Carleton
UNIVERSITY

SYSC5405: Pattern Classification and Experiment Design

Professor: James Green

Final Project Report

Group - 07

Drug-Target Interaction Prediction using K Nearest Neighbor Algorithm

Soham Patel
Student No: 101184663
Department of Electronics
Carleton University
sohampatel@cmail.carleton.ca

Aditi Biswas
Student No: 101193708
Systems and Computer Engineering
Carleton University
aditibiswas@cmail.carleton.ca

Raparthi Sai Gouthami Priyanka
Student No: 101212227
Systems and Computer Engineering
Carleton University
saigouthamipriyanka@cmail.carleton.ca

ABSTRACT

Drug and disease detection has been a very important activity in the field of medical sciences. The demand for drug industry grew year by year exponentially as new diseases come to the limelight every single year. Creating and experimenting with new drugs is a tedious and at the same time involves lot of financial burden and is a time-consuming procedure. When drugs interact with protein structures they react and produce reactions inside the host which can cure a disease which is why this is a potential research area for modern day researchers. It can save a lot of money, time and effort when existing drugs combined with a different protein target can cure new diseases. This is the primary motivation behind this project. A stable DTI is formed when a particular drug has strong binding affinity towards a protein target which can modify its properties and cause a desirable reaction. In this project we use the K-Nearest Neighbor classification machine learning algorithm to predict the likelihood of a drug combining with a protein target based on a variety of features. Further, a meta learning approach - bagging is implemented to increase the accuracy of the model. KIBA score can be considered as a measure of binding affinity which can be used as a significant metric to know if the pair interacts or not which is also predicted using the Linear regression algorithm. The final goal of this entire project is to design a classification algorithm that can be further used to train an ensembled reciprocal perspective model.

Key Words: *Drug-Target Interaction (DTI), KNN, Classification, KIBA score, Meta learning*

I. INTRODUCTION

The primary target of drugs is divided into four types of namely receptors, ion channels, enzymes, and carrier molecules[1]. Wet lab experiments to identify interactions are expensive but when replaced with computational methods using machine learning open various new possibilities.[2]

The initial datasets were extracted from Densely Annotated Video Segmentation (DAVIS) and Binding database. Binding DB has features where one can query the name of the protein target, chemical similarity, and substructure. The aim of this project is to design a cascaded meta model ensembled using the reciprocal (RP) feature vectors which can be used to perform the final prediction. The K-Nearest Neighbor is a supervised classification machine learning algorithm that can solve both classification and regression problems. The K value signifies the number of nearest neighbors to consider for a given data point. The final classification is based on voting process on the nearest possible neighbors. The class label that achieves the highest number of votes is assigned to the new data point.

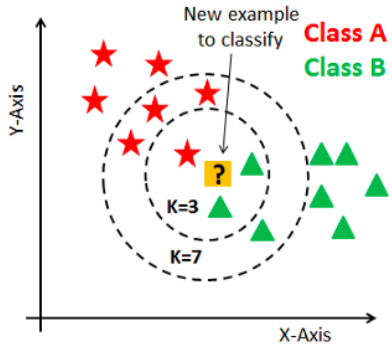

Figure 1: KNN algorithm's visual representation (Image Credit)

KNN algorithm has been proved to be efficient to numerous machine learning model. However, this algorithm has some pros and cons that are listed below:[3]

Pros: It is a non-parametric algorithm. It does not need to adopt any assumptions about the modeling problem. This strength about the algorithm makes it convenient to implement for

multiclass problems. Unlike most of the classifiers, it has only one hyperparameter (K value of the nearest neighbor) that needs to be tuned to get the best possible outcome from the classification model.

Cons: the computational time of the algorithm is comparatively slower. As the dataset grows, the speed of the algorithm decreases fast. Imbalanced dataset affects the algorithm as it tends to be biased towards the majority class. Handling class imbalanced dataset is a crucial factor before applying K-Nearest Neighbor algorithm.

II. METHODOLOGY

a) Simple Exploratory Data Analysis and Visualization:

Initially we used various techniques to visualize the data as data visualization is a crucial step to understand the dataset. Here, we looked out for missing values, range of features, correlation between features and if any class imbalance exists between the positive and negative class.

No missing value and outliers were found in the given training dataset. However, the dataset has huge class imbalance. The major class is the 'False' class which contains 86324 data samples, and the minority class is 'TRUE' class which contains 23155 data samples. The observed class imbalance ratio between both the classes is ¼. The graphical representation of the dataset's class imbalance is given below-

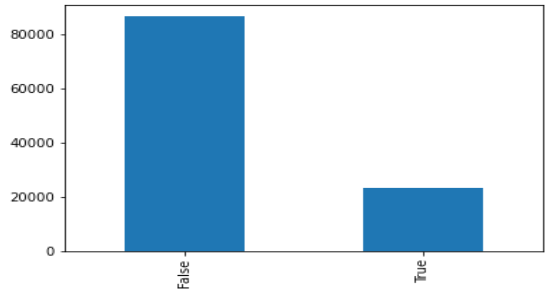

Figure 2: Class Imbalance

We visualize the features correlation by plotting heatmap. The correlation heatmap is plotted for the features of Group 26 which is shown in Figure 3.

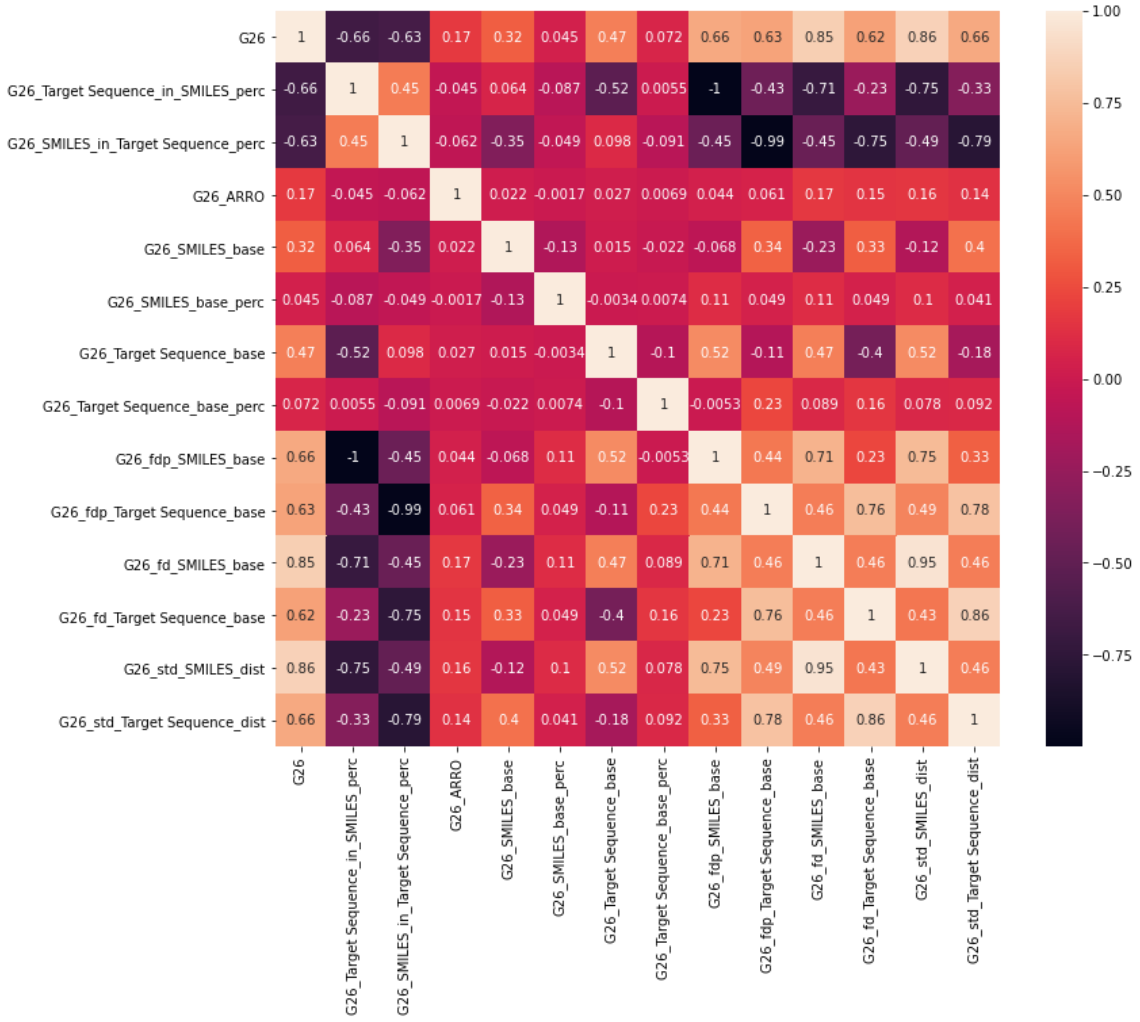

Figure 3: Correlation Heatmap

b) Experiment Design:

Jupyter Notebook is used for the environment setup and implementation of the project.

The experiment design is divided into two stages training and predicting as shown in Figure 4.

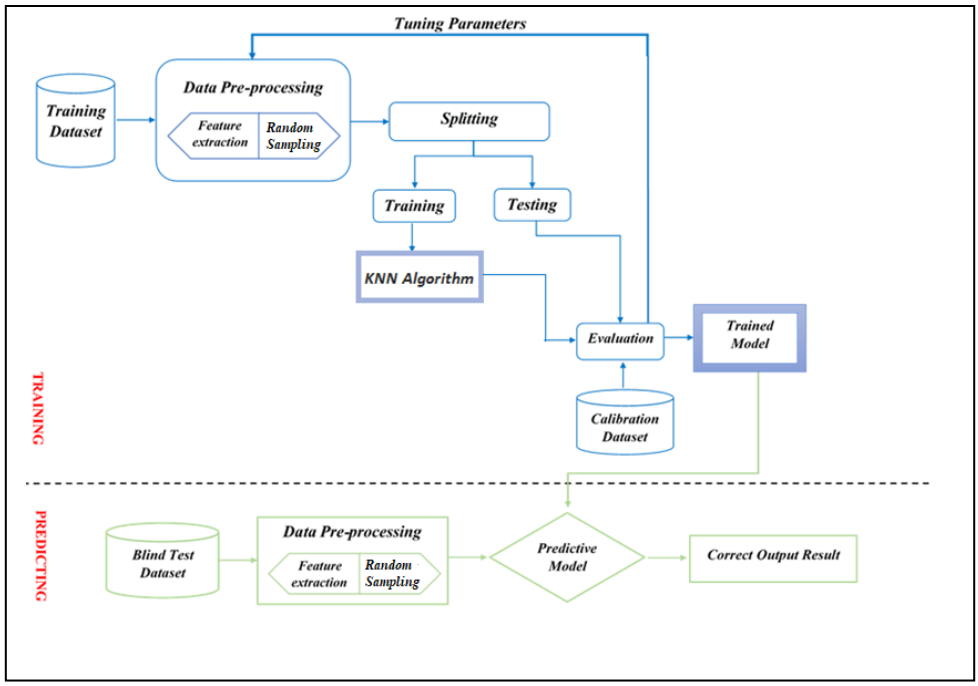

Figure 4: Experiment Design

c) Data Pre-Processing

Data Splitting: The primary training data file contained 109480 data samples. It is split primary further training and testing. We followed "Stratified Sampling" while splitting the dataset. "Stratified Sampling" in hold out test set means to divide the test dataset from the actual full dataset in such a way so that it has the right representation of class values. From the whole training dataset, for training we kept 100720 data samples and for testing 8759 data samples were kept aside. The size of test data was intentionally kept similar to the future blind test dataset.

Data Normalization: For normalizing the dataset Standard Scaler is used. The idea behind Standard Scaler is that it transforms the data in such a way that mean of the distribution is zero and standard deviation of one [4]. Standard Scaler is performed at feature level which means every feature of the dataset is normalized.

Class Balancing: As the dataset has a high-class imbalance, before proceeding to classification approach a balance of the classes is obtained. Random oversampling technique is used to balance

as it is very simple and straight forward. This technique balances the data by creating the replication of the minority class samples. This process does not cause any loss of the data sample's information. The random oversampling technique increased the minority class from 21% to 40%.

Feature Extraction/Selection: Principal Component analysis (PCA) is applied for extracting the mot significant features. PCA is a technique of collecting important features (in form of components) from a large set of features that are available in a data set[5]. PCA helps reduce the dimensionality of our feature space by dropping the least significant variables and keeping the major contributors intact. The variables that are shortlisted are independent of each other. PCA is based on eigen vectors which represent the direction in the scatterplot of data and eigen values represent magnitude, where bigger eigen values essentially correlate to more important directions [6].A feature subset of the initial dataset was selected which contains 107 features out of 336 feature set (excluding 'KIBA' and 'Label' feature column). However, the subset of the features selected by PCA was not used in final model as all feature set resulted in a better performance than the selected 107-feature subset.

d) Training, Validation and Testing:

Fold	Precision	Accuracy
Fold - 1	0.62	0.83
Fold - 2	0.62	0.83
Fold - 3	0.61	0.83
Fold - 4	0.62	0.83
Fold - 5	0.62	0.83

Table 1: K-Fold cross validation - testing

We performed a five-fold cross validation on our test data. The confusion matrices of testing in each fold are observed along with precision and accuracy.

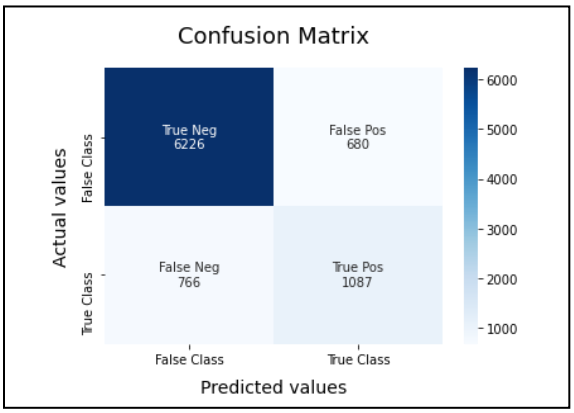

Figure 5: Fold-1

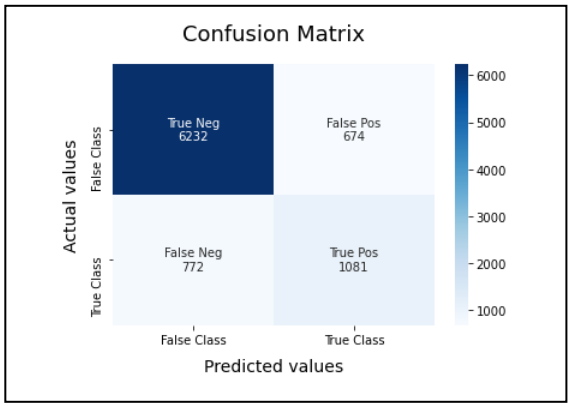

Figure 6: Fold -2

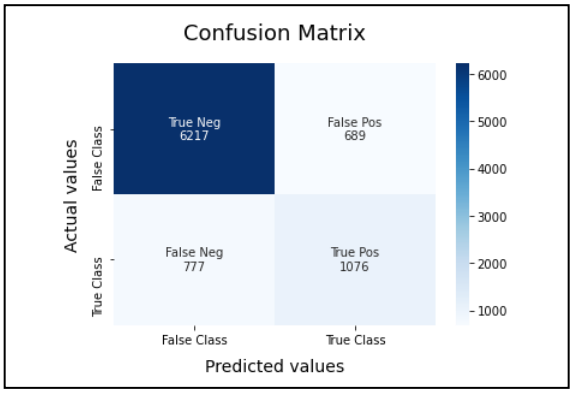

Figure 7: Fold -3

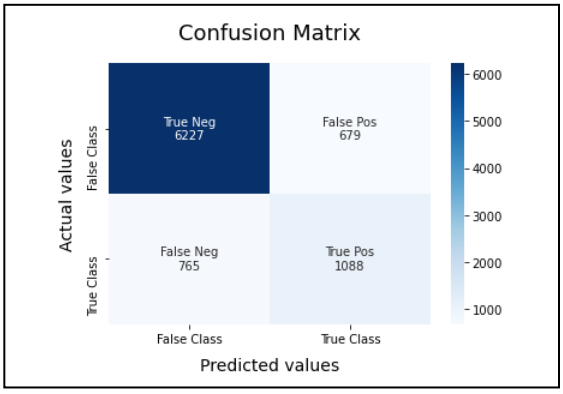

Figure 8: Fold - 4

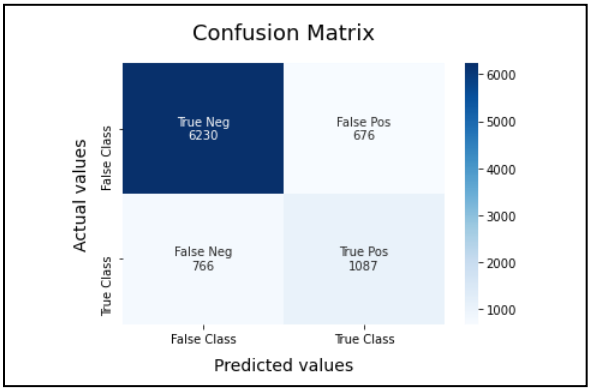

Figure 9: Fold - 5

As a part of hyperparameter tuning, the dataset is used for predicting the optimum K value. The classifier gave the maximum accuracy when the K value is nine and hence it is selected while creating the model.

III. META LEARNING APPROACH

'Bagging' is chosen as the meta learning approaching in this project. Bagging also stands for 'Bootstrap Aggregating'. The aim of this approach is to fit several independent models and calculate the average performance of all the models so that the variance will be reduced. Bagging approach is proven to fulfill these requirements [7]. However, to achieve this practically we need a huge dataset. Hence, in this scenario we depend on bootstrap samples to fit the models which are almost considered to be independent. Bootstrapping sampling involves creating multiple subsets of training data with replacement. Figure10 is a simple visual representation of the bagging approach.

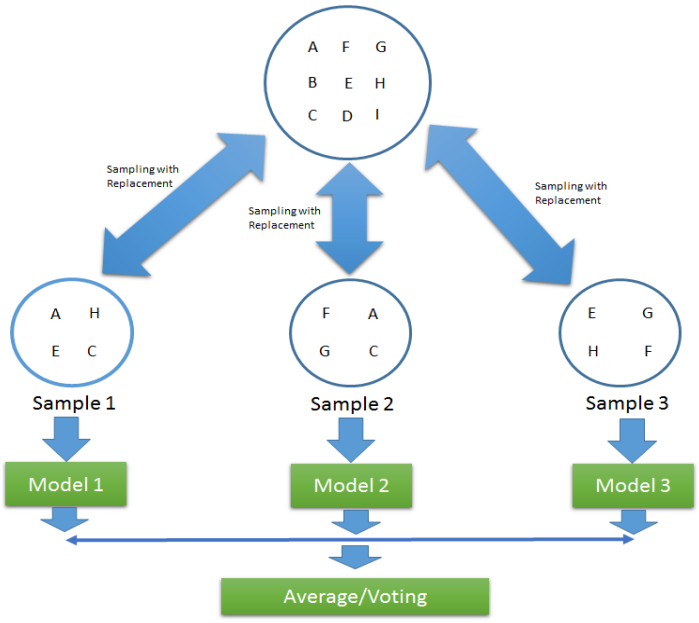

Figure 10: Visual representation of Bagging [[Aditi_ref_5]]

In our project we applied Bagging Classifier for our KNN model as a base estimator. The number of estimators was five.

IV. RESULT ANALYSIS

a) Estimate And Prediction of Precision-Recall without Meta Learning Approach:

Testing is performed on the test data set that we split earlier and a confusion matrix is plotted. The precision and accuracy for test data without meta learning approach is observed. A confusion matrix for the same is plotted as shown in Figure 12. The precision is observed to be 0.63 and accuracy is 0.84. Finally, the precision recall curve is plotted as shown in the Figure 11 to estimate the recall at 50% precision for the given classifier to better understand our model's performance with different thresholds and point out the best precision on recall>=50. The best precision observed is 0.698 along with the standard deviation of 0.119.

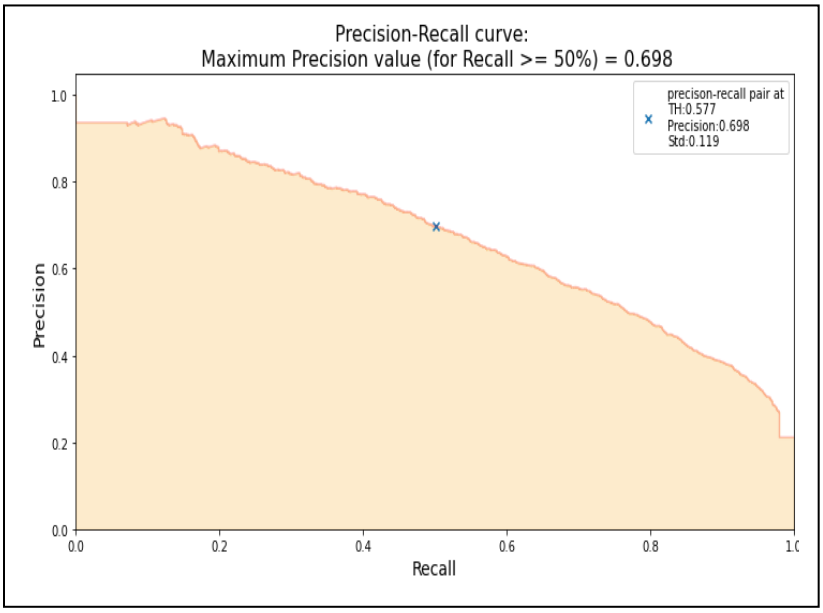

Figure 10: PR curve before Meta Learning

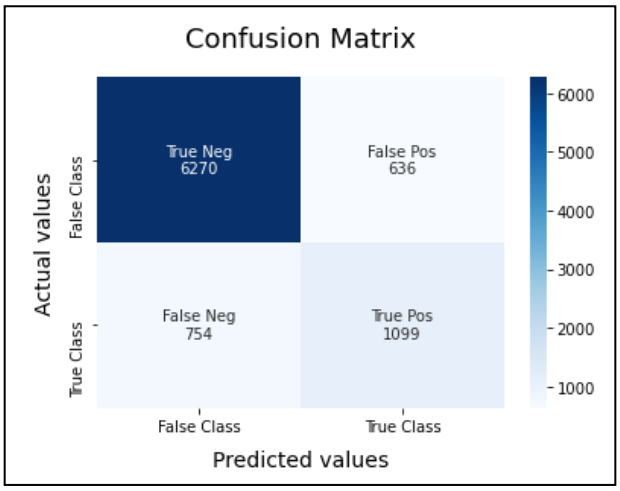

Figure 11: Confusion matrix before Meta Learning

b) Prediction of Precision-Recall with Meta Learning Approach:

Using the bagging approach did not improve the model as the precision declined when compared to the one achieved without the meta learning approach. The precision and accuracy achieved are 0.55 and 0.79 respectively.

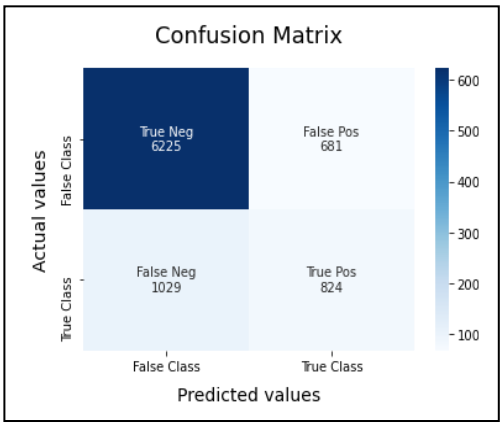

Figure 12: Confusion matrix after Meta Learning

PRECISION	ACCURACY
0.55	0.79

Table 2:Testing parameters

The PR curve is also plotted for the meta learning approach as shown in Figure 13.

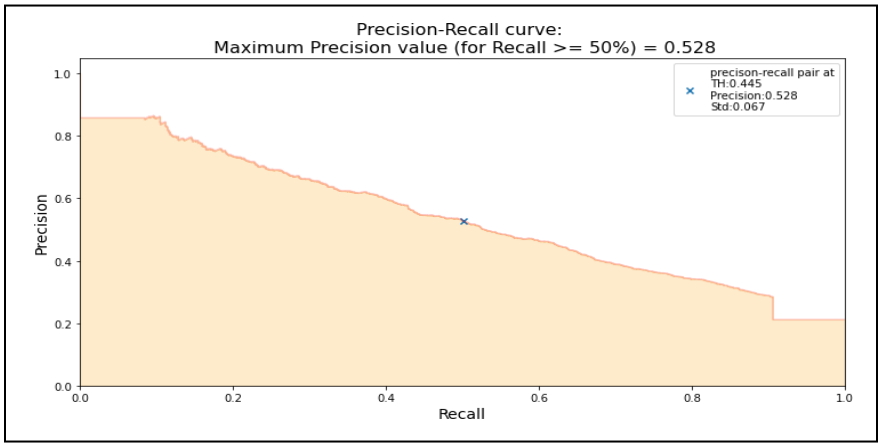

Figure 13: PR curve after Meta Learning

The estimate for prediction of precision value for future data was based on calculating the mean of all precision values from PR curve and standard deviation of the same. **Finally, the predicted precision for the given dataset is observed to be 0.497±0.186**

c) Actual Performance on Blind Test Data:

Our model outperformed when compared with the initially predicted precision and accuracy. The actual accuracy is found to be 0.635 when the model was tested on the blind dataset.

d) **Regression – KIBA score prediction:**

The prediction of KIBA score is carried out using the linear regression algorithm. The dataset is split in the ratio of 80:20 in which the 20% is used for testing the algorithm. Mean square error was calculated to evaluate the model. The estimated mean square error using the test data is 0.728 and the actual score on blind dataset is 0.7001. Other models and algorithms can also be used to improve the accuracy of prediction.

V. FUTURE WORK FOR MODEL PERFORMANCE IMPROVENT

1) We could try to check different feature selection techniques as a part of data preprocessing and see if it improves the model's performance. There are many other variables in data preprocessing that we could try differently like different split strategy, normalizing techniques, etc. to see if it makes any difference.
2) Keeping aside project rules, we can think of what techniques/ algorithms can perform well in our case of imbalanced large dataset. Some changes in training, testing strategies, different meta learning strategies may make a significant difference in model's performance.
3) The fundamental learning outcome from this project/ course is tradeoff and importance of focusing on correct performance metrics. We should give priority to aspects required for our problem statement and move forward with experiment. Let it be choosing extractor, classifier, strategies to split, meta learning, preprocessing or every other variable present in our process flow.

VI. REFERENCES

[1] H. McGavock, Ed., *How Drugs Work: Basic pharmacology for healthcare professionals*, 4th ed. London: CRC Press, 2016. doi: 10.1201/9781315385020.

[2] K. Sachdev and M. K. Gupta, "A comprehensive review of feature based methods for drug target interaction prediction," *J. Biomed. Inform.*, vol. 93, p. 103159, May 2019, doi: 10.1016/j.jbi.2019.103159.

[3] Genesis, "Pros and Cons of K-Nearest Neighbors," *From The GENESIS*, Sep. 25, 2018. https://www.fromthegenesis.com/pros-and-cons-of-k-nearest-neighbors/ (accessed Dec. 17, 2021).

[4] “StandardScaler, MinMaxScaler and RobustScaler techniques - ML,” *GeeksforGeeks*, Jul. 15, 2020. https://www.geeksforgeeks.org/standardscaler-minmaxscaler-and-robustscaler-techniques-ml/ (accessed Dec. 18, 2021).

[5] “PCA: Practical Guide to Principal Component Analysis in R & Python.” https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/ (accessed Dec. 18, 2021).

[6] “Understanding the Role of Eigenvectors and Eigenvalues in PCA Dimensionality Reduction. | by Joseph Adewumi | Medium.” https://medium.com/@dareyadewumi650/understanding-the-role-of-eigenvectors-and-eigenvalues-in-pca-dimensionality-reduction-10186dad0c5c (accessed Dec. 18, 2021).

[7] “Bagging — Ensemble meta Algorithm for Reducing variance | by Ashish Patel | ML Research Lab | Medium.” https://medium.com/ml-research-lab/bagging-ensemble-meta-algorithm-for-reducing-variance-c98fffa5489f (accessed Dec. 18, 2021).