



**University of Trento**

**Master of Science in Data Science**

**Academic Year 2021-2022**

**Big Data Technology**

**PROJECT REPORT**

**Students:**

*Hamid Omid*  
*Elisa Maccabiani*  
*Elisa Rigoni*

**Professors:**

*Daniele Miorandi*  
*Carlo Caprini*

# 1. PROJECT PURPOSE

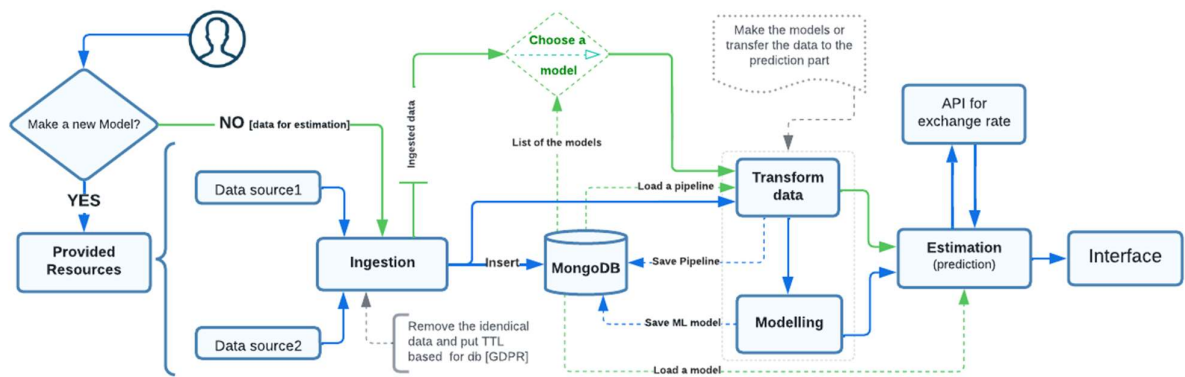
The aim of this project is to design and implement a big data system for estimating annual expenditure on technology of single individuals based on their socio-demographic data. Socio-demographics relate to a combination of social and demographic aspects that define people in a specific group or population. The core idea is that different socio-demographic factors affect the reasons why a customer chooses to invest in technological products and services, and the amount of money invested. Nowadays, having knowledge about personal annual spending on technology is relevant for many companies and organisations. In fact, by interpreting this data correctly, it is possible to discover patterns in consumer preferences and behaviours. We conducted research with the purpose of identifying the main socio-demographic aspects that can influence the individual expenditure in technology. After reading many papers and sociological studies we decided to focus our predictions on these characteristics: *age, sex, level of education, urban classification, number of children, working status, annual income, marital status, location, ownership, household size.*

Our different backgrounds allowed us to combine different knowledge and approaches to pursue a unique common objective through

an interdisciplinary collaboration. We accomplished the conclusion of this project by blending the use of different types of technologies, such as databases, data processing engines, machine learning algorithms, and also interface development.

## 1.1 Logical Pipeline

Logical pipeline for this project shown in the below chart. In this pipeline we have two main paths based on the given task by the user. First path which is followed by the **blue** line is for making a new model. And the **green** line is for using a made model for estimation. In the first path (making a new model), after importing the data sets to the pipeline, data will be ingested by the ingestion module. Ingestion module will remove unnecessary variables from the data set based on the user and system constraints (e.g. anonymisation<sup>1</sup>), and in the next step the data will be stored in our database. The difference for the second path is that we don't need to save the information in our database. To continue the first path we path the data to the transforming part to prepare the data for machine learning in spark<sup>2</sup>. In the next steps the model will be created and the new data can be estimated by the model that had been made in the previous step. Moreover, an API connection is provided that enables the user to change the currency based on currency



<sup>1</sup> Refer to privacy and regulations

<sup>2</sup> Refer to the Spark part [spark pipeline]

rate exchange. At the end, the outcome can be shown on the interface. As it can be seen for the second path we do not save the data in the database, the pipeline just ingest the data and pass the ingested data to the transforming and estimation part. The estimation part will load the chosen model and estimate the expenditures. In the table below the summary of the block's activation and functionality can be seen.

| Block              | Functionality  |
|--------------------|--|
| Ingestion          | Get the data from the sources, ingest it. Also will pass the ingested data to the transforming step.   |
| MongoDB            | It is the database and save the ingested data, pipeline model and ML model   |
| Transform the data | Will get the ingested data and after transforming will pass it to the estimation model or to modelling section. Also it can save or load the pipeline model to or from the database. |
| Modelling          | This section will get the transformed data and make a ML model. Also it can save the model on the database.  |
| Estimation         | This module can get the data and model and make the estimation. Also there is an ability to be connected to the API block to convert the currency.                                   |
| API                | This block uses an API to retrieve the update exchange rates.  |
| Interface          | This section will be a web interface for showup the output also will give the ability to work with pipeline with user interface  |

## 2. DATA

We decided to allow the user to input in our system both structured (e.g. CSV) and semi-structured data (e.g. Json). Given the existence of a wide range of socio-demographic characteristics that can be collected, there is a range of attributes that can be different from dataset to dataset. There are also common variables that are frequently collected, in this section we focus on understanding these indicators and how we can legitimately handle and analyse them. We created fake datasets

using different formats to simulate several types of inputs.

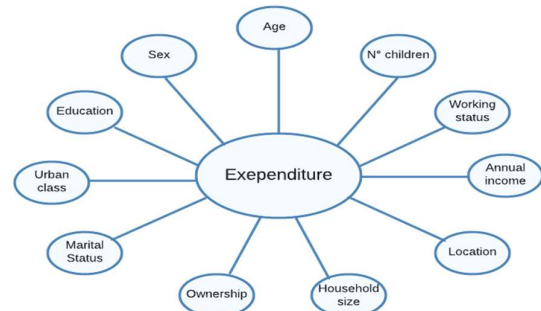
### 2.1 GDPR

Datasets about socio-demographics aspects also include personal data, because they are related to an identified or identifiable individual, and for this reason they are subject to specific constraints mandated by the GDPR in the EU. In particular, according to Article 17 of the GDPR, “*The data subject shall have the right to obtain from the controller the erasure of personal data concerning him or her without undue delay*”. Even the principles relating to processing of personal data (Art. 5, GDPR) should be taken into account, in particular the focus is on Art. 5, par 1.e) concerning the storage limitation: to ensure the respect for the rights the system uses a collection Time-To-Live for data storage. Moreover, in order to limit the number of people that have access to personal data collected in datasets and preserve the privacy of each individual, a process of anonymisation shall be implemented on every single record. Anonymisation is an important step because it allows data to be shared, so attributes like *Name*, *Surname* and *Email* will be removed to offer a good level of privacy.

### 2.2 Data models

#### Conceptual data model

The conceptual model shows a description of the relationship between the socio-demographic factors and the expenditure.




### Logical data model

The logical data model gives a description of attributes. After the ingestion, the data will have the following pattern:

Socio-Demographic Data

|                |   |
|----------------|---|
| ID             | int   |
| Age            | int [ 18 to 80 ]                                |
| Sex            | enum [ male, female, other ]                    |
| Education      | enum [ no, elementary, graduated ]              |
| Urban_class    | enum [ city, town, rural area ]                 |
| n_children     | int [ 0 to 10 ]                                 |
| Working_status | enum [ student, working, not working, retired ] |
| Annual_income  | float   |
| Marital_status | enum [ single, married, partnership, widowed ]  |
| Location       | string [ city name ]                            |
| Ownership      | enum [ owned, rented ]                          |
| Household_size | enum [ single, multi ]                          |



|              |       |
|--------------|-------|
| Expenditures | float |
|--------------|-------|

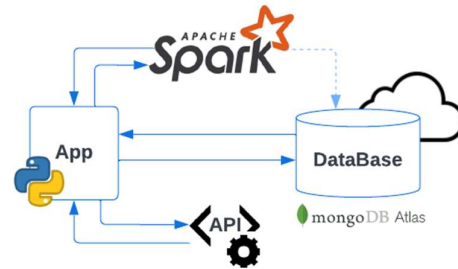
### Physical data model

The physical data model describes how data is represented in a computer-readable medium.

```
{
  "0": {
    "id": 57681,
    "sex": "F",
    "age": 52,
    "working_status": "working",
    "phone": "+49(0)1866 315955",
    "n_children": 0,
    "income_annual": 114084,
    "location": "Cuxhaven",
    "own_home": "yes",
    "marital_status":
  "relationship",
    "exp": 66164
  },
  ...
}
```

## 3. TECHNOLOGIES

In order to develop the system, it was necessary to use different types of technologies, such as MongoDB, Apache Spark, Python and API. The Technology diagram is shown, and it will be explained below.



### 3.1 MongoDB

As has already been said, the input data can be structured or semi-structured, hence it is preferable to have a flexible environment without a specific schema that can support both data formats. The idea of using MongoDB instead of SQL is strictly connected to this point. In fact, in SQL you need to define your tables and columns (schema on write) while in MongoDB you don't need to define the schema (schema on read), this ensures a more flexible system. In this way, it could be a perfect choice for saving the serialised or semi structured machine learning model [PMML] too. Moreover, MongoDB is an ideal choice because there are data with the potential for rapid growth, this means that data volume could get bigger exponentially. Since there is no need for relational operations, such as selection, projection or join, it would be of very little use to choose a SQL. Another important point is that the system created to estimate the expenditure on technology does not need an in-memory data store, which enables low latency and high throughput data access, because it only has to store data in a database. This consideration removes Redis from the list of the possible databases, conducting the choice to MongoDB. Lastly,

the scalability factor, which is the ability to expand or contract the capacity of system resources in order to support the different usage of the application, is well supported by MongoDB and this ability allows the system to store more data without sacrificing performance. Furthermore, MongoDB supports unique collection type like TTL (Time-To-Live) for data storage which will expire at a certain time, this quality allows to solve constraints connected to the process of personal data and to respect Art. 5, par 1.e) of the GDPR.

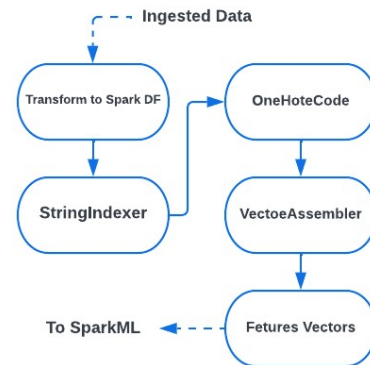
After choosing MongoDB, all values of data extracted from a dataset are transformed into the desired type and finally the observations are stored in the database. MongoDB Atlas has been used in this project on cloud infrastructure by which we can guarantee accessibility. By using as a service infrastructure, we transfer the expense of the maintenance but at the same time we can face new challenges like cyber security. For this project the accessing ip for connecting to the database is limited. Also propagate way of using permission for accessing the level of the database is important. Another challenge that we faced in this project was some problem with network configuration. We should provide a reliable network infrastructure. From our experience, some internet infrastructures based on their configurations were not a good choice as we had several problems during the project, but by changing the network infrastructure or system became reliable.

### 3.2 Apache Spark

Apache Spark is a data processing framework with two important qualities that are the key to the worlds of big data and machine learning. First of all, it can quickly execute processing tasks on very large datasets, for example it can perform task up to one hundred time faster than

MapReduce in certain situations. The second quality is the scalability: Spark can distribute data processing tasks across multiple computers, so it can be defined as a parallel data processing framework. In this way it can satisfy our scalability vision in this project which can be considered as a fundamental criterion of a big data project. Moreover, the ease of use is another advantage of Spark, indeed it supports a lot of languages and libraries proving to be quite versatile.

Spark has been used for making a machine learning model for prediction in this project. The data after ingestion will go to the spark pipeline that can be seen below.



In the first step Ingested data will transform to Spark dataframe to make it able for pipeline's next steps. The categorical variables will change to index and after that will convert to one hot code variable to be prepared for the ML model. At the last stage of the pipeline the variables will convert to feature vectors. This pipeline [pipeline.fit()] will save for the next usages [pipeline.transform()].

### 3.3 API

When the system obtains the prediction, it is also possible to convert the estimating expenditure in a different currency. This is done using currency exchange APIs, which gather currency updates from many data sources.

### 3.4 Pseudo Code

In the following picture the pseudo code can be seen. Generally, it contains 2 sections (main tasks) that can be implemented.

```
while True:
    if Ask_for_new_model:
        Load the data

        # Based on the constraints the data will be
        # ingested [e.g. Anonymisation(GDPR)].
        Ingest the data
        Ingested data will be stored in Database

        # Make a model based on data(train the model)
        # Output will be the trained pipeline model,
        # trained ML model, and outcome of Spark
        # pipeline Spark pipeline in this project = data
        #[Transforming TO spark df] > StringIndexer >
        # oneHoteCode > VectorAssembler > Features vectors
        Make a model
        Ask the new data and ingest the data
        Transform data throughout the Spark pipeline

        # Make the Estimation
        Make the estimation based on model
        Currency converting by API

    elif Ask_for_Use_a_model:
        A model will be chosen among the saved model
        Pipeline and ML model will be loaded
        Ask the new data and ingest the data
        Transform data throughout the Spark pipeline

        # Make the Estimation
        Make the estimation based on model
        Currency converting by API

    elif Ask_for_Exit:
        break

    else:
        continue
```

## 4. WEBSITE PROTOTYPE

After the completion of the system, we created a prototype of a possible web interface that allowed us to make the program we created more user-friendly. First of all, we developed a recognizable logo that conveys a perception about the service we provide.



Lately, we focused on developing a web interface that aims to make the user's interaction as simple and efficient as possible. We think that our prototype could improve the overall understanding of the design. We decided to display some focal points of the system we implemented, namely the steps in which the user has to load the datasets and the

final step in which it can save the model fitted to share it to future users (also providing further information that allows to improve user understanding).

## 5. FUTURE POTENTIAL IMPROVEMENT

There are some suggestions for future improvement in this project:

- Improve error handling
- Expand the new type of input (datasets from sql etc .)
- Develop the web interface
- Develop an API for giving the expenditure estimation service.
- Developing a module that can gather the data and make a suitable dataset