**Molecule Prediction**

**Project’s Report**

**Elaborated by:**

Safa Chaari

Mariem Mazouz

Abir Barouni

Ghofrane Soltani

Med Aziz Omrani

Aziz Tebessi



**Academic year: 2022-2023**

**Class: 4th DS 6**

**List of figures**

**Figure 1** :Methodology adopted\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**6**

**Figure 2** : Representation 2D\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**7**

**Figure 3** : Fingerprint \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **10**

**Figure 4**  :Graph with 4 nodes and 3 edges \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**10**

**Figure 5** : Representation 2D\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**11**

**Figure 6** : Used tools\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **12**

**Figure 7** : Framework Django \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **13**

**Figure 8** : Description of data\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**14**

**Figure 9** : Description of data\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**15**

**Figure 10** : Data info\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **15**

**Figure 11** : Missing values detection\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **18**

**Figure 12** : Result before doing the replacements\_\_\_\_\_\_\_\_\_\_ **21**

**Figure 13**  : Result after doing the replacements\_\_\_\_\_\_\_\_\_\_\_\_**21**

**Figure 14**  : converting features to numerical type\_\_\_\_\_\_\_\_\_\_\_**22**

**Figure 15** : Verification after the conversion\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**23**

**Figure 16** : splitting data\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**23**

**Figure 17** : Visualization of the features encoded\_\_\_\_\_\_\_\_\_\_\_**24**

**Figure 18** : Visualization molecular weight’s boxplot\_\_\_\_\_\_\_\_\_**25**

**Figure 19** : Visualization Polar Surface Area’s boxplot\_\_\_\_\_\_\_**26**

**Figure 20** : Applying IQR for Molecular Weight\_\_\_\_\_\_\_\_\_\_\_\_\_**27**

**Figure 21** : Applying IQR for Targets\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**28**

**Figure 22 :** CNN architecture\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**30**

**Figure 23** : VAE architecture \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **31**

**Figure 24**  : Gans architecture\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**31**

**Figure 25** : RNN architecture\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**32**

**Figure 26** : Bart architecture\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**32**

**Figure 27**  : Exemple D’output BART \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **33**

**Figure 28** : LSTM architecture\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**33**

**Figure 29**  : MolLSTM output\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**34**

**Figure 30** :MolGen architecture\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**35**

**Figure 31** :MolBERT architecture\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**36**

**Figure 32** : MolBERT output\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**37**

**Figure 33** : Roberta architecture\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**38**

**Figure 34** : output model\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**39**

**Figure 35** : MolBERT’s Score\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**39**

**Figure 36** : RoBERTa’s Score \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **39**

**Figure 37** : Interface of the team members of molminds company**43**

**Figure 38** : Interface of molecular prediction \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**44**

**Figure 38** : Interface of the prediction’s result \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**45**

**Table of contents**

[**Chapter 0: 7**](#_js4nhh6ye1pz)

[**Project overview 7**](#_5mpvixjf3ksr)

[0.1 Introduction 7](#_n3wh4dunhxah)

[0.2 Presentation of the project 7](#_wr69o8ooliws)

[0.2.1 project framework 7](#_rsr9de4f1ibx)

[0.2.2 Context and Issue 7](#_phocmccbqvf7)

[0.2.3 Purpose 7](#_ofu5ys581wkk)

[0.3 Methodology adopted 8](#_5y8xg2brgq3)

[0.3.1 IBM’s master plan phases 8](#_5v7icxqyz32k)

[0.3.2 Choice of methodology 10](#_ujfq2cgbiiys)

[0.4 Conclusion 10](#_4nl3s8ww0zme)

[**Chapter 1 11**](#_guedstax0k9)

[**Business understanding 11**](#_c55rvlgvsdws)

[1.1 Introduction 11](#_e9b5igibl31z)

[1.2 Business objectives 11](#_45bthaoxohk)

[1.2.1 Functional needs 11](#_n1gcdpuupok1)

[1.2.2 Non-functional needs 12](#_j4z5w44cemgy)

[1.3 Project planning 13](#_tbyrr0a0qq22)

[1.4 Used tools 14](#_j58t9m7mf2ns)

[1.5 Steps 15](#_nxgllctnlp5r)

[1.6 Conclusion 15](#_l9nysf68qbt8)

[**Chapter 2 16**](#_dgxf0n6790p7)

[**Data understanding 16**](#_dqvp7npebzrj)

[2.1 Introduction 16](#_y6pzylg40hy)

[2.2 Data source 16](#_bsq6qng15x3x)

[2.3 Features description 18](#_tfv28l56z5zm)

[2.4 Verifying data quality 20](#_8mab6mqyw0b4)

[2.5 Conclusion 21](#_hfq3xbdsii0n)

[**Chapitre 3 22**](#_nvx0hrb1jzh4)

[**Data Preparation 22**](#_nx49aaglrbqm)

[3.1 Introduction 22](#_62ydfecb55x7)

[3.2 Data cleaning 22](#_gp45uoxcc54l)

[3.3 Data transformation 23](#_9yn0735vkv9e)

[3.4 Data splitting 25](#_49h90ptbkv4b)

[3.5 Data encoding 26](#_c1smjq784qeh)

[3.6 Data preprocessing 27](#_22scggayxpx3)

[3.7 Conclusion 30](#_ldqiaxsm8280)

[**Chapitre 4 31**](#_ahyn5dsk2wux)

[**Modeling and Evaluation 31**](#_srz18kcs010d)

[4.1 Introduction 31](#_oalu01rzwqdd)

[4.2 Used Methods 31](#_1xtz3x1gisbq)

[4.3 Models 31](#_d8q0drvm95m7)

[4.3.1 Tested models : 31](#_vjm0pcz2bu57)

[4.3.2 Used models : 36](#_9i6yew67xbua)

[4.4 Evaluation 40](#_h13ykubr8zey)

[4.5 Conclusion 40](#_ko7jk6eneh6y)

[**Chapter 5: 41**](#_erbmnk2y854a)

[**Deployment phase 41**](#_g16mtvnrenmu)

[5.1 Introduction 41](#_1dz50074dy45)

[5.2 Concept and deployment strategy 41](#_narc5nwcc4ok)

[5.3 Deployment environment(tools): 41](#_1vvuufnm091w)

[5.4 the web application: 42](#_iw4ev6qjnedt)

[5.4.1 Web application: 42](#_hu5snr7a84hk)

[5.4.1.1 Interface of the molecule prediction: 42](#_8dogcc2kxuq8)

[5.5 Conclusion 43](#_i7k84p80j0qw)

[**Chapter 6: 44**](#_vnatots2c6lu)

[**Perspectives and contributions 44**](#_6pdf0e42xc4f)

[6.1 Perspectives 44](#_idsgal3lqsym)

[6.2 Academic and professional contributions 44](#_fggko6susbwq)

**General Introduction**

In the context of molecule prediction and data science, molecule design refers to the use of computational methods and algorithms to design new molecules with desired properties or functions. This involves using machine learning, artificial intelligence, and other data science techniques to analyze and model the relationships between molecular structures and their properties, and to generate new molecules that are predicted to have specific properties.

Molecule design in this context is often used in drug discovery, where researchers use computational models to predict the efficacy, toxicity, and other properties of candidate drug molecules. By designing new molecules with the desired properties, researchers can accelerate the drug discovery process and reduce the cost and time required to bring new drugs to market.

This report will represent the Molecule Design modifications that are applied on the dataset and the complete analysis on the same.

Our main goal will be to generate new molecules that are predicted to have specific properties.

For that , in chapter 0:In the project Overview, first there is the introduction of the methodology adopted, and then the determination of the goals of the project.

Next, the steps that are defined and the tools used.

After that, the dealing with the data understanding phase in which there is the exploration of the data, verification of its quality and the preparation for the next step which is the Data preparation phase that will be our third chapter.

Next moving to the chapter of modeling and evaluation, where there is the explanation of the different models worked with, and finally choosing the best one to be used in the Deployment phase which will be represented in the last chapter.

# **Chapter 0:**

# **Project overview**

## **0.1 Introduction**

In this chapter , the context and issues of the project will be represented , also its objectives and the methodology of the work adopted.

## **0.2 Presentation of the project**

First of all, the project framework will be exposed thereafter, the context and the problematic of the subject.

### **0.2.1 Project framework**

This project will be delivered 08/05/2023 as a Data Science project at ESPRIT: Private High School of Engineering and Technologies.

The start date was 23/01/2023.

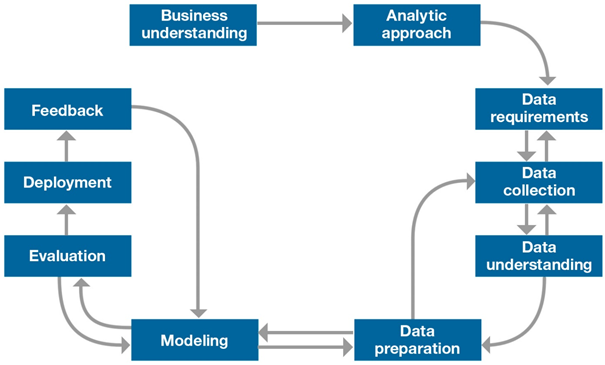
### **0.2.2 Context and Issue**

Molecule design consists of predicting properties of a molecule and generating new molecules with similar characteristics based on 31 attributes which includes AlgoP, Polar Surface Area (PSA) and HBA.

### **0.2.3 Purpose**

The purpose of this research is to develop and design an effective and efficient model for molecule prediction.

## **0.3 Methodology adopted**

****

**Figure 1: Methodology adopted**

IBM's master plan methodology is a process model that serves as the base for a [data science process](https://www.datascience-pm.com/data-science-process/). It has six sequential phases:

1. Business understanding – What does the business need?
2. Data understanding – What data do we have / need? Is it clean?
3. Data preparation – How do we organize the data for modeling?
4. Modeling – What modeling techniques should we apply?
5. Evaluation – Which model best meets the business objectives?
6. Deployment – How do we access the results?

### **0.3.1** **IBM’s master plan phases**

**1.Business Understanding:** The Business Understanding phase focuses on understanding the objectives and requirements of the project

**2. Data Understanding:** Associate the data with its signification from a business perspective, to determine precisely the data to be analyzed and to determine its quality.

**3. Data Preparation:**The data preparation step groups all activities required to construct, from raw data, a precise set of data to analyze. It thus includes data sorting based on selected criteria, data cleansing, and, most importantly, data recoding to ensure compatibility with any and all algorithms that will be used.

Digital data parametricity and its recoding into categorical data are extremely important and must be executed with the greatest care in order to ensure that used algorithms do not produce inaccurate results during the next step. All data must be centralized in a structured database known as a Data Hub.

**4. Modeling:** This is the actual Data Science step. Modeling includes selecting, configuring and testing various algorithms, as well as deciding on their sequence, which creates a model. The process is initially a descriptive one that generates knowledge and explains why things happened. It then becomes predictive and explains what will happen, and later prescriptive as it helps optimize future situations.

**5. Evaluation:** The aim of the evaluation step is to verify any models or knowledge obtained in order to ensure that they meet the objectives identified at the beginning of the process. The evaluation also informs model deployment decisions, or as required, model improvement ones. At this stage, the robustness and accuracy of developed models are tested.

**6. Deployment:** The final step of the process. It consists in implementing generated models for end users and its aim is to use modeling to format knowledge in such a way that it can be integrated into the decision-making process. Depending on objectives, deployment can thus range from the simple generation of a report describing knowledge obtained to the installation of an application that helps leverage the obtained model to predict unknown values for an element of interest.

## 

### **0.3.2 Choice of methodology**

IBM's master plan methodology will provide us with the most adequate solution for the needs of customers with a better workflow , it encourages best practices and allows projects to be replicated. This methodology provides a uniform framework for planning and managing a project.

## **0.4 Conclusion**

Now that the overall framework of the system has been detailed by presenting the problem, the solution that will be designed and developed, as well as the methodology to follow, the first phase of our methodology will begin , which is business understanding.

# **Chapter 1**

# **Business understanding**

## **1.1 Introduction**

The first step in the IBM Master Plan is business understanding, which is critical for determining the goals and objectives of the data mining project from a business perspective. This step involves identifying the key stakeholders, defining the business problem or opportunity, and establishing clear goals and success criteria for the project.

Throughout the project, there is a focus on the pre-established goals and objectives to develop an action plan that aligns with the business requirements.

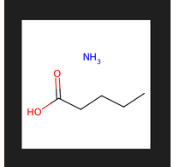
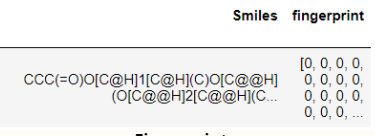
The Master Plan provides a structured approach for executing each phase of the project, including data understanding, data preparation, modeling, evaluation, and deployment, to ensure that the project remains aligned with the business goals.

## **1.2 Business objectives**

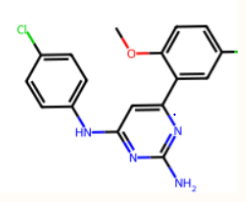
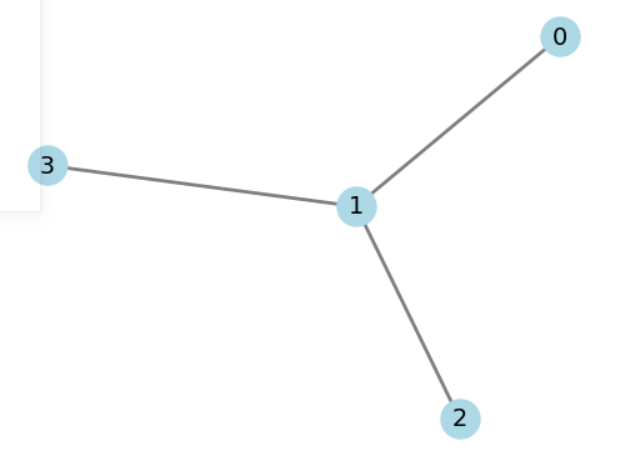
### **1.2.1 Functional needs**

A mechanism has been developed to make the entire process of molecular design verification is not only fast but also reliable by using deep learning methods:

* Valid molecule composition
* Identification of molecular characteristics
* Manipulate a huge amount of molecules.
* Computerize the process of generating molecule design.
* Visualization of molecules:

**Figure2 : Representation 2D Figure3 : Fingerprint**



**Figure4 : Graph with 4 nodes and 3 edges Figure5 : Representation 2D**

### **1.2.2 Non-functional needs**

* ***Robustness:***

the used model will be capable of dealing with invalid or erroneous input effectively.

* ***Reliability:***

It starts with data quality, ensuring that training and real-world datasets are complete, accurate, and relevant.

Not doing so may compromise the reliability of model output.

* ***User interface:***

An intuitive and user-friendly interface that allows scientists and engineers to input molecular data, run predictions, and analyze results.

## **1.3 Project planning**

Planning is an essential stage in project management as it involves identifying and organizing the necessary tasks to be completed throughout the project. This step is crucial in monitoring progress, fixing timelines, and managing resources to ensure efficient completion of the tasks. A well-designed plan established at the outset helps to achieve the set data mining and business goals. Efficient project management also ensures alignment between project objectives and the strategic objectives of the company.

1-Project Scope and Objectives

* Define the scope of the project, including the type of molecules and properties to be predicted
* Identify the objectives of the project, such as improving accuracy or reducing computation time

2-Data Preprocessing

* preprocess data for training and testing the deep learning model
* Ensure that the data is representative of the molecules and properties of interest
* Conduct exploratory data analysis to identify any data quality issues or biases

3-Model Selection and Development

* Identify appropriate deep learning models for the task at hand, such as convolutional neural networks or recurrent neural networks
* Develop and refine the deep learning model(s) using the preprocessed data
* Evaluate the model(s) using appropriate metrics, such as mean squared error or mean absolute error

4-Optimization and Tuning

* Optimize the model hyperparameters to improve performance and reduce overfitting
* Explore different optimization techniques, such as stochastic gradient descent or Adam optimization

5-Model Deployment and Integration

* Deploy the trained model into a production environment, such as a web application or API
* Conduct thorough testing to ensure that the model is functioning correctly

## **1.4 Used tools**

Our predictive modeling for molecule prediction is conducted using Python and its powerful open-source libraries through the utilization of Jupyter Notebook, which enables the code to be interactive and present the findings in a narrative format.

So,to simplify data analysis and transformation, to structure our data into an easily understandable format the import Pandas is Obliged. Also the leverage plotly for visualizing data exploration results in a more intuitive manner. Additionally, scikit-learn to split the dataset and build a predictive model to predict molecules.



**Figure6: Used tools**

* In the deployment phase the used framework is Django :



**Figure7 : Framework Django**

## **1.5 Steps**

Our project consists of several key steps:

The initial stage involves data discovery, exploration, and quality verification and also examines the data to gain insights into its structure and identify any potential data quality issues.

In the data preparation phase, Various techniques were used such as inclusion, cleaning, exclusion criteria, merging, and formatting to prepare the data for modeling.

In the modeling phase for predicting molecules properties, several algorithms are used to determine the most efficient one for the task at hand. So, various types of deep learning models, such as RoBERTa and molBERT , and fine-tuning are used with the preprocessed data. Also evaluating the models based on appropriate metrics, such as SIMILARITY coefficient.

The goal of this phase was to identify the most effective deep learning model that can predict the properties and the molecules accurately.

Finally, after the evaluation of the models are done , the deployment of the project is going to be the final step.

## **1.6 Conclusion**

In this chapter, the business objectives of the project were presented and outlined the steps taken during the Business Understanding phase. Moving forward, the subsequent chapter will delve into the second stage of the IBM master plan process, which is Data Understanding.

# **Chapter 2**

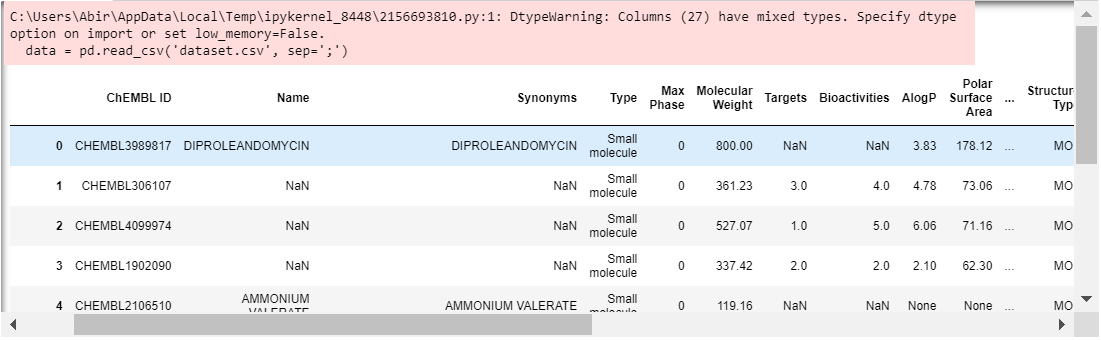
# **Data understanding**

## **2.1 Introduction**

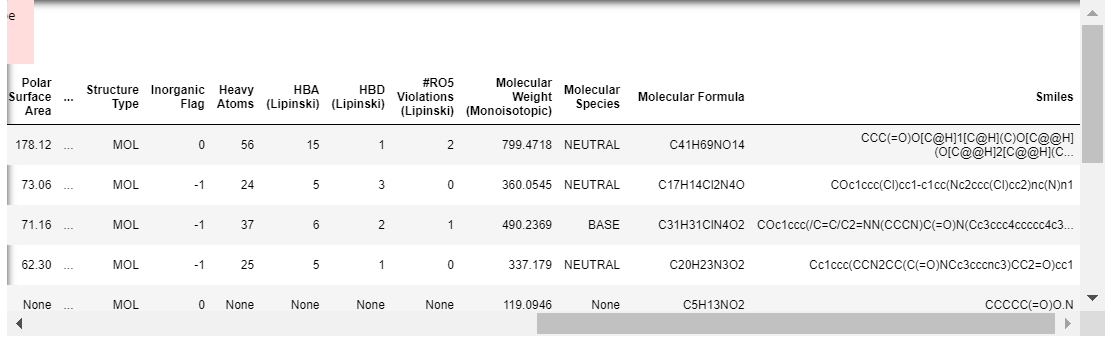
In this chapter, the second step of the IBM master plan process will be exposed. So first of all exploring the data , then visualizing and understanding each feature of it.

## **2.2 Data source**

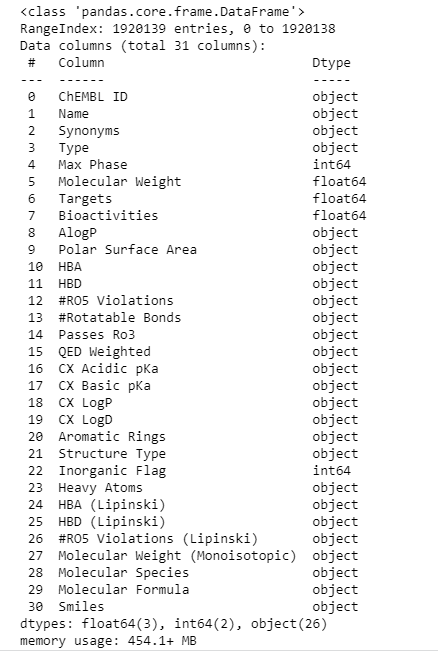
The dataset contains 1920139 rows andcomprises 31 features divided into 26 objects features and 2 int64 features and 3 float64. Features include 'ChEMBL ID', 'Name', 'Synonyms', 'Type', 'Max Phase', 'Molecular Weight', 'Targets', 'Bioactivities', 'AlogP', 'Polar Surface Area', 'HBA', 'HBD', '#RO5 Violations', '#Rotatable Bonds', 'Passes Ro3', 'QED Weighted', 'CX Acidic pKa', 'CX Basic pKa', 'CX LogP', 'CX LogD', 'Aromatic Rings', 'Structure Type', 'Inorganic Flag', 'Heavy Atoms', 'HBA (Lipinski)', 'HBD (Lipinski)', '#RO5 Violations (Lipinski)', 'Molecular Weight (Monoisotopic)', 'Molecular Species', 'Molecular Formula' and 'Smiles' which provide information on the properties, activities, and structures of small molecules and their targets. It is used extensively in drug discovery research and other fields such as computational biology and cheminformatics.



**Figure 8: Description of data**



**Figure 9: Description of data**

****

**Figure 10 : Data info**

## 

## **2.3 Features description**

Each feature, or column, represents a measurable piece of our data that can be used for analysis:

**AlgoP:** In the context of molecule design, "AlgoP" likely refers to the "ALGOrithmic Prediction" of physicochemical properties. AlgoP is a computational method used to predict various physicochemical properties of organic molecules, such as solubility, logP (partition coefficient), and pKa values.

**Polar Surface Area (PSA):** Is a molecular descriptor commonly used in molecule design to characterize the polarity and hydrogen bonding capabilities of a molecule. It represents the surface area of a molecule that is occupied by polar or charged atoms or groups, such as nitrogen, oxygen, and sulfur, and it is measured in square Angstroms(Å²). **TYPE:** There are many types of molecules that can be considered in molecule design. Some of the common types of molecules used in pharmaceutical, materials, and chemical industries include: Small molecules. **HBD (Hydrogen Bond Donor) and HBA (Hydrogen Bond Acceptor):** are two molecular descriptors commonly used in molecule design to predict the ability of a molecule to form hydrogen bonds with other molecules, particularly with protein targets in drug design.

**RO5: Rule of five:** The "Rule of Five" (RO5) is a widely used guideline in molecule design and drug discovery that helps to identify molecules that are likely to have good oral bioavailability and permeability across biological membranes. **Rotatable bonds**: Usedto quantify the number of single bonds in a molecule that can rotate freely around their axis. These bonds are typically located in the backbone or side chains of a molecule and allow it to adopt different conformations in solution or when binding to a target molecule.

**QED: Quantitative estimate of drug likeness:** Weighted is a molecular descriptor used in molecule design to assess the drug-likeness of small molecules. It is a scoring system that takes into account various physicochemical and pharmacological properties of a molecule and provides a single score that represents the molecule's overall drug-likeness.

**CX (Carboxylic acid) acidic pKa vs basic pka:** In organic chemistry, CX acidic pKa and CX basic pKa are terms used to describe the acidity or basicity of a functional group X attached to a carbon atom C.

The pKa value represents the acidity or basicity of the functional group in question. A lower pKa value indicates that the functional group is more acidic, while a higher pKa value indicates that it is more basic.

**CX LogP vs LogD:** They are terms used to describe the hydrophobicity or lipophilicity of a molecule, specifically the partition coefficient between a nonpolar and a polar solvent.

The LogP and LogD values are important in drug discovery and design as they can predict the molecule's ability to cross cell membranes and its distribution in different tissues.

The difference in molecule design between CX LogP and CX LogD lies in the pH of the medium used to measure the partition coefficient.

**Max Phase:** The MAX phases are a family of layered, hexagonal carbides and nitrides with the general formula Mn+1AXn (where n = 1 to 4), where M is an early transition metal, A is an A-group (mostly IIIA and IVA, or groups 13 and 14) element and X is either carbon or nitrogen.

**Molecular weight:** is an important factor when designing molecules, as it affects the physical properties of macromolecules. The molecular weight of a molecule is the sum of the atomic weights of its component atoms, and can be determined through mass spectrometry and thermodynamic or kinetic transport phenomena. Number average of molecular weight, weight average of molecular weight, Z-average molecular weight, viscosity average molecular weight, and distribution of molecular weight are all ways to calculate the molecular weight of polymers.

**Inorganic flag:** in molecule design refers to a feature or property of a molecule that identifies it as being inorganic in nature. Inorganic molecules are those that do not contain carbon-hydrogen bonds, and they often include metals, minerals, and non-carbon-based compounds.

**Molecular species:** refers to a specific type of molecule that can exist in different forms depending on its environment or chemical conditions. Specifically, a molecular species may be neutral, charged, or in a particular electronic or vibrational state.

**Heavy atoms:** In chemistry, heavy atoms are atoms with high atomic numbers, typically greater than or equal to 6 (carbon), which can be incorporated into a molecule's structure. Heavy atoms are often used in molecule design because they can influence the molecule's physical and chemical properties.

**Structure Type:** refers to the specific type of molecular structure that is being designed. There are two main types of molecular structures that are commonly considered in molecular design: sequence-based structures and molecular-based structures.

**The molecular formula:** is a shorthand notation used to describe the chemical composition of a molecule. It consists of the symbols of the elements present in the molecule and the number of atoms of each element, with subscripts to indicate the number of atoms. For example, the molecular formula for water is H2O, indicating that the molecule contains two hydrogen atoms and one oxygen atom.

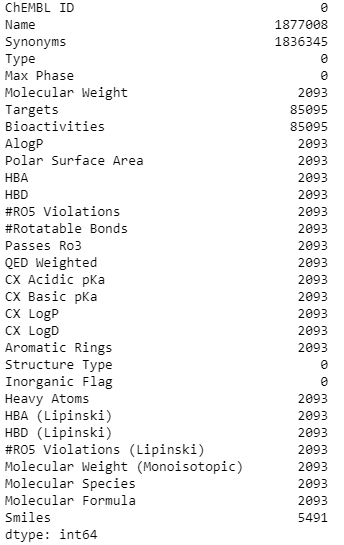
**Smiles:** **SMILES (Simplified Molecular Input Line Entry System)** is a chemical notation that allows a user to represent a chemical structure in a way that can be used by the computer. SMILES is an easily learned and flexible notation. The SMILES notation requires that you learn a handful of rules. You do not need to worry about ambiguous representations because the software will automatically reorder your entry into a unique SMILES string when necessary.

## **2.4 Verifying data quality**

**Detecting missing values in our data:**

**T**he total number of NaN data present was counted and the number of NaN or missing values in each column was found.

* Total number of NaN values: **3932987**



**Figure 11: Missing values detection**

**Detecting incorrect values in our data:**

Looking at unique values in categorical columns by printing them and checking if there are incorrect values like ”nan” :

* **#RO5 Violations has ['2' '0' '1' nan '3' '4'] values**
* **Passes Ro3 has ['N' ‘nan’ 'Y'] values**
* **HBA has ['15' '5' '4' '3' ‘nan’ '7' '8' '1' '2' '6' '12' '9' '14' '0' '10' '18' '11'**
* **'16' '13' '19' '17' '20' '21' '22' '27' '24' '23' '26' '25' '32' '28'**
* **'31' '30' '29'] values**
* **Bioactivities has [ ‘nan’ 4. 5. ... 2517. 1263. 2528.] values**
* **HBD has ['1' '2' ‘nan’ '3' '5' '0' '4' '14' '6' '10' '7' '8' '12' '11' '9' '17' '13'**
* **'15' '16' '19' '18' '20' '25'] values**
* **#RO5 Violations has ['2' '0' '1' ‘nan’ '3' '4'] values**
* **Molecular Species has ['NEUTRAL' 'BASE' ‘nan’ 'ZWITTERION' 'ACID'] values**

## **2.5 Conclusion**

The data comprehension stage, in which we explored the data, described it, and verified its quality, was completed by identifying missing, incomplete, or incorrect data.

# **Chapitre 3**

# **Data Preparation**

## **3.1 Introduction**

Data preparation may be one of the most difficult steps in any deep learning project as it is one of the key players in developing high-quality deep learning models.

It allows us to explore, clean, combine, and format data for sampling and deploying them.

It is essential as most deep learning algorithms need data to be in numbers to reduce statistical noise and errors in the data, etc.

In this topic, A data transformation will be discovered and the importance of data preparation will be learned.

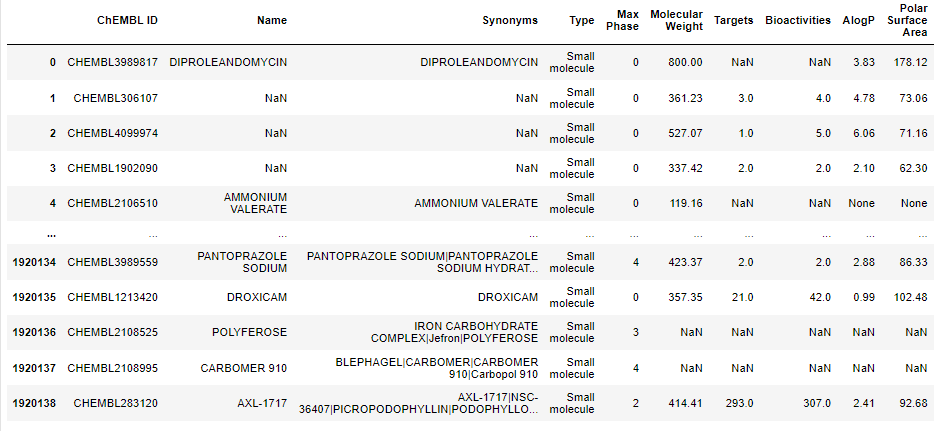
## **3.2 Data cleaning**

Data cleaning and validation techniques help determine and solve inconsistencies, outliers, anomalies, incomplete data, etc.

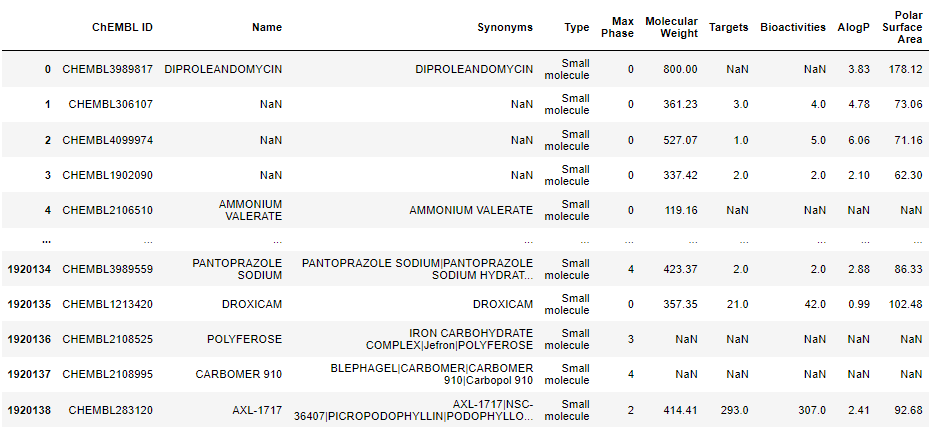
Clean data helps to find valuable patterns and information in data and ignores irrelevant data in the datasets. It is very much essential to build high-quality models, and missing or incomplete data is one of the best examples of poor data.

Since missing data always reduces prediction accuracy and performance of the model, data must be cleaned and validated through various imputation tools to fill incomplete fields with statistically relevant substitutes.

So in the first case, the start of replacing all of the ‘None’ and ‘NONE’ existing in the dataset with NaN so it will be easier to discover the missing values in order to deal with them.



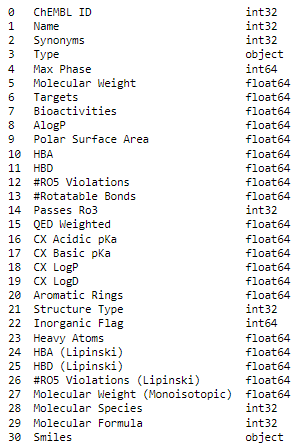
**Figure 12 : Result before doing the replacements**

****

**Figure 13: Result after doing the replacements**

## **3.3 Data transformation**

Then comes the phase of converting necessary columns to numerical type since they were all ‘objects’.



**Figure 14: converting features to numerical type**

* It was proceeded to convert the columns to their appropriate data type using the **'to\_numeric'** function.
* The categorical features were imputed using the 'mode' function.
* The numerical features were also imputed using the 'median' function.



**Figure 15: Verification after the conversion**

## **3.4 Data splitting**

Data splitting is an essential step in machine learning and data analysis that helps to ensure that the model is accurate, reliable, and can generalize to new, unseen data. By using separate subsets of the data for training, validation, and testing, we can improve the generalization of the model, identify potential problems, and fine-tune the model to achieve better performance.



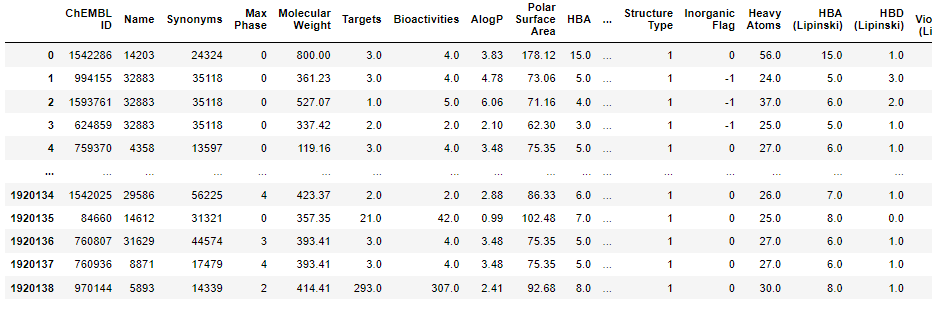
**Figure 16: splitting data**

## **3.5 Data encoding**

Data encoding is the process of transforming data from one representation to another. This is often necessary in data analysis and machine learning because different algorithms and models require different formats and types of data. Data encoding involves converting data into a specific format or representation that can be used by the algorithm or model being applied to the data.

For that, The categorical columns that needed to be converted were encoded using the LabelEncoder.

* Chembl Id
* Name
* Synonyms
* Passes Ro3
* Structure Type
* Molecular Species
* Molecular Formula



**Figure 17: Visualization of the features encoded**

The phase was concluded by deleting the column 'Type', which had the same value in all the rows of the data, and it was chosen to be deleted.

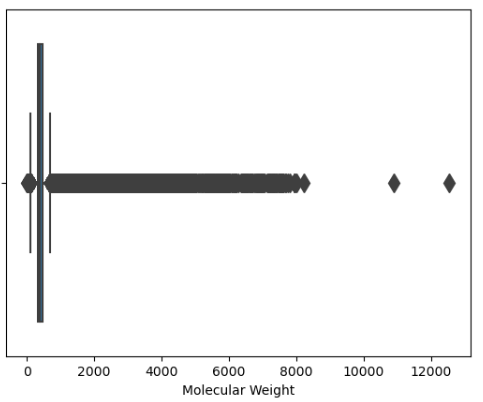
## **3.6 Data preprocessing**

Dealing with outliers is an important step in the data preprocessing phase.

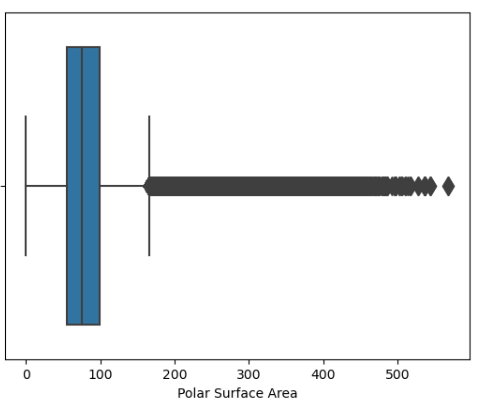
The identification of outliers can be demonstrated using statistical methods such as box plots and by calculating the IQR (interquartile range) of the data.

The first step that was taken and will be demonstrated is:

* The first thing that was done was visualizing the outliers via boxplots, as shown in the image below:



**Figure 18: Visualization molecular weight’s boxplot**

****

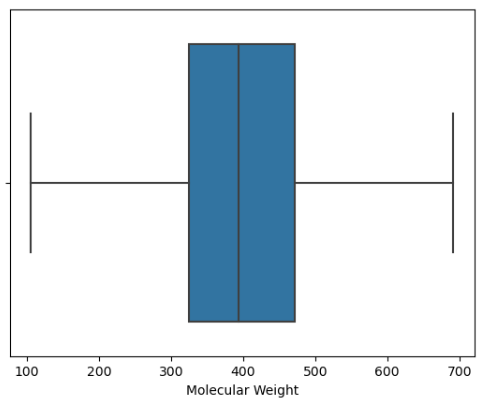
**Figure 19: Visualization Polar Surface Area’s boxplot**

The same process was then carried out with the following features: Targets, Bioactivities, AlogP, HBA, HBD, #RO5 Violations, #Rotatable Bonds, QED Weighted.

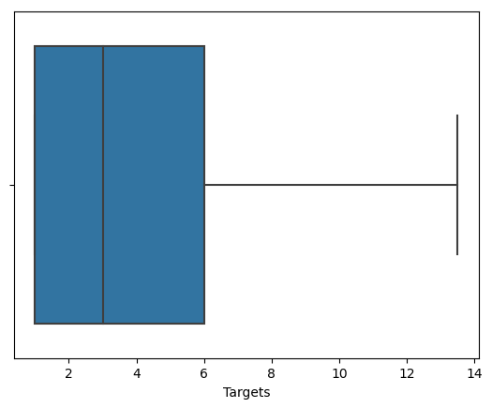
* Secondly, the outliers were dealt with as follows:

It is important to note that dealing with outliers should be done carefully and with a clear understanding of the data and the analysis being performed. In some cases, outliers may actually be valid data points that are important to the analysis, and in such cases, they should not be removed or transformed.

The IQR method was chosen for dealing with the outliers.



**Figure 20: Applying IQR for Molecular Weight**

****

**Figure 21: Applying IQR for Targets**

The same approach was applied to other features as well.

## **3.7 Conclusion**

All the methods used for preparing the data have been defined in this chapter, and as a result, the data has been cleaned and prepared for the next step, which is the modeling phase.

The following chapter will define all the algorithms that were used.

# **Chapitre 4**

# **Modeling and Evaluation**

## **4.1 Introduction**

Various deep learning techniques that are commonly used for developing predictive models for molecules will be explored in this chapter.The basics of molecular representations will be discussed first, including how molecules are represented as input data for deep learning models.

The different types of deep learning algorithms used for molecular modeling, including various deep learning techniques, will be explored next, followed by a discussion on the evaluation of the models in the final step.

## **4.2 Used Methods**

* Molecular fingerprints: As previously discussed, molecular representation encodes the presence or absence of certain chemical features or substructures in a molecule.
* SMILES (Simplified Molecular Input Line Entry System): is molecular representation that encodes the structure of a molecule as a string of characters, it was already present in our dataset.

Both of these methods were used as inputs to some of our modeling algorithms.

## **4.3 Models**

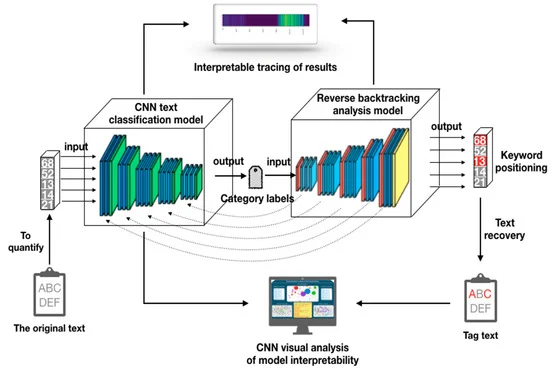
### **4.3.1 Tested models :**

Multiple models were experimented with during the modeling phase to discover the optimal solution with high precision.Nonetheless, not all of the models that were tried worked effectively, and several issues and errors were encountered with some of them.

As a result, it was necessary to discard those models and choose alternative ones that performed better for our task.

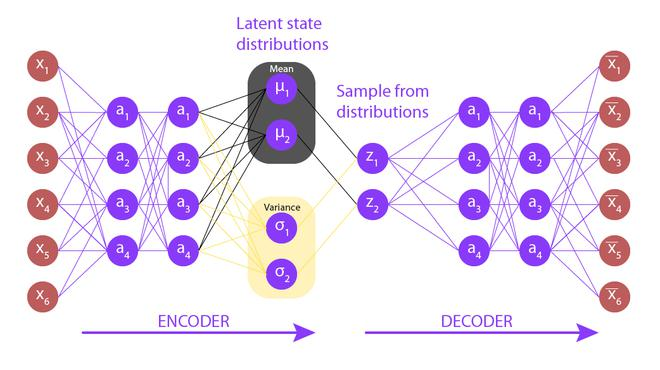
CNN: Regarding the CNN model, encoding of the input molecule is required, but a significant challenge was encountered during the decoding process.In order to decode the encoded molecule, the original SMILES string is required. However, the first SMILES string obtained was already encoded.This made it difficult to extract the necessary information for decoding,

Therefore, a workaround had to be developed in order to obtain the required SMILES string.



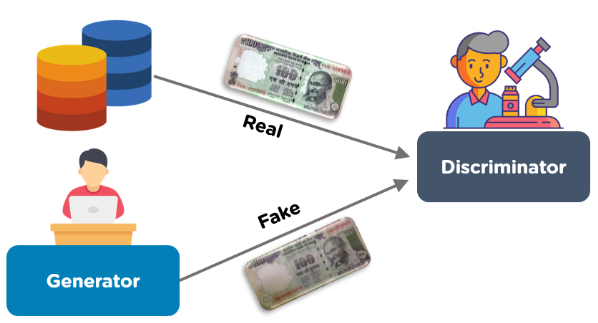
**Figure 22: CNN architecture**

VAE: The VAE was attempted to be used for autoencoding with decoding, but the process became increasingly complex and the issues encountered were unable to be resolved. Despite our efforts, the problem was not successfully solved, and alternative approaches had to be explored for the modeling task.



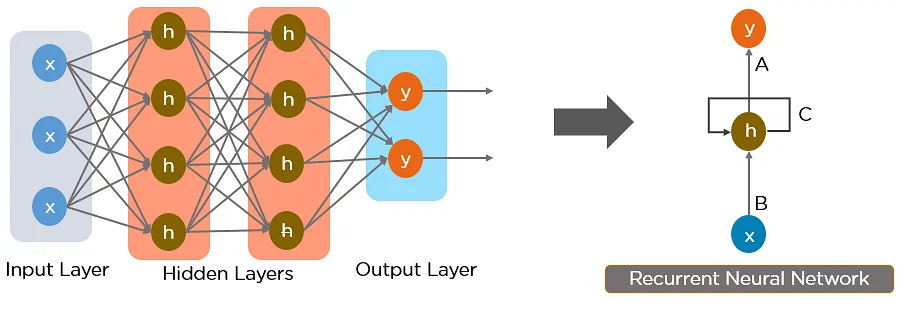
**Figure 23: VAE architecture**

GANS: Working with GANs can be challenging due to the significant computational resources and data required for effective training. It was found that the amount of resources required for training with GANs was impractical in our case, and the desired prediction results could not be achieved.



**Figure 24: Gans architecture**

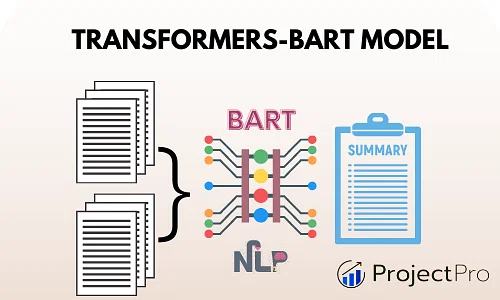
RNN: Generating SMILES strings with RNNs can be challenging due to the complexity of the chemical structures involved. RNNs need to learn the patterns and relationships between the molecular components in order to generate valid SMILES strings, which can be a difficult task.



**Figure 25: RNN architecture**

Bart:(Bidirectional and Auto-Regressive Transformers) model is a sequence-to-sequence model introduced by Facebook AI in 2019. It is based on the Transformer architecture and is pre-trained using a combination of denoising autoencoder and sequence-to-sequence language modeling objectives.

BART can be used for a variety of natural language processing tasks, including text summarization, question answering, machine translation, and text generation. It is capable of handling both autoregressive and non-autoregressive generation, which makes it more efficient than traditional sequence-to-sequence models that use autoregressive decoding.



**Figure 26: Bart architecture**

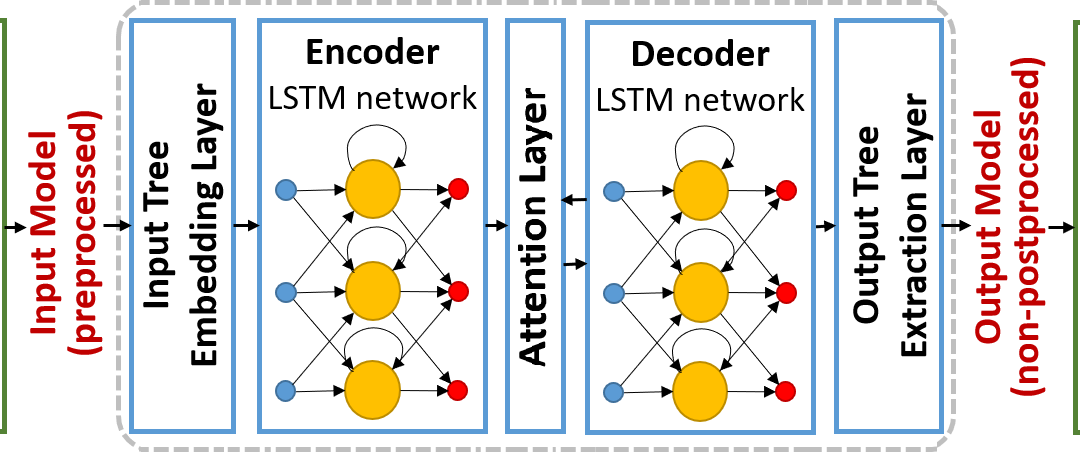
An attempt was made to generate SMILES representations of chemical compounds using the pre-trained BART model, but it was found that for a specific input sequence, the output SMILES generated by the model was invalid. The BART model was not able to be used for our prediction task due to the invalid output SMILES generated for a specific input sequence.

* This is the output:

****

**Figure 27: Exemple D’output BART**

MoLSTM: stands for Molecular Long Short-Term Memory, which is a type of neural network architecture used for generating new molecules with desired properties. It is based on the Long Short-Term Memory (LSTM) architecture, which is a type of recurrent neural network that can process sequences of data and remember information for long periods of time. In the case of MoLSTM, the input to the network is a molecular graph or a SMILES string that represents a molecule, and the output is a new molecular graph or SMILES string.



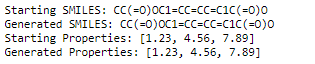
**Figure 28: LSTM architecture**

The code that works with the Molstm model will now be explained.

A MolLSTM (Molecule LSTM) model was defined and trained initially to predict molecular properties using SMILES string inputs. The trained model was then saved for future use. Specifically, the code first converts the input data into a format that can be used by the MolLSTM model, and defines the model architecture with specific parameters such as the maximum length of the SMILES strings and the number of latent dimensions. The model is then compiled with an optimizer and loss function, and trained on the input data for a specified number of epochs and batch size. Finally, the trained model is saved as a file with the extension ".h5".

After the MolLSTM model was trained and saved, a function was defined for generating new SMILES strings using the pre-trained model. The function takes a starting SMILES string and uses the MolLSTM model to predict the next characters in the sequence, based on the probabilities of each possible character. The function then appends the predicted characters to the starting string and continues the process until the desired length is reached. The generated SMILES string is returned as output.

This is example of the output:



**Figure 29: Molstm output**

MolGen: MolGen algorithms can be trained on large datasets of molecular structures and their corresponding properties, and can learn to generate new molecules that have similar properties to those in the training data. These algorithms can be used to accelerate the drug discovery process by generating new drug candidates that have a higher likelihood of being effective, or to design new materials with specific properties, such as high strength or thermal conductivity.

MolGen didn’t give us the correct form of smiles.

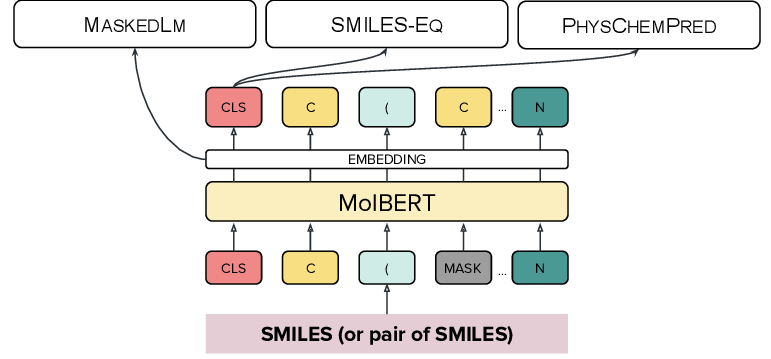


**figure 30:MolGen output**

## **4.3.2 Used models :**

MolBERT : is a pre-trained language model developed by researchers at the University of Washington that is specifically designed for processing molecular data. It is based on the popular BERT (Bidirectional Encoder Representations from Transformers) model, which is widely used for natural language processing tasks, but has been adapted and fine-tuned for molecular data.

MolBERT is trained on a large corpus of molecular data, including SMILES strings and molecular properties, and can be fine-tuned for specific molecular prediction tasks, such as predicting the properties of new molecules or screening for potential drug candidates.



**figures 31: MolBERT architecture**

Initially, Our specific dataset was used to train and fine-tune the MolBERT model, allowing it to learn and gain pre-training from the data

The code that works with the MolBERT model will be explained now.

This code defines two functions, predict\_properties() and generate\_smiles(), that can be used to predict the properties of a given molecule and generate new molecules that have similar properties.

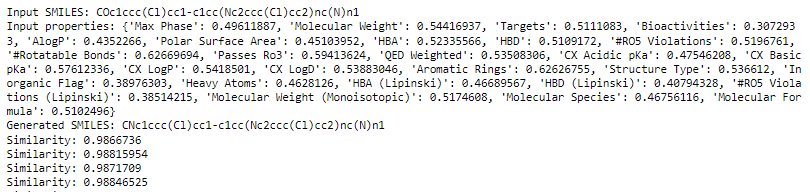
The predict\_properties() function takes a SMILES string (a string that represents the structure of a molecule) as input and returns a dictionary of predicted properties for that molecule. These properties include the maximum phase, molecular weight, targets, bioactivities, AlogP, polar surface area, and other molecular descriptors.

The generate\_smiles() function takes as input a dictionary of predicted properties for a molecule, a SMILES string for the original molecule, and a threshold value that determines how similar the generated molecules should be to the original molecule in terms of their predicted properties.

The function then generates new SMILES strings by mutating the original SMILES string and predicting the properties of the new molecules using predict\_properties(). If the predicted properties of a new molecule are sufficiently similar to the predicted properties of the original molecule , the new SMILES string is added to a list of generated SMILES strings. The function returns the list of generated SMILES strings.

The code then uses these functions to predict properties and generate new molecules for each molecule in our dataset. For each molecule, the code first predicts its properties using predict\_properties(). It then generates 10 new molecules with similar properties using generate\_smiles(). Finally, it calculates the cosine similarity between the predicted properties of the original molecule and the predicted properties of each generated molecule, and prints these similarities.

this is an example of the output of our model:



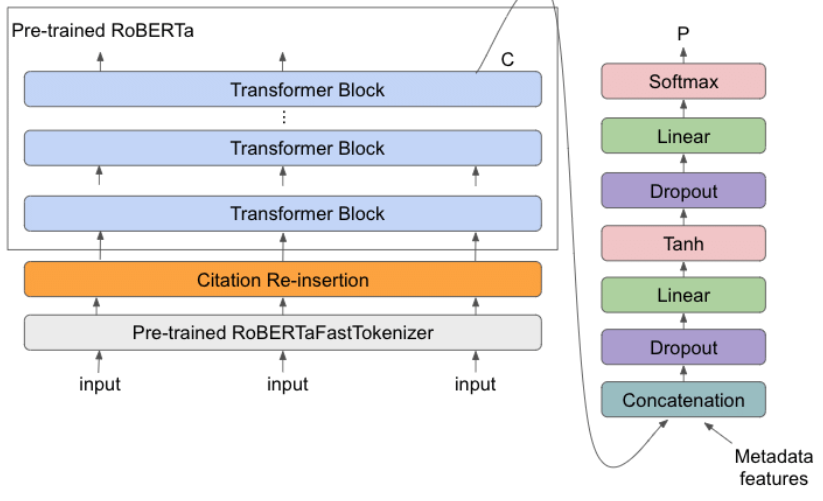
**figure 32:MolBERT output**

* RoBERTa:(Robustly Optimized BERT Pretraining Approach) is a type of deep learning model that uses a transformer architecture, specifically based on the transformer model introduced by Vaswani et al. (2017).

More specifically, RoBERTa is a variation of the BERT (Bidirectional Encoder Representations from Transformers) model that uses an improved pre-training procedure and more training data to achieve state-of-the-art results on a wide range of natural language processing (NLP) tasks.

The RoBERTa model is capable of taking as input various types of data, including natural language text, as well as structured data such as SMILES (Simplified Molecular Input Line Entry System) and predicted properties.

When used with SMILES and predicted properties, the RoBERTa model can be applied to a variety of tasks in cheminformatics, including drug discovery.



**Figure 33: Roberta architecture**

In the next paragraph an explanation for the RoBERTA model will take place:

First, the necessary libraries are imported, including NumPy, PyTorch, the RoBERTa tokenizer and model from the Transformers library, and the cosine\_similarity function from scikit-learn. The RDKit library is also imported for working with SMILES strings.

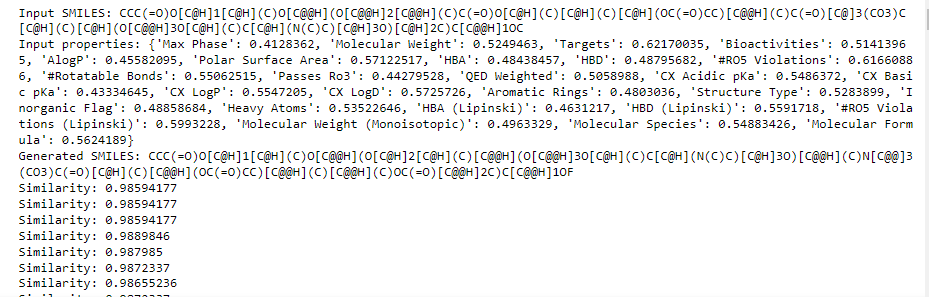
The RoBERTa model is then loaded, along with its tokenizer, using the "from\_pretrained" method. A function called "predict\_properties" is defined, which takes a SMILES string as input and uses the tokenizer and model to predict 29 different properties for the molecule. The properties and their corresponding predictions are returned as a dictionary.

A second function called "generate\_smiles" is defined, which takes the predicted properties from "predict\_properties", the input SMILES string, and optional threshold and maximum attempts values. The function generates up to 10 mutated SMILES strings with similar properties to the input using a random mutation strategy. The mutated SMILES strings are then compared to the predicted properties of the input using cosine similarity, and any SMILES strings with a similarity score greater than the threshold value are returned.

Finally, an example usage of these functions is provided in a for loop that iterates through a dataset of SMILES strings. For each SMILES string, the predicted properties are computed using "predict\_properties", and then up to 10 new SMILES strings are generated with similar properties using "generate\_smiles". The input SMILES string, input properties, generated SMILES strings, and similarity scores between each generated SMILES string and the input properties are printed to the console.

* The synthesis phase, which refers to the step where a mutated SMILES string is generated from the input SMILES string, was then integrated, and the feasibility of synthesizing the compound represented by the mutated SMILES was calculated.
* Then the meta synthesis phase as well , to check if the generated smiles can be converted back to its primary form or not.

Here is what the output looks like:



**Figure 34: output model**

## **4.4 Evaluation**

In the evaluation phase, similarity can be used to compare the similarity between two models.

One way to do this is to compare the predicted properties of the models with the input properties using **cosine\_similarity** method.

If the predicted properties of the two models are similar to the input properties, it can be an indication that the models are performing well.

In this project’s case, the similarity’s scores were compared between the two models: RoBERTa and MolBERT.

-> **Deduction:** MolbERT in this case, suits well the problematic and meets well its requirements, based on its cosine\_similarity as shown in the two pictures below:

In the next phase which is the deployment phase, MolBERT is going to be the chosen model.



**Figure 35: MolBERT’s Score**



**Figure 36: RoBERTa’s Score**

## **4.5 Conclusion**

Overall, this chapter provided an introduction to the field of molecular modeling and demonstrated how machine learning can be used to generate new molecules with specific properties.

# **Chapter 5:**

# **Deployment phase**

## **5.1 Introduction**

The last phase of a Data Science project is the deployment phase.

In this part of the report, an interface for generating predicted molecules would be presented.

The subject of project deployment and its realization in the form of results given just after having evaluated the various stages as well as the predefined processes would be presented.

## **5.2 Concept and deployment strategy**

Deploying deep learning models is the process of making models available in production environments, where they can provide predictions to other software systems. It is only after the model is deployed in production that they begin to add value to the customer, which makes deployment a critical step.

In our case the deployment plan will be a web interface that offers the following functionalities:

• An interface that contains a form for entering properties and Smiles information.

• An interface that contains the result of the model prediction

## **5.3 Deployment environment(tools):**

The main deployment tools used in this part are:

Django: a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so your application could be written without needing to reinvent the wheel since It’s free and open source.

## **5.4 the web application:**

The model will allow users to insert all the information about the molecule’s properties.

Among these informations, **'Max Phase', 'Molecular Weight', 'Targets', 'Bioactivities', 'AlogP', 'Polar Surface Area', 'HBA', 'HBD', '#RO5 Violations', '#Rotatable Bonds', 'Passes Ro3', 'QED Weighted', 'CX Acidic pKa', 'CX Basic pKa', 'CX LogP', 'CX LogD', 'Aromatic Rings', 'Structure Type', 'Inorganic Flag', 'Heavy Atoms', 'HBA (Lipinski)', 'HBD (Lipinski)', '#RO5 Violations (Lipinski)', 'Molecular Weight (Monoisotopic)', 'Molecular Species', 'Molecular Formula’** would be mentioned.

## **5.4.1 Web application:**

The main interface would have a form where the user can input the properties of the molecule they want to predict.

The properties could include structural information as well as chemical properties, as stated earlier.

Once the user inputs the properties and the smiles, the interface would use a deep learning model (in this case, it’s MolBERT ) to predict the chemical structure of the new smiles.

The interface could also display additional information about the predicted molecule, such as its predicted properties.

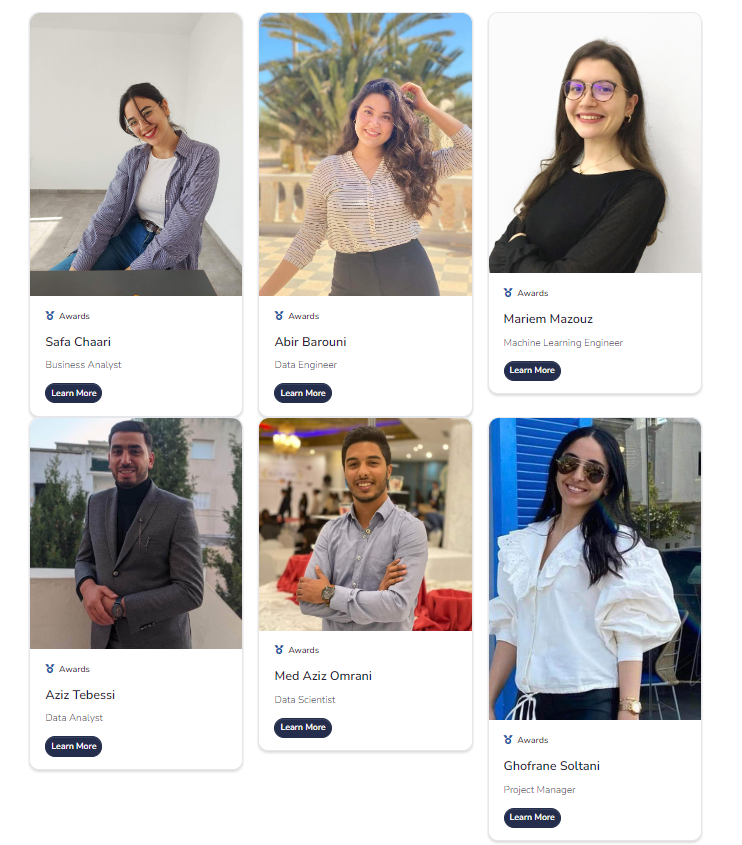
Finally, the interface could allow the user to save the predicted molecule to a database or export it to a file for further analysis.

This would allow the user to use the predicted molecule in downstream applications, such as drug discovery or materials science.

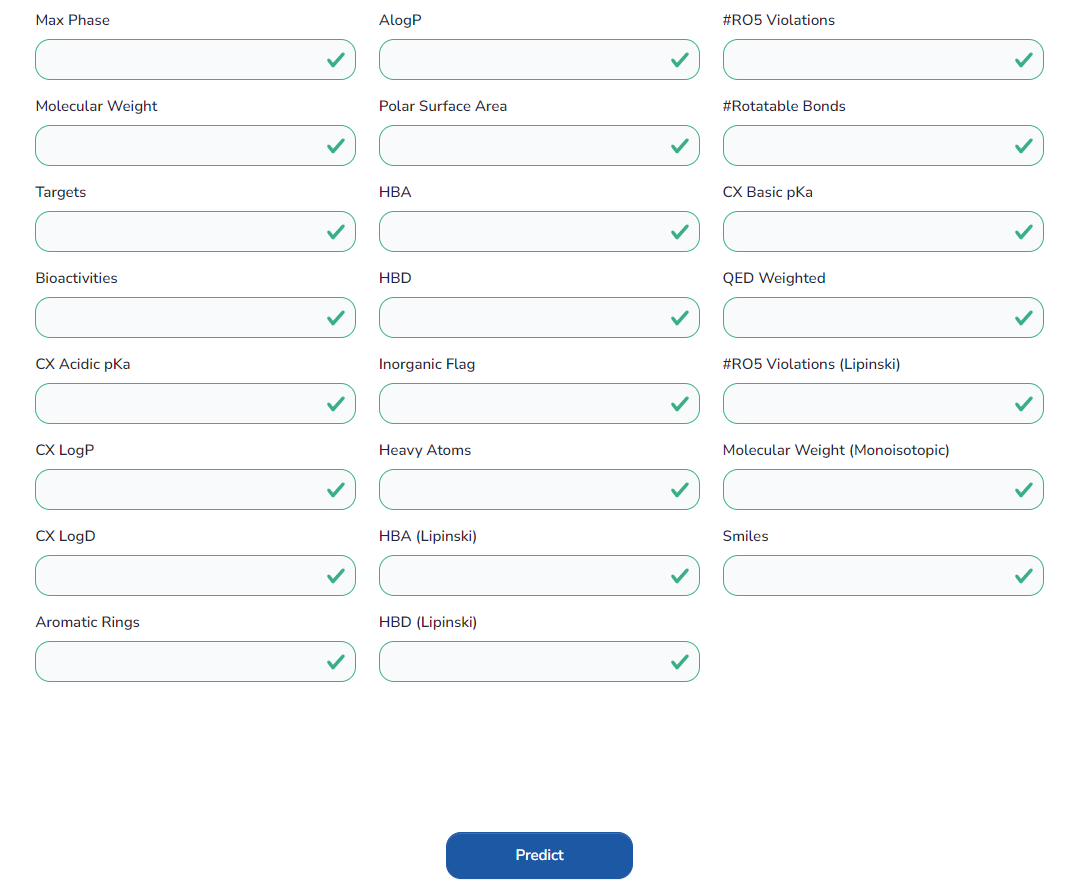
## **5.4.1.1 Interface of the molecule prediction:**

The following figure shows the interface of the form to be completed with the various customer information.

After submitting the form, our model created in the modeling phase will be loaded in the background and will do the necessary prediction to output as a result,a molecule prediction.

5

**Figure 37: Interface of the team members of molminds company**

****

**Figure 38: Interface of molecular prediction**

****

**Figure 39: Interface of the prediction’s result**

## **5.5 Conclusion**

In the IBM Master Plan methodology, the deployment phase is the final stage of the process.

The models developed were highlighted and the technologies used to put our results into production were specified. The technologies used were specified and the interactions with the system were described in order to utilize the results of the models.

# **Chapter 6:**

# **Perspectives and contributions**

## **6.1 Perspectives**

Predicting new smiles from existing ones using properties can have several perspectives, including:

1. Drug discovery: By predicting new smiles from existing ones, researchers can identify new molecules with specific properties that could be used as potential drug candidates. This can accelerate the drug discovery process by reducing the time and cost associated with traditional experimental methods.
2. Chemical synthesis: Predicting new smiles from existing ones can also aid in chemical synthesis by identifying new molecules with desirable properties that can be synthesized in the laboratory. This can help chemists to design more efficient and cost-effective synthetic routes.
3. Materials science: Predicting new smiles from existing ones can also have applications in materials science by identifying new molecules with specific properties that can be used as building blocks for the development of new materials.
4. Environmental science: Predicting new smiles from existing ones can also have applications in environmental science by identifying new molecules with specific properties that can be used in the development of new sustainable technologies.
5. Machine learning: Predicting new smiles from existing ones can also help to advance machine learning algorithms by providing large datasets for training and testing. This can lead to the development of more accurate and efficient predictive models.

## **6.2 Academic and professional contributions**

On an academic level, the project we did allowed us, among other things, to take part in the implementation of this deep learning project using an example of an existing and real database.

This project allowed us to concretely use what we learned theoretically and this by applying all the learning outcomes seen in class and from our own research.

Thanks to this experience we had the chance to strengthen our knowledge in the sectors affected by the project and above all to better understand the system and the prerequisites of the work through Data exploration .

It gave us the opportunity to learn new technologies and be more efficient in the field of machine learning and Web development , with the possibility of seeking and finding better solutions, always more optimal for the ambiguities encountered.

This allows us to better appreciate the versatility and the interest of the engineer-manager training that we had at ESPRIT.

Professionally speaking , the project allowed us to understand in depth the role of a data scientist and discover this huge world of data and modeling.

This gave us the opportunity to gain practical experience both in the elements of business and data science.

**General Conclusion:**

In conclusion, molecule prediction is a rapidly growing field.

As more data becomes available and Data Science techniques continue to advance, molecule prediction will become an increasingly powerful tool for scientists and engineers working to solve some of the most pressing challenges of our time.

Molecule prediction and data science will lead to accelerate the drug discovery process by predicting which molecules are likely to have the desired properties,

to help reduce the cost and time required by focusing on the most promising candidates from the start , to allow the customization of molecules properties and to help reduce the risk of adverse side effects in patients.

Overall, molecule design has the potential to revolutionize drug discovery and the development of new treatments for diseases by providing a faster, more cost-effective, and customized approach to molecule discovery.