1 Classification Models

Date: 2025–5–8
Objective: Explore efficacy of various classification models (KNNClassifier, RandomForestClassifier, GradientBoostedClassifier) to evaluate predicitive power.
Methods:

- **Algorithms/Tools:** `Scikit-learn` KNeighboursClassifier, RandomForestClassifier, GradientBoostedClassifier, ConfusionMatrix, ClassificationReport

1.1 Rationale:

In general classification models have the potential to be significantly more accurate than regression models and can work with smaller datasets, at the cost of losing contrast between the best performers in the dataset, and fringe cases that may be weak performers. Because these algorithms are all supervised, a threshold must be established to divide the continuous data into a binary yes/no. To accomplish this, all molecules with LXRβ and LXRα antagonist values less than 50% and greater than 75% were selected as hits. These values were chosen as any more extreme values would make to dataset too biased (as it stands these values still massively bias the dataset) and anything less extreme may defeat the purpose of the classification (too out of bounds of purpose to use). Seeing as the goal of this project is to quickly evaluate a set of molecules for their suitability, a classification system should work sufficiently to quickly separate potential molecules into positive and negative hits.

1.1.1 KNeighboursClassifier

As was the case in the initial regression sweep, the KNeighboursClassifier is a good, lightweight supervised machine learning algorithm that can be used to quickly assess if their are any clear patterns/emergent properties in the data. While this algorithm won't give clear insights into which features might be the most useful for determining if a molecule is a 'hit' or not, further investigation may be able to provide insights if the accuracy is sufficiently high.

1.1.2 RandomForestClassifier

Like the KNeighboursClassifier, Random Forest is a well known supervised machine learning algorithm which randomly constructs decision trees in varying orders to determine the most efficient/effect hierarchy of features which can predict a given label. Other than the mechanistic differences between KNeighbours, Random Forest enables visualization of the feature hierachy, giving insights into the most impactful features on the models decision.

1.1.3 GradientBoostClassifier

Gradient boost is another lightweight machine learning model which procedurally generates decision trees. Each generation of decision tree aims to predict and the residual of the previous iteration and is then weighted and incorporated into a total sum to minimize the loss function.

$$L(y_i, F_M(x_i)) = \sum_{i=1}^{n} \frac{1}{2}(y_i - F_M(x_i))^2$$

Where y_i is the observed label value, F_M denotes the final ensemble of decision trees, x_i is the feature set of a given sample, and n is the number of samples. A learning rate (typically denoted as $\nu = 0.1$) is used to modulate the effect of each decision tree iteration and may be increased or decreased depending on granularity of the desired prediction ability and dataset size. The classification model takes this general approach but requires a binary label set for input. Gradient boosting has the unique feature of improving off of the previous iteration weak learner to make an over all strong learning model.

Results:

1.2 Classification and Metrics

Each classification algorithm was trained on the same dataset, and the training set vs testing set was varied to measure accuracy. To turn the 2D LXRα/β data into a 1D binary, thresholds were established for each column. Molecules with LXRα inhibition values at 100nm greater than 75% and LXRβ inhibition values at 100nm less than 50% were considered positives or hits and given a score of 1. Values which failed at least one

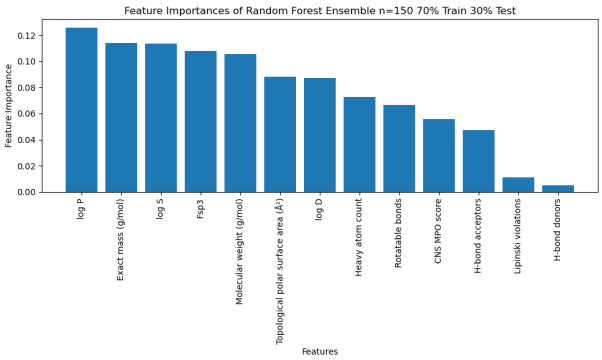
of these criteria were considered negatives and given a score of 0. While ideally no LXRα activity would be detected (e.g. a 100%) this combined with the LXR$\beta < 50\%$ drastically limited the size of the positive pool, rendering the dataset too biased to use. To evaluate these models 3 metrics were used; specificity, sensitivity, and accuracy. Specificity is indicative of a models ability to spot negatives and therefore is given by the ratio of true negatives over total negatives in the training set. Sensitivity denotes a models ability spot positives and is given by the ratio of true positives over total positives in the training set. Accuracy is a combined measure of total correctness and given by the ratio of true positives plus true negatives over the total predictions made in the training set.

1.3 KNN Classification

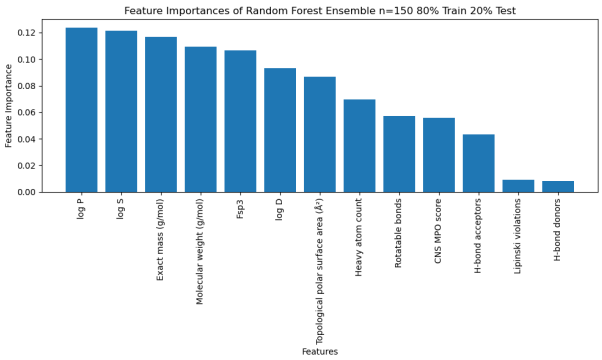
After an initial evaluation of the efficacy of the three models, it was found that KNN mildly outperformed Gradient Boost and Random forest on the small dataset. As such 25 KNN models were trained, and their metrics averaged. On average specificity scored $95.2\% \pm 4.3\%$ for each of the models with no discernible difference between them. Sensitivity was non-existent, averaging $7.6\% \pm 10.7\%$ due to the lack of hits in the dataset and did not improve when test datasize increased. To further illustrate this point, the total accuracy was recorded at roughly $86.2\% \pm 4.2\%$ for each of the models despite the low sensitivity, showing that even with the reduced constraints on LXRα/β activity the training set was still too biased. Due to the extremely low presence of positives in the testing set, the (training, test) split was changed $(80\%, 20\%) \rightarrow (70\%, 30\%)$, which showed a negligible effect on the three metrics.

1.4 Random Forest Classification

As stated in the previous section, the RandomForestClassifier struggled with the small dataset, having a remarkable selectivity ($\approx 95\%$), but poor sensitivity ($\approx 0\%$). Similar to the KNN classifier, to determine if this low sensitivity was a result of the small testing pool the same test split transformation was applied $(80\%, 20\%) \rightarrow (70\%, 30\%)$ which had minimal effect on the 3 metrics. Additionally, the number of trees in the forest (iterations of the approximation loop) was increased from 100 to 150. This again had a negligible impact on the metrics but did have minor changes to the classification hierarchy shown in fig. 1.



(a) RandomForest 150 trees, 70:30 training:testing split



(b) RandomForest 150 trees, 80:20 training:testing split

Figure 1: Comparison of the feature importance of the Random Forest Classifier models trained on a 70:30 (left) and an 80:20 (right) training:testing split. Log P has the highest importance in both models. Exact Mass appears as second highest importance in the 70:30 model and third highest in the 80:20 model. Log S appears as third highest in the 70:30 model and second highest in the 80:20 model.

Figure 1 shows that Log P is preserved as the feature with the highest importance between the 80:20 and 70:30 training split models. Of note, Exact Mass and Log S switched places from 2^{nd} and 3^{rd} most important in the 70:30 split respectively (but with a very similar importance) to 3^{rd} and 2^{nd} respectively (with a more discernible difference). Overall this reliance on Exact mass, Log P, and Log S may be indicative of some relation between aromaticity and/or hydrophobicity in the decision tree, but due to the low selectivity of the model and the similarity of the compounds it is likely not a reliable distinguishing factor between LXRα/β.

1.5 Gradient Boost Classification

Like the KNN classifier and Random Forest Classifier, the Gradient Boost classifier struggled with a high selectivity but poor specificity. To explore alternative classification mechanisms, two Gradient Boost models were trained on distinguishing LXRα inhibitors and LXRβ inhibitors respectively. The intention was, given an accurate enough reading, a compound may be screened for LXRα/β specificity and sensitivity independently. This decision greatly assisted in reducing the bias in the datasets and resulted in significantly higher sensitivity scores, but reduced specificity and accuracy significantly. Interestingly, there was a notable difference in the LXRα/β metrics. The model trained to spot molecules with low inhibition of LXRα had a specificity of 50%, a selectivity of 76% and a total accuracy of 62%. The model trained to spot molecules with high inhibition LXRβ had a specificity of 86%, a selectivity of 61% and a total accuracy of 74%. This means that if a combined model returns a hit there is a 46% chance if it being a true hit, e.g. $P(1|1) = 46\%$ and if it returns not hit, e.g. $P(0|0) = 43\%$.

1.6 Discussion & Takeaways:

Each model suffered from the extreme bias of the dataset. Methods exist (such as SMOTE) to systematically synthesize data points that can be used to build a rebalanced dataset from a heavily biased one. There are several draw backs however; SMOTE functions by taking two arbitrary 'hits' in the biased dataset and drawing a line between them in N-space (where each column of continuous data is regarded as a spatial dimension). an arbitrary point is then picked on this line and is considered a new 'hit'. This approach assumes that the label set is correlated to the feature set through some form of clustering, which may or may not be the case. it is true that KNN did perform the best out of the three algorithms tested, but due to the low sensitivity this may not be indicative of the data being correlated through clustering and some statistical anomaly. Additionally synthetic data cant be used in the testing set and may only be used when training the model. Due to the very small number of positives in the testing set, a proper estimation of model sensitivity is likely not possible. Clustering using the individual LXRα/β inhibition data may yield more reliable results, as was the case with the dual Gradient Boost Classifiers. Random Forest struggled with the biased dataset and was not able to adequately distinguish negatives from positives. An interesting focus on properties that related to hydrophobicity (Log P and Log S) was seen as well as a focus on exact mass. Exploration of using the dual classifier approach with Random Forest may yield more conclusive results. Additionally, a comparison of different model outputs may be used in a combined model. For instance Gradient Boost and KNN may be used in parallel and the result analyzed to see the probability of a hit given Gradient Boost and KNN both giving hits, e.g. $P(1|A = 1, B = 1)$ where A and B are the model outputs. Additionally the same classification approaches may be used on the machine readable structure set using MACCS, Morgan Fingerprinting, or some other substructure analysis/similarity search.