Notre démarche actuelle consiste à tester différentes approches comportant deux grands axes :

Le grand axe 1. Battre le syba a son propre jeu :

L'idée ici est d'utiliser les mêmes bases de données (avec les mêmes labels binaires) utilisées par syba pour construire différents modèles plus puissants (sur l'une des deux erreurs (faux positifs et faux negatifs) ou même les deux en même temps).

Une première étape consiste à tester plusieurs modèles simples, puis à se concentrer sur les plus prometteurs.

Modèles :

Git (pas encore mis à jour) : https://github.com/MazighLahianiX/PSC

1. Régression logistique : Marc

2. Réseau de neurones : Courte introduction de Marc

Nous avons commencé ici par l'approche la plus simple :

2.1. Réseaux neuronaux denses : comme leur nom l'indique, les couches sont entièrement reliées (denses). Chaque neurone d'une couche reçoit les données de tous les neurones de la couche précédente - ils sont donc connectés de manière dense.

Un simple réseau constitué de deux couches denses donne déjà des résultats plus prometteurs que ceux de la régression logistique, comme le montre le tableau résumant les erreurs de prévision.

Cependant, la modification des paramètres du modèle (nombre d'époques, taille des batch, ajout de couches) ne semble pas améliorer les capacités de prédiction, c'est pourquoi nous devons envisager une autre modele toujours en utilisant toujours les réseaux de neurones.

2.2. Réseau neuronal récurrent : Avec LSTM comme architecture .

Un réseau neuronal récurrent (RNN) est une classe de réseaux neuronaux artificiels où les connexions entre les nœuds forment un graphe dirigé le

long d'une séquence temporelle. Cela lui permet de présenter un comportement temporel dynamique.

Long short-term memory (LSTM) est une architecture de réseau neuronal récurrent (RNN) utilisée dans l'apprentissage profond.

Une unité LSTM commune est composée d'une cellule, d'une porte d'entrée, d'une porte de sortie et d'une porte d'oubli. La cellule stocke des valeurs à des intervalles de temps arbitraires et les trois portes régulent le flux d'informations entrant et sortant de la cellule.

Les réseaux LSTM sont bien adaptés à la classification, au traitement et à la prévision basés sur des données de séries chronologiques, car il peut y avoir des écarts de durée inconnue entre des événements importants dans une série chronologique. En chimie_informatique par exemple, les LSTM sont utilisés pour générer des nouvelles molécules, faire des prévisions sur la molécule suivante dans une séquence de SMILE, ou encore classer les molécules selon certains critères.

Les LSTM ont été développés pour résoudre le problème de disparition du gradient qui peut être rencontré dans la formation des RNN traditionnels. L'insensibilité relative à la longueur des décalages est un avantage des LSTM par rapport aux autres RNN.

Comme nous pouvons le voir dans le tableau des résultats, les architectures LSTM sont probablement les plus prometteuses, mais contrairement aux DNN, les réseaux LSTM dans notre cas sont très sensibles aux différents paramètres (nombre d'époques, taille des batch, ajout de couches), mais leur entrainement sur nos bases de données prend en moyenne deux à trois jours, et donc trouver le meilleur paramétrage peut être très complexe et coûteux, la deuxième partie du PSC se concentrera sur ce problème, qui consiste à construire le réseau le plus optimal.

3. Naïveté multinomiale de Bayes :

Une approche qui on a peine testé et qui devrait être explorée plus est de rester sur le modèle de syba, donc sur des modèles Bayes naïfs avec différentes implémentations afin d'essayer d'améliorer les prédictions, le seul court test qui a été fait de tels modèles n'a pas été très prometteur et donc cette approche ne sera pas beaucoup considérée dans le reste du projet.

Le modèle naïf multinomial de Bayes de la bibliothèque sklearn met en œuvre l'algorithme naïf de Bayes pour les données distribuées multinomiales, et est l'une des deux variantes naïves classiques de Bayes (l'autre étant l'approche utilisée par syba) utilisées dans la classification des textes (où les données sont généralement représentées par des

comptages de vecteurs de mots) comme ce que l'on fait pour la transformation de SMILES en vecteur boolean.

Excellent lien pour les détails de l'implémentation.

https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB

Le grand axe 2 : Nouvelles approches ou nouvelles bases :

(Une partie pour les chimistes, des bases de données nouvelles et plus intéressantes ? Mieux labeliser ?)