Application of Machine Learning Classification Algorithms in Predicting Biological Response to Molecule Characteristics

**Researchers:** Rogers Kiome, Nimat Ibrahim, Alfred Aminayanate Bob-Manuel, Yaw Adjei Darko, Oriih Gloria, Damilola Idris, Ketul Patel, Miracle Uche

# Introduction

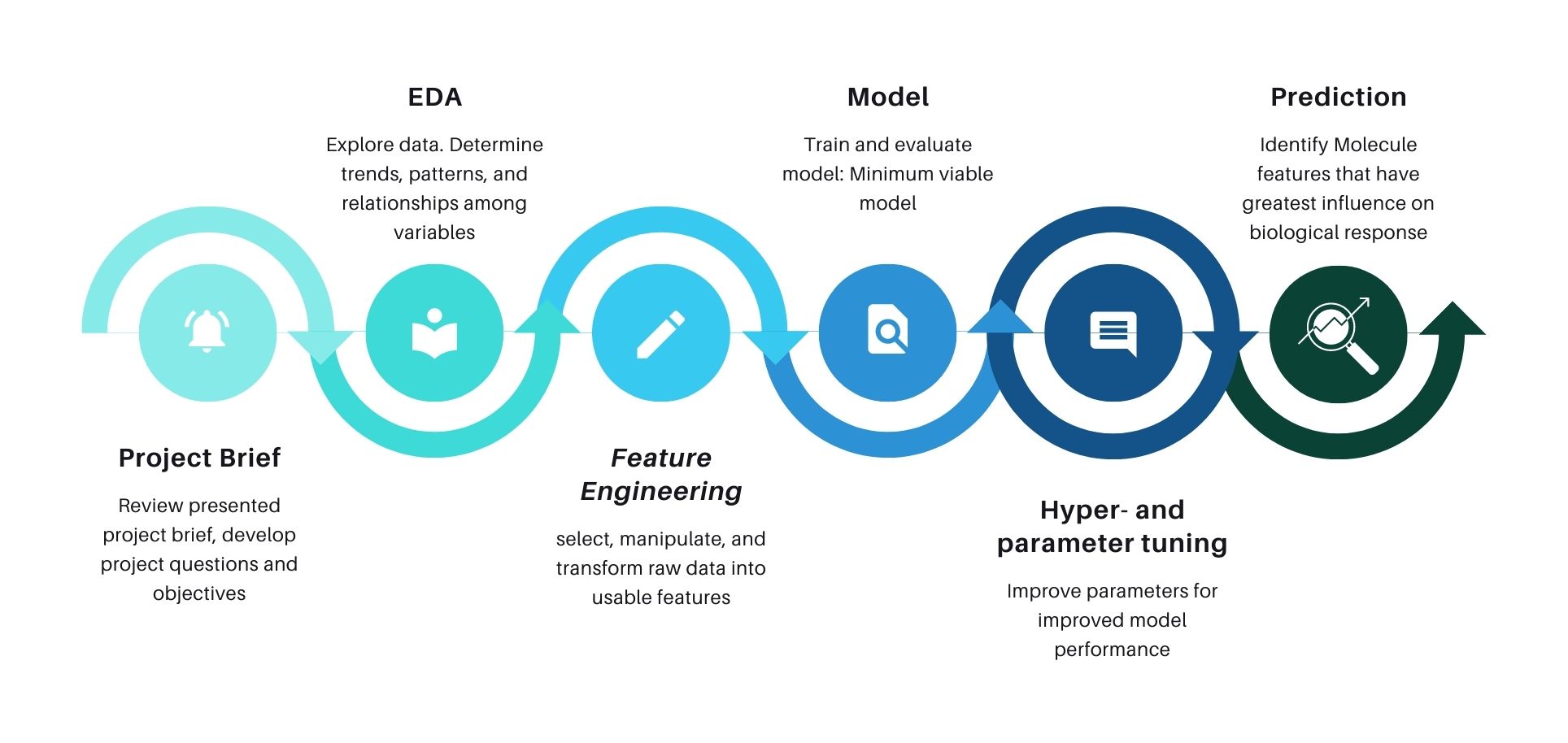
Understanding molecular functions is known as the science of life. At the molecular level, many biological responses in organismal systems take place. An accurate understanding of the life sciences depends on an understanding of molecular functions. The catalytic and binding activities that take place at the molecular level are demonstrated by molecular functions. Furthermore, it is well known that living cells can sense mechanical pressure and convert it into biological reactions.

Through the development of models and predictive analyses, the machine learning technique is a systematic procedure that seeks to understand the molecular basis of biological activity. The last ten years have seen significant advancements in the field of molecular biology, with significant implications for biotechnology and human health. The classification approach in machine learning, which has until now eluded researchers, offers quick and easy ways to analyse chemical and biological interactions in cells.

## Aim and Objectives

The goal of this project is to create a good model that can associate molecular information to an actual biological response and predict biological activity of each molecule given a new dataset that the model has never seen before.

## Flow Process



**Project Brief:** A formal introduction and brief overview of project plan, asking the right questions and implementing strategies for the data extraction and exploration

**Exploratory Data Analysis (EDA):** exploring the features of the dataset to determine trends, patterns and relationship amongst variables to generate insights and informed decision

**Feature Engineering:** This comprises selecting, transforming, and manipulating raw data into usable features. This is an effective technique or procedure because it improves predictive models by highlighting trends and isolating relevant information.

**Modelling**: This includes developing the model, training and testing it, and selecting the most suitable model for the project.

**Hyper and Parameter Tuning:** This entails optimizing the model's parameters for improved performance. This is by far the most crucial component of the process since it involves direct control of the behavior of the training algorithm and has a substantial impact on the overall performance of the model being trained.

**Prediction**: This is where the project's goal of finding molecular properties that have the greatest influence on biological response is met.

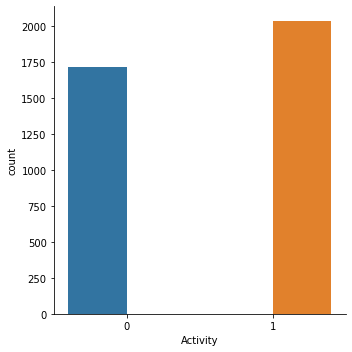
**Model deployment:** The validated and final model should be deployed to the web at this point so that anybody can make live predictions.

## Data Source

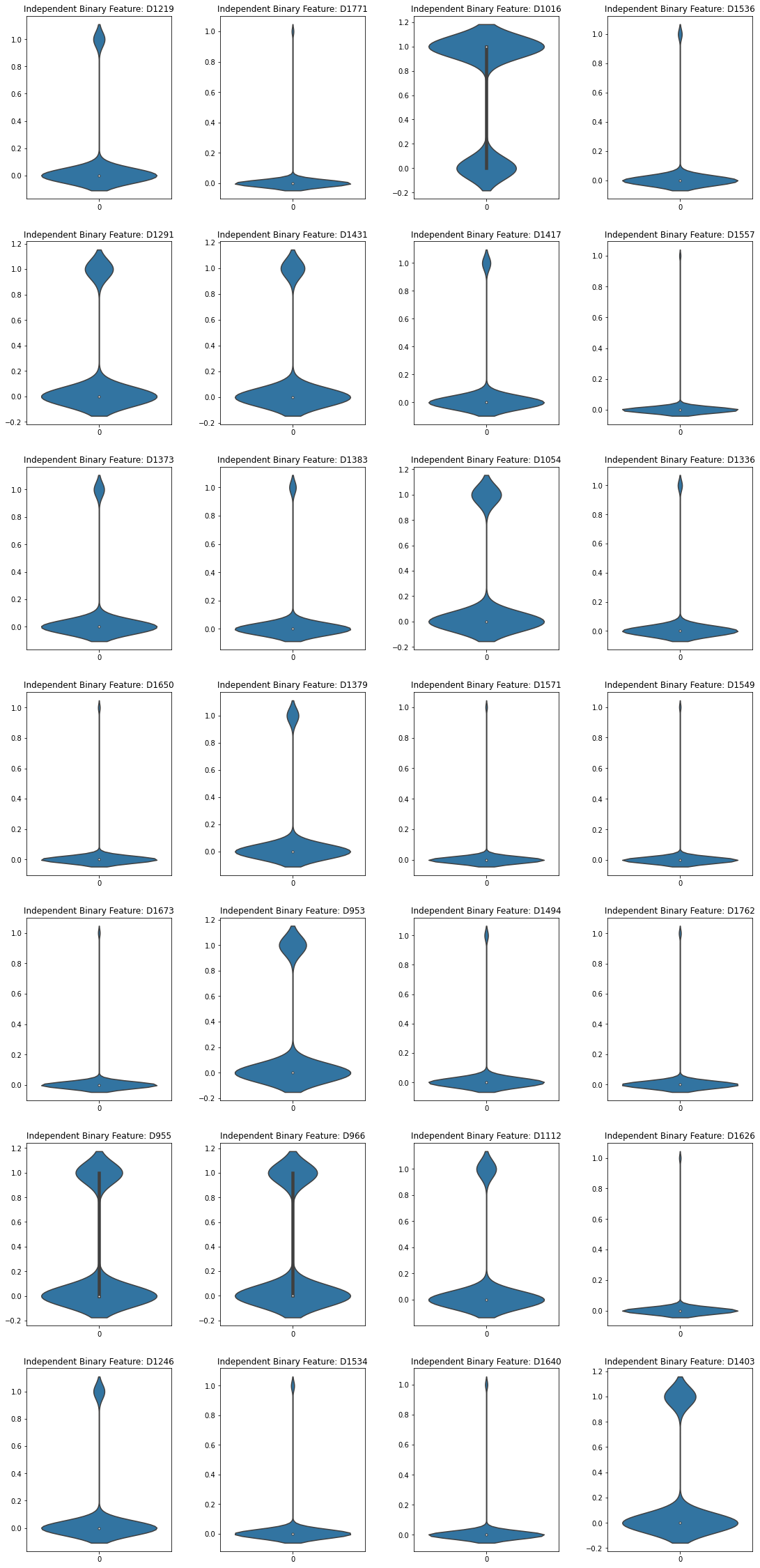
The dataset used to solve this classification problem was obtained from Kaggle's database, which can be found here. [‘https://www.kaggle.com/c/bioresponse’]

# Data Preparation

Since this problem involves binary classification, it is expected that the dataset will contain a target feature (a sample can only consist of one of two things) that involves using a range of features (pieces of information in a given dataset) about a molecular function to predict whether or not a biological response occurred. The dataset had 2501 rows and 1777 columns, with 1 target column labeled "Activity" and 1776 columns labeled "D1- D1776" (known as descriptors). The dataset had already been normalised, and it was predicted that one hot encoding was performed on the dataset as 835 of the features were binary features. Also, the dependent variable, “Activity” had binaries (1,0) which then necessitated the adoption of classification algorithms for machine learning task. Next, a check of class imbalance in the “Activity” variable was conducted, where it was observed, as shown in the plot below, that the classes were approximately balanced.

****

The plot below describes the distribution of features; it tells us that certain values are distributed almost equally in each variable, while others are distributed unevenly, leading to the assumption that the dataset may contain outliers, because the majority of these values will be far from the mean; this can also be seen with a box plot and kinetic density estimator (kde), which will tell us whether the data is right or left skewed.



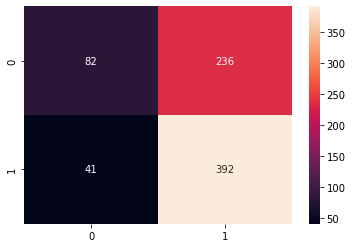
# Minimum Viable Model

"A Minimum Viable Model (MVM), and the process that surrounds it, strives to maximize the early understanding of an ML/AI problem domain in a real-world setting while limiting investment of time and resources," according to (Jeff Bienkowski, 2019). The main question, though, is what this can achieve for us in this project.

This is a planning process in which we try to see the performance characteristics of the models (with visuals, of course) that can handle the classification problem, as well as the quantity and quality of the data. For example, the various models will be tested, and the various hyperparameters will be tuned to get the best accuracy score, and a comparison of the models will be performed to determine which has the best score before it is selected as the most suitable model for the dataset, thereby achieving the project's goal.

## Naïve Bayes

The Naïve Bayes model had an accuracy score of 63% (0.6311584553928096), and a log loss error of 12.739586905970421 which should be better.



# The confusion matrix of the Nave Bayes shows the actual value and predicted value. We can see that the Nave Bayes did not perform well, with a predicted value of 236 of false positive of classifying the molecule description to the actual value of 0, when it wasn't 0, and a false negative Type II error of 41, predicting 41 values of classification of 1.

On submission of the generated CSV to Kaggle, the score was (lower is better):

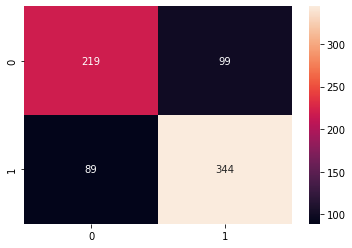
* Public score: 13.11529
* Private score: 12.34188

The leading score was 0.37355.

It is seen that the model performs better on test data than on the validation data. Better models or more feature engineering and hyperparameter tuning was employed to improve the score.

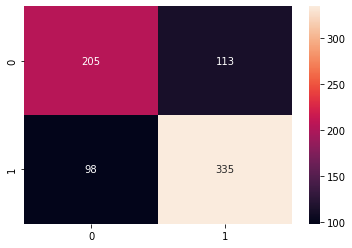
## Logistics Regression

The confusion matrix predicts 99 for the False Positive values (Type I Error) of the classified binary of the target value of "0," with a False Negative of 89 for the classification binary of 1. Overall, the model predicted 219 and 344 correct True Positive and True Negative values for the values of 0 and 1, with a log loss error of 8.646297100384551, 84% which is pretty good.



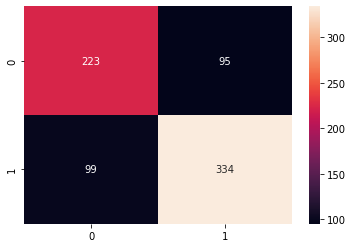
## K-Nearest Neighbors

The confusion matrix of the K-Nearest Neighbors model predicts values of 113 and 98 for the false Positive and False Negative (Type I and II Error) of the classification binary values of 0 and 1, respectively, indicating that the model correctly predicted the binary values of 0 and 1 with scores of 219 and 344, with a log loss error of 9.7040 and an accuracy score of 74 %.



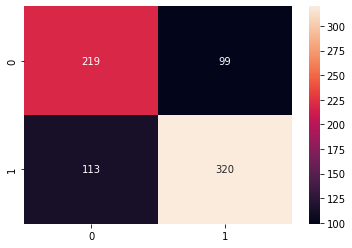
## Support Vector Machine

Although the Support Vector Machine model outperforms the preceding models in terms of accuracy and prediction. Its classified binary value prediction is 223 and 334 for True Positive and Negative, respectively, and 95 and 99 for False Positive and False Negative (Type I & II Error). Surprisingly, its accuracy score is 74%.



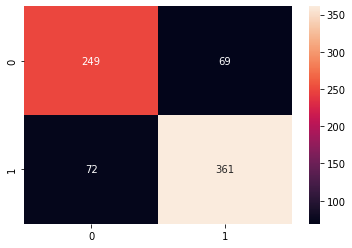
## Decision Tree Classifier

The Decision Tree Classifier clearly indicates the anticipated values of 219 and 320 respectively for the True Positive and Negative of the target features, while its faulty predictions are 99 and 113 respectively which are once again the False Negative and Positive (Type I & II Error) (Type I & II Error). With an accuracy score of 71%.



## Random Tree Classifier

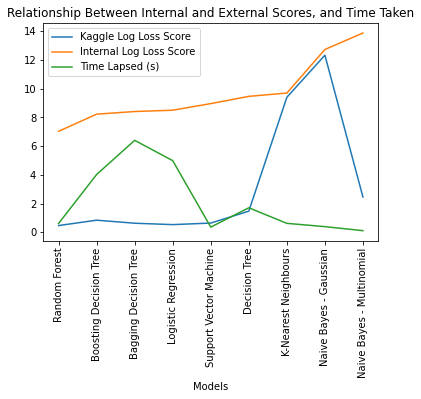
As we can see, the Random Tree Classifier model has an accurate prediction of 249 and 361 appropriate predictions of the categorized binary values of 0 and 1, but a lower score of False Negative and Positive (Type I & II Errors) where the model predicts incorrect values for the target feature. With an accuracy score of 81%.



So, after looking at the various models and their various performance measurements, we can see that, surprisingly, the Random Tree classifier has the best predicted outcome, but logistics regression has the highest accuracy score, and this is after tuning the hyper-parameter to get the best out of the model scores.

## **Internal and External Scores and Time Taken**

The graph below was displayed to show the link between the internal scores and external scores (i.e., the scores from Kaggle competition on this dataset) and the time spent to derive these scores, as a result of the Kaggle competition, it takes longer to derive the log loss scores of the Nave Bayes model, which in this case is the performance measurement, although internally within the SourceForge Project phase Bagging Decision Tree required the most time to derive the log loss score. Although internally SourceForge had better scores.

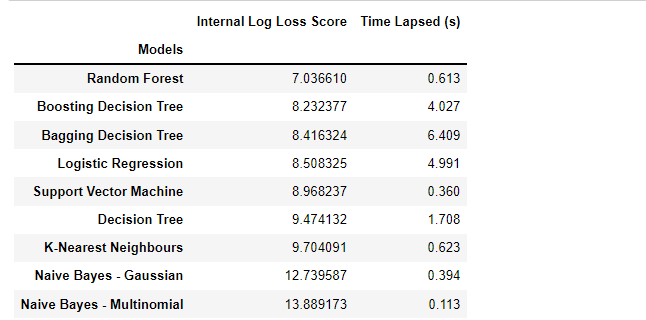


## Result

To clarify the results of this experiment, the table below clearly shows the results of the various models for both the Kaggle scores (public and internal), this is the log loss score on submission to Kaggle, as shown below, from best to worst performing (lower is better).

To summarize the findings, it was still thought to be a good option to also perform more feature engineering methods such as Elastic Net regularization, which had an intercept of 0.53367, as the score and intercepts for both options were the same, it can be concluded that the binary columns do not need to be excluded from the regularization process. Also, comparing the scores of the regularization with the model scores for untuned data, it was evident that regularization is marginally beneficial, as it would arrive at number 5 in the table shown above for Kaggle scores. It had:

* Private score of 0.68822
* Public score of 0.69185



# 

# Hyperparameter And Parameter Tuning

To improve the performance of the models, the model hyperparameters were tuned. Two best performing models were selected for this step. These models are random forest and XGBoost. The tuning process was carried out in two steps:

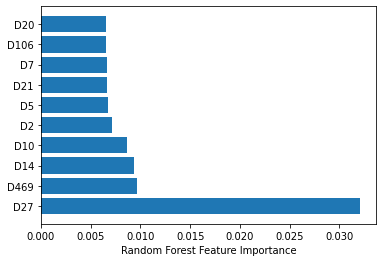
Step 1: Rough tuning using RandomSearchCV

Step 2: Fine tuning using GridSearchCV

The tuned Random Forest performed better than the tuned XGBoost. For the tuned Random Forest the log loss score on unseen data in Kaggle improved from 0.46893 to 0.41477.

# Feature Importance

Using the best performing model, the tuned Random Forest, the most influential features were identified. The top most influential features are shown below, starting with the least influential of the ten to the most influential.



# Deployment

Pickle is a useful Python tool that allows you to save your ML models, to minimise lengthy re-training and allow you to share, commit, and re-load pre-trained machine learning models. We pickled the tuned and fitted random forest model for use in the development of a web-based prediction system.

The web-app was developed using python’s 3rd party Streamlit library, and it was deployed to the Streamlit Cloud through a GitHub repo.

The app can be accessed through this link:

https://kiomegm-molecule-biological-response-appddd-enr6dz.streamlit.app/

# Conclusion and Recommendations

The process of determining the biological response of molecular functions in the provided split dataset is depicted in the analysis's outcome; however, this results in an instance in which a new dataset can be provided to the created or selected model to enable it to make accurate predictions with a 74 percent accuracy rate.

This may open the door to additional research on the subject, which would be extremely beneficial to the scientific research community.

Thanks for reading.

# **Team Members**

1. Rogers Kiome
2. Nimat Ibrahim
3. Alfred Aminayanate Bob-Manuel
4. Yaw Adjei Darko
5. Oriih Gloria
6. Damilola Idris
7. Ketul Patel
8. Miracle Uche