Mechanism of action

Rishav Raj
IIITD,
CSAI, 2021556

Parth Kaushal
IIITD,
CSAI, 2021548

Shubham Pal
IIITD,
CSAI, 2021564

Garv Makkar
IIITD,
CSAI, 2021530

October 2023

1 Abstract

Discovering drugs has always been a crucial research domain. There have been various developments, and in today's time, with increasing technology, there has been rapid growth in this field. Our project aims to predict a molecule's biological activity, which is referred to as mechanism-of-action. The mechanism of action is an essential aspect of drug development. It can help scientists in the process of drug discovery. The problem has been approached with various machine learning algorithms. Since drugs can have multiple MoA annotations, it is a multi-label classification problem. The data provides insights into the activity of genes and the responses of cells to various drugs. Various preprocessing steps have been done, like removing skewness from the data, applying PCA, and scaling the data. It was realized that removing outliers increased the loss calculated. Models implemented are Naive Bayes, Logistic Regression, SVM, AdaBoost, Random Forest, CNN, ANN and kNN with the respective cross-entropy loss (aka log loss) as 0.17579, 0.01717, 0.02716 , 0.01983, 0.0165, 0.1733, 0.03114 respectively. Within these models, it was concluded that the CNN has performed the best.

2 Introduction

This report explores a machine learning project led by The Connectivity Map in partnership with MIT, Harvard, LISH, and the NIH. It aims to improve how we predict how drugs work, which is vital for developing new medicines. In the past, medicines were discovered by chance or traditional methods with a partial understanding of how they worked. We aim to understand how diseases work and find specific targets in our bodies. The Mechanism of Action (MoA) of a drug refers to the biological process by which a drug produces its therapeutic effects by understanding how it interacts with the body's biological systems. This is done by treating human cells with the drug and using computer programs to compare the results to a vast database of genes. The database used, which is provided by a Kaggle competition, has MoA information for over 5,000 drugs. Using the training data, we aim to create a program that can automatically tell us a drug's MoA. Since drugs can have multiple MoAs, we will use various multi-label classification machine-learning algorithms. The database provides the inputs like gene expression data and cell viability data.

3 Literature Survey

Classification of drugs based on mechanism of action using machine learning techniques — Discover Artificial Intelligence

This paper discusses the various machine learning models and their accuracies on the mechanism of action (MoA) dataset. This paper is related to our work as we are working on the same dataset; hence, seeing what methods have already been used will be helpful. The accuracy of the machine learning model was evaluated by applying the log loss function. The machine model that was tested in this paper is BRkNN(Type A and Type B) (Binary Relevance K Nearest Neighbors), ML-KNN (Multi-label K-Nearest Neighbors), and a custom Neural Network using Keras. The log loss for BRkNN-a was 0.11, for BRkNN-b 0.28, for ML-KNN it was 0.11, and using the custom neural network, the best result was obtained, around 0.017. The paper also describes integrating the neural network model into a web application using the Flask framework.

Mechanism of Action (MoA) Prediction - Kaggle Competition

This paper aims to propose various multi-label classification machine learning algorithms to predict a molecule's biological activity, which is referred to as mechanism-of-action, or MoA for short. This paper aligns with our work as we aim to solve the same problem statement using the same dataset. First, data exploration was done, and then for feature engineering, the data that did not generate MoA was excluded, PCA was applied to use the relevant features and feature augmentation was also done. Further, cross-validation was performed to evaluate the models like Neural Network, TabNet, and ResNet, having log loss of 0.0159, 0.0150, and 0.0147, respectively. We would explore more models and perform more data preprocessing as this paper needed more work.

Large-scale comparison of machine learning methods for drug target prediction on ChEMBL

The effectiveness of deep learning approaches compared to other machine learning and target prediction approaches in drug development tasks is discussed in the paper. The paper mentioned that the lack of large-scale studies and hyperparameter selection bias are some challenges that arise in evaluating the effectiveness of deep learning in drug discovery. Hence, a nested cluster-cross-validation strategy was used. The deep learning methods like FNN were better than other methods like SVM, RF, KNN, NB, SEA, GC, Weave, and SmilesLSTM. Hence, for our aim of drug target prediction, this paper gives a detailed comparison of various models and how deep learning has an edge.

4 Dataset and Preprocessing

4.1 Dataset Description

This dataset has been sourced from the Laboratory for Innovation Science at Harvard and is made available through a Kaggle competition. It combines information regarding gene expression and cell viability data. Specifically, it provides insights into the activity of genes and the responses of cells to various drugs. The data is generated using a novel technology that allows simultaneous measurement of how different types of human cells react to multiple drugs across a set of 100 different cell types. This technological advancement aids in addressing the challenge of identifying cell types that are most compatible with specific drugs.

As is customary in such datasets, it has been divided into two distinct parts: a training set and a testing set. The primary objective is to develop a computer program using the training data that can predict and assign one or more categories related to the mechanisms of action (MoA) to each case within the test set. It is essential to note that drugs can belong to multiple MoA categories, making this task a multi-label classification problem. Please bear in mind that the labels for the testing data are unavailable, and therefore, the test data will be generated by splitting the training data.

Within the training data, there exists an optional set of MoA labels that are not present in the test data and are not considered when evaluating the performance of models. The training data is organized into four separate comma-separated files:

- train features.csv: This file contains features for the training set. The features labeled with ’g-’ represent gene expression data, while those labeled ’c-’ represent cell viability data. The ’cp type’ column indicates whether the samples were treated with a compound (’cp vehicle’) or a control perturbation (’ctrl vehicle’). Control perturbations have no MoAs. The ’cp time’ and ’cp dose’ columns indicate the treatment duration (24, 48, or 72 hours) and dose (high or low).

- train drug.csv: This file contains an anonymous drug ID specific to the training set.

- train drug.csv: This file contains an anonymous drug ID specific to the training set.

- train targets scored.csv: This file contains binary MoA targets that are used for scoring and evaluation.

- train targets nonscored.csv: This file includes additional (optional) binary MoA responses for the training data. These responses are not used for prediction or scoring.

4.2 Evaluation

The log loss has been used as an evaluation metric. Predictions will be made for each unique sigid in the test data to determine the probability of a positive response for every MoA target. This means you’ll make a total of ”M * N” predictions, where ”N” is the number of sigid observations in the test data multiplied by the number of scored MoA (M) targets.

Each prediction, denoted as ”p,” represents the likelihood of a positive MoA response for a specific sigid.

In the context of evaluating these predictions, the term ”y” stands for the actual ground truth, where it equals 1 if there is a positive response and 0 if there isn’t.

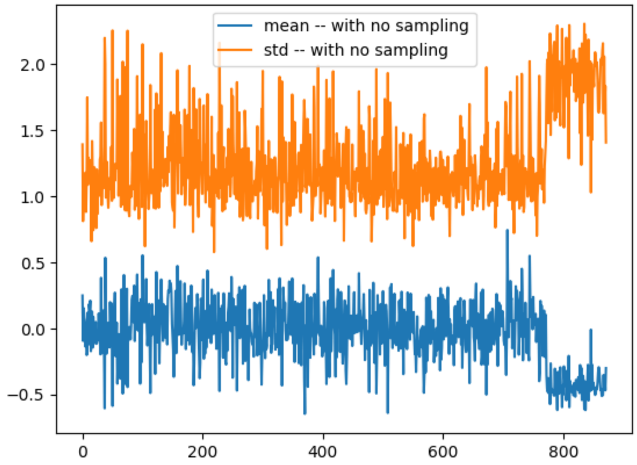
$$\mathrm{L} = -\frac{1}{M}\sum_{i=1}^{M}\frac{1}{N}\sum_{j=1}^{N}\left[y_{ij}\log(p_{ij}) + (1-y_{ij})\log(1-p_{ij})\right]$$

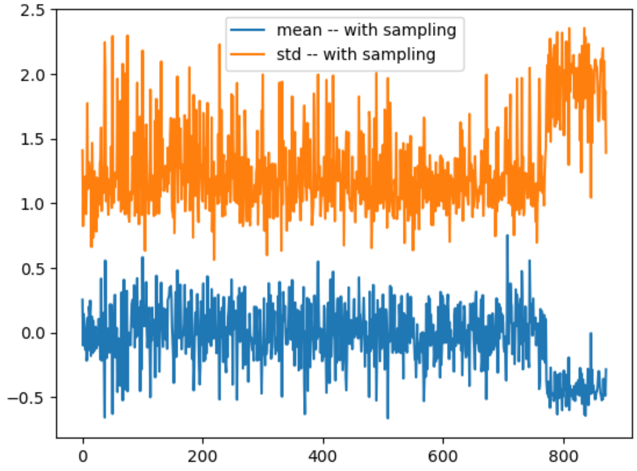
4.3 Preprocessing

The dataset consists of 23,814 rows, with each row representing a distinct data point. Each data point is described by 875 features, which encompass a range of characteristics or attributes. These features are employed for making predictions or conducting analyses on the dataset.

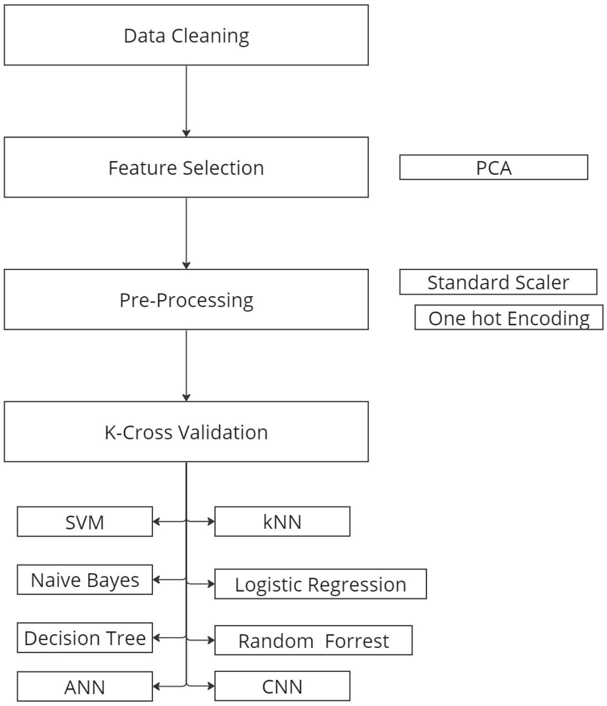
In this dataset, these attributes likely represent specific categories or classes that are relevant to the data.To handle these discrete attributes (‘cptime‘, ‘cpdose‘, and ‘cptype‘), a technique known as one-hot encoding has been applied. One-hot encoding is a method used to convert categorical data into a binary (0 or 1) format, creating new binary columns for each distinct category.

The remaining 872 features in the dataset are numerical in nature. These features likely represent continuous variables or measurements. To ensure that all features are on the same scale, a process known as standardization has been applied. This scaling ensures that no single feature dominates the learning process, and it can be particularly important for certain machine learning algorithms that are sensitive to feature scales. To enhance the computational efficiency of models like SVM, the training dataset, comprising around 23,000 rows, underwent a strategic sampling process. The objective was to alleviate the computational burden associated with large datasets. To gauge the impact of sampling on the dataset characteristics, the mean and standard deviation were meticulously analyzed before and after the sampling operation. The resulting plots convincingly demonstrate that the sampling process did not significantly alter the essential statistical properties of the dataset. This observation not only underscores the effectiveness of the sampling strategy in reducing computational complexity but also assures that the sampled dataset remains representative of the original dataset.





5 Methodology



5.1 Gaussian Naive Bayes - Baseline Model

In the context of using the Naive Bayes technique, an underlying assumption is that all features are independent of each other. However, when we analyzed the distributions of individual features through plotting, we observed that some of these features followed a roughly normal or Gaussian distribution. In contrast, the majority of features exhibited significant skewness, where their distributions were notably skewed to either the left or the right.

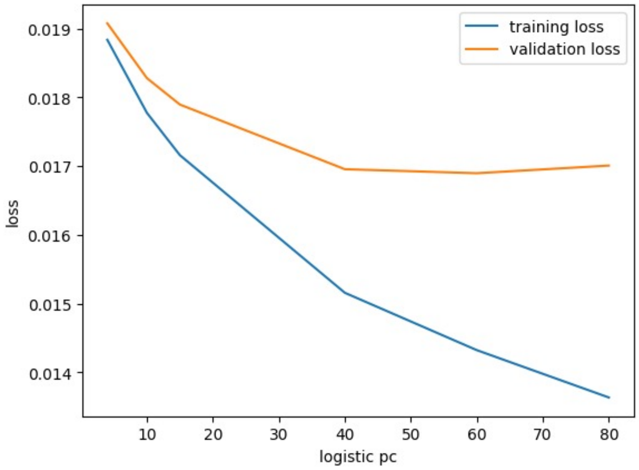
The Naive Bayes model was employed as the primary analytical tool. In order to enhance its performance and optimize the feature space, Principal Component Analysis (PCA) was utilized as a preprocessing technique. The parameters for PCA were carefully selected through a rigorous cross-validation process to ensure optimal dimensionality reduction.

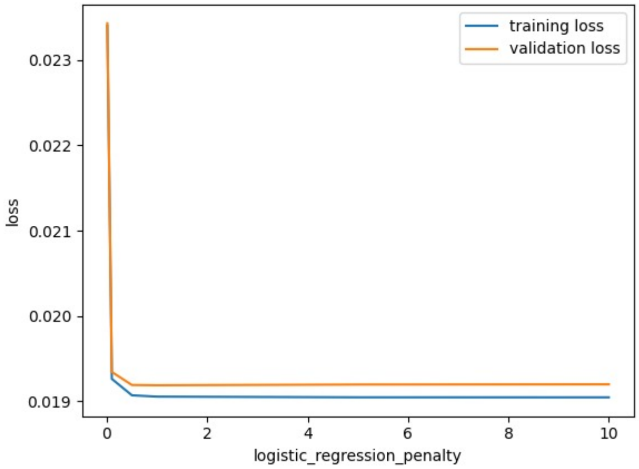
During the experimentation phase, the efficacy of Local Outlier Factor (LOF) as a preprocessing step was evaluated by monitoring the loss curves associated with different LOF parameters. Subsequently, a decision was made to eliminate LOF preprocessing from the model pipeline due to its limited contribution to model performance.

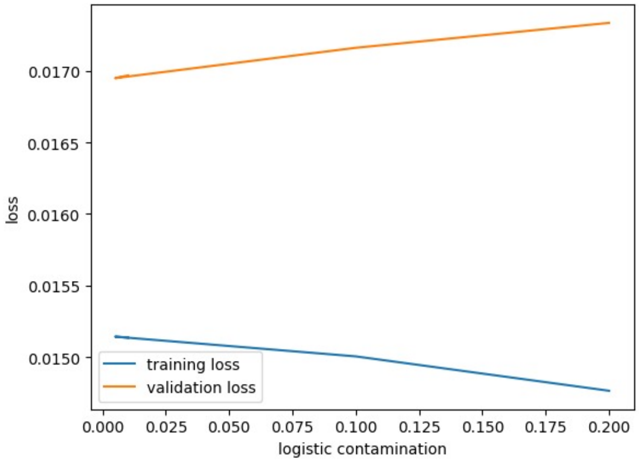
5.2 Logistic Regression

In the context of multi-label classification, a conventional approach involves constructing individual classifiers for each label, thereby independently modeling the presence or absence of each label for a given instance. Nonetheless, prior to employing these classifiers, critical preprocessing steps are implemented to improve model performance and robustness.

One of the key preprocessing steps is dimensionality reduction using Principal Component Analysis (PCA). The optimal number of principal components, a vital parameter in PCA, is determined through rigorous k-fold cross-validation, with k set to 3.



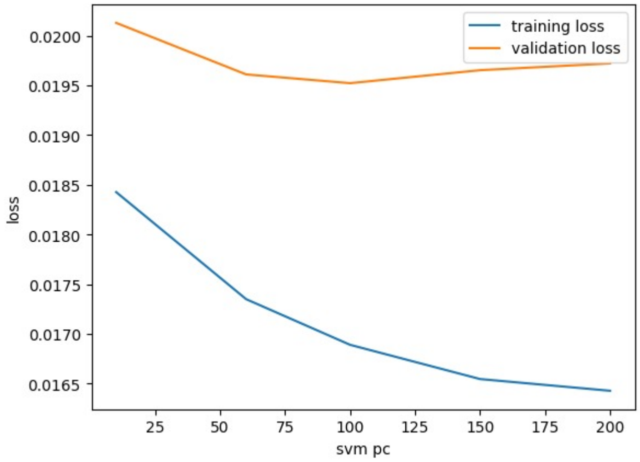


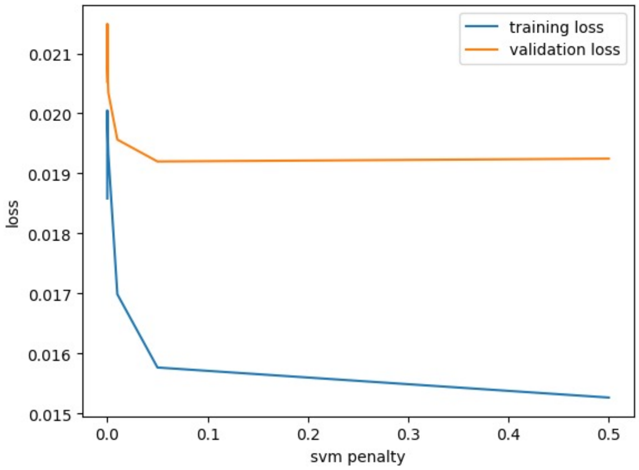


Furthermore, for the purpose of enhancing the model's resilience and its ability to identify outliers or anomalies within the dataset, the Local Outlier Factor (LOF) method is initially applied. LOF serves as a valuable tool for detecting data points that deviate significantly from the overall distribution, contributing to the model's overall robustness. However, upon a comprehensive evaluation, it was observed that the model's performance with LOF and without LOF yielded remarkably similar results. As a result, a decision was made to remove the LOF outlier detection method from the model.

5.3 SVM

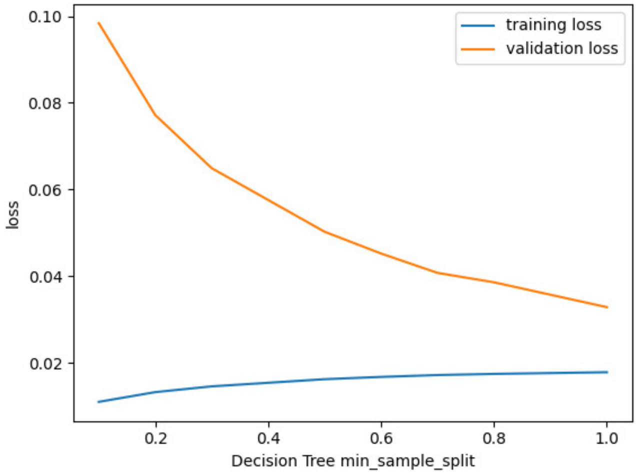
In our study, an analysis of the dataset revealed inherent non-linearity in class distributions within the feature space. This observation led to the adoption of a Support Vector Machine (SVM) equipped with a Radial Basis Function (RBF) kernel. This kernel choice, known for its prowess in handling non-linear data, leverages its ability to implicitly map the data into a higher-dimensional space where it becomes linearly separable. Recognizing SVM's sensitivity to feature scales, especially with RBF kernels, all features were normalized using StandardScaler to ensure zero mean and unit variance; care was taken to retain genuine data variability and prevent overfitting. The dataset's inherent biases posed another challenge. This was managed by meticulous tuning of SVM hyperparameters, specifically the contamination and neighbors parameters, ensuring the classifier did not unduly favor the majority class. Principal Component Analysis (PCA) was incorporated, given the high dimensionality of post-one-hot encoding. This reduced computational demands and addressed the curse of dimensionality, with varied principal components being tested for optimal balance between efficiency and information retention. Through these strategic preprocessing and tuning steps, the SVM with RBF kernel was adeptly tailored to our specific dataset, aiming for optimal classification performance.

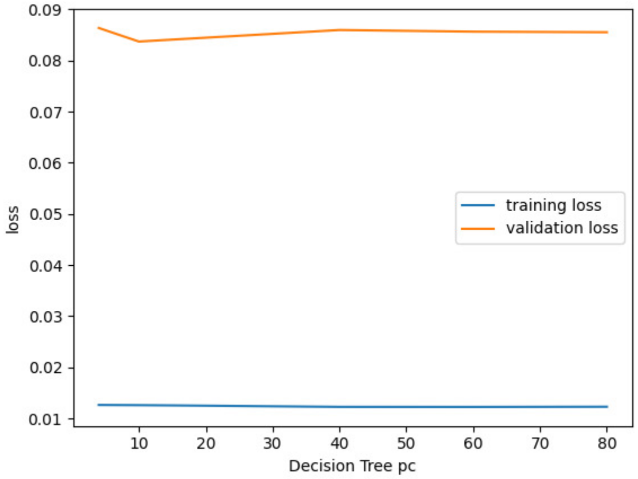




5.4 Decision Tree

A decision tree serves as a predictive modeling instrument, delineating potential decision outcomes through a sequential set of conditions. Its structure, akin to a flowchart, features nodes representing decisions or attribute tests, with branches denoting diverse potential outcomes.

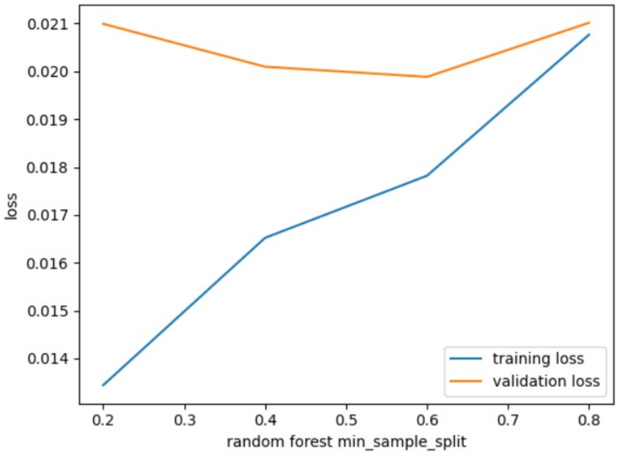


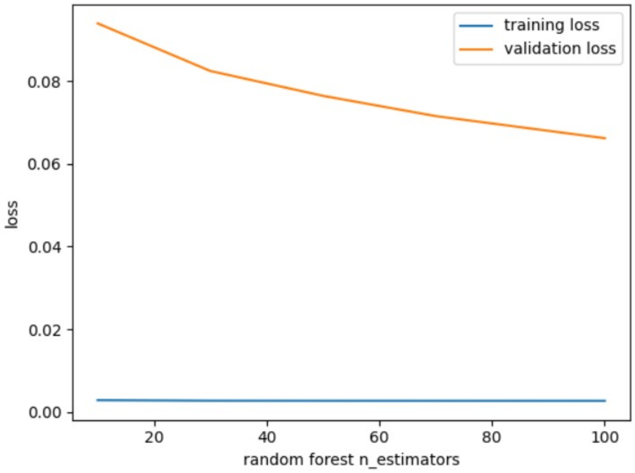


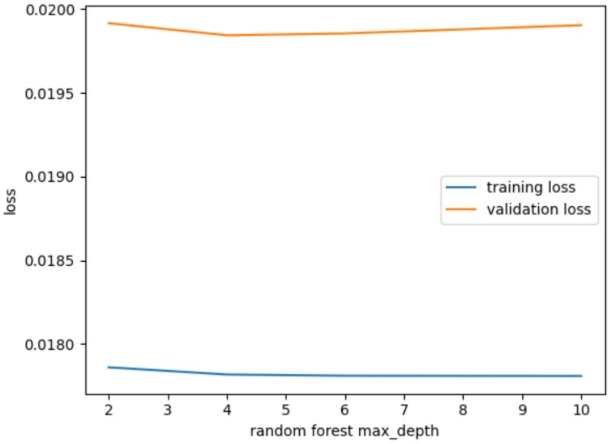
Optimal parameters for the model were discerned through k-fold cross-validation. However, upon subjecting the model to testing, it became evident that the decision tree struggled to yield satisfactory performance on the testing dataset.

5.5 Random Forest

Random Forest stands out as an ensemble learning technique designed to enhance predictive accuracy and mitigate overfitting issues often associated with individual decision trees. By constructing a diverse ensemble of trees, Random Forest leverages the strength of each tree trained on a random subset of both data and features, subsequently combining their predictions for improved robustness. This collective approach significantly bolsters generalization capabilities, rendering Random Forest a potent tool for addressing classification and regression challenges in machine learning. In comparison to standalone Decision Trees, which might struggle with generalization on unseen data, Random Forest excels in handling such scenarios adeptly. The ensemble method effectively addresses the limitations of individual trees, offering superior model variance and mitigating overfitting concerns.





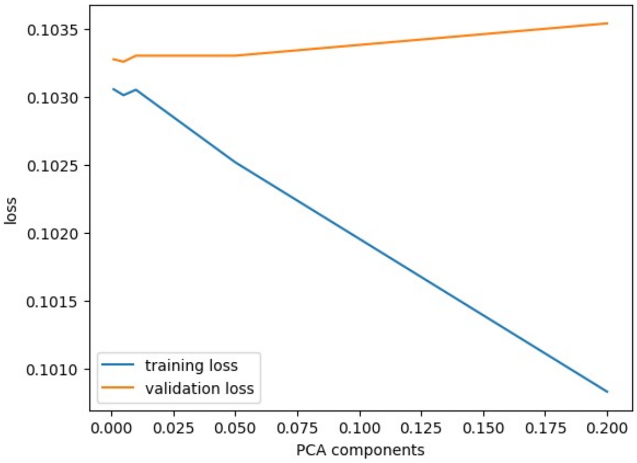


During the model tuning process through cross-validation, a systematic approach was employed. Initially, the hyperparameter `nestimators` was optimized, determining the optimal number of trees in the ensemble. Subsequently, the `min sample split` parameter was fine-tuned, followed by the optimization of `max depth`. This iterative manual refinement process was repeated multiple times to ascertain the optimal configuration of parameters. This meticulous technique, proven effective with Random Forest, was similarly applied to other methods, ensuring a comprehensive optimization approach across various machine learning models.

5.6 ANN

Artificial Neural Networks (ANNs) are machine learning models inspired by the human brain, comprising interconnected nodes organized into layers. They excel at capturing complex patterns and relationships in data, making them versatile for tasks like classification and regression. During training, ANNs adjust their weights to improve predictive capabilities based on labeled examples.

The training of the Artificial Neural Network (ANN) did not involve a sampling process. Notably, superior performance was achieved with decision tree and random forest models. This underscores the effectiveness of decision tree and random forest algorithms in handling high-dimensional data, showcasing their capabilities in comparison to ANN. For PCA value of 30, Solver algorithm as Adam, Activation Loss as Identity, Hidden Layers (3, 3, 3, 3), and 10,000 iterations, we received our optimal results.



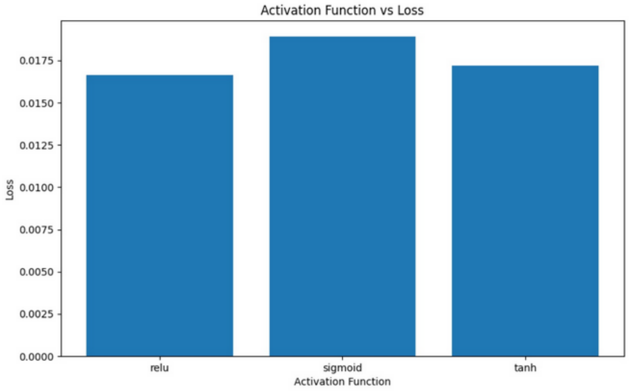
5.7 KNN

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm used for classification and regression tasks. The algorithm makes predictions based on the majority class (for classification) and assigns each instance to the one nearest it. As we increased the neighbors, our testing loss decreased. We checked up to 400 neighbors and got the optimal value at 400 nearest neighbors.

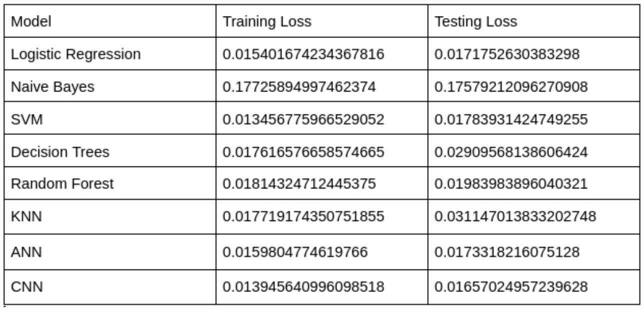
5.8 CNN

Convolutional Neural Networks (CNNs) are specialized deep learning models designed for processing and analyzing visual data. They excel in image recognition and feature extraction by leveraging convolutional layers to capture hierarchical patterns. CNNs have demonstrated remarkable success in various computer vision tasks, such as image classification and object detection.

Given the inherently high-dimensional nature of image datasets, CNNs are well-suited for such complex data structures. With approximately 875 features, CNNs excel in identifying and extracting the most relevant features, resulting in superior performance compared to other models employed in our analysis.



6 Result and Analysis



Model	Training Loss	Testing Loss
Logistic Regression	0.015401674234367816	0.0171752630383298
Naive Bayes	0.17725894997462374	0.17579212096270908
SVM	0.013456775966529052	0.01783931424749255
Decision Trees	0.017616576658574665	0.02909568138606424
Random Forest	0.01814324712445375	0.01983983896040321
KNN	0.017719174350751855	0.031147013833202748
ANN	0.0159804774619766	0.0173318216075128
CNN	0.013945640996098518	0.01657024957239628

The performance evaluation of various machine learning models reveals that Naive Bayes, Decision Tree, and KNN exhibit poor results, indicating them as baseline models. Notably, Decision Tree displays high variance, indicative of overfitting. In contrast, the Random Forest model outperforms the Decision Tree due to its ensemble approach, showcasing improved results. Logistic Regression, Support Vector Machines (SVM), and Artificial Neural Networks (ANN) demonstrate notably strong performance.

Among these, Convolutional Neural Networks (CNN) stand out by delivering the best performance on unseen data. This efficacy can be attributed to its effectiveness in handling high-dimensional data, where the large number of features is pivotal. The CNN excels in learning intricate features, contributing to its robust performance in dealing with previously unseen data.

7 Conclusion

In conclusion, this research paper was undertaken with the primary objective of developing predictive models for the prediction of Mechanism of Action (MOA). Throughout this study, several algorithms have been proposed and investigated as potential tools for MOA prediction. One notable aspect of this paper is the emphasis placed on explaining the rationale behind selecting specific parameters for the proposed models.
This explanatory approach is substantiated by graphical representations, which provide visual insights into the underlying data patterns and the significance of certain model parameters. Such elucidation enhances the transparency and comprehensibility of the model development process, fostering a deeper understanding of the chosen methodologies. In summary, this research endeavor has not only contributed novel algorithms for MOA prediction but has also strived to provide a clear and rational basis for the parameter selections made, thus enriching the interpretability and applicability of the proposed models in the domain of Mechanism of Action prediction.

8 References

[1] H. L. Gururaj, Francesco Flammini, H.A. Chaya Kumari, G.R. Puneeth, B.R. Sunil Kumar. Classifcation of drugs based on mechanism of action using machine learning.

[2] Lombardi Alessandro, Polvani Niccolo, Zacchei Filippo. Department of Mathematics, EPFL, Lausanne, CH. Mechanism of Action (MoA) Prediction - Kaggle Competition.

[3] Andreas Mayr, Gunter Klambauer, Thomas Unterthiner, Marvin Steijaert,b Jorg K. Wegner, Hugo Ceulemans, Djork-Arne Clevert and Sepp Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL