# CLASSIFICATION OF MICE BASED ON PROTEIN EXPRESSION LEVELS

# Overview

This project focuses on analyzing protein expression levels in the cerebral cortex of mice to classify them based on genotype, behavior, and treatment. The aim is to identify subsets of proteins that can discriminate between these classes and to understand the biological mechanisms underlying learning and memory, particularly in the context of Down syndrome.

# Objectives

The objective of this project is to analyze protein expression levels in the cerebral cortex of mice to classify them into distinct categories based on their genotype, behavior, and treatment. The project specifically aims to identify subsets of proteins that can distinguish these categories and to elucidate the biological mechanisms underlying learning and memory, with a particular focus on Down syndrome.

# Methodologies

**Data source**: Real-time data set was provided focusing on protein expression level and impact of genotype, behavior and treatment on mice.

**Tools and Technologies**

- Programming Language: Python

Libraries:

- Pandas: For data manipulation and analysis.
- NumPy: For numerical computations.
- Scikit-learn: For machine learning algorithms and model evaluation.
- Matplotlib and Seaborn: For data visualization.
- IDE: Jupyter Notebook or any other Python IDE

# Introduction

**Background**

Down Syndrome (DS) is a genetic disorder caused by an extra copy of chromosome 21, leading to intellectual disabilities and developmental delays. Associative learning, crucial for adapting to the environment, is often impaired in individuals with DS. Protein expression studies in mouse models (Ts65Dn mice) help uncover the molecular mechanisms underlying these cognitive deficits. These studies measure protein levels in the brain, providing insights into the biological processes affected by DS. Using mice models, which share genetic and biological similarities with humans, researchers can identify key proteins and pathways that are altered in DS. This knowledge aids in understanding the condition and evaluating potential treatments, such as memantine, to improve cognitive function and quality of life for those with DS.

# Dataset Description

**Dataset Characteristics**

The dataset contains protein expression levels measured in the nuclear fraction of the cerebral cortex in mice. It includes both control and trisomic (Down syndrome) mice subjected to a context fear conditioning task.

- Type: Multivariate
- Subject Area: Biology
- Associated Tasks: Classification, Clustering
- Feature Type: Real
- Instances: 1080
- Features: 80

# Data Preprocessing

Data preprocessing is the process of transforming raw data into a useful, understandable format, resolving issues like inconsistent formatting, human errors, and incompleteness. It's crucial for data mining and machine learning projects, affecting the success and performance of machine learning models by making data completer and more efficient for analysis.

- Handling Missing Values
- Normalization/Scaling
- Encoding Categorical Variables

# Exploratory Data Analysis (EDA)

EDA is used to discover patterns, spot anomalies, test hypotheses, and check assumptions with the help of summary statistics and graphical representations.

- **Summary statistics:**

Libraries like pandas and numpy were used to calculate and examine mean, median, standard deviation, etc. for each protein to understand the distribution and variability of the data.

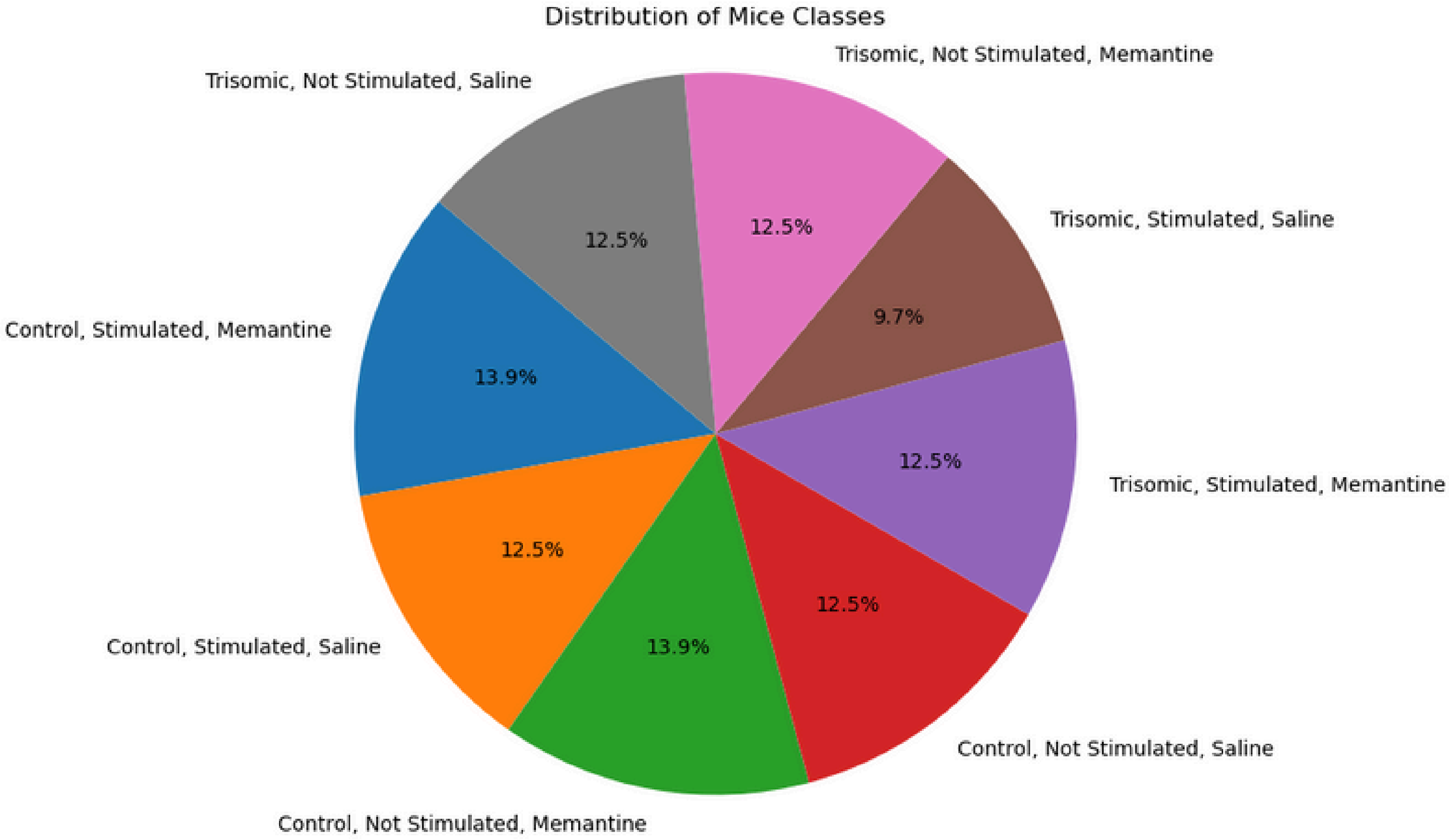
- **Visualizations of data distribution:**

Libraries like matplotlib and seaborn were used to visualize the data in the form of histograms (for numerical columns), pie charts (distribution of mice classes), bar plot (for categorical variables & distribution of target variables), heatmap (protein expression level by classes & for correlation matrix of protein expression), box plot (protein expression levels & for outlier detection & for numerical variables by target variables). These visualizations helped in providing valuable insights and identifying any patterns or anomalies in the data.

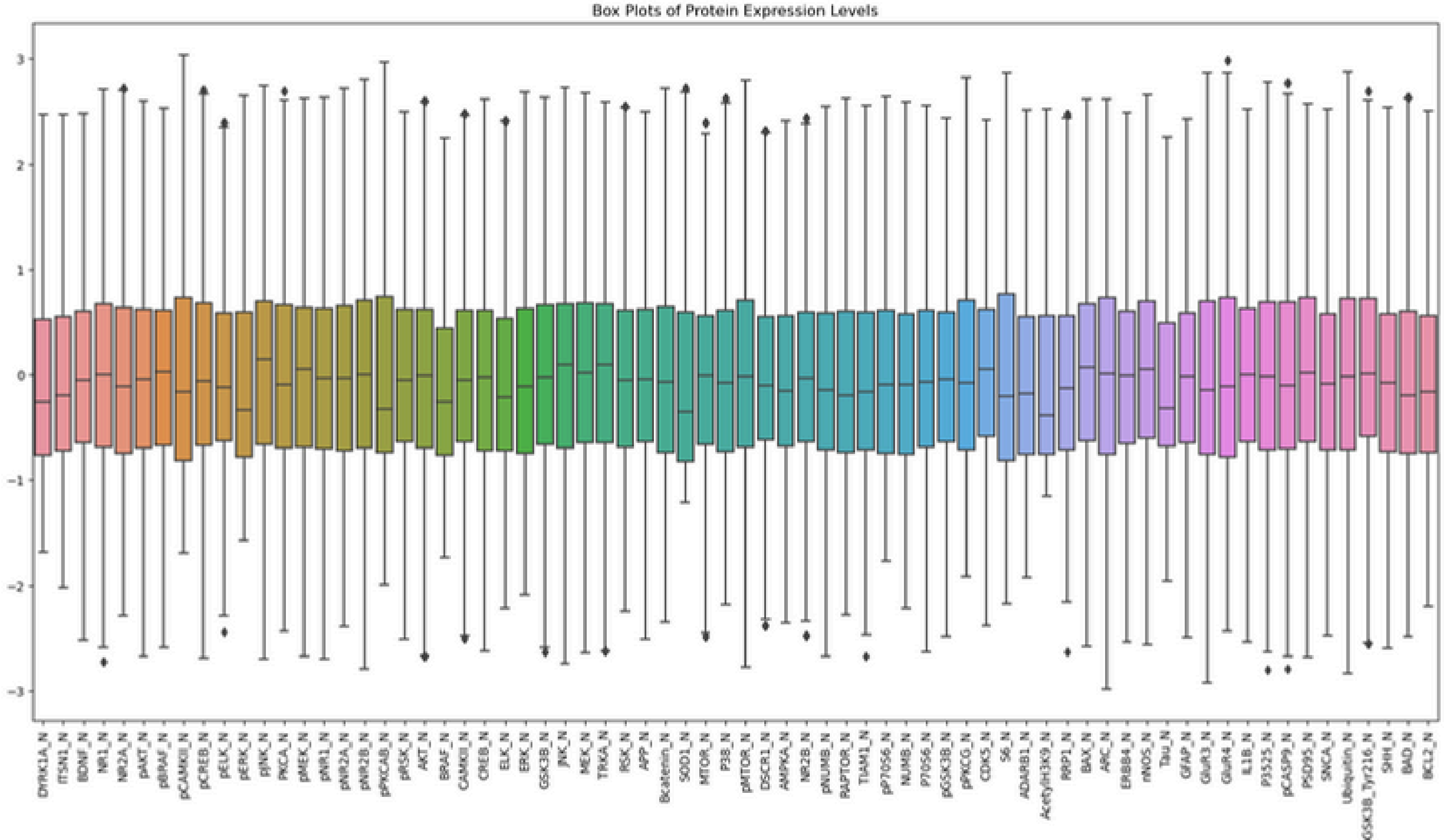
- **Correlation analysis:**

Libraries like panda, matplotlib and seaborn were used to analyze the correlations between different proteins to understand their relationships. Correlations were visually represented in the form of heatmap

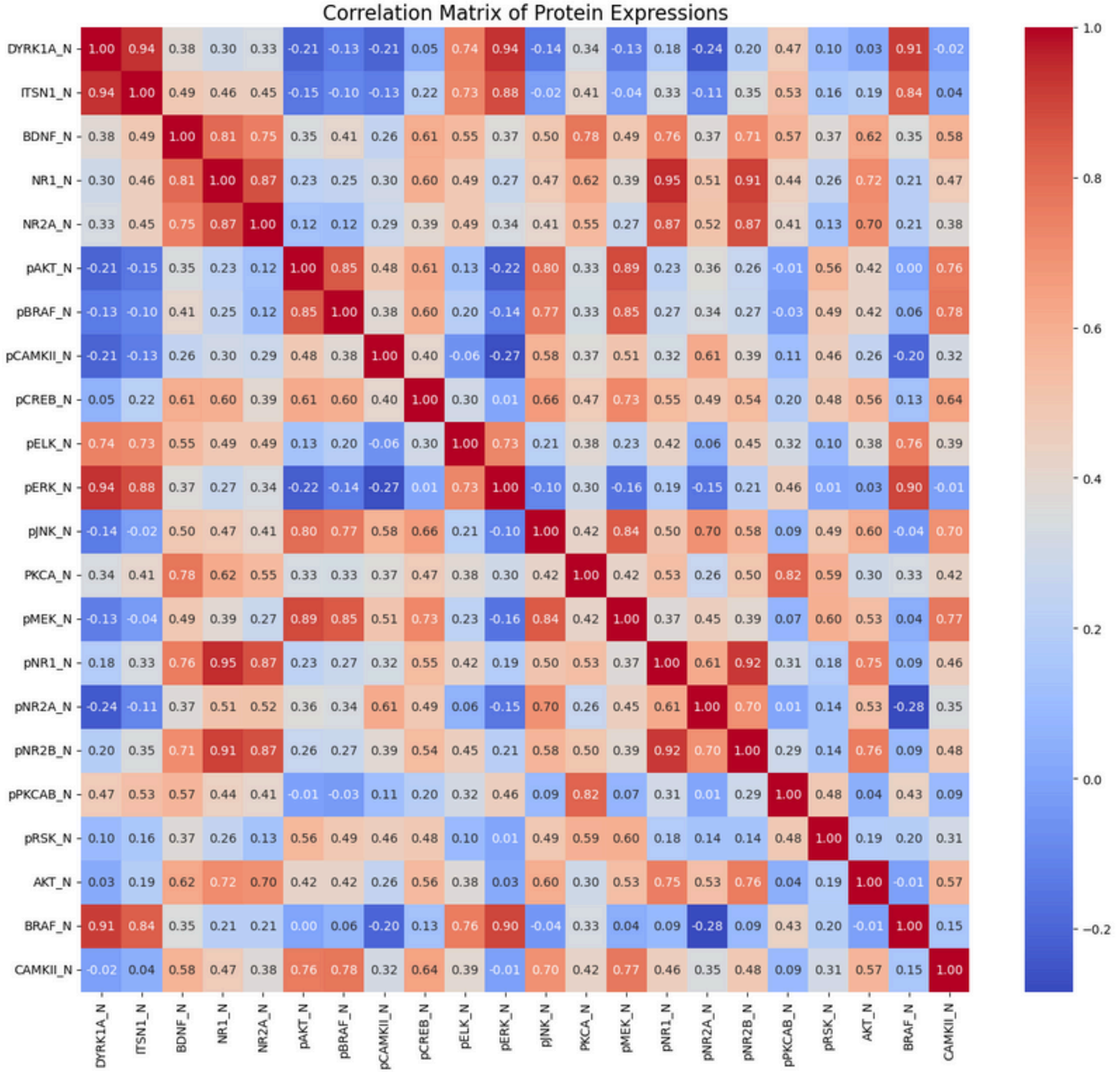
# Piechart of  Distribution of Mice Classes



# Box Plots of Protein Expression Levels



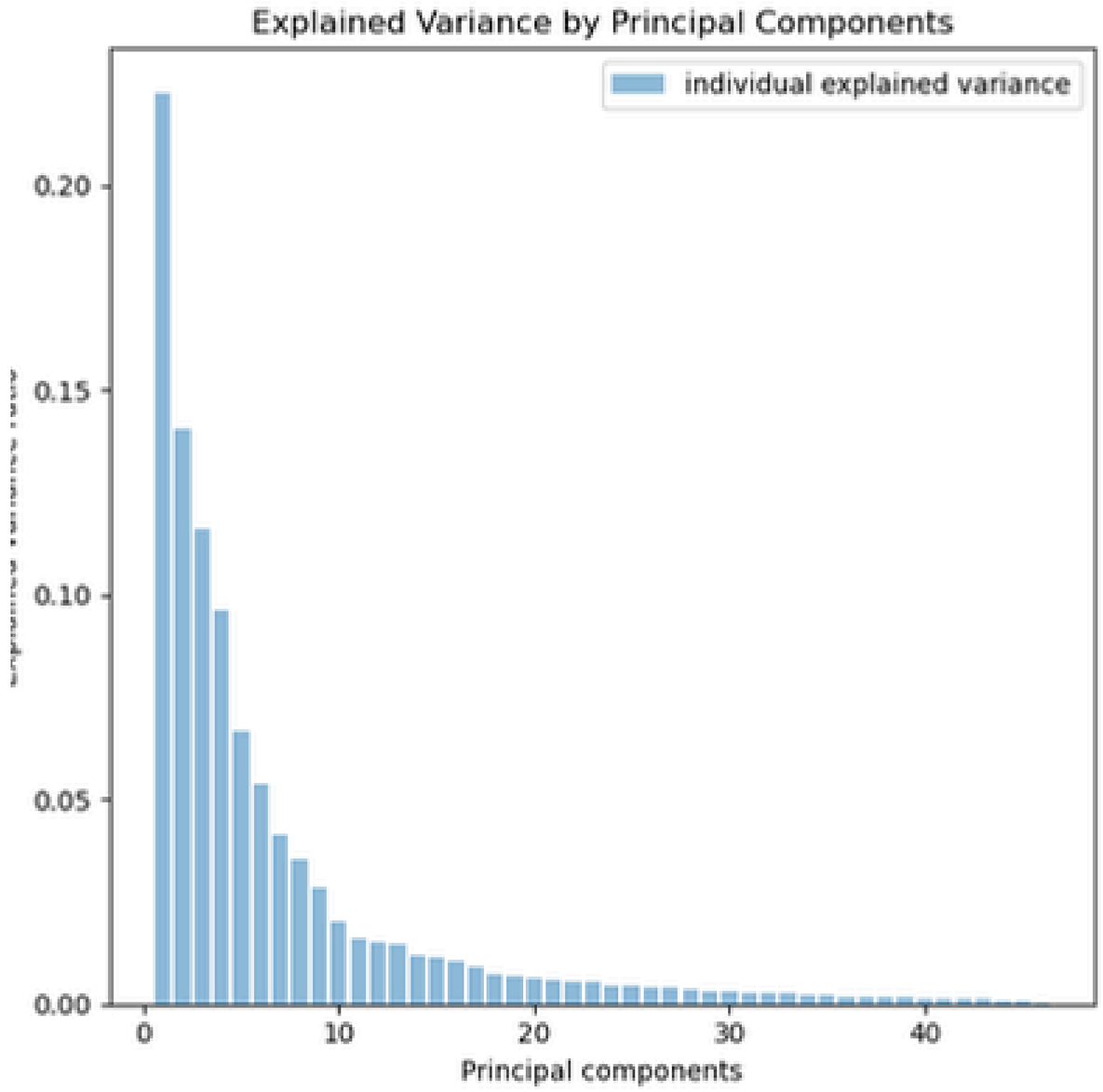
# Correlation Matrix of Protein Expressions

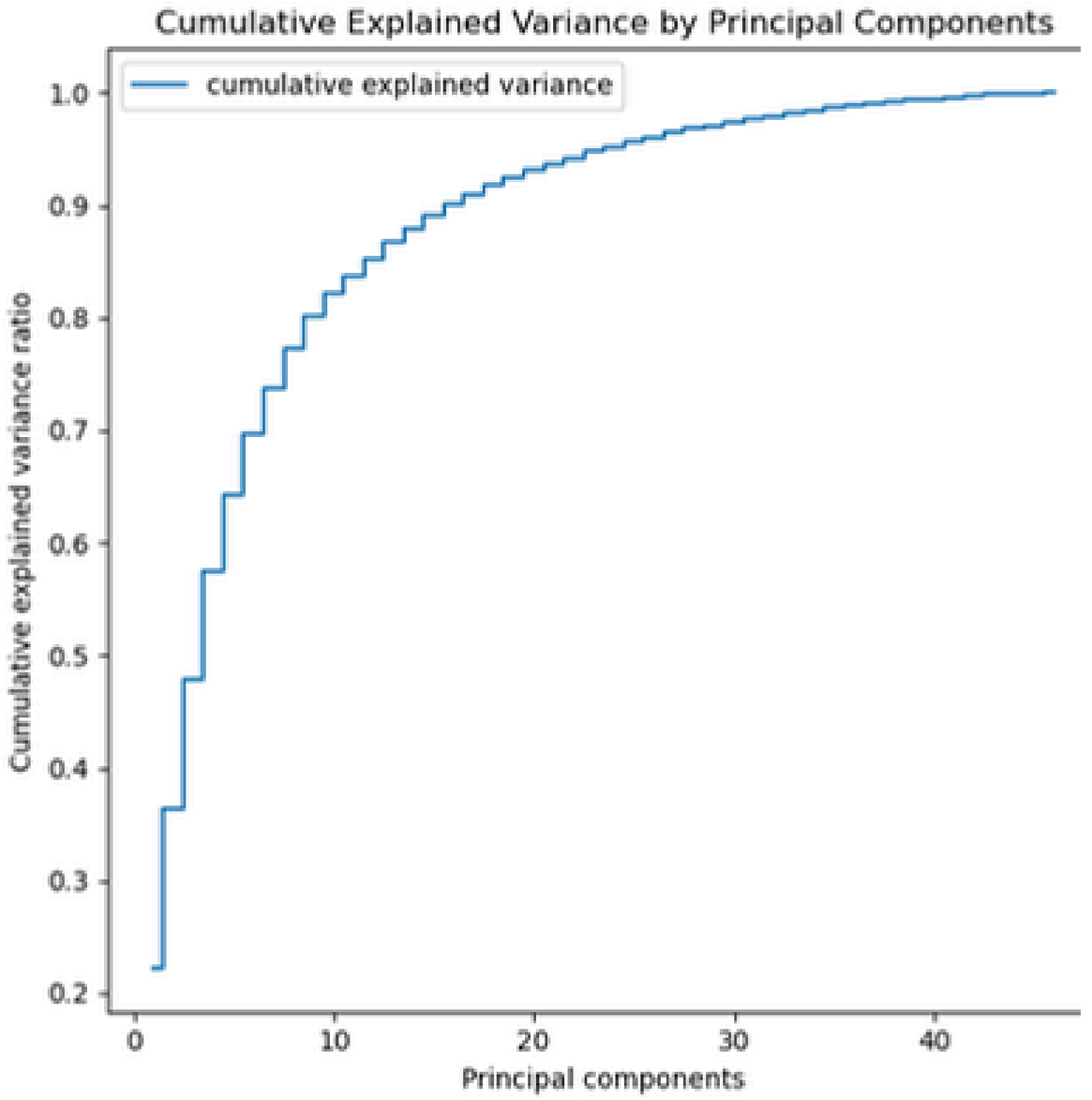


# Feature Selection

- **Techniques used (correlation analysis, mutual information, feature importance from models):** Evaluating and selecting the most relevant features for our machine learning model is crucial for building effective models.
- **Pearson Correlation Coefficient:** Measures the linear relationship between two variables. Values range from -1 to 1, where 1 indicates a strong positive correlation, -1 indicates a strong negative correlation, and 0 indicates no correlation.
- **PCA(Principle Component Analysis):** Through this we choose components which explain the most of the variance of the data. 20 components explained 93% of the variance

- **Results of feature selection and key proteins identified:** After feature selection there are many features that are removed from the model which made our model more effective and efficient.


Explained Variance by Principal Components
individual explained variance
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
Principal components


Cumulative Explained Variance by Principal Components
cumulative explained variance
Cumulative explained variance ratio
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
10
20
30
40
Principal components

# Model Training and Evaluation

Model training involves selecting appropriate machine learning algorithms, splitting the dataset into training and testing sets, tuning hyperparameters, and training the models. In this project, several models were evaluated to find the best-performing one for classifying mice based on protein expression levels.

- **Description of models used (Random Forest, SVM, Neural Networks):**

1. Random Forest: A robust ensemble learning method that builds multiple decision trees and merges them to get a more accurate and stable prediction.
2. Support Vector Machine (SVM): A powerful classifier that finds the optimal hyperplane which best separates the data into different classes.
3. Neural Networks: A deep learning model that can capture complex relationships in the data through multiple layers of neurons.

# Model Training and Evaluation

- **Model training process including data splitting and hyperparameter tuning:**
  Various libraries were imported from scikit-learn that included LogisticRegression, SVC, KNeighborsClassifier, RandomForestClassifier were used to train the model to get maximum accuracy and precision. KNC provided the best results with maximum accuracy of 94.79%, precision of 95.16% and R2_Score of 93.31%.

1. **Data Splitting:**
   The dataset was split into training and testing sets using an 70:30 ratio to ensure sufficient data for both model training and evaluation.
2. **Hyperparameter Tuning:**

- Grid Search and Random Search methods were employed to find the best hyperparameters for each model.
- Parameters used are parameters = {
  'n_neighbors': [3, 5, 7, 9, 11],
  'weights': ['uniform', 'distance'],
  'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
  'p': [1, 2] }

3. **Evaluation metrics and confusion matrix analysis:**

- **KNeighborsClassifier**

The Random Forest model was trained and evaluated on the dataset. The performance metrics indicate how well the model distinguishes between the classes. Here are the detailed **results before hyperparameter tuning**:

In [78]:
```
from sklearn.neighbors import KNeighborsClassifier
knc = KNeighborsClassifier()
```

In [79]:
```
knc.fit(X_train,y_train)
y_pred = knc.predict(X_test)
```

In [80]:
```
print("Performance of model")
print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
print(f"Precision: {precision_score(y_test, y_pred, average='weighted')}")
print(f"R2_Score: {r2_score(y_test, y_pred)}")
```

```
Performance of model
Accuracy: 0.9475308641975309
Precision: 0.951679953188038
R2_Score: 0.9331836047164515
```

# Results after hyperparameter tuning

In [81]:
```
clf.best_params_
```

Out[81]:
```
{'algorithm': 'auto', 'n_neighbors': 3, 'p': 1, 'weights': 'distance'}
```

In [82]:
```
knc_best = KNeighborsClassifier(algorithm = 'auto', n_neighbors = 3, p = 1, weights = 'distance')
```

In [83]:
```
knc_best.fit(X_train,y_train)
y_pred_best = knc_best.predict(X_test)
```

In [84]:
```
print("Performance of model")
print(f"Accuracy: {accuracy_score(y_test, y_pred_best)}")
print(f"Precision: {precision_score(y_test, y_pred_best, average='weighted')}")
print(f"R2_Score: {r2_score(y_test, y_pred_best)}")
```

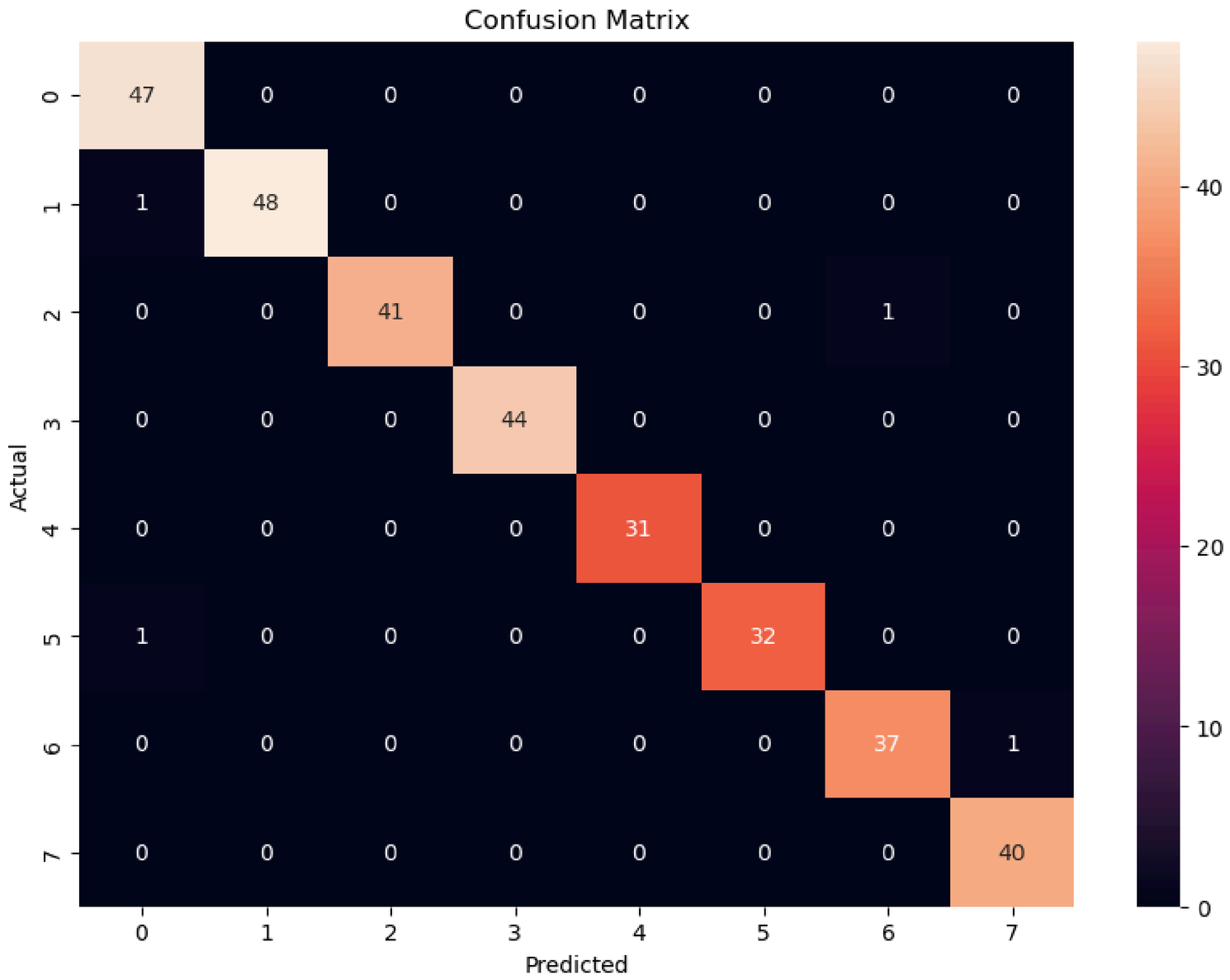
```
Performance of model
Accuracy: 0.9845679012345679
Precision: 0.9851987353206867
R2_Score: 0.9752947782144863
```

# Results and Discussion:

- **Interpretation of the results:**

**Random Forest:** The Random Forest model achieved high accuracy and balanced precision and recall, indicating it is effective in distinguishing between classes. The confusion matrix shows a good balance between true positives and true negatives.
**Support Vector Machine (SVM):** The SVM model showed slightly lower accuracy compared to Random Forest but was still effective. The confusion matrix indicates slightly higher false positives compared to Random Forest.
**KNeighborClassifier(KNC):** The k-nearest neighbors (KNN) algorithm is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

# Confusion Matrix:



- **Identification of key discriminant proteins.**

The Random Forest and SVM models identified several key proteins with high feature importance or significant coefficients, suggesting their critical roles in distinguishing between genotypes.
KNC model helped in finding the key proteins and distinguishing between their genotypes, behavior and treatment.

- **Discussion on the impact of genotype, behavior, and treatment on protein expression.**

**Genotype:**

Significant differences in protein expression profiles were observed between genotypes, indicating specific proteins are associated with Down syndrome.

The identified key proteins may be involved in pathways dysregulated in Down syndrome.

**Behavior:**

The analysis showed correlations between certain behavioral metrics and protein expression levels, suggesting that behavioral characteristics can influence or reflect underlying protein expression patterns.

**Treatment:**

Different treatments resulted in varying protein expression profiles, highlighting potential therapeutic targets. Treatments that normalized protein expression closer to non-Down syndrome profiles were particularly noteworthy.

- Biological significance and potential implications for Down syndrome research.

Pathophysiological Insights:
The identified key discriminant proteins provide insights into the molecular mechanisms underlying Down syndrome. For instance, proteins involved in synaptic function, neurodevelopment, or cellular metabolism could be highlighted.
Novel Targets:
These proteins could serve as biomarkers for diagnosing Down syndrome or monitoring disease progression and response to treatment. They might also offer new therapeutic targets for intervention.
Broader Implications:
The findings can influence future research directions, encouraging studies on related chromosomal disorders or broader neurological conditions.
Clinical Applications: The results could be translated into clinical practice through the development of targeted therapies or personalized medicine approaches for individuals with Down syndrome.

# Conclusion

- Summary of key findings.

All three models effectively differentiated protein expression profiles by genotype.

Key discriminant proteins were identified, providing insights into Down syndrome.

Protein expression is influenced by genotype, behavior, and treatment.

- Limitations of the study.

Sample size may limit the generalizability of findings.

Potential confounding factors not fully accounted for.

- Recommendations for future research.

Conduct validation studies with larger cohorts.

Explore functional roles of key discriminant proteins.

Investigate longitudinal effects of treatments on protein expression

# THANK YOU!

## TEAM F members

- Shruti
- Vikhyat
- Sanidhya
- Sneha
- Kavya
- Rupal
- Viraj
- Shanmugam
- Rishi
- Dhruv
- Abhinav
- Jai
- Ankit
- Sparsh
- Udhaymaayei