esprit
Se former autrement
HONORIS UNITED UNIVERSITIES

MOLECULE DESIGN FOR DRUG DISCOVERY

Members: Aymen Khiari

Oussama Boussetta

Anas Ben Brahim

Ayoub Mouelhi

Rihab Ben salem

Supervisor: Dorra Trabelsi

Emna Ben Mansour

2022/2023

OPENORA
Where innovation meets openness

Table of content

Table of figures

List of tables

Abstract

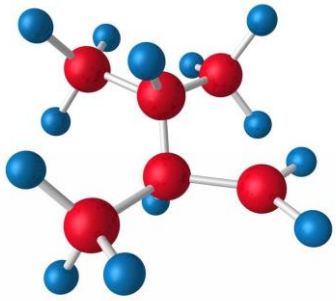
This report presents a systematic approach to molecule design that was developed to generate potential molecules based on specific client-requested properties. The research team explored the mechanics behind molecule design and provided a thorough retrosynthesis analysis to enable easy synthesis of the requested molecules. Despite encountering challenges in this novel and rapidly evolving domain, the team remained committed to delivering high-quality results and gained valuable insights into the field of molecule design. This paper contributes to advancements in the field and lays the foundation for future research in this area.

P.S In the world of machine learning and artificial intelligence, the quality and performance of models are highly dependent on the hardware used to run them. Unfortunately, not all individuals or organizations have access to the necessary resources to run these models efficiently. This lead to a range of difficulties, including slower processing times, inaccurate results, and a lack of scalability.

Chapter1: Business understanding

Introduction:

Molecule discovery refers to the process of identifying and isolating new chemical compounds and studying their properties and structure. This can involve synthesizing new compounds through chemical reactions, screening natural products, or using computational techniques to predict the properties of novel molecules. The goal of molecule discovery is to find molecules with desirable properties, such as activity against disease targets, that can be developed into drugs or other useful products.



1. Presentation of the project

Molecule discovery is a crucial step in the drug development process, as new molecules with specific properties can lead to the creation of new treatments for a variety of medical conditions. The discovery process can involve a combination of laboratory experimentation, computational modelling, and high-throughput screening techniques.

In the laboratory, scientists use synthetic chemistry to create new compounds, modify existing ones, or isolate naturally occurring molecules. They then evaluate the properties and activities of these molecules to determine their potential for further development.

In addition to drug development, molecule discovery can also lead to the creation of new materials, agrochemicals, and other products. For example, new molecules with specific properties, such as high melting points or low flammability, can be used to create advanced materials with improved performance.

The discovery of new molecules is also important for expanding our understanding of the natural world and how chemical compounds interact with each other and with biological systems. This knowledge can lead to new insights into the underlying mechanisms of disease, as well as new ways to diagnose and treat a wide range of medical conditions.

Molecule discovery is a complex and multi-disciplinary field, requiring expertise in synthetic chemistry, biology, pharmacology, and computer science. Collaboration between scientists and researchers from different backgrounds is often essential for making important discoveries and developing new products.

Overall, the field of molecule discovery is constantly evolving, with new technologies and techniques being developed to facilitate the identification of new and useful compounds. Its significance in shaping the future of medicine, materials science, and many other fields cannot be overstated.

1.1 Study of the existent:

The purpose of this section is to conduct a study of the existing literature, reports, and data related to the topic at hand, to establish a baseline understanding of the subject matter, identify gaps in knowledge, and guide future research and decision-making.

1.1.1 Traditional Process

The traditional process of molecule discovery typically involves several stages, including:

- Target identification: The first step in the traditional process of molecule discovery is to identify a target of interest. This can be a disease target, a material property, or any other desired property or activity.
- Lead identification: Once the target has been identified, the next step is to find a molecule that has the potential to interact with the target. This can be done through a variety of methods, including screening natural products, synthesizing new compounds, or using computational techniques to predict the properties of novel molecules.
- Lead optimization: The lead molecule is then modified and optimized to improve its properties and increase its activity against the target. This can involve making structural changes to the molecule, changing its solubility, or optimizing its pharmacokinetics.
- Pre-clinical testing: Once a promising molecule has been identified, it is subjected to pre-clinical testing to determine its safety and efficacy. This typically involves in vitro and in vivo testing in animal models.

- Clinical trials: If the molecule passes pre-clinical testing, it can then proceed to clinical trials in humans. This stage of the process is crucial for determining the safety and efficacy of the molecule in humans, as well as its pharmacokinetics.
- Regulatory approval: If the molecule is shown to be safe and effective in clinical trials, it can then be submitted for regulatory approval. This stage of the process can take

several years and requires significant investment, but it is essential for bringing new treatments to the market.

This traditional process of molecule discovery can be time-consuming and expensive, but it has the potential to lead to important breakthroughs in medicine and other fields.

1.1.2 Limit of the existent:

The traditional process of molecule discovery has several limitations, including:

- **Time and cost**: The process of discovering new molecules and developing them into safe and effective treatments can take many years and require significant investment. This can make it difficult for small companies and organizations to compete in the field.
- **Inefficiency:** The traditional process of molecule discovery can be inefficient, with many molecules being discarded at early stages of the process due to lack of activity or other issues. This can result in a significant waste of resources and time.
- **Limited target space:** The traditional process of molecule discovery is often limited by the target space, as it is difficult to identify and study all potential targets. This can result in missed opportunities for discovering new and innovative treatments.
- **Lack of diversity:** The traditional process of molecule discovery often relies on synthesizing and testing compounds that are like known active molecules. This can result in a lack of diversity in the types of molecules that are studied, limiting the potential for discovering new and innovative treatments.
- **High attrition rate:** The high attrition rate in the traditional process of molecule discovery, with many molecules failing at later stages of development, can result in significant financial losses for companies and organizations.

Despite these limitations, the traditional process of molecule discovery remains an important part of the drug development process, and new technologies and techniques are being developed to improve its efficiency and reduce the time and cost involved.

1.1.3 Proposed solution:

While traditional methods of drug discovery have proven successful, they are hindered by limitations as mentioned before.

However, recent advances in artificial intelligence (AI) and machine learning have enabled a shift towards more efficient and effective approaches to drug discovery, and molecule design.

Our approach involves developing Deep learning models that utilize molecular properties as input to generate a molecular structure represented as a string using SMILES[1] notation.

Once we generate those SMILES, we will apply a retrosynthesis[2] approach to explore multiple possible pathways that can lead to the target molecule.

1.2 Functional Requirement:

Functional requirements define the features and capabilities that a product or system must have to meet the needs of its users and stakeholders.

1.2.1 Business requirements:

1.2.1.1 Cost reduction:

Using some specific computational techniques we can reduce the amount of labor, time and lab process required for new molecule structures by using machine learning and artificial intelligence: Utilize machine learning algorithms to automate and optimize various stages of the design process, reducing the time and resources required to perform manual tasks.

1.2.1.2 Suggestions of New Molecules with specific proprieties

Harnessing the power of AI, the model can suggest or rather more specifically can predict new molecules based on what it learned. As the researcher can input the specific properties of a desired molecule which can be determined based of its characteristics.

Harnessing the power of AI, the model can suggest or rather more specifically can predict new molecules based on what it learned. As the researcher can input the specific properties of a desired molecule which can be determined based of its characteristics.

[1] SMILES: A standardized notation system used to represent organic molecules as a linear sequence of characters.

[2] Retrosynthesis: A well-established technique used in organic chemistry to design synthetic routes for complex molecules by breaking them down into simpler building blocks.

1.2.1.3 Accelerate the Process of discovering new Molecules:

Computers are fast, very fast, therefore utilizing a machine learning algorithm will be crucial for labs as it can save a huge amount of time by guiding the researchers to the right direction instead of excruciating the traditional process of trial and error.

To accelerate the process of discovering new molecules by having high-throughput screening to rapidly test large numbers of potential candidates, reducing the time required to identify promising molecules and automate and optimize various stages of the design process, reducing the time and resources required to perform manual tasks.

1.2.2 Data Science Requirements:

1.2.2.1 SMILES:

SMILES (Simplified Molecular Input Line Entry System) is a notation used to represent molecular structure using an ASCII string. The SMILES notation uses a simple set of rules to describe the atoms and bonds in a molecule, such as indicating the atomic symbol for each atom and using parentheses to indicate the connectivity between atoms.

Example: $C_{10}H_{10}N_2O_2S$

Figure 1 : SMILE Exemple

1.2.2.2 Retrosynthetic analysis:

Retrosynthesis, also known as retrosynthetic analysis, is a method used in the design of complex organic molecule synthesis. It's a reverse reaction that involves breaking down the

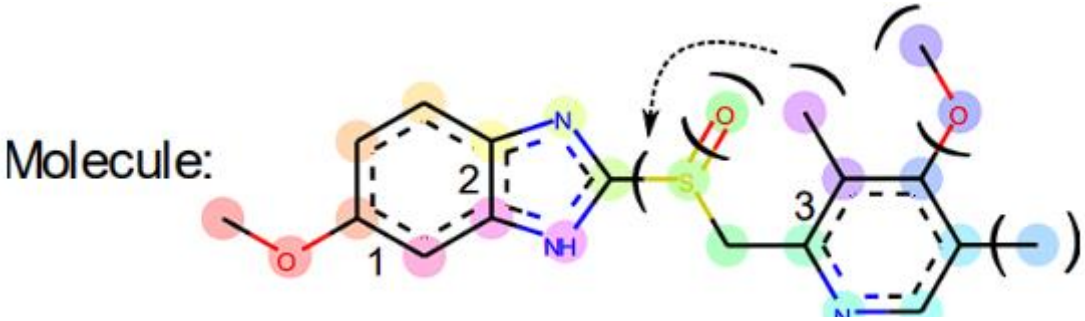


Figure 2 : Structural Representation

target molecule into a series of simpler structures, creating a pathway that leads our model to generate new molecules.

1.3 Non-functional requirements:

The non-functional requirements are not mentioned directly by the client but they're rather implied to be developed for various reasons.

In our case, these requirements are defined as follows:

- **Performance:** The models will process data (SMILEs) efficiently and return results within a specified time frame.
- **Scalability:** The model will be able to handle increasing amounts of data over time.
- **Usability:** The client side's user interface will be user-friendly and intuitive for researchers, laboratory scientists and end-users.
- **Maintainability:** The system will be easy to maintain and update over time.
- **Reliability:** The system must produce consistent and accurate results.
- **Availability:** The system must be always accessible to users.
- **Interoperability:** The system must be compatible with other systems and tools used by the organization.
- **Cost:** The project must be completed within budget constraints.
- **Compliance:** The system must comply with relevant regulations and standards, such as data protection laws.

Conclusion

In conclusion, the business understanding of the drug discovery process is essential for informed decision-making and successful outcomes. It involves functional and non-functional Requirements.

By considering these factors, pharmaceutical companies can navigate the complexities of drug discovery and increase their chances of bringing safe and effective treatments to market.

Chapter2: Data Understanding

Introduction

In this chapter focuses on the essential aspects of data in our project. We explore how to determine data requirements, collect relevant data, and gain a comprehensive understanding of it. By establishing clear data requirements, collecting appropriate data, and analyzing it effectively, we lay the groundwork for successful data-driven decision-making in subsequent stages of our project.

2.1 Data requirements

In the data requirement phase, we define the specific data needed to solve the problem at hand. This involves chemical property data, structural information, biological activity data and toxicity.

These data elements are crucial for assessing drug-likeness, understanding structure-activity relationships, predicting efficacy and safety, and optimizing molecule design.

By incorporating these data requirements into the drug discovery process, researchers can make informed decisions, prioritize promising molecules, optimize drug candidates, and accelerate the process.

2.1.1 Data Collection

In this phase, we will gather the necessary data by identifying data sources, creating a data collection plan, executing it to retrieve the data, validating its quality, and storing it securely to ensure that we have the relevant data needed for our modeling lately.

For our case, potential sources have been identified, including APIs and websites such as **Chembl, PubChem,** and **Zinc** as mentioned in the pictures below:

Figure 4 : ChemBL

Figure 3 : PubChem

Figure 5 : Zinc

These platforms provide access to a wide range of small molecule data, its proprieties as well as their SMILES representation which can be collected via downloading it or using web scraping in different type of files to be used later to achieve our objectives.

After working on data collecting, we obtain those results from different websites with 28 features which can help us to enhance the quality of the outcomes:

Table 1 : data sources' quantity

ChEMBL	PubChem	ZINC
2.3 million **rows**	9 million rows	300K Smiles representation

To make it more flexible we used MongoDB which supports the collection of different file types by offering a data model.

It allows us the storage of various formats, such as JSON, XML, images, videos, and documents, making it suitable for managing diverse file types for a unified database system for a better data understanding.

2.1.2 Data Understanding

Understanding the features of the data is essential for selecting appropriate statistical and machine learning models, as well as for performing data cleaning, feature engineering, and data visualization tasks. By analyzing each feature in detail, including its distribution, range, mean, median, and standard deviation, we can gain insights into the characteristics of the data and identify potential issues such as missing values, outliers, or data inconsistencies. Furthermore, by examining the relationships between different features and the target variable, we can determine which features are most important for predicting the outcome of interest and develop a predictive model that accurately captures the underlying patterns in the data.

Table 2 : Features understanding

Feature	Description
ro5	Rule of 5 - guideline for evaluating drug-like properties of a molecule
ro5(lipinski)	Lipinski's Rule of 5 - like the Rule of 5, but more widely used
rotatable bonds	Number of bonds in the molecule that can rotate freely
algop	A measure of a molecule's lipophilicity
aromatic rings	Presence of cyclic structures with alternating double bonds
Bioactivities	Observed biological activities of the molecule
CX Acidic pKa	A measure of the molecule's acidity
CX Basic pKa	A measure of the molecule's basicity
CX LogD	A measure of the molecule's distribution between water and a lipid-like environment
CX LogP	A measure of the molecule's lipophilicity
HBA	Number of hydrogen bond acceptors in the molecule

Feature	Description
HBA (Lipinski)	Number of hydrogen bond acceptors according to Lipinski's Rule of 5
HBD	Number of hydrogen bond donors in the molecule
HBD (Lipinski)	Number of hydrogen bond donors according to Lipinski's Rule of 5
Heavy Atoms	Number of non-hydrogen atoms in the molecule
Inorganic Flag	Binary flag indicating whether the molecule contains inorganic atoms
Max Phase	Maximum phase of clinical trials the molecule has reached if any
Molecular Formula	Chemical formula of the molecule
Molecular Species	Whether the molecule is neutral, an anion, or a cation
Molecular Weight	Mass of the molecule
Molecular Weight (Monoisotopic)	Mass of the molecule using the most abundant isotope for each element
Passes Ro3	Binary flag indicating whether the molecule passes the Rule of 3
Polar Surface Area	A measure of the surface area of the molecule that is polar or charged
QED Weighted	A measure of the drug-likeness of the molecule based on a quantitative estimate of drug-likeness
Smiles	A standardized string of characters representing the structure of the molecule
Structure Type	The type of molecule, such as a protein, nucleic acid, or small molecule
Targets	Known or predicted targets of the molecule, such as proteins or receptors
Type	The type of the molecule, such as a drug candidate or reference compound

2.1.3 Data preparation

The first step in the data science process is data understanding. This involves gaining a thorough understanding of the data that will be used for analysis. By doing so, data scientists can identify any issues with data quality, detect errors and anomalies, and understand how the data is distributed. This step is crucial for ensuring the accuracy and reliability of the analysis.

2.1.3.1 SMILE VALIDATION

validating the SMILES strings in the dataset. SMILES stands for Simplified Molecular Input Line Entry System and is a notation system for representing the chemical structure of molecules. After that a check for duplicate SMILES in necessary as it will severely increase the learning curve for the model

- ⇨ In this dataset there was only one invalid SMILE

2.1.3.2 Drop unrelated features

Table 3 : Unrelated features

ChEMBL ID	An index id used by the Chembl database engine system.
Name	The name of each compound
Synonyms	A synonym of that name of the compound
Inchi Key	A different type of representing a molecule which wasn't used in this project

2.1.3.2 Plots

We plotted histogram of features to find out a distribution law:

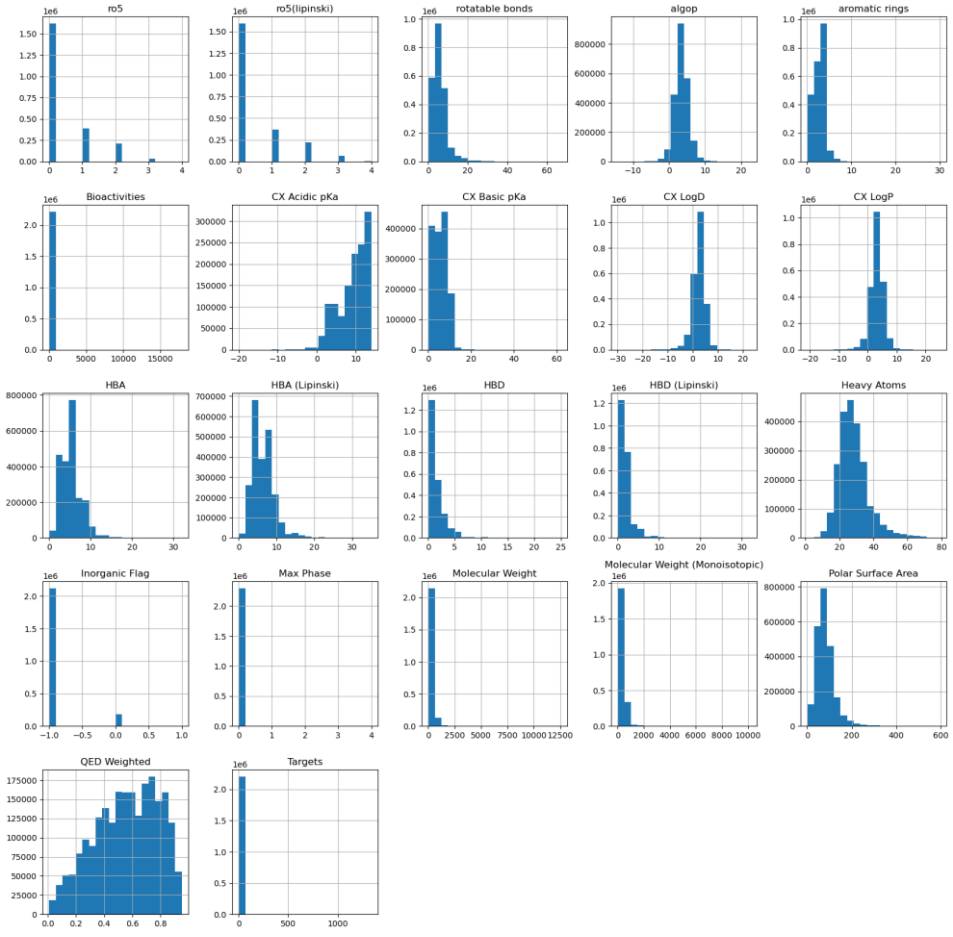


Figure 6 histogram of various features

⇨ Unfortunately, no feature seemed to follow a distribution law.

2.1.3.3 Imputing NA values

a) Imputing the NA values in with same index:

Handling missing data is an important part of data pre-processing in machine learning. When many columns have missing values in the same rows, it can be challenging to decide how to handle these missing values. One approach is to eliminate the rows that contain missing values. This approach can be effective if the number of rows with missing values is relatively small compared to the total number of rows in the dataset. However, it is important to carefully consider the impact of removing these rows on the quality and quantity of the data.

b) Imputing pka acidic & pka basic

In our analysis, we utilized the features algop, HBA, HBD, and Polar Surface Area to impute the values of pKa acidic and pKa basic. These features were selected based on the observation that they typically have an undirected relation with both pKa acidic and pKa basic. This means that while the values of these features may be associated with changes in the pKa values, it's unclear whether the features themselves directly cause the changes in pKa or whether other underlying factors are responsible for the observed patterns. Despite this uncertainty, we found that incorporating these features in the imputation process helped to improve the accuracy of the imputed pKa values. However, further investigation and analysis would be necessary to fully understand the nature of the relationship between these variables and to explore other potential features that could be used to improve the imputation accuracy even further.

- ⇨ We then incorporate **a linear model** to impute **pka basic** and **pka acidic**

c) Imputing CX LogD & CX logP

In this analysis, we aimed to impute the missing values of `CX LogD` and `CX LogP` in our dataset using a linear regression model. First, we split our data into two sets: a training set that contained all the rows where both `CX LogD` and `CX LogP` were present, and a test set that contained all the rows where at least one of these variables was missing.

We then defined a set of features that we believed might be useful in predicting the missing values of `CX LogD` and `CX LogP`. We used these features to train a linear regression model on the training set.

Next, we used the trained model to predict the missing values of `CX LogD` and `CX LogP` in the test set. To do this, we used the `SimpleImputer` function from the `sklearn.impute` module to fill in the missing values in the test set with the mean value of the corresponding feature

across the training set. We then passed the resulting imputed test set through the linear regression model to predict the missing values of `CX LogD` and `CX LogP`.

Finally, we combined the training and test sets back into a single dataset that contained the imputed values for `CX LogD` and `CX LogP`. By imputing the missing values using a linear regression model and a set of relevant features, we aimed to improve the accuracy and completeness of our dataset for downstream analyses.

d) Imputing Bioactivities

applies a linear regression model to impute missing values for the "Bioactivities" feature. The dataset is split into training and test sets, where the training set contains all instances with non-missing "Bioactivities" values, and the test set contains instances with missing values. The set of features used in the model includes several molecular properties such as "CX Acidic pKa", "CX LogP", and "Polar Surface Area", among others.

A linear regression model is trained on the training set, and then used to predict the missing "Bioactivities" values in the test set. The imputation is performed using a SimpleImputer with a mean imputation strategy. The predicted values are inserted into the test set. Finally, the training and test sets are concatenated into a single dataset containing the imputed values for the "Bioactivities" feature.

e) Imputing molecular species

The implemented method uses a random forest classification model to predict missing values in the "Molecular Species" feature of a given dataset. Firstly, a new dataset is created without any NaN values in the "Molecular Species" column. The features and target variables are separated, with "CX Acidic pKa" and "CX Basic pKa" as features and "Molecular Species" as the target variable. The data is then split into training and testing sets, with 20% of the data being reserved for testing.

A random forest classification model is then trained on the training set using 100 estimators and a random state of 42. This trained model is then used to predict the missing "Molecular Species" values in the original dataset. To evaluate the performance of the model, the accuracy score metric is used, which compares the predicted values to the actual values in the testing set. Finally, the accuracy score is printed to the console.

f) Imputing Type

he first step involves pre-processing the data, which includes removing rows with missing "Type" values and unnecessary columns such as "Smiles", "Molecular Formula", and "Structure Type". Categorical variables are encoded using **LabelEncoder**.

The data is then split into training and testing sets using **train_test_split**. The model is trained on the training set using the **RandomForestClassifier** algorithm, with a random state of 42.

After training, the model is used to predict the "Type" of the molecules in the test set. The accuracy of the model is then evaluated using **accuracy_score**, which compares the predicted values to the actual values in the testing set. Finally, the accuracy score is printed to the console.

g) Imputing Target

SimpleImputer is a class in Scikit-learn library that provides a simple strategy for imputing missing values in a dataset. It works by filling in missing values with either the mean, median, most frequent value, or a constant.

- ⇨ We used **Mean** to impute this feature.

h) Data encoding:

We have 4 categorical features that need to be encoded.

- **“Structure Type" &"Passes Ro3":** these 2 features have two possible values, so a simple label encoding is sufficed.
- **“Molecule Species” & “Type”:** In this scenario, binary encoding was chosen over label encoding for two features that have a cardinality of more than two. Label encoding could result in the model assigning higher priority to categories with higher numerical encoding, which would be unfavourable in this case. Therefore, binary encoding was used to avoid this issue. This encoding method converts the categorical variables into binary code, where each category is represented by a unique combination of binary digits.

i) Outlier imputation

To identify the percentage of outliers in each feature, we conducted a statistical analysis. The results of the analysis are as follows:

- **"Ro5":** 1.48%
- **"Ro5(lipinski)":** 3.00%
- **"Rotatable bonds":** 3.68%
- **"Algop":** 2.70%
- **"Aromatic rings":** 9.81%
- **"Bioactivities":** 11.03%
- **"CX Acidic pKa":** 3.09%
- **"CX Basic pKa":** 0.04%
- **"CX LogD":** 3.51%
- **"CX LogP":** 3.05%
- **"HBA":** 4.97%
- **"HBA (Lipinski)":** 1.74%
- **"HBD":** 8.14%
- **"HBD (Lipinski)":** 11.25%
- **"Heavy Atoms":** 3.28%
- **"Inorganic Flag":** 8.01%
- **"Max Phase":** 0.28%
- **"Molecular Weight":** 3.40%
- **"Molecular Weight (Monoisotopic)":** 3.31%
- **"Polar Surface Area":** 4.11%
- **"QED Weighted":** 0.00%

These values were used to determine whether any feature has a significant number of outliers.

Based on the previous step of identifying outliers in the dataset, we used the QSAR (Quantitative Structure-Activity Relationship) method to detect and remove outliers using an isolation forest algorithm. Specifically, we used the implementation of the **IsolationForest** class from the Scikit-learn library.

The Isolation Forest algorithm is a popular unsupervised anomaly detection algorithm that works by isolating observations that are rare and different from most of the data points. The

algorithm creates a tree structure where outliers are isolated in the shortest path lengths. The contamination parameter is set to 0.1, meaning that we allowed up to 10% of the data to be considered as outliers.

By using the Isolation Forest algorithm, we were able to detect and remove the outliers from the dataset, which can improve the performance of the machine learning model.

j) Feature selection:

To determine the most important features for our model, we used correlation analysis. This involved calculating the correlation coefficient between each feature and the target variable. The correlation coefficient ranges from -1 to 1, with a value of 1 indicating a perfect positive correlation and a value of -1 indicating a perfect negative correlation. By analyzing the correlation coefficients, we were able to identify the features that had the strongest correlation with the target variable. These features were the most important for our model, as they had the greatest influence on the target variable. We then selected these features to be included in our final set of input features for the machine learning model.

Conclusion

In conclusion, understanding the data we work with is crucial for successful data analysis. It involves gathering information about the data, such as its source, format, size, and quality, to ensure that it is suitable for the analysis we want to perform. Proper data understanding also involves identifying any potential issues or biases in the data and addressing them appropriately before proceeding with analysis. By investing time and effort into understanding our data, we can make more informed decisions and generate more accurate insights, leading to better outcomes for our projects and businesses.

Chapter 3: Model & Evaluation

Introduction

In this chapter, we discuss the models and evaluation methods used in our molecule design project. We explore various generative models, including Variational Auto Encoder (VAE), GPT-2, Generative Adversarial Network (GAN), and Conditional Variational Auto Encoder (CVAE), as well as Message Passing Neural Network (MPNN) for retrosynthesis. We present a comparison of these models' performance using Top-K accuracy metrics and showcase their results in a table. Additionally, we describe our deployment strategy for making our model accessible to clients using the Django framework and simple front-end technologies. Overall, this chapter provides a comprehensive overview of the models and methods we used in our project.

3.1 Models

3.1.1 Properties to SMILES:

- **Variational Auto Encoder (VAE)** is a type of generative model that can learn to generate new data samples that are like the training data. VAEs use an encoder network to map the input data to a low-dimensional latent space and a decoder network to generate new data samples from points in the latent space. The decoder is trained to generate data samples that are like the input data, while the encoder is trained to ensure that points in the latent space are well distributed and can be easily sampled to generate new data samples.
- **GPT-2** (Generative Pre-trained Transformer 2) is a large-scale transformer-based language model developed by OpenAI. It is trained on a large corpus of text data and can be fine-tuned for a variety of natural language processing tasks, including language modeling, text classification, and text generation. GPT-2 uses a transformer-based architecture that allows it to capture long-range dependencies and generate coherent and fluent text.
- **GAN** (Generative Adversarial Network) is a type of generative model that can learn to generate new data samples that are like the training data. GANs consist of two networks: a generator network that generates new data samples and a discriminator network that tries to distinguish between the generated data samples and the real ones. The generator is trained to generate data samples that can fool the discriminator, while the

discriminator is trained to distinguish between the generated and real data samples. GANs have been used to generate realistic images, videos, and audio.

- **CVAE** (Conditional Variational Auto Encoder) is an extension of the VAE model that can learn to generate new data samples conditioned on additional input information. In a CVAE, the encoder network maps the input data and the conditioning information to a latent space, and the decoder network generates new data samples conditioned on both the latent space and the conditioning information. CVAEs have been used for a variety of tasks, including image generation, text generation, and speech synthesis, where the conditioning information could be the class label, image attributes, or speaker identity.

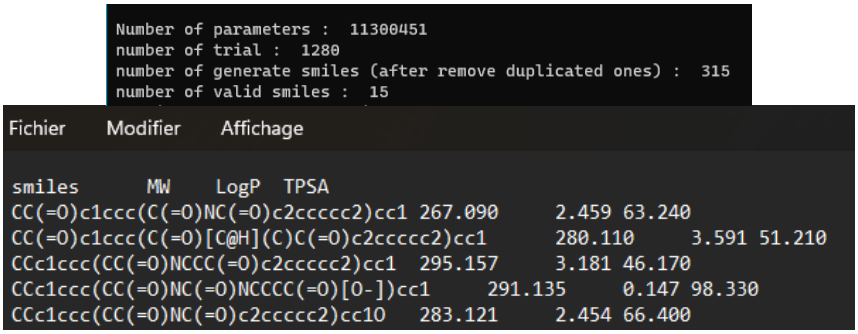


Figure 7 : CVAE test

3.1.2 Retrosynthesis:

MPNN stands for Message Passing Neural Network, which is a type of neural network architecture commonly used for modeling graph-structured data. In MPNN, information is exchanged between neighboring nodes in a graph through message passing, which allows the network to capture local dependencies in the graph structure.

Table 4 : model comparison

Method	Top-1	Top-3	Top-5	Top-10	Top-50
GLN	52.5	69.0	75.6	83.7	92.4
G2Gs	48.9	67.6	72.5	75.5	
GraphRetro	53.7	68.3	72.2	75.5	
AT	53.5	69.4	81.0	85.7	

Method	Top-1	Top-3	Top-5	Top-10	Top-50
MEGAN	48.1	70.7	78.4	86.1	93.2
MPNN	53.4	77.5	85.6	92.4	98.4

3.1.3 Top K accuracy

Table 5 : top k accuracy

Metric	Top-1 Exact accuracy	MaxFrag accuracy
Top-3 Exact	0.775	0.813
Top-5 Exact	0.856	0.889
Top-10 Exact	0.924	0.943
Top-50 Exact	0.984	0.988
MaxFrag accuracy	0.534	0.572

3.2 Deployment

To make our model accessible to clients, we decided to deploy it using the Django framework. Given that both our model and Django are developed in Python, integrating the two is straightforward, resulting in a seamless deployment experience.

Our Django application will handle incoming client requests and retrieve the predicted results, which will then be returned to the client. This allows clients to easily access our model without worrying about the technical details of integrating with it.

For the client-side interface, we decided to keep it simple by using vanilla JavaScript, HTML, and CSS. By avoiding the use of complex front-end technologies, we were able to keep our project lightweight and focused on its core functionality. This choice also ensures that clients with different levels of technical expertise can easily interact with our application.

Overall, our deployment strategy ensures that our model is easily accessible to clients while keeping the user experience simple and intuitive. By leveraging the strengths of Django and simple front-end technologies, we were able to develop a streamlined and effective solution for deploying our model.

In addition to integrating the model with Django and using simple front-end technologies, we have also utilized Docker to containerize each model of the retro model and properties to smiles model. Docker allows us to package the models along with their dependencies, ensuring consistency and portability across different environments.

By containerizing the models, we can easily deploy and scale them on various platforms, such as cloud servers or local machines, without worrying about compatibility issues. Docker provides isolation for each container, ensuring that the models and their dependencies are self-contained and do not interfere with each other.

Moreover, Docker allows us to manage the deployment and dependencies of the models efficiently.

By using Docker, we can easily distribute and deploy the containerized models, simplifying the setup process for clients. They can simply pull the container image and run it on their local machine or deploy it to their desired infrastructure.

Overall, integrating Docker into our deployment strategy enhances the portability, scalability, and manageability of our models. It allows us to encapsulate the models and their dependencies, making it easier for clients to access and utilize them without worrying about complex installation processes or conflicting dependencies.

Conclusion

In this chapter, we discussed the models and evaluation methods used in our molecule design project. We presented various generative models such as VAE, GPT-2, GAN, CVAE, and MPNN for retrosynthesis and compared their performance using Top-K accuracy metrics. We also showcased our deployment strategy using the Django framework and simple front-end technologies. Overall, this chapter provided a comprehensive overview of the models and methods we used in our project. Our deployment strategy ensured that our model is easily accessible to clients while keeping the user experience simple and intuitive. By leveraging the strengths of Django and simple front-end technologies, we were able to develop a streamlined and effective solution for deploying our model.

General Conclusion

Overall, this project presented many challenges since the domain of molecule design is still relatively new and rapidly evolving. However, it was a valuable experience for the team to explore this uncharted territory and apply our knowledge in a way that has the potential to save lives.

Through this project, we gained a deep understanding of the mechanics behind molecule design and developed a systematic approach to generating potential molecules that met our client's desired properties. Additionally, we provided a thorough retrosynthesis analysis, which will enable the client to easily synthesize the requested molecules.

Despite the difficulties we encountered, we remained committed to delivering high-quality results and learning from our experiences. We are confident that our efforts will pave the way for future advancements in this field and we look forward to continuing to push the boundaries of what is possible in molecule design.

References

For data Collection:

[1] https://pubchem.ncbi.nlm.nih.gov

[2] https://www.ebi.ac.uk/chembl/

[3] https://zinc15.docking.org

Retrosynthesis:

[4]https://europepmc.org/backend/ptpmcrender.fcgi?accid=PMC8549044&blobtype=pdf

RDKit documentation:

[5] https://www.rdkit.org/docs/source/rdkit.Chem.rdMolDescriptors.html

Conditional β-VAE:

[6]https://chemrxiv.org/engage/apigateway/chemrxiv/assets/orp/resource/item/626b4332368ab64701913771/original/conditional-vae-for-de-novo-molecular-generation.pdf?fbclid=IwAR1PX4K0jLx4c3d5Jt1HTmBgxcE8kFVGZBBn1lGLXfdn14fg48NPU1yuZ-U

GNN:

[7]https://towardsdatascience.com/drug-discovery-with-graph-neural-networks-part-1-1011713185eb

[8] https://courses.nvidia.com/courses/course-v1:DLI+X-FX-16+V1/

Django:

[9] https://docs.djangoproject.com/en/4.2/