Machine Learning Project Presentation

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Rishav Raj 2021556
Shubham Pal 2021564
Parth Kaushal 2021548
Garv Makkar 2021530

Motivation

Drug Discovery Significance:
Our healthcare and pharmaceutical industries are critically dependent on the discovery of new drugs.

Enhancing Drug Development:
MoA of drugs help in drug development as they provide really good insights about how they work in the body.

Transition from Traditional Methods:
This project will hopefully provide some help in transitioning from traditional methods to new and proven methods using machine learning.The data analysis methods present today are much more powerful than earlier methods.

Prediction of Biological Activity:
The project's primary goal is to predict the biological activity of molecules, specifically their mechanism of action (MoA). MoA is a fundamental aspect of drug development.

Literature Review

Paper 1: Classification of Drugs Using Machine Learning

- This study machine learning models and their performance on the Mechanism of Action (MoA) dataset .
- Models evaluated: BRkNN Type A, BRkNN Type B, ML-KNN, and a custom Neural Network using Keras.
- Log loss results obtained from the models were : <model(log-loss)> : BRkNN-a (0.11), BRkNN-b (0.28), ML-KNN (0.11), and custom neural network (0.017).
- Describes integration of the custom neural network into a web application using Flask.

Literature Review

Paper 2: MoA Prediction in Kaggle Competition

- Aim was to propose multi-label classification machine learning algorithms for predicting MoA.
- Data exploration, feature engineering (PCA, feature augmentation), and cross-validation were used.
- Log loss results obtained from the models were : <model(log-loss)> : Neural Network (log loss 0.0159), TabNet (log loss 0.0150), and ResNet (log loss 0.0147).
- The study highlights the need for further model exploration and data preprocessing.

Literature Review

Paper 3: Large-Scale Comparison of Machine Learning Methods

- Compares deep learning approaches to other machine learning methods in drug target prediction.
- Addresses challenges in evaluating deep learning in drug discovery.
- Utilizes a nested cluster-cross-validation strategy.
- Finds deep learning methods (FNN) outperforming SVM, RF, KNN, NB, SEA, GC, Weave, and Smiles LSTM.
- Provides valuable insights for drug target prediction.

Dataset Description

- **Source of the data**

  The Laboratory for Innovation Science at Harvard. Available through a Kaggle competition.

- **About the data**

  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.

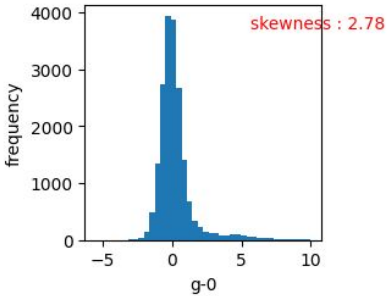
- **Type of Problem**

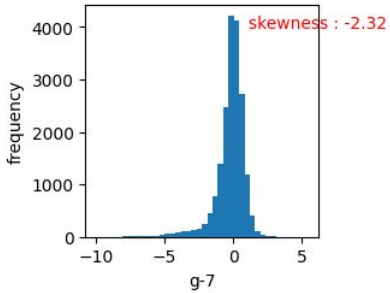
  Drugs can belong to multiple MoA categories, making this task a multi-label classification problem.

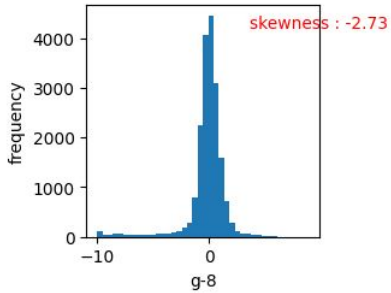
Dataset Description

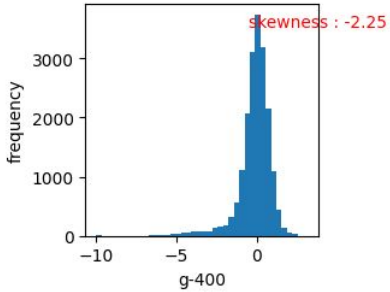
- Training data is divided into four files:
  - train_features.csv
  - train_drug.csv
  - train_targets_scored.csv
  - train_targets_nonscored.csv
- Testing features for prediction is given in test_features.csv.
- **train_features.csv** - Features for the training set. Features g- signify gene expression data, and c- signify cell viability data. cp_type indicates samples treated with a compound (cp_vehicle) or with a control perturbation (ctrl_vehicle)
- **train_drug.csv** - This file contains an anonymous drug_id for the training set only.
- **train_targets_scored.csv** - The binary MoA targets that are scored.
- **train_targets_nonscored.csv** - Additional (optional) binary MoA responses for the training data. These are not predicted nor scored.
- **test_features.csv** - Features for the test data. One must predict the probability of each scored MoA for each row in the test data.

Dataset Description – Visualization









Most of the features are highly skewed.

Data Pre-processing

Is data preprocessing required?

For our dataset, yes preprocessing is required.

1. One hot encoding will be needed for discrete features

   Discrete attributes : cp_type, cp_dose, cp_time

2. As the number of features is a lot, the model will become complex, hence we there is a need to apply PCA to reduce the number of features.
3. We don't want to allow a single feature to be dominant, hence standardization is required too.

Data Pre-processing

- There are a total of 23,814 rows in the dataset and 875 features. These features serve the purpose of making predictions and conducting analyses.
- Out of the 875 features, three of them are discrete attributes: 'cptime,' 'cpdose,' and 'cptype.' To address these discrete attributes, we have employed one-hot encoding.
- Feature scaling, standardization has been implemented to ensure uniformity.
- Preprocessing algorithms such as PCA and LOF were used and finally cross validation was on each model for finding the best possible parameters.
- Optimal PCA values are mentioned with respective models.
- While applying models, we realized that removing outliers is giving no benefit.

Evaluation Metric

For every `sig_id` you will be predicting the probability that the sample had a positive response for each `<MoA>` target. For N `sig_id` rows and M `<MoA>` targets, you will be making $N \times M$ predictions. Submissions are scored by the log loss:

$$\text{score} = -\frac{1}{M}\sum_{m=1}^{M}\frac{1}{N}\sum_{i=1}^{N}\left[y_{i,m}\log(\hat{y}_{i,m}) + (1-y_{i,m})\log(1-\hat{y}_{i,m})\right]$$

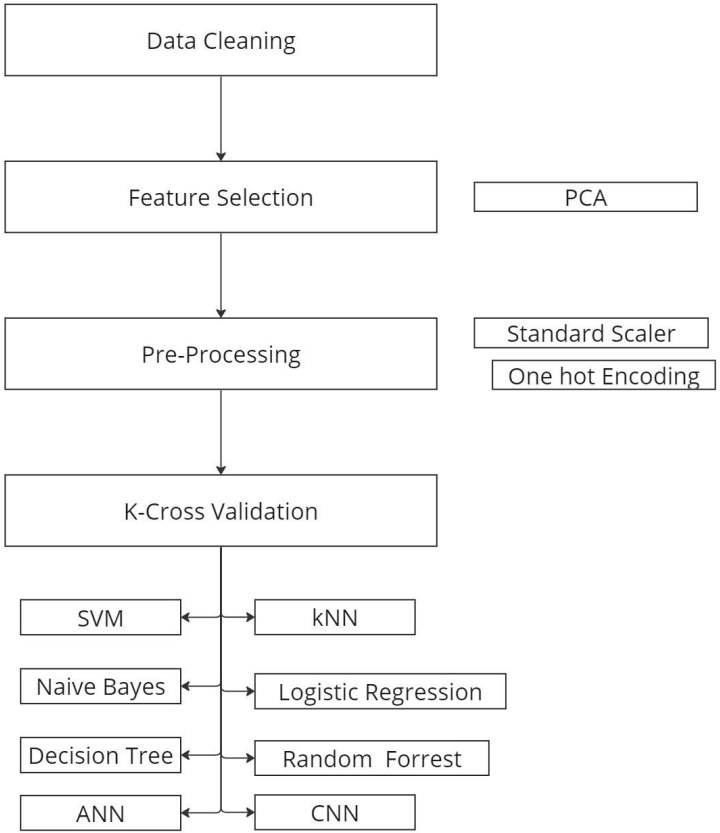
where:

- N is the number of `sig_id` observations in the test data ($i = 1, \ldots, N$)
- M is the number of scored MoA targets ($m = 1, \ldots, M$)
- $\hat{y}_{i,m}$ is the predicted probability of a positive `MoA` response for a `sig_id`
- $y_{i,m}$ is the ground truth, 1 for a positive response, 0 otherwise
- $log()$ is the natural (base e) logarithm

Note: the actual submitted predicted probabilities are replaced with $max(min(p, 1-10^{-15}), 10^{-15})$. A smaller log loss is better.

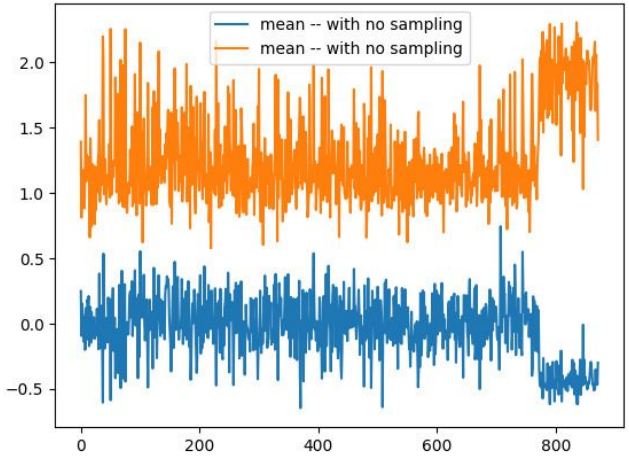
(Screenshot From Kaggle)

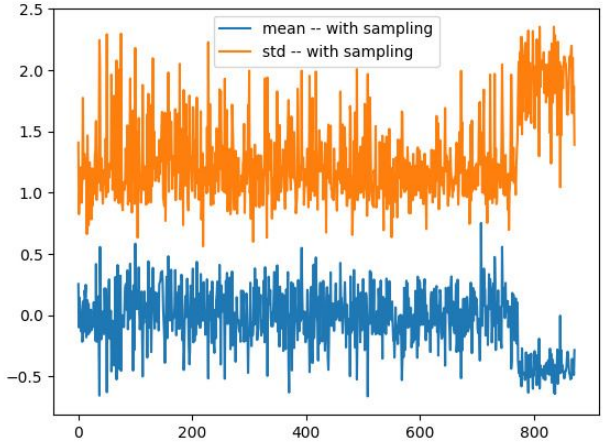
Methodology – Flowchart



Sampling

- Training data has 23000 rows.
- Computationally expensive.
- 5000 data points has been sampled.
- Mean and standard deviation has been plotted, before and after sampling. It shows the quality of sampling.





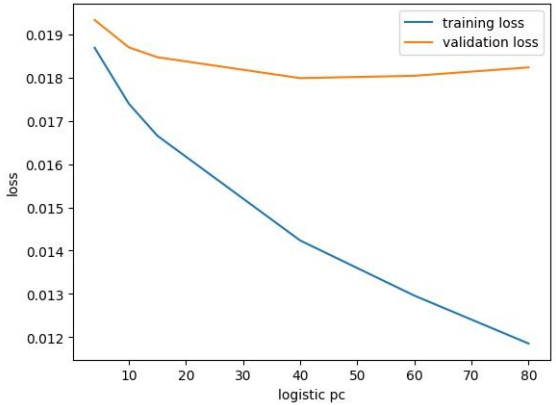
Methodology – Naive Bayes

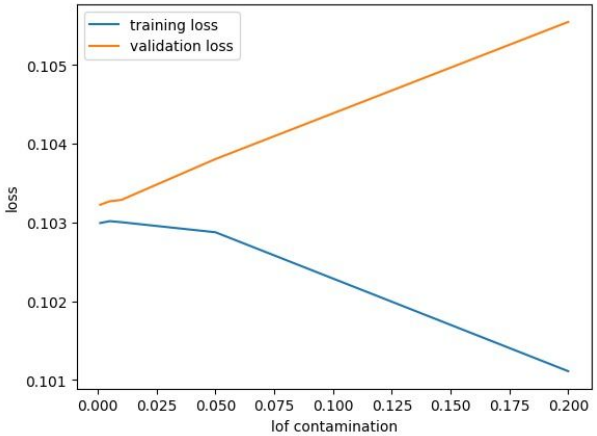
- Assumes that the the features are independent and follow a gaussian distribution
- We found out that our distribution was skewed to either left or right, to remove this skewness correction was performed using Box-Cox transformation
- Skewness removal did not improve model performance.
- Principal Component Analysis method used for dimensionality reduction.
- Optimal parameters for PCA was detected through an extensive k-cross validation.

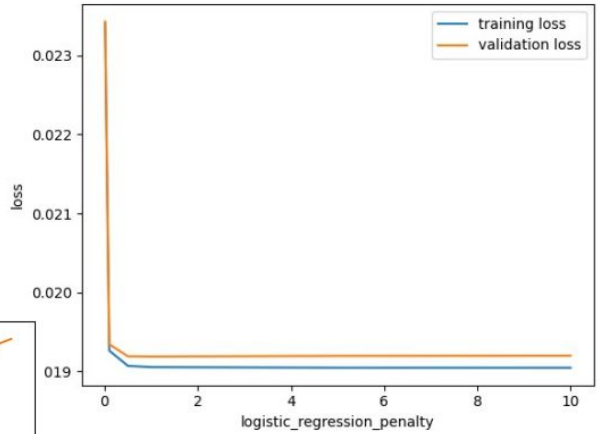
Methodology – Logistic Regression

- Individual Classifiers created for each label.
- Local Outlier Factor method was used for outlier detection.
- Principal Component Analysis method used for dimensionality reduction.
- Optimal parameters for regularization (Lasso), PCA and LOF were detected through an extensive k-cross validation.
- Optimal PCA value: 20.
- Optimal Lasso penalty value = 0.5
- On evaluation it was found that LOF was not causing any significant, improvement in the model performance. Therefore the decision was made to remove it.

Methodology – Logistic Regression

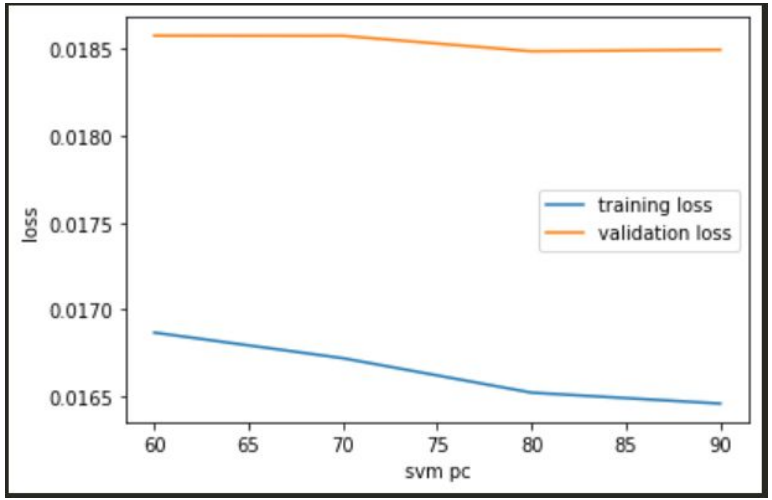






Methodology – SVM

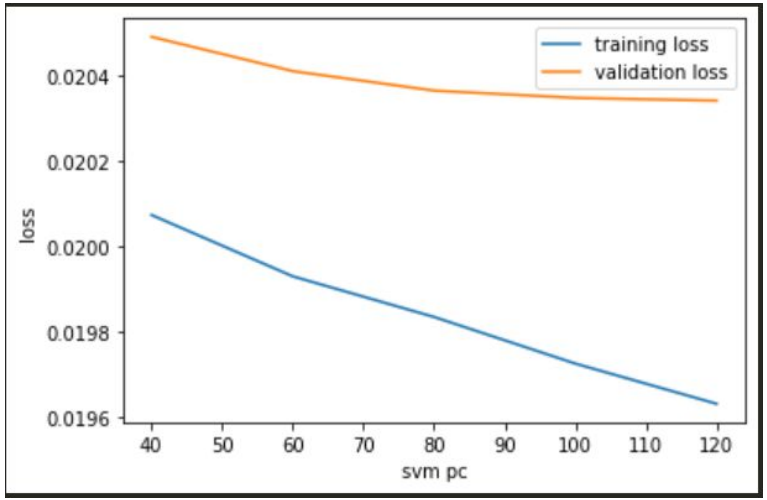
- Poly Kernel has been used.
- Optimal parameters for PCA = 80
- Optimal PCA was chosen on the basis of this graph




```
Test Loss : 0.01898077930151102
Training Loss : 0.016454399064474323
```

Methodology – SVM

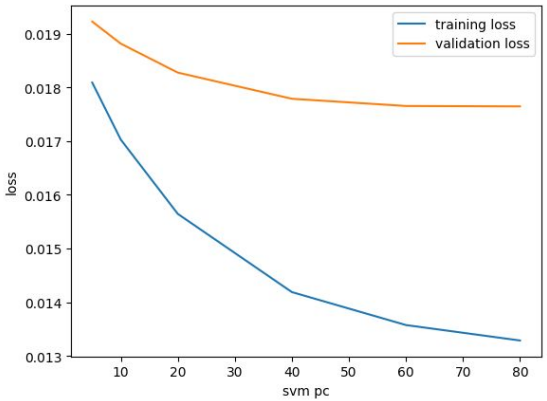
- Sigmoid Kernel has been used.
- Optimal parameters for PCA =80
- Optimal PCA was chosen on the basis of the given curves

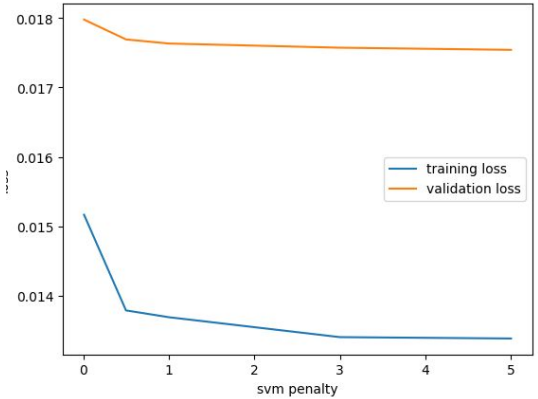



```
Test Loss : 0.020609936651957136
Training Loss : 0.019853900801617176
```

Methodology – SVM

- RBF Kernel has been used.
- Optimal parameters for PCA and penalty was chosen on the basis of given curves.



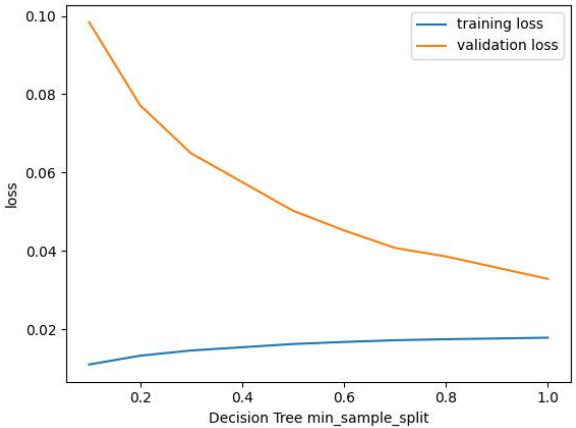


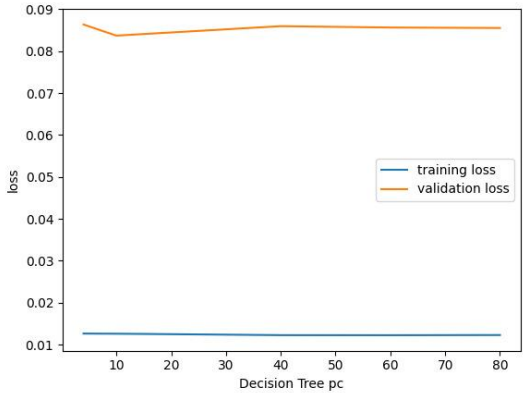
Training Loss:
0.013456775966529052

Testing Loss:
0.01783931424749255

Methodology – Decision Tree

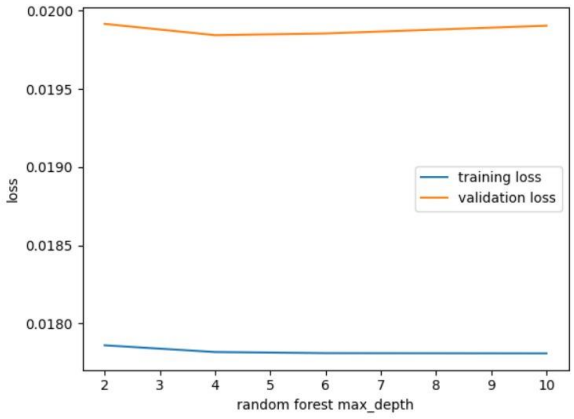
- Optimal parameters for the model were discerned through k-fold cross-validation.
- However, upon subjecting the model to testing, it became evident that the decision tree struggled to yield satisfactory performance on the testing dataset.

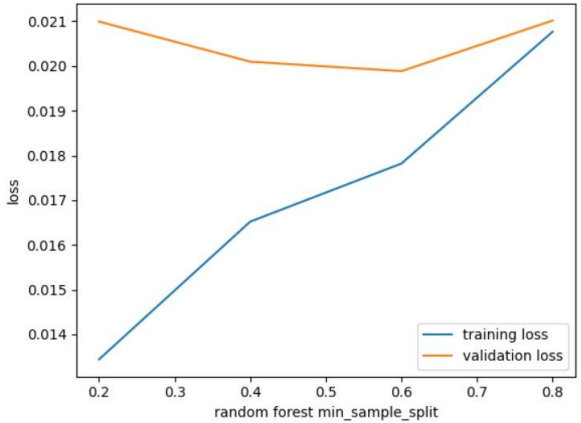


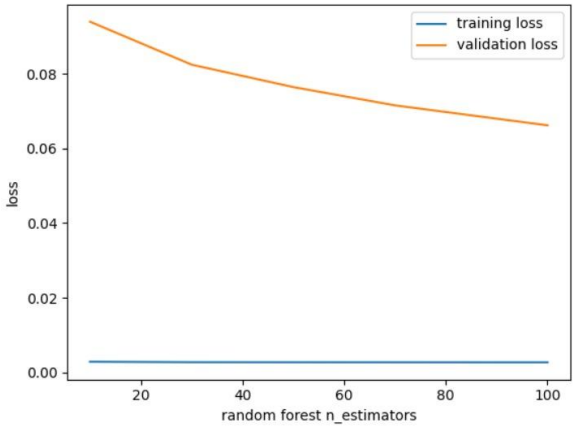


Methodology – Random Forest

- Random Forest stands out as an ensemble learning technique designed to enhance predictive accuracy and mitigate overfitting issues often associated with individual decision trees.



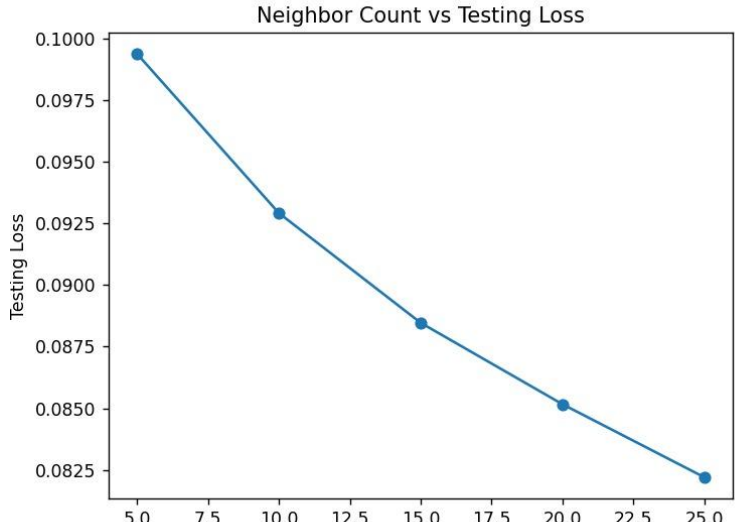


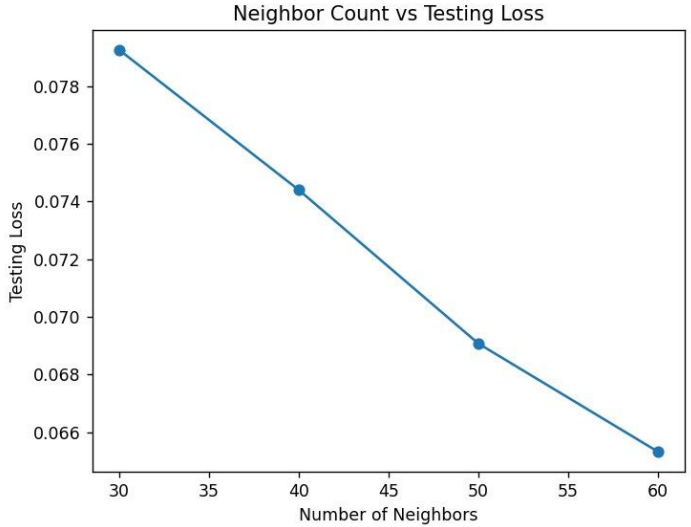


Methodology – KNN

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.

For PCA = 30





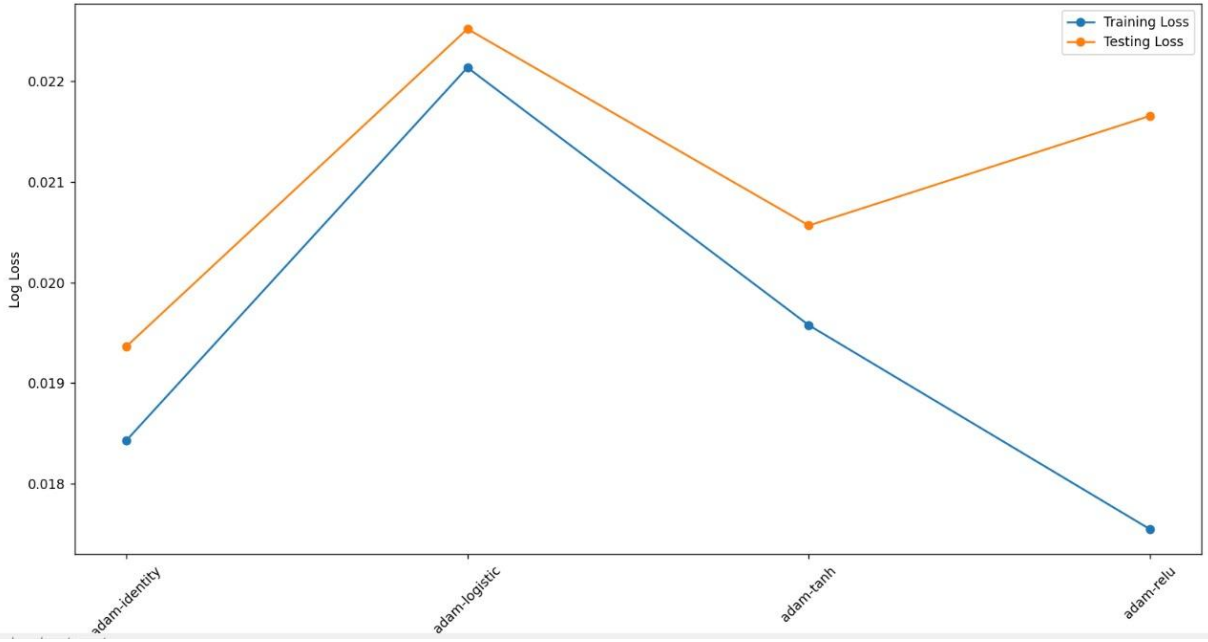
Methodology – KNN

For PCA = 30 , further increasing neighbours

```
Processing columns with 100 neighbors: 100%|
Training loss with 100 neighbors: 0.0151655785785431628
Testing loss with 100 neighbors: 0.055781618734113615
Processing columns with 200 neighbors: 100%|
Training loss with 200 neighbors: 0.016547287423825798
Testing loss with 200 neighbors: 0.042445795312423 16
Processing columns with 300 neighbors: 100%|
Training loss with 300 neighbors: 0.0172680889260712 62
Testing loss with 300 neighbors: 0.035630836614364 01
Processing columns with 400 neighbors: 100%|
Training loss with 400 neighbors: 0.0177191743507518 55
Testing loss with 400 neighbors: 0.0311470138332027 48
```

Methodology – 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.



Best hyperparameter for PCA = 10:
Solver = Adam
Activation Loss: Identity
Hidden Layer:
(3, 3, 3, 3)
Iterations = 10K

```
Testing solver: adam, activation: identity
Processing columns: 100%|
Training loss : 0.018427575241924947
Testing loss : 0.0193611342120623
Testing solver: adam, activation: logistic
Processing columns: 100%|
Training loss : 0.022133686722382254
Testing loss : 0.022516801245881478
Testing solver: adam, activation: tanh
Processing columns: 100%|
Training loss : 0.01957519956495432
Testing loss : 0.020565508829363914
Testing solver: adam, activation: relu
Processing columns: 100%|
Training loss : 0.017548207823863026
Testing loss : 0.021654468817714562
```

Methodology – ANN

- For PCA = 30
  Solver = Adam,
  Activation Loss: Identity,
  Hidden Layer: (3, 3, 3, 3),
  Iterations = 10K

  Training loss : 0.0155901942833462 94
  Testing loss : 0.017467805442214736

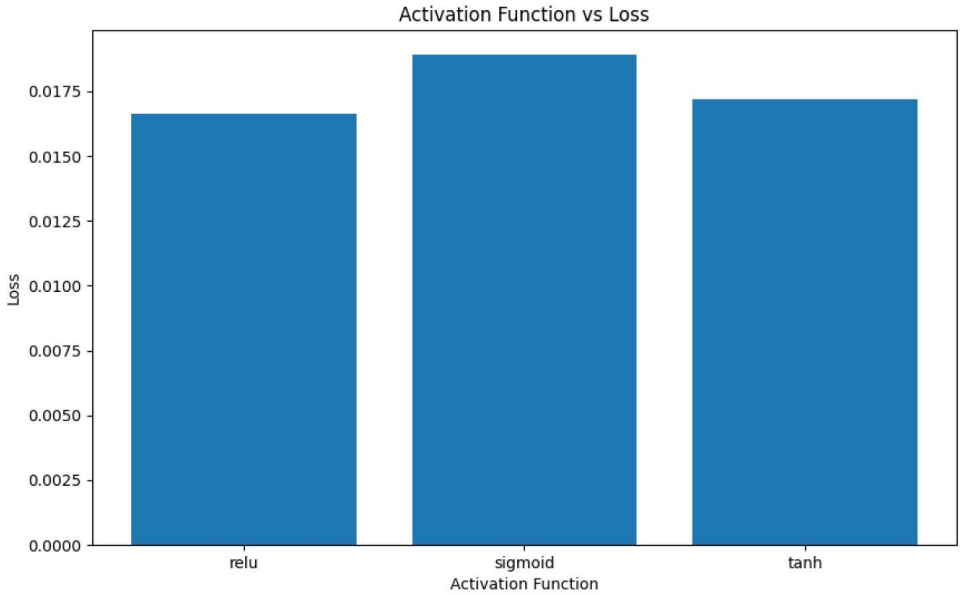
Methodology – CNN

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.

For PCA = 50
Activation: relu
loss: 0.0166413187980651 86

Activation: sigmoid
loss: 0.018917141482234

Activation: tanh
loss: 0.017188021913170815



Result and Analysis

- **Comparison of Models on the basis of their Training and Testing Loss**

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

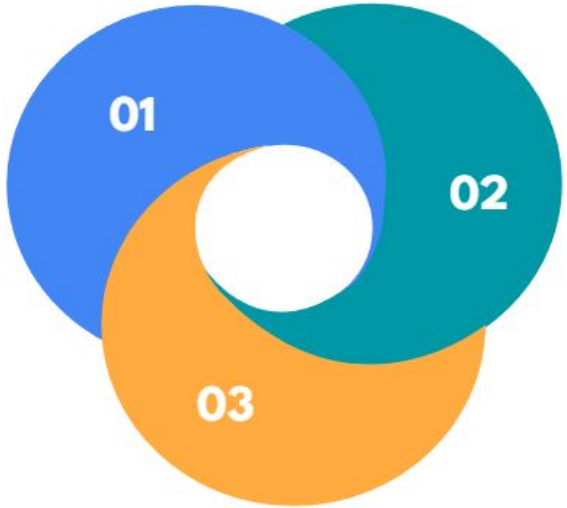
Conclusion

- Results of Naive Bayes, Decision Tree, KNN are poor (baseline models)
- High variance for Decision Tree, overfitting.
- Random Forest gives better results than decision tree because of ensembling.
- Logistic regression, SVM, ANN are performing significantly good.
- CNN is giving best performance on the unseen data.

TIMELINE

The proposed timeline is being followed and the concepts which have been covered in class after the mid semester presentation like Decision Tree, Random Forest, KNN, ANN and CNN were applied.

Future Scope: We will implement new concepts by studying on our own. We will apply other deep learning models and further improve the results.



Challenges Faced:

The major challenge we faced was the size of the dataset, running each the code each time required a lot of time.

CONTRIBUTIONS

The tasks of motivation, literature review, and dataset description, hyperparameter tuning (respective ,models) and understanding of results were collectively completed by our team.

- Rishav Raj: SVM, Logistic Regression, Data Visualization, Decision Tree, Random Forest
- Parth Kaushal : Naive Bayes, SVM, CNN, ANN
- Shubham Pal : Logistic Regression, SVM, Random Forest,KNN
- Garv Makkar : SVM, Data Visualization,ANN,KNN