# Problem Statement

Drugs are typically small organic molecules that achieve their desired activity by binding to a target site on a receptor. The first step in the discovery of a new drug is usually to identify and isolate the receptor to which it should bind, followed by testing many small molecules for their ability to bind to the target site. This leaves researchers with the task of determining what separates the active (binding) compounds from the inactive (non-binding) ones. Such a determination can then be used in the design of new compounds that not only bind, but also have all the other properties required for a drug (solubility, oral absorption, lack of side effects, appropriate duration of action, toxicity, etc.).

The goal is to to **develop predictive models that can determine given a particular compound whether it is active (1) or not (0)**. A molecule can be represented by 100000 **binary** features which represent their topological shapes and other characteristics important for binding.

# Dataset

## Caveats:

The dataset has an imbalanced distribution i.e., within the training set there are only 78 actives (+1) and 722 inactives (0). No information is provided for the test set regarding the distribution.

## Description

Input matrix has shape of 800-100000 and the test data have a matrix of shpe 350-100000. Data is a **sparse matrix** of 100000 binary attributes so for eaach observation only the indices of the non zero column are provided.

# Application Architecture and Module Division

# Code Explanation

## Data ingestion

The data was not in a conventional csv format but the data was in a tax file. Since our data have 100000 columns and all are of binary nature so the indices of the columns for each observation, that are non zero are provided. To tackle this problem the data was read from a classical text file and was converted to data frame using Data_Getter class. I have not converted this data frame to csv due to its size.

I split data for training and testing for models and when this was done therewas found no mis balance in the classes of the data. Proportion of class 1 was .0953125 for training data and it was .10625 for the test set

## Model Selection

There are two challanges here in modelling the classifier:

1. Number of features is very high
2. Data is imbalanced as there are only 78 active cases and 722 observations are inactive, therefore F1 score is preferred over accuracy score for assessing the predictions.

The approaches used for dimensionality reduction are:

- Principle Component Analysis
- **Sparse PCA** (With L1 penalty tuning) For PCA I will code in notebook, in directory model_eda, for the number of components taken and the vriance described by them.

ML models tried for classification are:

1. KNN
2. Decision Tree
3. Adaboost
4. Random Forest

Most of the algorithms, being non-parametric, won't require **SMOTE(Synthetic Minority Oversampling Technique)** or any such oversampling technique. But I will use SMOTE in some cases.

## Dimenstion Reduction

- **PCA**: I used PCA with 500 principle components that described total of 71% of the variance in the data. And I applied DecisionTree, RandomForest, Adaboost, KNNeighbourClassifier. The results are pretty amazing and

| Models | Accuracy Score | F1-0 | F1-1 |
|---|---|---|---|
| KNN | 0.91875 | 0.94 | 0.00 |
| RandomForest | 0.925 | 0.96 | 0.43 |
| DecisionTree | 0.89375 | 0.94 | 0.37 |
| AdaBoost | 0.94375 | 0.97 | 0.73 |

I find the Adaboost technique very appealing because for category 1 we get best f1 score score.

## Model Tuning

I used GridSearchCV to crossvalidate and get the best adaboost estimator whic is found for the best n_estimators = 300 and the classification_report is

| F1 Score | F1-0 | F1-1 |
|---|---|---|
| 0.95 | 0.97 | 0.75 |

## Logging Framework

I have implemented a logger in every class i formed and App_Logger class is present in logger.py The logging file is present in directory, log_files. Every time a function is operated the actions are logged in that file with the date and time.

## Deployment

I have not deployed it on cloud instead i have produced a gui web application using flask web framework. It can be used using the run.py all the requirements to use the application are in requirements.txt.

## Note

1. I will make csv of reduced matrix and the model produced also to make the app smooth but the training will take more time.
2. For training at leas 500 observations will be needed for pca to work properly.
3. I have not uploaded the object of pca model as it was of size, that is not allowed to be uploaded. But dont worry will be created iteslf in model directory