研究プロジェクト後半 秋山研究室 レポート

提出日：2021 年 5 月 28 日

系：情報工学系

学籍番号：19B12223

氏名：玉野　史結

1　1-A) イブプロフェン構造式

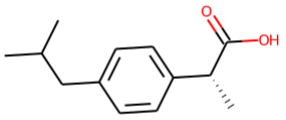

図 1: 実習 1-A イブプロフェンの構造式出力

2　1-D) $\ln K_a$ 値を計算する関数の設計

Tajimi らの論文 [1] によると，Supplementary Table S1. に含まれる f_b 値は実数値のものもあれば $f_{bmin} - f_{bmax}$ のように範囲で指定されているものもある。範囲指定されている場合には

$$f_b = \frac{f_{bmin}+f_{bmax}}{2}$$

としたと記述されている。また

$$\ln K_a = C \ln \frac{f_b}{1-f_b}$$

と定義されており，定数 C は先行研究 [2] より 0.3 と設定，また $\ln K_a$ 値の発散を防ぐため f_b は $f_b \times 0.99 + 0.005$ としていた。

そのため，関数 `calc_lnka(fb)` の最初のステップは，引数 `fb` に対してまずデータのタイプが `float` であるか否かを確認し，さらに「`-`」で区切った時のリストの長さが 1 より大きいかを確認することとした。 `float` でなく，かつ区切ったときのリストの長さが 1 よりも大きい場合， f_b は範囲で指定されているということになるため，リストの 1 つ目の要素 (*i.e.* f_{bmin}) とリストの 2 つ目の要素 (*i.e.* f_{bmax}) の平均をとったものを f_b とした。当てはまらない場合は値が与えられているため f_b の値はそのままである。その後， f_b の値に 0.99 を乗算し 0.005 を加算することによりスケーリングを行なった。

このようにして得られた f_b を `scaled_fb` と表現することとし，関数の出力を，

$$0.3 \times \ln(\frac{\texttt{scaled\_fb}}{\texttt{1-scaled\_fb}})$$

として関数を定義した。

3　1-E) 学習器に Lasso 回帰を用いた際のハイパーパラメータ等について

ここでは，以下の 5 点について記述する。

- 最適化したハイパーパラメータ
- 各ハイパーパラメータの探索範囲
- ハイパーパラメータ探索方法
- 最良ハイパーパラメータ
- 最良ハイパーパラメータにおける cross validation 中の $\ln K_a$ 値の RMSE と $\ln K_a$ 値の相関係数 R

3.1　最適化したハイパーパラメータ

`sklearn.linear_model.Lasso` におけるパラメータおよびその意味は，以下である。 [3]

- *alpha*
  alpha はラグランジュ乗数を表しており，以下の式の第 2 項の係数となっている。overfitting を抑制する効果がある一方，大きすぎると underfitting 問題が発生するため，最適化の必要があると考えた。

$$\frac{\|y - Xw\|_2^2}{2 \times n} + alpha\|w\|_1 \tag{1}$$

  なお， y は真値， Xw は予測値， w は重み， n はサンプル数を表す。

- *fit_intercept*
  fit_intercept は切片を計算するか否かを決める *bool* 値である。データが原点を通る直線でプロットされることが確実である場合には *False* で良いが，本課題では default 値でもある *True* とした。

- *normalize*
  normalize は正規化を行うか否かを決める *bool* 値である。本課題では，説明変数に標準化を施すのみで，正規化は行わずに進めることとした。従って，本課題では default 値でもある *False* とした。

- *precompute*
  precompute は計算速度を高めるためにグラム行列を用いるか否かの指標である。 本課題では標準化を行った後に学習させるため，グラム行列は相関行列となる。 *sparse input* であればスパース性の保持のため *False* となるが，本課題では *sparse input* でない上に，用いる必要がないと判断し *True* とした。

- *copy_X*
  copy_X は overwrite を行わないか否かを決める *bool* 値である。本課題では，*True* とし，overwrite を行わないようにした。

- *max_iter*
  max_iter は iteration の回数を指定するパラメータである。最適化する必要があると考えた。

- *tol*
  tol は (1) 式の最適化をどの程度寛容するかを表す指標である。この値よりも小さくなるまでイテレーションが行われる。最適化の必要があると考えた。

- *warm_start*
  warm_start は前回学習させた学習器の情報を再利用するか否かの指標であり，本課題では平等に評価を行うために *False* とした。

- *positive*
  positive は係数を正とするか否かの指標であるが，本課題では係数が負の場合も十分にありうると考え，default 値でもある *False* とした。

- *random_state*
  random_state は以下で説明する *selection* が *random* の際に，どのような擬似乱数に基づいて係数選択を行うかを指定する。本課題では *cyclic* を用いたため，このパラメータは設定しなかった。

- *selection*
  selection は *random* か *cyclic* をとるパラメータであり，回帰における係数の設定方法について指定する。 *random* は特に， *tol* が 1e-4 よりも大きい場合に収束が早くなる。後述するが，本課題では *tol* の探索範囲を 1e-6 から 1e-4 としたため，このパラメータは default 値の *cyclic* とした。

以上より，最適化するハイパーパラメータは *alpha* , *max_iter* , *tol* の 3 つとした。

3.2 各ハイパーパラメータの探索範囲

- *alpha*
  alpha が大きくなればなるほどパラメータの複雑さ (*i.e.* (1) 式の $\|w\|_1$) が与える影響が大きくなる一方で，モデルのデータへの適合具合が軽視されることになる。default 値では 1.0 のため，本課題では *alpha* の探索範囲を 1e-2 から 1.5 まで探索を行い，適切な *alpha* を求めた。

- *max_iter*
  max_iter は，*tol* を 適切に設定すれば比較的重要性が低いと考え， default 値が 1000 であるため，1000 から 100000 までとした。

- *tol*
  tol は default では 1e-4 であるが， *alpha* の値を 1e-2 から 1.5 までと， default 値よりは平均してやや小さくなる値でとっているため，寛容性を低くしても良いと考え，値を小さくし，1e-6 から 1e-4 に設定した。

3.3 ハイパーパラメータ探索方法

今回の課題においてハイパーパラメータ探索には optuna[4] を用いて， cross validation における RMSE 値の平均をとり，その値が最小となるように 100 回探索を行った。

3.4 最良ハイパーパラメータ

探索を行った 3 つに対し，最良ハイパーパラメータは以下のようになった。

$alpha = 0.010001550247043559$
$max\_iter = 18633$
$tol = 2.0186369425523876\text{e-}05$

3.5 最良ハイパーパラメータにおける cross validation 中の $\ln K_a$ 値の RMSE と $\ln K_a$ 値の相関係数 R

cross validation 中の $\ln K_a$ 値の RMSE 並びに $\ln K_a$ 値の相関係数 R を以下に示す。

- cross validation 1
  RMSE = 0.5209547078388901
  R = 0.7107256587155896

- cross validation 2
  RMSE = 0.4606982018660353
  R = 0.7605366763431665

- cross validation 3
  RMSE = 0.4281213900359186
  R = 0.7542478496501217

- cross validation 4
  RMSE = 0.49896753871626187
  R = 0.7249485193449345

- cross validation 5
  RMSE = 0.4763822826075803
  R = 0.7516799440924689

4 1-F) Cyclic peptide drug の予測における $\ln K_a$ 値の RMSE , $\ln K_a$ 値の相関係数 R

1-E で選択されたハイパーパラメータを用いて Small molecule データを教師データとした Lasso 回帰で Cyclic peptide drug の予測を行なった結果を以下に示す。

RMSE = 891905781196.9539
R = −0.16718436183577068

5 1-G) E) と F) の予測精度の比較・考察

予測精度は F) で得られたものの方が低かった。 Lasso 回帰は線形回帰であるため，予測値が真値より一定値離れてしまうことは考えられるが，その点を考慮しても大幅に真値と離れているものが存在し，それらは $\ln K_a$ を非常に小さい値であると予測していた。テストデータにおける説明変数を確認したところ，*Ipc* という記述子に対してのみ非常に大きな値をとっており，重みは −0.018062564152947512 であったことから，この記述子が外れた予測に影響していると考えた。それぞれの値を以下に示す。

- Acetyl-Daptomycin
  $Ipc = 1.90252689\text{e+}03$
  $\ln K_a = -0.5870781$
  $prediction = -3.50131812\text{e+}01$

- Daptomycin
  $Ipc = 1.20007563\text{e+}06$
  $\ln K_a = 0.51222289$
  $prediction = -2.16764776\text{e+}04$

- Telavancin
  $Ipc = 1.03358327\text{e+}11$
  $\ln K_a = 0.63662153$
  $prediction = -1.86691641\text{e+}09$

- Dalbavancin
  $Ipc = 2.41744875\text{e+}14$
  $\ln K_a = 0.75673178$
  $prediction = -4.36653231\text{e+}12$

- Oritavancin
  $Ipc = 8.80504378\text{e+}12$
  $\ln K_a = 0.51222289$
  $prediction = -1.59041668\text{e+}11$

予測値の値の大小と *Ipc* の値の大小が関係していることから， *Ipc* が誤った予測に関与していることは間違いないと考える。この *Ipc* 値が異常に大きくなることは Github の issue にてあげられていた。[5] これによると，`avg=True` とすれば良いと書かれており，改めて 2D 記述子 を並べたベクトルを構成する関数を作り変えた。考察段階で *Ipc* という記述子は `Descriptors.descList` において index が 40 であることが分かっていたため，その値のみ再び `avg=True` という引数を加えて再計算した予測値が以下となった。

RMSE = 0.7231702603632365
R = 0.4526782165593785

並外れた値では無くなったが，やはり Cyclic peptide drug に対する予測精度の方が低かった。修正後の 2D 記述子 を用いて計算を行った結果，真値との差を確認すると，ある分子に対して大きく外れているというわけではなかったが，誤差の平均を計算すると 0.15672424808532037 となり，全体として $\ln K_a$ 値をやや小さく予測していることが分かった。 Cyclic peptide は Small molecule と比較すると大きな分子であり，立体構造による影響がより大きくなる一方で 2D 記述子では平面的な構造の数値化しか行えていないためにこのような予測制度になっていると考えた。

6 1-H) 重要な記述子についての考察

重要な記述子を Lasso 回帰において係数の絶対値が 0.075 より大きいものであると定義する。その時，重要な記述子は RDKit の 2D descriptors 208 の内，6 に絞られた。それらを係数の絶対値の大きいものから順に示す。

- *MolLogP* (weight : 0.2846387781835837)
  MolLogP は Wildman-Crippen LogP 値を表している。[6] $\log P$ は分子の有機層と水層中の平衡状態における濃度比率である分配係数 P の対数値であり，大きい値であると有機層における濃度が高いすなわち脂溶性が高いことを示す指標となっている。[7] Wildman-Crippen LogP とは Wildman と Crippen が考案した原子単位での $\log P$ を計算し，分子を構成する原子を分子内相互作用等も考慮して分類し，その寄与度合を用いて，和をとることで算出される値である。[8]

従って，この記述子の重みの絶対値が大きいということは，脂溶性と $\ln K_a$ 値，すなわち血漿タンパク質との結合率 f_b との関係が大きいということであり，かつ係数が正であるから脂溶性が大きい程 $\ln K_a$ 値が大きくなる，すなわち f_b が大きくなるということが小分子データから学習されていることになる。

- *FpDensityMorgan1* (weight : −0.14901095942588066)
  FpDensityMorgan1 を考える上で，まず Morgan Fingerprint について考える。Morgan Fingerprint は ECFP に相当するものであり，原子からの一定結合距離にある部分構造を数えていくものである。[9] また，その距離 (radius) は 2 に設定されており， *FpDensityMorgan1* の *1* はソースコードより対象とする環境から取り除く原子の ID であると考えられる。[10][11] すなわち， *FpDensityMorgan1* は ECFP の密度の内，原子 ID が 1 のものを取り除いた指標であると考える。

  この記述子の重みの絶対値が大きいということは，原子から半径 2 の構造のフィンガープリントが f_b に関与しており，原子 ID が 1 のものを取り除いたものが指標として重要であることが分かった。一方で，原子 ID 1 のものを除去することでどのような恩恵があるのか分からなかったが，[12] を参考にすると，C 原子に結合している原子が (C × 3, H × 1) の場合に原子 ID が 1 になっていた。参考サイトにおける原子 ID が全分子に共通のものであるとするならば，先述した原子は Fingerprint Density を計算する上で密度を高くしてしまう，もしくは低く計算してしまい，f_b を予測する上でその影響が比較的大きくなると考えた。

- *HallKierAlpha* (weight : −0.11381840054960113)
  HallKierAlpha は Hall と Kier によって提案された指標であり，対象とする原子の共有結合半径と混成軌道の両方の影響をエンコードしたものであり， sp^3 混成軌道の場合に値が 0 となる。[13]

  従って，この記述子の重みの絶対値が大きいということは，f_b を予測する際には共有結合半径や原子の混成軌道といった指標が重要だと分かり， f_b が血漿タンパク質との結合率であることを考えると，確かに結合においてその半径・混成軌道は重要な指標であると考えられるため学習ができていると考えた。

- *fr_quatN* (weight : −0.09347316929887078)
  fr_quatN は 窒素原子が電子を 1 つ失った 4 価窒素の数を表している。[14]

  従って，この記述子の重みが負であるということは， 4 価窒素の数が多いとき f_b が小さく予測されることになる。血漿タンパク質は負に荷電している [15] ことを考えると正の電荷を帯びている 4 価窒素の数が多い場合には結合しやすくなると考えたが，分子全体として電荷を打ち消しあっていると仮定した場合，局所的に正電荷を帯びている場合には局所的に負電荷を帯びている部分があるということも意味し，負電荷を帯びた部分からは反発力により結合しないと考える。以上より， 4 価窒素を多く有する構造は正電価を帯びた窒素原子は立体的にみた場合に分子の内側，負電荷を帯びた原子がその外側に位置しているのではないかと考える。

- *VSA_EState10* (weight : 0.08555745188948738)
  VSA 記述子は van der Waals surface area 記述子を指し，これは MOE 独自の記述子であり，$\log P$，モル屈折，電荷の特性に関し，ある一定の範囲内の分子の表面積を記述子の値と

して表したものである。[16] *VSA_EState* は特に置換基や官能基による寄与を，表面積の寄与を bin として作成したものであり，*VSA_EState10* は記述子の値が 11.00 以上のものを指す。[17] なお，似た記述子に *EState_VSA* があり，これは置換基や官能基による寄与を bin として表面積の寄与の値を取ったものである。

この記述子の重みが大きいということは置換基や官能基の寄与が大きい場合には f_b が大きくなると予測されており，結合という観点から考えると，置換基・官能基による障害は確かに影響をもたらしていると考えられる。そのため，学習が適切に進んだと考える。

- *qed* (weight : 0.07588883863605525)
  qed は薬らしさを 771 個の傾向医薬品データセットを用いて定量化したものとなっており，分子量，$\log P$，水素結合ドナーの数，水素結合アクセプターの数，極性表面積，回転可能結合数，芳香環の数，必要でないとされている構造である忌避構造の数の 8 つの記述子を用いてモデル化されたものとなっている。これらの記述子を情報エントロピー

  $$H(p) = -\sum_{i=1}^{M} p_i \log_2 p_i$$

  が最大となるように係数を決め，上位 1000 のエントロピーを与える係数の平均を取ったものが *qed* となっている。[18] これらのことより，各記述子の値ではなくその重みが記述子の値となっているということになる。

  以上より，*qed* の重みが大きいということは，薬らしさを表現する各記述子の重みが大きいほど f_b が大きくなると予測されていることとなる。すなわち，drug-like な分子は血漿タンパク質との結合率が高いと考えられる。

7 1-I) Lasso 回帰の代わりに Random Forest Regression を行った結果

7.1 ハイパーパラメータ探索までのコードについて

7.1.1 データ取得

実行時， Small molecule の csv データが格納されたファイル名を `-small_mol` の引数として，Cyclic peptide drug の csv データが格納されたファイル名を `-cyclic_pep` の引数として指定することで，csv ファイルを DataFrame として取得することができるようにするため，`argparse` ライブラリと `pandas` ライブラリを用いた。すなわち，`argparse.ArgumentParser` にて作成したインスタンス `parser` に引数を加えていくことで，`parser.parse_args()` メソッド内のクラスがファイルに相当するためこれを `pandas.read_csv()` の引数と指定すれば DataFrame として csv データを取得できる。

7.1.2 説明変数・目的変数の作成

説明変数を作成するには SMILES 式を 2D 記述子を並べたベクトルに変換する必要がある。その処理を行うクラス・メソッドを別ファイルに作っておく。まず，SMILES 式を引数にとるクラス

を作成する。SMILES 式を 2D 記述子に変換するため， RDKit の `rdkit.ML.Descriptors.MoleculeDescriptors.MolecularDescriptorCalculator()` の引数に `rdkit.Chem.Descriptors.descList` の 1 つ目の要素 (*i.e.* インデックス 0) を格納したリストを指定することで，記述子計算機を作成する。作成した計算機の `Calcdescriptors()` メソッドにクラス内の SMILES 式を指定することで全 2D 記述子が計算されたタプルが得られるため，目的変数として学習器に用いることができるように `numpy.array()` に変換して返すことで 2D 記述子を並べたベクトルを返す関数を作成する。これを各サンプルに適用する。さらに，`sklearn.preprocessing.StandardScaler()` により標準化のためのインスタンスを作成し，`fit_transform` を説明変数に適用することで標準化を行い，説明変数を作成する。
また目的変数は，DataFrame 内の f_b に上述した `calc_lnka` 関数を各サンプルごとに適用することで作成できる。

7.2 Random Forest Regressor のハイパーパラメータについて

7.2.1 最適化したハイパーパラメータ

Random Forest Regression はアンサンブル学習の内，バギングという、訓練データを分割し，特徴量を選択した後にそれぞれの弱学習器を並列的に用いて結果を統合することで予測値を出すという手法である。[19] 以下に図で示す。

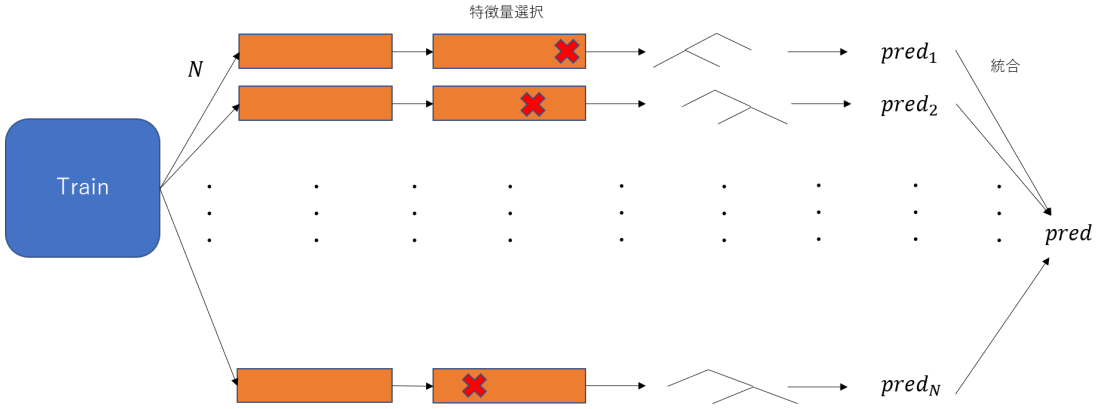

図 2: Random Forest Regression の内部処理を図式化したもの

`sklearn.linear_ensemble.RandomForestRegressor` におけるパラメータおよびその意味は，以下である。 [20]

- *n_estimators*
  これはバギングに用いる木の数である。この数は最適化する必要があると考えた。

- *criterion*
  criterion はデータを分割する際の指標であり，不純度を評価している指標である。不純度が大きいことは分割したデータがあるクラスに偏っていることを意味し，不純度が小さくなるように分割する必要がある。このパラメータは *mse* または *mae* のどちらかをとるが，本課題では default である *mse* を用いた。

- *max_depth*
  これは決定木の深さを調節するパラメータである。この値を最適化するよりは後述する *min_samples_split* の最適化を行うことで overfitting 等の抑制を行うこととしたため，default でもある *None* とした。

- *min_samples_split*
  これはノードを分割するために必要な最小サンプル数である。過学習を防ぐ必要もあり，最適化する必要があると考えた。

- *min_samples_leaf*
  これはノードを構成するために必要な最小サンプル数であり，仮に *min_samples_split* が 5 であり，分割する前のノードに 5 つサンプルが含まれていたとしても， *min_samples_leaf* が 3 であれば分割できないことになる。この値も最適化する必要があると考えた。

- *min_weight_fraction_leaf*
  これはノードの重みの総和に対する最小の重みの割合を考える指標であり，この値を設定すると少なくとも *min_weight_fraction_leaf* より大きな重みが必要であるということになる。このパラメータは重要性が低いと考え，最適化を行わず default の 0.0 とした。

- *max_features*
  これは最適な分割を行うために考慮すべき特徴量の数を指定するパラメータであり，取り得るパラメータは *auto*（*i.e.* 全ての特徴量を使う。），*sqrt*（*i.e.* $\sqrt{(\text{全特徴量の数})}$ を使う。），*log2*（*i.e.* $\log_2$(全特徴量の数) を使う。），int （*i.e.* int を使う。）， float （*i.e.* （全特徴量の数）$\times$ float を使う。）である。このパラメータは最適化する必要があると考え，本演習では float を指定した。

- *max_leaf_nodes*
  これは最大のノードの数である。 default では *None* であるが，過学習の抑制のため最適化することとした。

- *min_impurity_decrease*
  この値は決定木の成長を停止するための閾値となる値であり，不純度がこれより小さくなると分割を停止するというものである。以下の計算式で表されるパラメータとなっている。

  $$\frac{N_t}{N \times (impurity - \frac{N_{tR}}{N_t} \times (right\_impurity) - \frac{N_{tL}}{N_t} \times (left\_impurity))}$$

  N はサンプル数，N_t は現在のノードのサンプル数，N_{tR} は右側の子ノードのサンプル数，N_{tL} は左側の子ノードのサンプル数を表しており， *impurity* は不純度である。本演習ではノードの数やノード内のサンプル数等で過学習を抑制するため，この値まで最適化する必要性がないと判断し， default である 0 を用いた。

- *boot_strap*
  これは木を作成する際にブートストラップを用いるか否かの指標であり，*True* とした。

- *oob_score*
  oob とは out of bag の略であり，各 bootstrap においてサンプリングされなかったサンプル

を指す。これを学習の正確度計算に用いるか否かの指標となっており，本課題では *True* とした。

- *warm_start*
  これは Lasso 回帰同様，前回学習させた学習器の情報を再利用するか否かの指標であり，本課題では平等に評価を行うために *False* とした。

- *ccp_alpha*
  ccp とは cost-complexity pruning の略である。木は大きく複雑であるほど誤分類を防ぐことができる一方で汎化性能が低下してしまう。そのためこの *ccp_alpha* とノードの数の乗算値を罰則項として用いることで，木の刈り込みを行うというものである。最適化する必要があると考えた。なお，効果的な α は以下のように定義されていた。[21]

$$\alpha_{eff}(t) = \frac{R(t) - R(T_t)}{|T| - 1} \tag{2}$$

  ここで $R(t)$ は刈り込み後の誤分類の数，$R(T_t)$ は刈り込み前の誤分類の数，$|T|$ は決定木の大きさを表す。

- *max_samples*
  ブートストラップを行う上でサンプル数の最大数を指定するものであるが，サンプル数を制限することによる本演習における有用性が感じられず， default である *None* とした。

以上より，最適化するハイパーパラメータは *n_estimators* , *min_samples_split* , *min_samples_leaf* , *max_features* , *max_leaf_nodes* , *ccp_alpha* の 6 つとした。

7.2.2 各ハイパーパラメータの探索範囲

- *n_estimators*
  木の数が多ければ多いほど，バリアンスを低くすることはできる可能性があるが，トレードオフの関係であるバイアスに影響を及ぼしてしまう。今回は default 値の 100 よりは大きな値を探索する一方で，大きすぎてはいけないことから探索範囲を 100 から 1000 とした。

- *min_samples_split*
  この値は最小で 2 であるが，どこまで大きな値を取れば良いかを考える際に，2 つに均等に分類されたときの一方のノードのサンプル数が 10 より大きいとまだ分類の余地があるのではないかと考えたため，最大値は 2×10，すなわち探索範囲は 2 から 20 とした。

- *min_samples_leaf*
  上述した通り，一方のノードのサンプル数 10 より大きいとまだ分類の余地があると考えたため，探索範囲は 1 から 10 とした。

- *max_features*
  必要な特徴量の数は，半分は必要ではないかと考え， float で 0.5 から 1.0 とした。

- *max_leaf_nodes*
  訓練データは約 1200 個である。そのためノードの数はそれよりも小さくなるはずであり，1200 に近い値であると過学習の恐れがあるため最大値を 1000 とし，最小値は 100 としてこの範囲で探索を行った。

- *ccp_alpha*
  (2) より効果的な α を考えると，決定木のサイズが大まかに 1000 であると仮定し，分子が最小で 1 であることを考慮すると，最適な値は 1e-3 あたりでないかと考える。概算した結果を含んだ範囲を探索するように，1e-5 から 1e-1 までを探索することとした。

7.2.3 ハイパーパラメータ探索方法

Lasso 回帰同様にハイパーパラメータ探索には optuna [4] を用いて， cross validation における RMSE 値の平均をとり，その値が最小となるように 100 回探索を行った。

n_estimators は int 指定で 100 - 1000, *min_samples_split* も int 指定で 2 - 20, *min_samples_leaf* も int 指定で 1 - 10, *max_features* は 0.5 - 1.0 の範囲で対数一様分布からサンプリングした値, *max_leaf_nodes* は int 指定で 100 - 1000, *ccp_alpha* は 1e-6 -1e-1 の範囲の対数一様分布からサンプリングした値を回帰のパラメータとした。cross validation を行う際には，
`sklearn.model_selection.cross_val_score` を用いて，引数には学習器，小分子の説明変数，目的変数，スコアリングには 2 乗平均平方根誤差の負の値を用いる *neg_root_mean_squared_error* を使い，cross validation 数は 5 としてスコアを算出した。この返り値は cross validation の数だけ scoring の値が格納された list となるため，optuna において最小化してほしいスコアである RMSE 値となるようにそれらの平均を取り，このままでは負の数であるため -1 をかけることで正の数に変換して optuna で最小化したいスコアである RMSE 値とした。

7.2.4 最良ハイパーパラメータ

探索を行った 7 つに対し，最良ハイパーパラメータは以下のようになった。

n_estimators = 460
min_samples_split = 2
min_samples_leaf = 1
max_features = 0.6400954926671139
max_leaf_nodes = 575
ccp_alpha = 1.0658736175592873e-05

7.2.5 最良ハイパーパラメータにおける cross validation 中の $\ln K_a$ 値の RMSE と $\ln K_a$ 値の相関係数 R

cross validation 中の $\ln K_a$ 値の RMSE 並びに $\ln K_a$ 値の相関係数 R を以下に示す。なお，最良パラメータを用いるには optuna の Attributes の 1 つである `best_params` を利用する。この Attribute の返り値は dictionary であるため，回帰のパラメータとして用いるにはアンパックを行うため `**` をつける必要がある。また cross validation を行うため `skleran.model_selection.KFold` を用いた。`KFold.split()` の引数に訓練データの説明変数，目的変数を入れ，繰り返し処理を行うことで cross validation が実行される。この際， `KFold.split()` では index が取得できるため，繰り返し処理内で訓練データのインデックスを指定して，さらに訓練データと検証データに分割し， validation を行う。学習結果は `sklearn.metrics.mean_squared_error` の引数に検証データの目的変数と予測値を指定して計算した後，平方根を取れば RMSE を求めることができる。また，`numpy.corrcoef()` メソッドの引数に同じく検証データの目的変数と予測値を指定すると分散共分散行列が計算されるため，非対角成分を取得することで相関係数 R を求めることもできる。以上のプロセスをリストに都度追加していき， cross validation が終了したのちにリストを出力することで値を得ることができる。

- cross validation 1
  RMSE = 0.4644035832147438
  R = 0.7889564461816483

- cross validation 2
  RMSE = 0.4427123851402675
  R = 0.7880714111267291

- cross validation 3
  RMSE = 0.37761214578583624
  R = 0.8132106435791429

- cross validation 4
  RMSE = 0.43001360403166555
  R = 0.8070416146526004

- cross validation 5
  RMSE = 0.42854181803491354
  R = 0.8117764861589346

7.3 Cyclic peptide drug の予測における $\ln K_a$ 値の RMSE , $\ln K_a$ 値の相関係数 R

選択されたハイパーパラメータを用いて Small molecule データを教師データとした Random Forest 回帰で Cyclic peptide drug の予測を行なった結果を以下に示す。なお，この際に用いる教師データは Small molecule 全体であるため，学習器には Small molecule 全体のデータを用いる必要がある。また， RMSE および R を求めるには cross validation 内で行ったことと同じプロセスを辿れば良い。

RMSE = 0.6531016693402263
R = 0.5388666099065988

7.4 重要な記述子についての考察

重要な記述子を `sklearn.ensemble.RandomForestRegressor.feature_importances_` において値が 0.02 より大きいものであると定義する。その時，重要な記述子は RDKit の 2D descriptors 208 の内，7 に絞られた。それらを大きいものから順に示す。なお，出力する際には 2D 記述子のリストを予め取得しておき，`sklearn.ensemble.RandomForestRegressor.feature_importances_` を `zip` で繰り返し処理を行い，importance が 0.02 より大きい場合に出力するようにすれば良い。

- *MolLogP* (importance : 0.310359580332063)
  MolLogP は Lasso 回帰同様に最も重要な記述子であるということが分かり，脂溶性と血漿タンパク質との結合率 f_b との関係がやはり大きいということが確認できる。

- *SMR_VSA7* (importance : 0.047524561084776436)
  SMR_VSA は先述した MOE 独自の記述子のうち，分子屈折率 (MR) と表面積の寄与を用いた記述子となっており， *SMR_VSA7* はその値が 3.05 以上 3.63 未満のものを指している。[22] [23]

- *BCUT2D_MRHI* (importance : 0.03212854034578604)
  BCUT2D は原子量， Gasteiger 電荷という電子の電気陰性度を基にして決定する点電荷 [24], MolLogP，MolMR を対角成分としたベクトルで，それぞれの固有値の最大値と最小値が要素となっている。[25] *BCUT2D_MRHI* は MR の最大固有値を指している。

- *SlogP_VSA6* (importance : 0.03018716865299929)
  SlogP_VSA は先述した MOE 独自の記述子のうち，$\log P$ の値と表面積の寄与を用いた指標となっており， *SlogP_VSA6* はその値が 0.15 以上 0.20 未満のものを指している。 [22] [23]

- *PEOE_VSA6* (importance : 0.02604793965526905)
  PEOE_VSA は MOE 独自の記述子のうち，部分電荷と表面積の寄与を用いた記述子となっており， *PEOE_VSA6* はその値が -0.10 以上 -0.05 未満のものを指している。[22] [23]

- *SMR_VSA10* (importance : 0.02151527420149223)
  SMR_VSA10 は，先述した *SMR_VSA* の値が 4.00 以上のものを指している。[22] [23]

- *VSA_EState6* (importance：0.021178404488366054)
  VSA_EState6 は先述した *VSA_EState* の値が 6.00 以上 6.07 未満 のものを指している。[17]

以上の記述子より， Random Forest Regression において Lasso 回帰と異なる重要な記述子の多くは，分子屈折率や電荷に関するものであった。すなわち血漿タンパク質との結合率 f_b が分子屈折率と関係がある一方で，その関係は線形関係ではないために Lasso 回帰では重要視されなかった可能性が考えられる。このことは bin が最大の場合や最小の場合以外にも重要度が大きい記述子があることからも示唆される。また VSA 記述子の重要度が高くなっており，表面積あたりの特性が予測精度に関与している可能性が高いと考える。

7.5　目的変数を変えた時の予測精度の比較・考察

予測精度は Cyclic peptide drug を目的変数としたときの方が Lasso 回帰同様に低かった。なお Lasso 回帰の場合と異なり，並外れた値ではなかったが，同じ修正を施したもので計算を行っても予測精度が上がることはなかった。結果を以下に示す。

RMSE = 0.6450219425012329
R = 0.5299471326326767

真値との差が 1 より大きかったものが以下の 2 分子である。

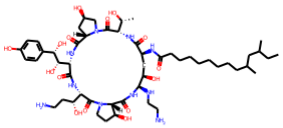

図 3: Caspofungin の構造式出力

$true - pred = 1.4978040040756992$

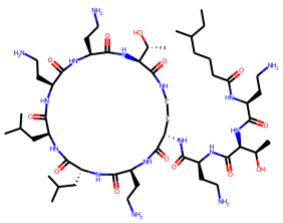

図 4: Colistin の構造式出力

$true - pred = 1.2772413714104842$
一方で真値との差が 0.1 より小さかったものが以下の 2 分子である。

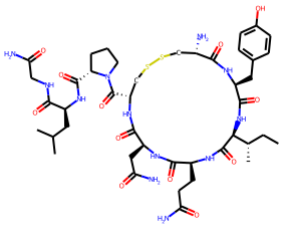

図 5: Oxytocin の構造式出力

$true - pred = -0.028241018662841177$

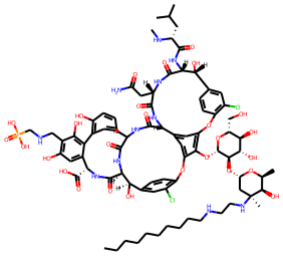

図 6: Telavancin の構造式出力

$true - pred = -0.002621245656653448$
これらの違いを構造式から確認することが困難であったため，それぞれの記述子の値を比較し，前者と後者でやや値に差が見られたものについてその記述子と値の一部を示す。なお，ここでの差は統計的に有意なものを確かめたのではなく，主観的なものとなっている。

- *BCUT2D_MWHI*
  Caspofungin：16.33303999707714，Colistin：16.274319729818163
  Oxytocin：33.11460794525803，Telavancin：35.49679579150821

- *BCUT2D_MRHI*
  Caspofungin：5.979840534258863，Colistin：5.979840534258863
  Oxytocin：8.763128354458999，Telavancin：7.5119995728238464

- *SMR_VSA10*
  Caspofungin：41.35025810546054，Colistin：64.97897702286656
  Oxytocin：86.56677297564003，Telavancin：78.11720550066529

- *SlogP_VSA12*
  Caspofungin：0.0，Colistin：0.0
  Oxytocin：21.587795952773448，Telavancin：30.79764210725292

BCUT2D_MWHI と *BCUT2D_MRHI* が後者 2 分子において値が大きいことは，それぞれの最大固有値がより大きく，情報を多く含んでいる分子の方が予測精度が高いことを意味している。また *SMR_VSA10* や *SlogP_VSA12* はそれぞれの指標の bin において無限大を含む指標になっている（*i.e.* 上限がない）。VSA は先述したようにある一定範囲の分子の表面積を記述子としたものであることから，テストデータ内の一部の分子の記述子の値を見た限りでは，ある表面積の大きさ内の特性だけでなく，表面積自体と予測精度が関係しているのではないかと考える。

8　2-c) 予測精度の向上のために行ったこと

予測精度の向上のため，用いる記述子について考えた。その際，記述子の最適化による予測精度の向上が見られたという論文があった。[26] この論文を参考にし，実装を試みたものを以下に示す。

- Dragon 7.0 [27] の利用
- CORALSEA [28] の利用
- Random Forest Regression における重要度に基づいて記述子を絞り込む

8.1　Dragon 7.0 の利用

Dragon 7.0 は 2D 分子記述子の 1 つであり，RDKit 以外の記述子も論文で用いられることが多いことがわかった。しかしながら， Dragon 7.0 はサービスが終了しており，実装が困難であったため，他の分子記述子を用いることを試みた。 論文内でも Dragon にはある記述子の計算ができないといった旨が記されており，そのような不具合を解消した分子記述子計算ソフトウェア mordred [29] というものを用いることにした。使用の際には，`mordred` から `Calculator` ， `descriptors` をインポートすることで利用できる。`Calculator(descriptors)` にて計算機のインスタンスを作成し，そのインスタンスの引数に SMILES 式をとることで分子記述子がおよそ 1600 - 1800 個計算される。しかしながら，このソフトウェアは不完全であり，一部エラーとなっている記述子が存在する。そのためエラーとなる記述子は説明変数から排除した。さらに *bool* 値の分子記述子も存在し，その場合は *True* ならば 1， *False* ならば 0 とした。

検証のため，課題 1 で最適化したハイパーパラメータで Random Forest 回帰を行ったが，検証できなかった。分子によって計算できる記述子が異なるため，説明変数の大きさが一致していないのではないかと考えた。活用には時間を要すると考え，実装は断念した。

8.2 CORALSEA の利用

CORALSEA という SMILES 式から 最適な記述子の計算を行うソフトウェアがある。論文内では正しい正規化のために用いられており，こちらの利用も検討したが，ソフトウェアの動かし方の理解に時間を要すると考え断念した。

8.3 Random Forest Regression における重要度に基づいて記述子を絞り込む

簡単な検証のため，課題 1 で `feature_importances_` Attribute を用いて，重要度が 0.01 より大きくなった記述子 13 個のみを用いて RMSE, R を計算した。この検証では，重要度の高い記述子のインデックスを取得し，関数内で逐一インデックスを指定するという非効率的な方法で実装した。結果を以下に示す。

RMSE = 0.6479761913510614
R = 0.547988855813419

なお，記述子を絞り込む前の結果を再度示す。

RMSE = 0.6450219425012329
R = 0.5299471326326767

比較すると，RMSE 値は精度が低下しているが，R 値は向上している。そのため，記述子を絞り込むという方法は有効である可能性が高いと考えた。

論文内では，標準偏差が 0 の記述子や別の記述子とのピアソン相関係数が 0.95 を上回るような記述子を取り除くということを行っていた。また，R のパッケージを用いて，Random Forest を適用し，設定した閾値を下回る重要度の変数を除いていくということも行なっていた。

本課題では，標準偏差が 0 の記述子を取り除くという点と，設定した閾値を下回る重要度の変数を取り除くという点の実装を行った。なお，モデルは課題 1 で最適化したものを利用した。

まず，Small molecule データを行列化した訓練データにおいて，列ごとに標準偏差を計算し，値が 0 となるもののインデックス，すなわち列番号を格納しておく。その列を削除していくが，この際に列番号の小さいものから削除した場合にはずれが生じてしまう。そのため列番号の大きいものから削除していくことでエラーを防いだ。学習後に `feature_importances_`を用いて重要度を計算する。今回設定した閾値の値は 0.01 であるため 0.01 を下回る列番号を格納しておき，標準偏差の場合同様に列を削除していく。一方で削除する列がない場合には全ての重要度が閾値以上であることを意味するため，繰り返し処理を止め，予測精度を計算・出力するようにした。結果を以下に示す。

RMSE = 0.6383771362346949
R = 0.5433755466603287

記述子を絞り込む前よりも RMSE，R 共に精度は僅かに上昇した。一方で期待したほどの上昇ではなかった。ここでパラメータの 1 つである *warm_start* を *True* とすることで前回の学習を利用し，より精度の高い学習ができるのではないかと考え実装した。結果を以下に示す。

RMSE = 0.6408462854827193
R = 0.560401097512674

相関係数 R は僅かに上昇し，今回のように繰り返し処理で学習器を更新していく際には用いた方が良いと分かった。

本課題では，主成分分析や論文内で行われていた相関係数，独立性の検定に基づいた記述子の絞り込みを行っていないため，その点を行うことで僅かに精度が上昇する可能性がある。また，RDKit の 2D 記述子は総数 208 であり，論文内で用いられた Dragon よりも少なく，使用できなかった mordred と比較しても少ない。そのため記述子の絞り込みによる効果が十分に発揮されなかったと考える。また，記述子の取捨選択を行なったのちにハイパーパラメータの探索を行っておらず，最適なパラメータとなっていない可能性もあるため，パラメータの探索を行うことで精度が上昇すると考えられる。

また，論文内では pH が 7.4 におけるイオン状態を考慮したり，Plasma Protein Binding 値は *in vivo* で算出されたもののみを利用するなど行っていた。さらに，非結合率 f_u を指標としており，結合率だけでなく非結合率も考えることで予測精度が上昇するのではないかと考えた。

参考文献

[1] Takashi Tajimi *et al.* "Computational prediction of plasma protein binding of cyclic peptides from small molecule experimental data using sparse modeling techniques", *BMC Bioinformatics* **19**(Suppl 19): 527, 2018. doi: 10.1186/s12859-018-2529-z.

[2] IngleBL *et al.* "Informing the human plasma protein binding of environmental chemicals by machine learning in the pharmaceutical space: applicability domain and limits of predictability." *JChem Inf Model.* 2016;56(11):2243-52.

[3] sklearn.linear_model.Lasso
URL : `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html`（アクセス日：2021/5/21）.

[4] Optuna
URL : `https://optuna.org`（アクセス日：2021/5/21）.

[5] IPC Descriptor gives suspiciously large values
URL : `https://github.com/rdkit/rdkit/issues/1527`（アクセス日：2021/5/24）.

[6] rdkit.Chem.Crippen module
URL : `https://www.rdkit.org/docs/source/rdkit.Chem.Crippen.html`（アクセス日：2021/5/23）.

[7] ケモインフォマティクスと LogP 計算：CLogP の C は"calculated"ではありません
URL : `https://future-chem.com/logp/`（アクセス日：2021/5/23）.

[8] Scott A. Wildman *et al.* "Prediction of Physicochemical Parameters by Atomic Contributions", *Journal of Chemical Information and Computer Sciences* **39**: 868-873, 1999. doi: 10.1021/ci990307l.

[9] RDKit でフィンガープリントを使った分子類似性の判定
URL : https://future-chem.com/rdkit-fingerprint/（アクセス日：2021/5/23）.

[10] rdkit/rdkit/Chem/Descriptors.py
URL : https://github.com/rdkit/rdkit/blob/master/rdkit/Chem/Descriptors.py（アクセス日：2021/5/23）.

[11] rdkit/rdkit/Chem/Draw/SimilarityMaps.py
URL : https://github.com/rdkit/rdkit/blob/c2fb57c19f8bac4aac07f4d0915ece8d95f7c963/rdkit/Chem/Draw/SimilarityMaps.p（アクセス日：2021/5/23）.

[12] 【RDKit】Morgan フィンガープリントの生成ルールを解釈してみた
URL : https://qiita.com/oki_kosuke/items/9a7b52911c4ef9d4edad（アクセス日：2021/5/23）.

[13] Réka Laczkó-Rigó *et al.* "Structural dissection of 13-epiestrones based on the interaction with human Organic anion-transporting polypeptide, OATP2B1", *The Journal of Steroid Biochemistry and Molecular Biology* **200**: 2020. doi: 10.1016/j.jsbmb.2020.105652.

[14] Electronic Supplementary Material (ESI) for CrystEngComm.
URL : http://www.rsc.org/suppdata/ce/c4/c4ce01912a/c4ce01912a1.pdf（アクセス日：2021/5/23）.

[15] 血液成分であるタンパク質の種類について【電気泳動による血清タンパク分画による分類】
URL : https://totthi.com/blood-component/（アクセス日：2021/5/23）.

[16] MOE/ケモインフォマティクス
URL : https://www.molsis.co.jp/lifescience/moe/qsar/（アクセス日：2021/5/23）.

[17] rdkit.Chem.EState.EState_VSA module
URL : http://rdkit.org/docs/source/rdkit.Chem.EState.EState_VSA.html（アクセス日：2021/5/23）.

[18] RDKit で薬らしさを定量的に評価する
URL : https://future-chem.com/rdkit-qed/（アクセス日：2021/5/23）.

[19] ランダムフォレスト (Random forest) とは?機械学習モデルを分かりやすく解説!!
URL : https://nisshingeppo.com/ai/random-forest/（アクセス日：2021/5/24）.

[20] sklearn.ensemble.RandomForestRegressor
URL : https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html（アクセス日：2021/5/24）.

[21] 1.10. Decision Trees
URL : https://scikit-learn.org/stable/modules/tree.html（アクセス日:2021/5/24）.

[22] ChemDes/MOE-type descriptors (60)
URL : http://www.scbdd.com/chemdes/list10/（アクセス日：2021/5/25）.

[23] rdkit.Chem.MolSurf module
URL : http://rdkit.org/docs/source/rdkit.Chem.MolSurf.html（アクセス日：2021/5/26）.

[24] 佐々木皓平 他 . 高精度な分子動力学計算のための静電相互作用パラメータの検討
URL : http://www.molsci.jp/2018/pdf/1P103_m.pdf（アクセス日：2021/5/26）.

[25] rdkit.Chem.rdMolDescriptors module
URL : http://rdkit.org/docs/source/rdkit.Chem.rdMolDescriptors.html（アクセス日：2021/5/26）.

[26] Cosimo Toma *et al.* "QSAR Development for Plasma Protein Binding: Influence of the Ionization State", *Pharmaceutical Research* **36**: 28, 2019. doi: 10/1007/s11095-018-2561-8.

[27] Kode. Dragon (Software for Molecular Descriptor Calculation) version 7.0. Kode srl; 2016.

[28] CORAL-QSAR/QSPR
URL : http://www.insilico.eu/coral/（アクセス日：2021/5/27）.

[29] 森脇寛智 *et al.* "分子記述子計算ソフトウェア mordred の開発", ケモインフォマティクス討論会 39, 2016. doi: 10.11545/ciqs.2016.0_Y4.