Task 1: Baseline Modeling — The Standard ML Pipeline

Weeks: 1–2
Title: *"Let's Build a Model!"*

Objectives:

- Clean and standardize chemical data
- Apply a standard cheminformatics ML pipeline
- Train multiple models and evaluate their performance

Deliverables:

- Jupyter notebook containing well-commented code and proper documentation of analysis logic
    - Make sure to run your notebook in one go. This ensures that the sequence of the code is correct and anyone can run it after you.
    - **Push your code to the GitHub repo before the next session**

Description

You are a cheminformatics research and development (R&D) team in a pharmaceutical company. The CEO wants to reduce the experimental costs by embracing the promising new trend of machine learning (ML) and artificial intelligence (AI).

The task is to look for potential new antagonists of the **D(2) dopamine receptor**, a type of GPCR used as a target for drugs that, among other things, seek to influence the central-nervous-system (CNS). An external vendor of small, non-toxic drug-like molecules, offers a huge catalogue of potential candidates to buy from. The CEO, instead of buying and trying one candidate after the other, wants to guess which ones have the best chance of working by using known antagonists.

The CEO gives you a dataset of ~ 12 thousand molecules (see attached). It consists of three pieces of information, a molecule represented as a SMILES string the activity value and the activity type. You will train a model on these molecules so that the CEO can use it to inspect the vendor's molecules.

1st internal meeting

You held an internal meeting together to discuss how you can approach this task.

A team member mentioned the need to make sure that these **SMILES** strings are **sanitized** and **standardized.** Another mentioned that it is important to have a numerical representation of each SMILES string because ML models can only process numbers.

But how can you convert SMILES into numbers?

A team member read that a tool called RDKit provides useful descriptors that can describe a molecule numerically[1]. This can be interesting to explore.

Another team member mentioned that **Extenced Connectivity Fingerprints (ECFP)** can also be used. It's also implemented in RDKit with the morgan fingerprint algorithm.

Now, the only thing left is to train an ML model. But which one?

You decided to try your luck with two powerful families of modeling, XGBoost and a neural network (NN) version called multi-layer perceptron (MLP)

You split the tasks between you, and you go modeling.

- **You will split the dataset into 80% training and 20% testing**
- **You will train four models (each representation with each model)**
- **You will check the performance of each model on the test set and decide which model and representation pair works best.**

Meeting the CEO

You request a meeting with the CEO to show her your results. You structure a jupyter notebook that walks through the steps you did for training the model, you explain your choices and decisions, and what should the CEO do next.

- **What was your training scheme? How and why did you split the data?**
- **What were the representations and models you used? Why did you use them?**
- **How did you decide which representation and model to use eventually? What analysis help you pick?**
- **How well will this model perform if the CEO gives you new molecules?**

[1] One of the descriptors is named "Ipc", we recommend that you do NOT use it!

The CEO challenge

The CEO was quite impressed, but before making a final decision, she gives you a new internal dataset and asks you to show your model performance on it as well.
If the results remained the same, this would make her very confident.

- **You will need to save your trained model in a format that allows you to load it and use them for testing new molecules**
- **You will receive these two new datasets 3 days before the next lecture**
- **You will report the results of your chosen model in your notebook.**

Important notes for your deliverables

The bare minimum you can do is to have a functional Jupyter notebook that

1. Sanitizes and standardizes SMILES
2. Represent SMILES in RDKit descriptors and Morgan Fingerprints
3. Train the XGBoost and MLP models
4. Use different evaluation metrics to assess the performance of your models

If you have questions alongside your implementation journey (and I hope you do), simply note them down in this spreadsheet and move on.

Once you have your bare minimum product and you want to learn more, go back to your questions and answer them one by one.

Each answer will help you understand your work better, and therefore, present it and explain it better.

A cheat hint!

If you copy and paste this document to a language model, it should give you such bare minimum functional code, notebook, and presentation outline.

I am saying this to emphasize that this is NOT the important part. It's your questions and digging deeper to understand what is happening...