AI Project Report

Classification of Mice Protein Expression Data Using Machine Learning

CSE422 – Artificial Intelligence

Department of Computer Science and Engineering
BRAC University

Submitted by:

Israt Jahan Hira (ID:24241155, Department of CS)
Nusrat Jahan Snigdha (ID: 22101812, Department of CSE)

Submitted to:

Asif Hasan [AHC]
Asif Shahriar [SHAH]

Table of Contents

1. Introduction

In this project, we aim to classify mice samples based on their protein expression profiles. The dataset was provided by the faculty as part of our course project. Protein expression profiling is a crucial task in biological and medical research, particularly for understanding diseases such as Down syndrome. The classification task aims to determine the class of each sample (combination of genotype, treatment, and behavior) based on the measured expression of 77 proteins. The motivation behind this project is to apply AI models to identify meaningful patterns and predict classes based on protein expression features in mice.

2. Dataset Description

General Overview

- Dataset: Mice Protein Expression Data
- Total Samples: 1,080 rows
- Total Features: 82 (77 proteins + 5 categorical/ID columns)
- Output: Class (combination of Genotype, Treatment, and Behavior)
- Problem Type: Classification. Because the goal is to predict categorical labels (classes) based on numeric inputs.

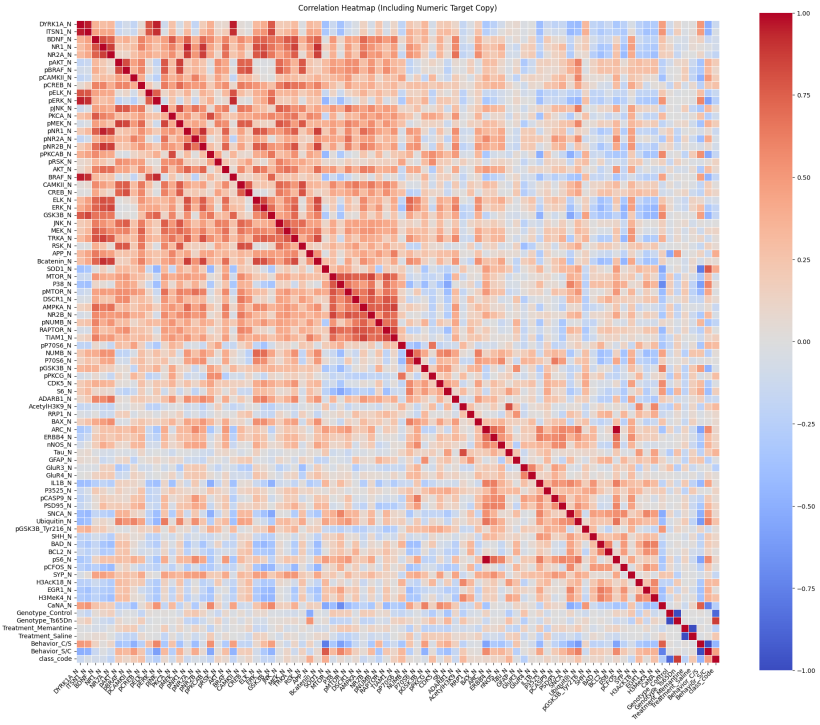
Feature Types

- Quantitative: 77 protein expression values
- Categorical: Genotype, Treatment, Behavior, MouseID, and Class

The target column is “class”, which consists of 8 unique combinations (e.g., c-CS-m, c-SC-s, etc.) representing different experimental conditions.

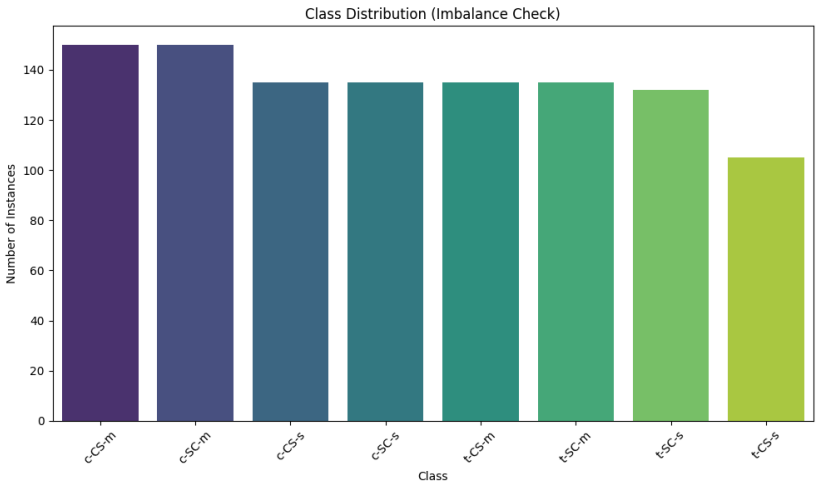
Correlation Analysis

Using a Seaborn heatmap, we visualized correlations among the 77 protein features. While some proteins were moderately correlated, there was no extreme multicollinearity. This suggests each protein provides some unique contribution.



From the heatmap, we noticed some small clusters of proteins that moved together across samples, possibly reflecting co-expression under similar biological regulation.

Class Distribution

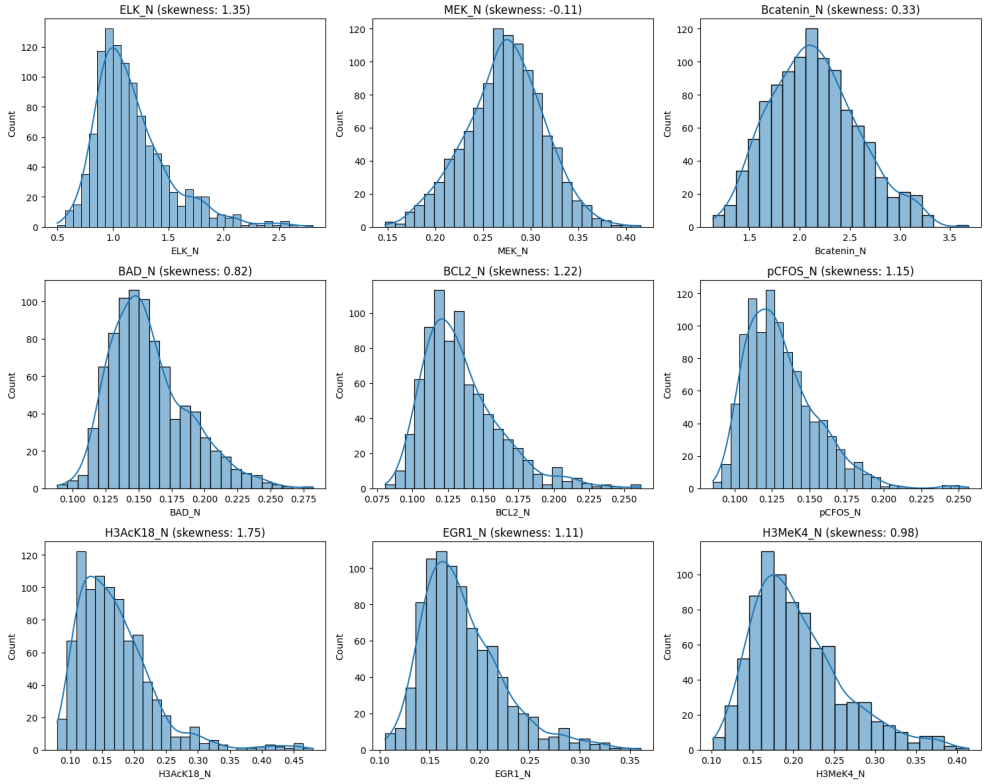


The bar chart of the target class labels revealed the dataset is imbalanced. Some classes like “c-CS-s” were overrepresented compared to others like “t-CS-s”. This imbalance could potentially bias model performance.

Exploratory Data Analysis (EDA)

We conducted a detailed EDA to understand the underlying structure and patterns in the protein expression data:

- We reviewed the dataset using head (), info (), and describe ().
- We detected missing values, duplicates, inconsistent data types.
- We interpreted skew in numerical features using histograms with KDE curves and box plot.



These insights guided our preprocessing and modeling strategy.

3. Dataset Preprocessing

- Null Values: The column 'DYRK1A_N' had missing values, and rows with missing values were dropped. Other columns such as 'MEK_N' and 'Bcatenin_N' were imputed using the mean, while 'ELK_N', 'BAD_N', 'BCL2_N', and others were filled using the median.
- Categorical Encoding: 'Genotype', 'Treatment', and 'Behavior' were categorical features and were one-hot encoded using pandas' get_dummies function.
- Unnecessary feature: We dropped MouseID as it is unnecessary for ml training.

- Scaling: Features were standardized using StandardScaler to ensure uniform feature range, crucial for KNN, Logistic Regression, and Neural Networks.

4. Dataset Splitting

The dataset was split into 70% training and 30% testing sets using stratified sampling. Stratification ensured that the class distribution in both sets was representative of the full dataset.

- Training samples: 756
- Testing samples: 324

5. Model Training & Testing

The following models were trained on the dataset:

• K-Nearest Neighbors (KNN)
• Decision Tree
• Logistic Regression
• Neural Network (implemented using TensorFlow/Keras with two hidden layers, Relu activation in the hidden layer, dropout (30%) for regularization, and softmax activation in the output layer)

Each model was trained on the standardized training data and tested on the holdout test set. Model predictions were evaluated using accuracy, precision, recall, and F1-score.

6. Model Comparison

In this project, we trained four different classifiers—K-Nearest Neighbors (KNN), Decision Tree, Logistic Regression, and a Neural Network — to classify mice samples based on their protein expression features. Below is a summary of their performance on the test set:

Model	Accuracy	Precision	Recall	F1 Score
K-Nearest Neighbors	0.978	0.979	0.978	0.978
Decision Tree	1.000	1.000	1.000	1.000
Logistic Regression	1.000	1.000	1.000	1.000
Neural Network	1.000	1.000	1.000	1.000

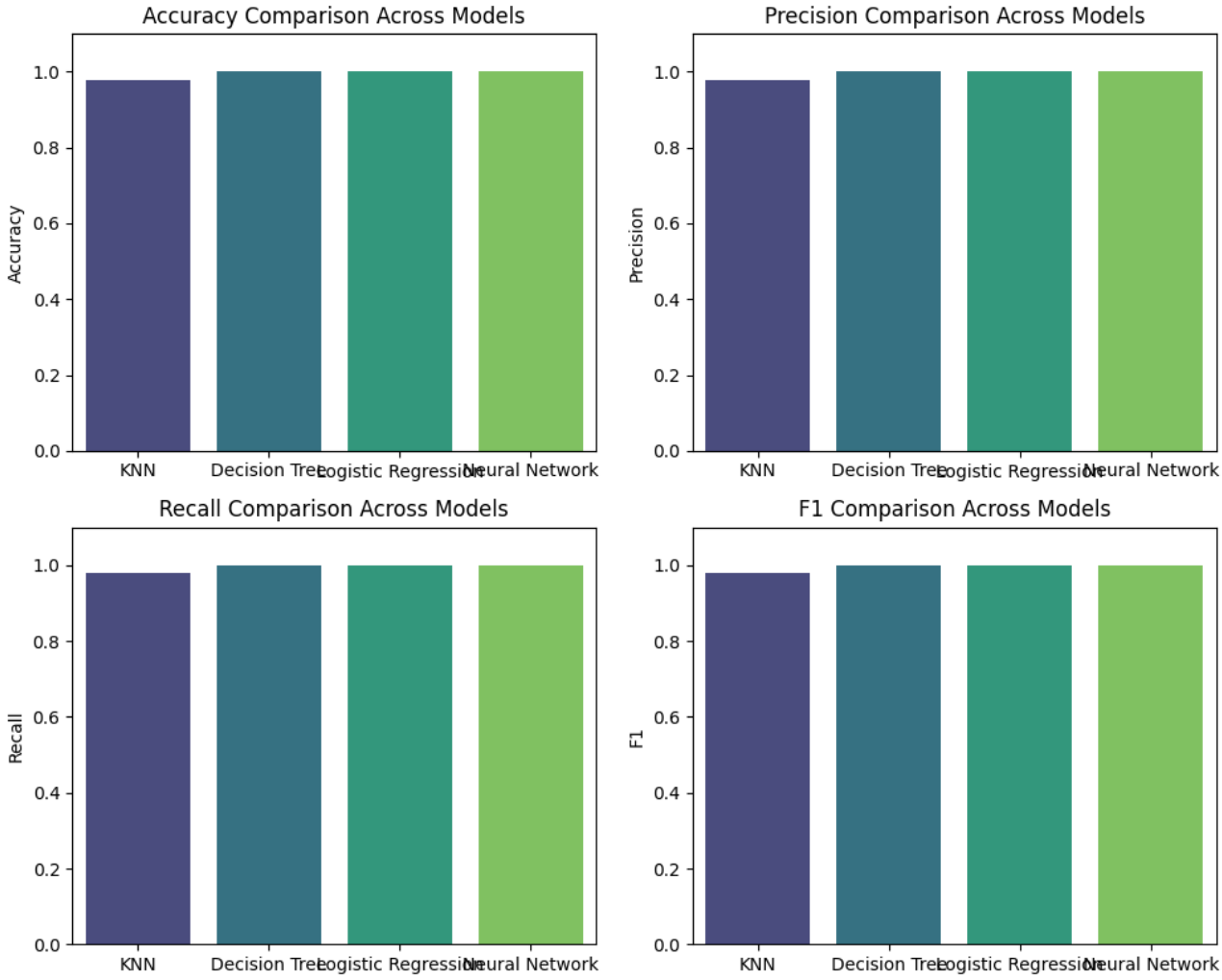
The performance metrics indicate that three of the models (Decision Tree, Logistic Regression, and Neural Network) achieved perfect classification scores on the test set. KNN also performed extremely well, with nearly perfect metrics.

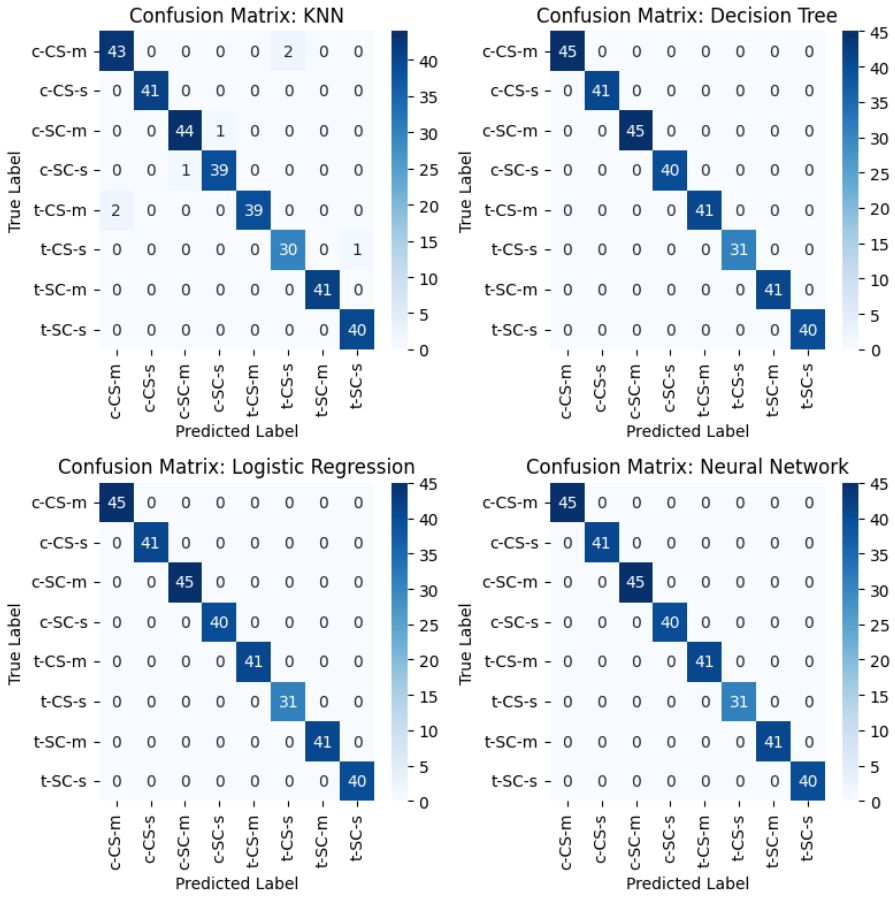
However, such flawless results—especially for Decision Tree and Neural Network—raise a potential red flag for overfitting. This is particularly relevant given the structure of the dataset:

- Total samples: 1077
- Total features: 82 (relatively high feature-to-sample ratio)
- Multiclass classification (8 classes with imbalance)

Therefore, the perfect results are likely an indication of overfitting.

Performance of the models was compared based on classification metrics. A bar chart was used to visualize and compare accuracy, precision, recall, and F1 score. Confusion matrices were plotted for each model using seaborn heatmaps.





The Neural Network outperformed the classical models in most metrics, likely due to its capacity to model complex non-linear relationships among features. KNN and Logistic Regression showed decent performance but were slightly inferior. Decision Tree performed adequately but showed signs of overfitting.

7. Conclusion

Among the models evaluated, Logistic Regression, Neural Network, and Decision Tree all achieved perfect accuracy and F1 scores on the test set, while KNN was slightly below but still excellent. This suggests that the Mice Protein Expression dataset is highly discriminative with clear patterns between the classes.

Challenges faced included dealing with missing values and class imbalance, both of which were addressed through preprocessing and stratified splitting.

However, due to the high dimensionality of the data (82 features vs. 1077 samples) and the perfect results, we suspect some level of overfitting, especially for models like

Decision Tree and Neural Network which are prone to it. KNN's strong yet slightly lower performance appears more realistic and generalizable.

Key Takeaways:

- All models performed exceptionally well, especially after proper preprocessing.
- Perfect results likely signal overfitting or data leakage and should be validated with cross-validation.
- KNN provides a good balance between performance and robustness.

To further confirm model reliability, we recommend running cross-validation and rechecking data leakage. Additionally, exploring feature importance could help reduce dimensionality and improve generalization. This project gave us hands-on experience with a real-world biological dataset and reinforced how data preparation and model selection critically impact performance.