivial. Similar to any software development lifecycle, several steps need to be taken into consideration.

### Problem Analysis

The first step like any other software is to define properly the problem to be solved. During this stage several roles need to participate together to build an analysis of the requirements for the system. The most important roles are that of the domain expert, who can describe the use cases and a knowledge engineer who can write the problem description.

Once the problem has been properly defined and the required knowledge gathered the implementation phase takes place.

### Exploratory Data Analysis

This process involves analyzing and summarizing data to better understand any relationships that may exist. Exploratory Data Analysis provides a preliminary understanding of the data, which can be used to develop and refine the expert system's rules and decision-making processes. This is a discipline on its own and the most common steps are:

**Data Collection**: Without data sourcing it is impossible to start. Data may come from various sources in disparate formats, frequencies, and quality.

**Understanding the Dataset**: The first step is to understand the data available; this can be done by visual inspection, summarizing the data or using statistical methods to gain any valuable insights.

**Cleaning the Dataset**: Data is not perfect, it might contain outliers, missing data, null values, wrong values. Part of the outcome of analyzing the data should be to identify any steps required to clean the data.

**Identifying Correlations**: This is especially true when working with multiple data sources but not necessarily.  The same dataset might contain various correlations.

**Choosing the right methods**: Whilst the data might be cleaned and correlated, different values might be in different scales. For example, a dataset about houses might contain the size of the house and the price both on different numeric scales. Choosing the proper normalization methods is important to be able to work with the data, especially in an AI scenario.

### Knowledge base implementation

Once the analysis of the problem is defined and the data is cleaned and normalized the next step is to import this information into the knowledge base of the expert system. Again this requires several steps, mainly:

**Convert the data into Facts**: Depending on the software used the data needs to be converted into possible different fact types. For example if the dataset is about identifying plants. There might be distinct facts about habitats of plants and about characteristics of plants such as shape and form.

**Writing the rule-base**: The problem description obtained in the first step is transformed into rules based on the model of the facts obtained in the previous step. Rules might depend on each other creating an inter relationship. Some facts might require higher priority and will require this to be implicitly specified (for example clips uses the salience property).

## Creating the user interface

Once the system is in place, a user interface needs to be delivered in order for users to be able to interact with it. The usability of the user interface will most likely be a determine factor in the success of the project. At this stage the medium for the delivery of the interface needs to be decided (this could be any combination of mobile, web or tick client)

Design: Dedicated designers might be involved to provide usability guidance in this step.

Implementation: Depending on the medium for delivery, various specialists might be involved in this step. The user interface needs to interact with the backend system as well and various options exist in this area.

Testing

During this phase the expert system is tested to ensure proper functionality. This requires again various steps from quality assurance testing to end user testing.

## Keeping Up to Date

Data changes with time, and an expert system must keep up to date with new facts. Knowledge acquisition involves various techniques but in general boils down back to the data sourcing.

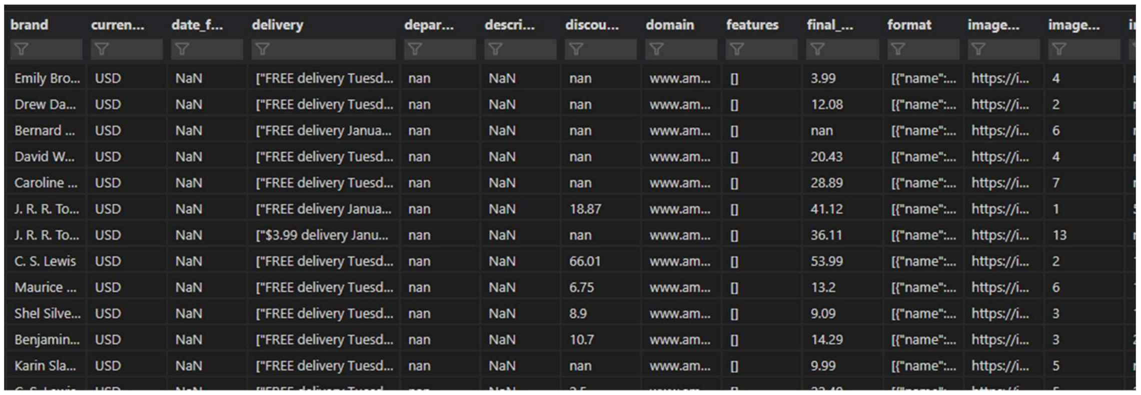
## Assignment Dataset

Following the example provided for the assignment, after importing the data; some of the columns have no relevant data, some contain JSON formatted data, while some contain partial data. For example, date published was only available for 3 rows out of the entire dataset.

Importing the data is done in a very simple through the pandas library,  by providing the file, or an immediate url as demonstrated below:

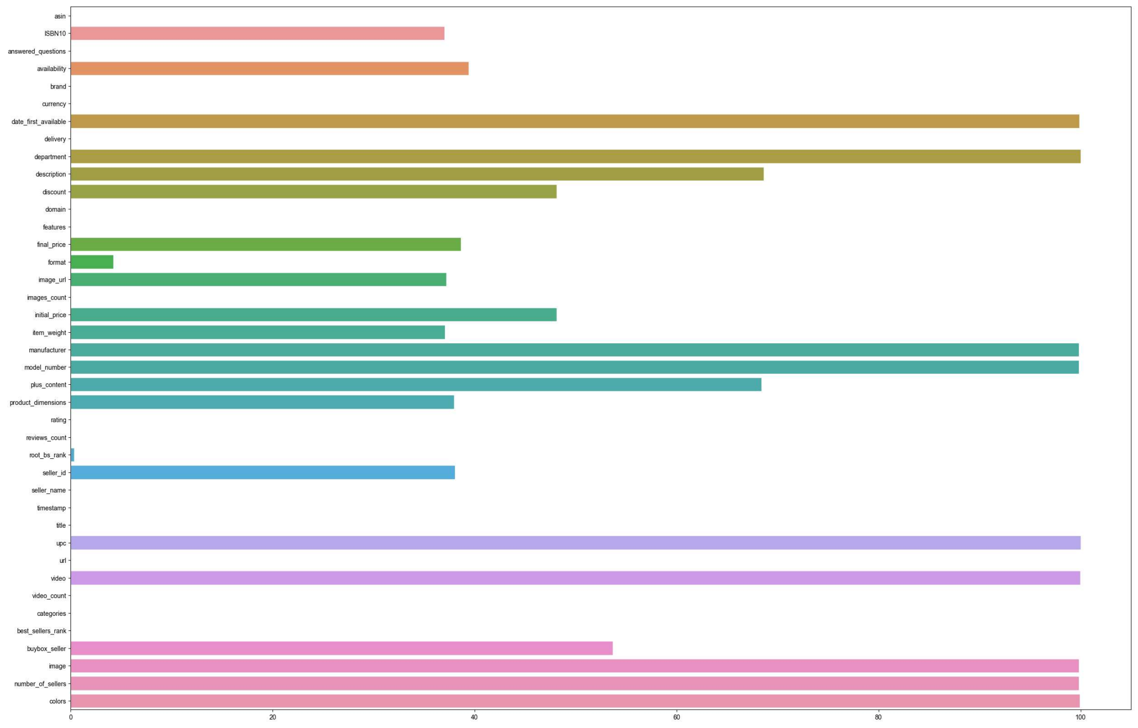
```python
baseds = pd.read_csv('https://github.com/luminati-io/Amazon-popular-books-dataset/blob/main/Amazon_popular_books_dataset.csv?raw=true')

print(baseds.shape)

baseds.head(baseds.shape[0])
```

(Note I use head(shape[0]) as this works like getting both the first portion of the dataset and the bottom portion, giving you better visibility). The data can be further analyzed in the data explorer as per figure below. As can be seen in the figure above the dataset starts with 2269 rows and 40 columns.



| brand | curren... | date_f... | delivery | depar... | descri... | discou... | domain | features | final_... | format | image... | image... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Emily Bro... | USD | NaN | ["FREE delivery Tuesd... | nan | NaN | nan | www.am... | [] | 3.99 | [{"name":... | https://i... | 4 |
| Drew Da... | USD | NaN | ["FREE delivery Tuesd... | nan | NaN | nan | www.am... | [] | 12.08 | [{"name":... | https://i... | 2 |
| Bernard ... | USD | NaN | ["FREE delivery Janua... | nan | NaN | nan | www.am... | [] | nan | [{"name":... | https://i... | 6 |
| David W... | USD | NaN | ["FREE delivery Tuesd... | nan | NaN | nan | www.am... | [] | 20.43 | [{"name":... | https://i... | 4 |
| Caroline ... | USD | NaN | ["FREE delivery Tuesd... | nan | NaN | nan | www.am... | [] | 28.89 | [{"name":... | https://i... | 7 |
| J. R. R. To... | USD | NaN | ["FREE delivery Janua... | nan | NaN | 18.87 | www.am... | [] | 41.12 | [{"name":... | https://i... | 1 |
| J. R. R. To... | USD | NaN | ["$3.99 delivery Janu... | nan | NaN | nan | www.am... | [] | 36.11 | [{"name":... | https://i... | 13 |
| C. S. Lewis | USD | NaN | ["FREE delivery Tuesd... | nan | NaN | 66.01 | www.am... | [] | 53.99 | [{"name":... | https://i... | 2 |
| Maurice ... | USD | NaN | ["FREE delivery Tuesd... | nan | NaN | 6.75 | www.am... | [] | 13.2 | [{"name":... | https://i... | 6 |
| Shel Silve... | USD | NaN | ["FREE delivery Tuesd... | nan | NaN | 8.9 | www.am... | [] | 9.09 | [{"name":... | https://i... | 3 |
| Benjamin... | USD | NaN | ["FREE delivery Tuesd... | nan | NaN | 10.7 | www.am... | [] | 14.29 | [{"name":... | https://i... | 3 |
| Karin Sla... | USD | NaN | ["FREE delivery Tuesd... | nan | NaN | nan | www.am... | [] | 9.99 | [{"name":... | https://i... | 5 |

After analyzing the fill rate of columns using a bar chart, which represents the percentage missing for each column. It can be clearly seen that while ASIN was 100% filled (had no missing data) the isbn10 had 30% of the data missing, thus is less reliable.



In order to generate the graph seaborn barplot  is used and the count of na fields is obtained through pandas

```
dataset_stats = dict()
#the inverse of the row count, to compute the % na
r = 1 / baseds.shape[0]

#go through each column, and get the % of fields that are na. since this is
(0-1) mutiply by 100 to make it more human
for column in baseds.columns:
    dataset_stats[column] = r * baseds[column].isna().sum() * 100

#some basic settings to make the plot visible.
fig, ax = plt.subplots(figsize=(30, 20))
sns.set(font_scale=1)

#sns doesn't like array seems to prefer lists.
sns.barplot(y=list(dataset_stats.keys()), x=list(dataset_stats.values()),
orient = "h")
```

While some other columns where completely missing any data and could be dropped as provided no value.

```
#drop the columns we don't need, since there are quite a few,
#this is split on multiple lines to make it more readable.
baseds.drop(['answered_questions','currency','date_first_available',
'department', 'domain','features'], inplace=True, axis=1)
```

```
baseds.drop(['manufacturer','model_number','plus_content','product_dimensions'
, 'upc', 'video','video_count'], inplace=True, axis=1)
baseds.drop(['buybox_seller','image','number_of_sellers','colors',
'description', 'delivery'], inplace=True, axis=1)
```

Following this step, two other cleanups where required. The first one was based on an assumption that the user is actually buying a book and only wants to see those books in stock (if this is not the case the line can be commented out.). The second checks that there are no empty cell values in the '*best_seller_rank*' column, since this the main column the rule engine will use. Another check could have been to validate the json in that column, however this is done during fact loading.

```
rowindex = baseds[baseds['availability'] != 'In Stock.'].index
baseds.drop(rowindex, axis=0, inplace=True)

#one of the rows has this cell empty.
baseds.dropna(subset=['best_sellers_rank'], inplace=True)

#cleanup the memory
del rowindex

baseds.shape
```

Knowledge and Inference

To be able to work an expert system requires to transform the underlying dataset to facts. Facts are then used by the inference subsystem which would apply rules to draw conclusions. This implies that the more accurate facts are the more accurate will be the decisions.

Clips accepts facts as strings. Therefore, the data needs to be transformed into a format acceptable by the system.

```
def get_list_as_clips(items):

    result = ""
    for i in items:
        result += f' "{cleanstr(i)}"'

    return result

def cleanstr(s):
    return re.sub("['\"]", "", s)
```

While a simple string formatter is adequate care needs to be taken for reserved clips characters. In the assignment, rudimentary helper functions have been provided to manage this. While these can be improved to manage all cases and provide escape replacements, this implementation was adequate for this use case.

Each book can be in multiple categories, and this is represented as different facts in clips. A helper method is provided to convert the pandas row into a multiple clips facts.

```
#helper function to convert a pandas row to a fact.
#unforunatly i could not manage to find a more efficient way.
#there is a way to convert to dictionary and load the dictionary inside clips
#however this requires to initiaze the fact too, which made it impossible.
def to_clips_fact(env, r):

    #parse the json
    jcategories = json.loads(r["categories"])
    #all categories have books category.
    if ('Books' in jcategories): jcategories.remove('Books')
    #clean special chars
    jcategories = [cleanstr(item) for item in jcategories]

    jrankcategorie = json.loads(r["best_sellers_rank"])

    #each row rapresents a book, and each book might be in multiple category
ranks
    #this will create a distinct fact per rank.
    for n in jrankcategorie:
        rcat = n['category'].split(' / ')

        #clean any chars that might break clips.
        rcat = [cleanstr(item) for item in rcat]

        #again rank category have books in all of them.
        if ('Books' in rcat): rcat.remove('Books')

        factstr = f'(amazonbook (asin "{r["asin"]}") (ISBN10 "{r["ISBN10"]}")
(title "{cleanstr(r["title"])}")  (categories
{get_list_as_clips(jcategories)}) (rankcategories {get_list_as_clips(rcat)})
(rank {n["rank"]}) )'
        #print(factstr) #for debugging issues..
        env.assert_string(factstr)
```

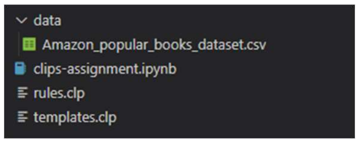
The helper function also checks if the Books super category is present, and if so, removes it.

```
for _, row in baseds.iterrows():
    to_clips_fact(environment,
row)
```

Finally the whole dataset is enumerated through using a simple for loop on the rows and each row calls the helper method to convert it to and feed it to clips.

## Data Model

The raw data is loaded into pandas, and following the data analysis, redundant columns are removed. Furthermore, books not in stock are removed. This was an assumption made that the recommender is trying to sell books. Should this not be required, it can be easily turned off. The outcome of this step reduces the data from 2269x40 to 1249x21 rows and columns.



In the next part, clips source files are loaded. The templates and rules are kept in separate files. This makes it easier to sperate concerns, and if the project was bigger more separation could be done to improve maintainability.

An iteration is done on all the rows in the data source, and each row is transformed into multiple facts (named amazonbook). This is due to the JSON data in the '*best_sellers_rank'* field, since a book might be in more than one category.

$$\begin{tikzcd}
 & \boxed{\text{Category: Science Fiction}} \arrow[r] & \boxed{\text{Fact: The Silmarillion In Science Fiction}} \\
\boxed{\text{Book: The Silmarillion}} \arrow[ur] \arrow[dr] & & \\
 & \boxed{\text{Category: Fantasy}} \arrow[r] & \boxed{\text{Fact: The Silmarillion In Fantasy}}
\end{tikzcd}$$

Thus, as depicted above, a single book which is associated in two categories will be represented as two different facts.

A helper function to remove special characters is used to clean out any characters that clips might interpret in a different manner; this is done using python RegEx library. Another helper function converts a list in python to a list in clips. The category 'Books' was cleaned out as well as this is a master category and was present in all the rows. Finally categories in ranks where delimited by ' / ' which was split into different categories and passed as a multislot field to clips. Each category is a subcategory of the previous one. An improvement can be made to consider a tree of categories, rather than individual categories.

The user request is evaluated and passed through the helper functions to clean. Each entry from the user is added as a fact in clips (named request). Each request has an id, this is an assumption that the system should be able to handle multiple requests and correlation needs to be done between different requests.

Once the pre-processing phase is done, the run command in clips is called to evaluate all the input.

This is done into two phases and represented by two distinct rules. The first rule extracts the related categories for books inputted by the user. This is done using the '*member'* test function which takes a slot as input for the first parameter and checks if it is contained in the multislot presented as the second parameter. This creates a new fact (named ExtractedTopics)

```
(defrule extract_topics
   "Extract topics for list of book titles supplied."
   ?a <- (request (id ?id) (preferences $?preferences))
   ?b <- (amazonbook (rankcategories $?rankcategories) (title ?title))
   ;The member$ function will tell if a single field value is contained in a
multifield value.
```

```
   (test (member$ ?title $?preferences))
   =>
   (assert (extracted_topics (requestid ?id) (categories $?rankcategories)))
)
```

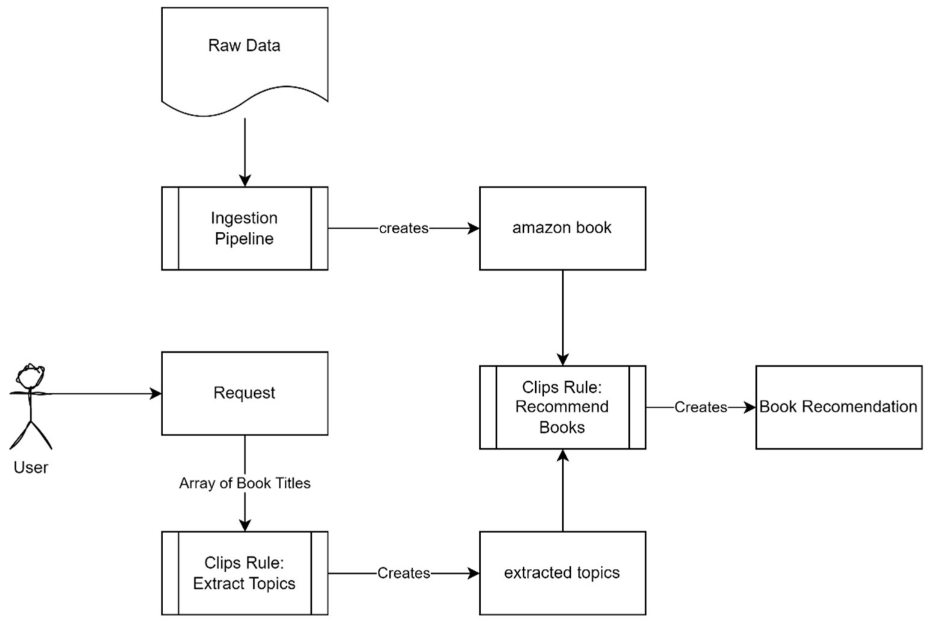
The second phase will intersect the list created in the first phase with the list of categories for each book fact. Once all the categories for a book presented by the user are contained in the rank categories presented a new fact is created (named book recommendation). This fact will be the result presented to the user and also contains the rank of this book for further sorting by the user interface. Finally, the recommended book is removed from the fact list so that it is not captured again.

```
(defrule recommend_books
   "Recomend books based on topics of previous list."
   ?u <- (extracted_topics (requestid ?id) (categories $?categories))
   ?b <- (amazonbook (asin ?asin) (ISBN10 ?isbn10) (rankcategories
$?rankcategories) (rank ?rank))
   ;This function checks if one multifield value is a subset of another; i.e.,
if all the fields in the first multifield value are also in the second
multifield value.
   (test (subsetp $?categories $?rankcategories))
   =>
   (assert (book_recommendation (requestid ?id) (asin ?asin) (rankcategories
?rankcategories) (rank ?rank) ))
   ;Remove the recomendation from the list so it is not presented again.
   (retract ?u ?b)
)
```

Finally since the same book might be represented by multiple facts, the last rule remove books that have been recommended

```
(defrule retract_books_recomended
   ?a <- (book_recommendation (asin ?asin))
   ?b <- (amazonbook (asin ?asin))
   =>
   (retract ?b)
)
```

The figure below represents the workflow described above.

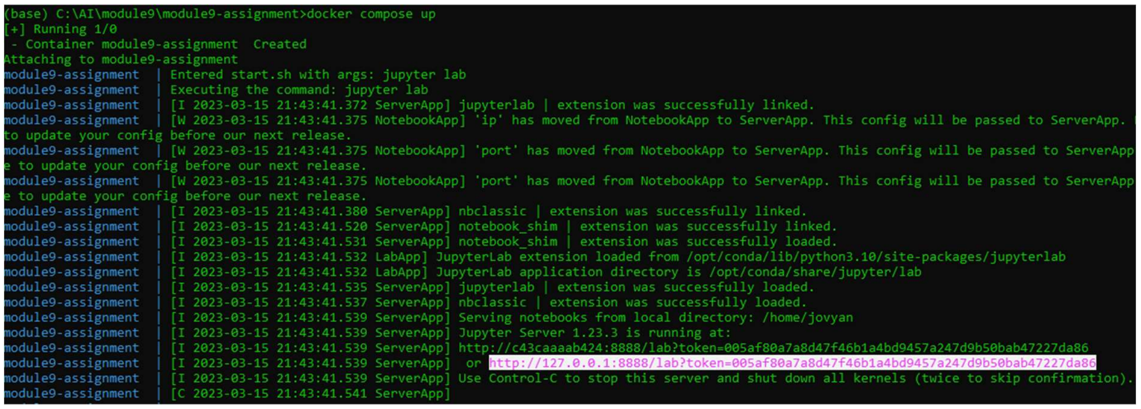


## Additional Notes

Git Repository: https://github.com/andrewspiteri/module9-assignment

## Running in Docker

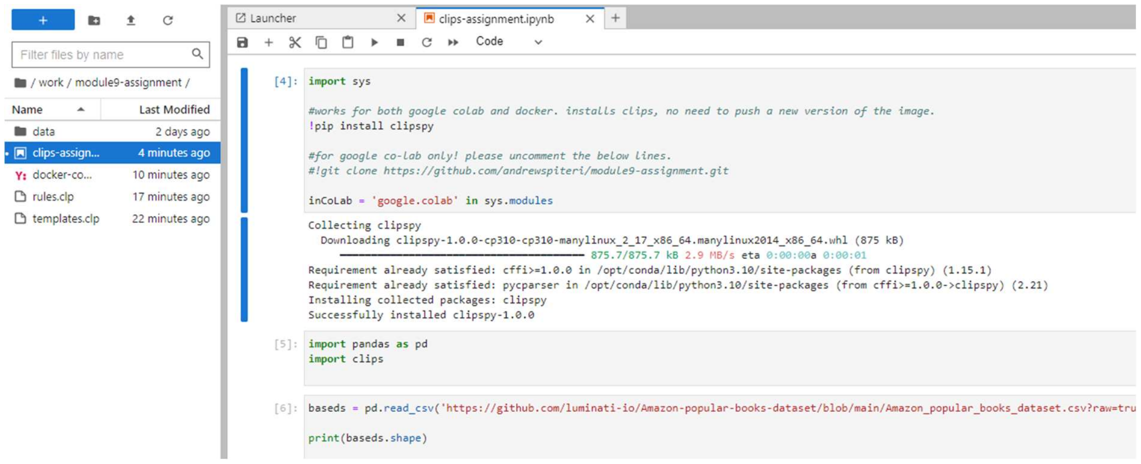
The system can be run using the command “docker compose up” in the folder where the git repository was cloned. No other special requirements are needed.



```
(base) C:\AI\module9\module9-assignment>docker compose up
[+] Running 1/0
 - Container module9-assignment  Created
Attaching to module9-assignment
module9-assignment  | Entered start.sh with args: jupyter lab
module9-assignment  | Executing the command: jupyter lab
module9-assignment  | [I 2023-03-15 21:43:41.372 ServerApp] jupyterlab | extension was successfully linked.
module9-assignment  | [W 2023-03-15 21:43:41.375 NotebookApp] 'ip' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp.
to update your config before our next release.
module9-assignment  | [W 2023-03-15 21:43:41.375 NotebookApp] 'port' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp
e to update your config before our next release.
module9-assignment  | [W 2023-03-15 21:43:41.375 NotebookApp] 'port' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp
e to update your config before our next release.
module9-assignment  | [I 2023-03-15 21:43:41.380 ServerApp] nbclassic | extension was successfully linked.
module9-assignment  | [I 2023-03-15 21:43:41.520 ServerApp] notebook_shim | extension was successfully linked.
module9-assignment  | [I 2023-03-15 21:43:41.531 ServerApp] notebook_shim | extension was successfully loaded.
module9-assignment  | [I 2023-03-15 21:43:41.532 LabApp] JupyterLab extension loaded from /opt/conda/lib/python3.10/site-packages/jupyterlab
module9-assignment  | [I 2023-03-15 21:43:41.532 LabApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
module9-assignment  | [I 2023-03-15 21:43:41.535 ServerApp] jupyterlab | extension was successfully loaded.
module9-assignment  | [I 2023-03-15 21:43:41.537 ServerApp] nbclassic | extension was successfully loaded.
module9-assignment  | [I 2023-03-15 21:43:41.539 ServerApp] Serving notebooks from local directory: /home/jovyan
module9-assignment  | [I 2023-03-15 21:43:41.539 ServerApp] Jupyter Server 1.23.3 is running at:
module9-assignment  | [I 2023-03-15 21:43:41.539 ServerApp] http://c43caaaab424:8888/lab?token=005af80a7a8d47f46b1a4bd9457a247d9b50bab47227da86
module9-assignment  | [I 2023-03-15 21:43:41.539 ServerApp]  or http://127.0.0.1:8888/lab?token=005af80a7a8d47f46b1a4bd9457a247d9b50bab47227da86
module9-assignment  | [I 2023-03-15 21:43:41.539 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
module9-assignment  | [C 2023-03-15 21:43:41.541 ServerApp]
```

Once the module docker is up and the url together with the token is opened in the browser the assignment is mounted under /work/module9-assignment

It must be noted that neither google colab, my local pc, nor docker image required a separate installation of clips to work other than the pip install part.



| Name | Last Modified |
|---|---|
| data | 2 days ago |
| clips-assign... | 4 minutes ago |
| docker-co... | 10 minutes ago |
| rules.clp | 17 minutes ago |
| templates.clp | 22 minutes ago |

```
[4]: import sys

     #works for both google colab and docker. installs clips, no need to push a new version of the image.
     !pip install clipspy

     #for google co-lab only! please uncomment the below lines.
     #!git clone https://github.com/andrewspiteri/module9-assignment.git

     inCoLab = 'google.colab' in sys.modules

     Collecting clipspy
       Downloading clipspy-1.0.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (875 kB)
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 875.7/875.7 kB 2.9 MB/s eta 0:00:00a 0:00:01
     Requirement already satisfied: cffi>=1.0.0 in /opt/conda/lib/python3.10/site-packages (from clipspy) (1.15.1)
     Requirement already satisfied: pycparser in /opt/conda/lib/python3.10/site-packages (from cffi>=1.0.0->clipspy) (2.21)
     Installing collected packages: clipspy
     Successfully installed clipspy-1.0.0

[5]: import pandas as pd
     import clips

[6]: baseds = pd.read_csv('https://github.com/luminati-io/Amazon-popular-books-dataset/blob/main/Amazon_popular_books_dataset.csv?raw=tru

     print(baseds.shape)
```

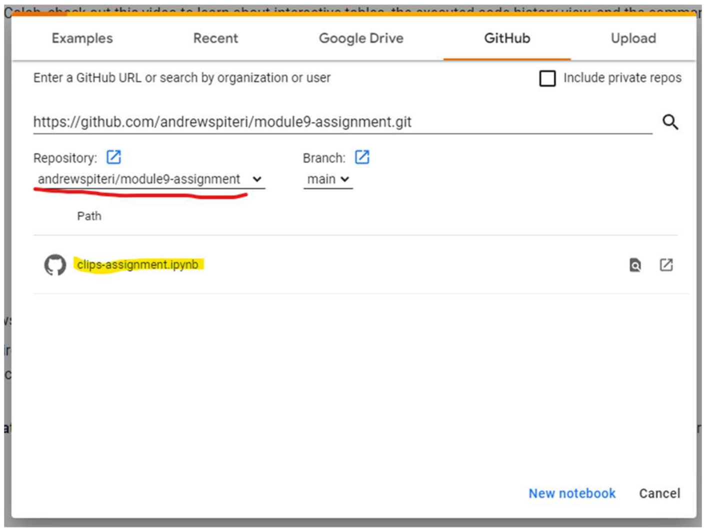
## Running in Google Co-Lab

In order to run in google colab, choose the github options, and paste the following URL:

https://github.com/andrewspiteri/module9-assignment.git

Make sure you choose module9-assignment (colob has the habit of picking a random repository instead of the one you pasted)

And click on clips-assignment notebook.



Enter a GitHub URL or search by organization or user

https://github.com/andrewspiteri/module9-assignment.git

Repository: andrewspiteri/module9-assignment

Branch: main

clips-assignment.ipynb

Uncomment the following line to download the code in the vm:

```
#for google co-lab only! please uncomment the below lines.
#!git clone https://github.com/andrewspiteri/module9-assignment.git
```

References

(KMS 2003) Knowledge management system (2003) Knowledge Management System - an overview | ScienceDirect Topics. Available at: https://www.sciencedirect.com/topics/social-sciences/knowledge-management-system (Accessed: March 9, 2023).

Samadi Soufi et.al (2013): Investigate the Effect of Expert Systems Application on Management Performances. Available at: https://journal-archieves31.webs.com/478-482.pdf (Accessed: March 15, 2023).

Meseguer, P. and Verdaguer, A. (1996), Expert system validation through knowledge base refinement. Int. J. Intell. Syst., 11: 429-462. https://doi.org/10.1002/(SICI)1098-111X(199607)11:7<429::AID-INT2>3.0.CO;2-O

Edwards, John & Duan, Yanqing & P.C, Robins. (2000). An analysis of expert systems for business decision making at different levels and in different roles. European Journal of Information Systems. 9. 36-46. 10.1057/palgrave.ejis.3000344.