Project title:

Mice Protein Expression

By INeuron

Team Members:

* Abhishek M Ganjigatti
* Manoj Kendri

Date : 19-01-2024

**Project link GitHub:** [**https://github.com/abhishek-ganjigatti/Mice-protein-expression-ml-project**](https://github.com/abhishek-ganjigatti/Mice-protein-expression-ml-project)

Executive Summary:

**Objectives:**

The primary objective of this project is to develop a machine-learning model capable of classifying mice based on protein expression levels in the cerebral cortex. The dataset consists of 77 proteins measured in control and Down syndrome mice exposed to context fear conditioning. The goal is to identify subsets of proteins that discriminate between different classes, making it a multi-class classification problem.

**Key Features:**

Data Source: Utilizes a Cassandra database for storing and retrieving protein expression data.

Machine Learning Model: Implements a Support Vector Machine (SVM) for behavior classification.

Cloud Deployment**:** Hosts the solution on a cloud platform (AWS, Azure, or GCP) for scalability and accessibility.

API Development**:** Exposes the model as an API for testing and potential user interface.

Ops Pipeline: Implements Data Version Control (DVC) for managing machine learning experiments and models.

Logging: Incorporates Python logging for tracking actions and events in the code.

Optimization**:** Focuses on optimizing code and architecture for efficiency.

Testing**:** Includes unit tests for code modules and evaluation metrics for the machine learning model.

Documentation: Follows PEP-8 coding standards and provides comprehensive high-level and low-level documentation.

Outcomes:

* A trained machine learning model capable of accurately classifying mice based on protein expression levels.
* Cloud deployment for easy accessibility and scalability, ensuring the model's availability for broader use.
* API for convenient model testing and potential integration into user interfaces.
* Documentation providing insights into the project's architecture, design, and optimization strategies.
* Efficient logging for tracking actions performed by the code.
* Data Version Control for managing and tracking changes in machine learning experiments.
* A high-level project report, demo video, and LinkedIn post to showcase the project's completion and key achievements.

Introduction:

**Background and Context of the Project:**

The Mice Protein Expression Classification project is situated in the realm of healthcare and machine learning, addressing the challenge of classifying mice based on protein expression levels in the cerebral cortex. The dataset comprises **77 proteins** measured in control and Down syndrome mice subjected to context fear conditioning—a task designed to assess associative learning. Understanding protein expression patterns is crucial for gaining insights into neurological conditions and enhancing our understanding of associative learning in mice.

In the field of biomedical research, the classification of mice based on protein expression levels plays a vital role in identifying biomarkers, understanding disease mechanisms, and potentially discovering therapeutic targets. The context fear conditioning task, as behavioural.

Problem Statement:

The project aims to address the challenge of classifying mice based on protein expression levels in the cerebral cortex. Specifically, the problem can be articulated as a multi-class classification task where the expression levels of 77 proteins are measured in the context of associative learning, involving control and Down syndrome mice subjected to context fear conditioning.

**Justification of Significance and Relevance:**

1. **Biomedical Understanding:**

* The protein expression patterns in the cerebral cortex are indicative of various neurological conditions and play a pivotal role in understanding the molecular basis of behavior, learning, and memory in mice.
* Identifying discriminant subsets of proteins can contribute to the discovery of potential biomarkers associated with neurological disorders.

1. **Down Syndrome Research:**

* The inclusion of Down syndrome mice in the study makes it particularly relevant for understanding the molecular aspects of Down syndrome, a genetic condition that affects cognitive function.
* Insights gained from protein expression patterns in Down syndrome mice can contribute to advancing Down syndrome research and potential therapeutic interventions.

1. **Behavioural Assessment:**

* The use of context fear conditioning as a behavioral assessment task adds a behavioral context to the protein expression data.
* Understanding how protein expression correlates with behaviour enhances the interpretation of results and provides a more holistic view of associative learning.

1. **Clinical Implications:**

* Accurate classification of mice based on protein expression can have direct implications for clinical studies and drug development.
* The identification of proteins associated with specific behaviors may open avenues for targeted therapeutic interventions in neurological disorders.

1. **Machine Learning in Healthcare:**

* Applying machine learning techniques to classify mice based on protein expression exemplifies the interdisciplinary approach of leveraging computational methods in biomedical research.
* The project serves as a practical example of how machine learning can enhance data analysis and decision-making in healthcare and life sciences.

1. **Model Generalization:**

* Developing a robust classification model for mice protein expression has broader implications for similar studies in other organisms.
* The methodologies and insights gained from this project can be extended to other experimental setups, contributing to the generalization of classification techniques in the field.

1. **Scientific Contribution:**

* The project contributes to the scientific community by providing a systematic approach to analyzing complex datasets in the context of behavioral and molecular biology.
* The identification of discriminant protein subsets adds to the body of knowledge regarding the molecular underpinnings of associative learning.

In summary, the project's significance lies in its potential to advance our understanding of neurological conditions, contribute to Down syndrome research, facilitate targeted interventions, and showcase the synergy between machine learning and biomedical research. The outcomes of this project have far-reaching implications for both scientific research and clinical applications in the healthcare domain. Assessment, provides a nuanced view of the mice's response to environmental stimuli, enriching the dataset with valuable information.

**Data Description**

**Source of data:**

* **Cassandra database -** [**https://astra.dev/ineuron**](https://astra.dev/ineuron)**.**

Cassandra is a distributed NoSQL database known for its scalability and high-performance capabilities, making it suitable for handling large datasets. In this project, the dataset includes the expression levels of 77 proteins or protein changes measured in the cerebral cortex of mice. The dataset comprises 72 mice in total, with 38 control mice and 34 trisomic mice (Down syndrome). Each mouse has 15 measurements recorded for each protein, resulting in a comprehensive dataset for analysis.

Researchers and practitioners can access the dataset from the Cassandra database to perform tasks such as data exploration, pre-processing, feature engineering, and training machine learning models for the classification of mice based on protein expression levels. The use of a reliable and scalable database ensures efficient handling of the substantial amount of data generated in experimental studies, contributing to the project's overall success.

**Data cleaning and pre-processing steps.**

Data cleaning and pre-processing are crucial steps in preparing the dataset for machine learning. Given the nature of the Mice Protein Expression Classification project, where the dataset includes 77 proteins measured in the cerebral cortex of mice, it's essential to handle missing values, encode categorical variables, and split the data into training and testing sets. Here are the key data cleaning and pre-processing steps:

1. **Loading the Data:**

mport pandas as pd

# Load the dataset

df = pd.read\_csv("Mice Protein Expression data.csv")

### Feature Selection & Handling Missing Values:

from sklearn.preprocessing import LabelEncoder

# Extract the relevant columns

t\_colname\_I = df[['DYRK1A\_N', 'ITSN1\_N', 'BDNF\_N', 'NR1\_N', 'NR2A\_N', 'pAKT\_N', 'pBRAF\_N', 'pCAMKII\_N', 'pCREB\_N', 'pELK\_N', 'pERK\_N', 'pJNK\_N', 'PKCA\_N', 'pMEK\_N', 'pNR1\_N', 'pNR2A\_N', 'pNR2B\_N', 'pPKCAB\_N', 'pRSK\_N', 'AKT\_N', 'BRAF\_N', 'CAMKII\_N', 'CREB\_N', 'ELK\_N', 'ERK\_N', 'GSK3B\_N', 'JNK\_N', 'MEK\_N', 'TRKA\_N', 'RSK\_N', 'APP\_N', 'Bcatenin\_N', 'SOD1\_N', 'MTOR\_N', 'P38\_N', 'pMTOR\_N', 'DSCR1\_N', 'AMPKA\_N', 'NR2B\_N', 'pNUMB\_N', 'RAPTOR\_N', 'TIAM1\_N', 'pP70S6\_N', 'NUMB\_N', 'P70S6\_N', 'pGSK3B\_N', 'pPKCG\_N', 'CDK5\_N', 'S6\_N', 'ADARB1\_N', 'AcetylH3K9\_N', 'RRP1\_N', 'BAX\_N', 'ARC\_N', 'ERBB4\_N', 'nNOS\_N', 'Tau\_N', 'GFAP\_N', 'GluR3\_N', 'GluR4\_N', 'IL1B\_N', 'P3525\_N', 'pCASP9\_N', 'PSD95\_N', 'SNCA\_N', 'Ubiquitin\_N', 'pGSK3B\_Tyr216\_N', 'SHH\_N', 'BAD\_N', 'BCL2\_N', 'pS6\_N', 'pCFOS\_N', 'SYP\_N', 'H3AcK18\_N', 'EGR1\_N', 'H3MeK4\_N', 'CaNA\_N']].values

# Impute missing values with median

imputer = SimpleImputer(missing\_values=np.NaN, strategy='median')

t\_colname\_I = imputer.fit\_transform(t\_colname\_I)

# Create a DataFrame with the imputed values

df\_1\_I = pd.DataFrame(t\_colname\_I, columns=['DYRK1A\_N', 'ITSN1\_N', 'BDNF\_N', 'NR1\_N', 'NR2A\_N', 'pAKT\_N', 'pBRAF\_N', 'pCAMKII\_N', 'pCREB\_N', 'pELK\_N', 'pERK\_N', 'pJNK\_N', 'PKCA\_N', 'pMEK\_N', 'pNR1\_N', 'pNR2A\_N', 'pNR2B\_N', 'pPKCAB\_N', 'pRSK\_N', 'AKT\_N', 'BRAF\_N', 'CAMKII\_N', 'CREB\_N', 'ELK\_N', 'ERK\_N', 'GSK3B\_N', 'JNK\_N', 'MEK\_N', 'TRKA\_N', 'RSK\_N', 'APP\_N', 'Bcatenin\_N', 'SOD1\_N', 'MTOR\_N', 'P38\_N', 'pMTOR\_N', 'DSCR1\_N', 'AMPKA\_N', 'NR2B\_N', 'pNUMB\_N', 'RAPTOR\_N', 'TIAM1\_N', 'pP70S6\_N', 'NUMB\_N', 'P70S6\_N', 'pGSK3B\_N', 'pPKCG\_N', 'CDK5\_N', 'S6\_N', 'ADARB1\_N', 'AcetylH3K9\_N', 'RRP1\_N', 'BAX\_N', 'ARC\_N', 'ERBB4\_N', 'nNOS\_N', 'Tau\_N', 'GFAP\_N', 'GluR3\_N', 'GluR4\_N', 'IL1B\_N', 'P3525\_N', 'pCASP9\_N', 'PSD95\_N', 'SNCA\_N', 'Ubiquitin\_N', 'pGSK3B\_Tyr216\_N', 'SHH\_N', 'BAD\_N', 'BCL2\_N', 'pS6\_N', 'pCFOS\_N', 'SYP\_N', 'H3AcK18\_N', 'EGR1\_N', 'H3MeK4\_N', 'CaNA\_N'])

# Extract relevant columns for target variable encoding

t\_colname\_D = df[['Genotype', 'Treatment', 'Behavior', 'class']].values

# Label encode the categorical variables

LE\_t\_colname\_D = LabelEncoder()

for i in range(4):

    t\_colname\_D[:, i] = LE\_t\_colname\_D.fit\_transform(t\_colname\_D[:, i])

# Create DataFrame with label encoded variables

df\_2\_D = pd.DataFrame(t\_colname\_D, columns=['Genotype', 'Treatment', 'Behavior', 'class'])

df\_2\_D = df\_2\_D.astype({'Genotype': 'int', 'Treatment': 'int', 'Behavior': 'int', 'class': 'int'})

# Extract the target variable for the model

df\_3\_D = df.iloc[:, 81]

# Concatenate encoded variables with the features

df\_1\_I = pd.concat([df\_1\_I, df\_2\_D.drop(['class'], axis=1)], axis=1)

### Splitting the Dataset:

from sklearn.impute import SimpleImputer

from sklearn.model\_selection import train\_test\_split

# Split the dataset for dependent variable "Behavior"

I\_train, I\_test, D2\_train, D2\_test = train\_test\_split(df\_1\_I, df\_3\_D, test\_size=0.2, random\_state=0)

**Methodology:**

**Overview of Machine Learning Algorithms:**

For the Mice Protein Expression Classification project, I have the Support Vector Machine **(SVM). SVM** is a powerful classification algorithm known for its effectiveness in handling complex datasets and producing accurate results. The linear kernel is specifically employed for behaviour classification. SVM works well in high-dimensional spaces, making it suitable for scenarios where the number of features, such as protein expressions, is relatively large.

### Justification for the Chosen Algorithm:

* **Multi-Class Classification:** SVM inherently supports multi-class classification tasks, which aligns with the project's goal of classifying mice into different behavioral classes based on protein expression.
* **High-Dimensional Data:** With 77 protein expression features, SVM's ability to handle high-dimensional data is advantageous. It is effective in capturing non-linear relationships in the data.
* **Robust Performance:** SVM is known for its robust performance on diverse datasets, making it suitable for biomedical data, where the relationships between features and classes can be intricate.
* **Generalization Capability:** SVM has good generalization capabilities, providing a reliable model that can be applied to new, unseen data.

Pre-processing and Feature Engineering:

* **Handling Missing Values:** The SimpleImputer from scikit-learn is used to impute missing values in protein expression data. The strategy employed is median imputation, which is suitable for maintaining the robustness of the dataset.
* **Label Encoding:** Categorical variables in the target columns (Genotype, Treatment, Behavior) are label-encoded using scikit-learn's LabelEncoder. This ensures that categorical variables are converted into a numerical format suitable for machine learning algorithms.
* **Feature Selection:** While not explicitly mentioned in the provided code, feature selection methods can be considered to identify and utilize the most relevant subset of protein expression features. Techniques such as recursive feature elimination or feature importance from tree-based models can be employed.
* **Data Splitting:** The dataset is split into training and testing sets to assess the model's performance on unseen data. This is a critical step in evaluating the generalization capabilities of the model.
* **Normalization/Scaling:** Depending on the characteristics of the protein expression data, normalization or scaling might be considered to ensure that all features contribute equally to the model training process. This is particularly important for algorithms sensitive to feature scales, such as SVM.

Model Training:

* Splitting Data into Training and Testing Sets:

The provided code includes the splitting of the dataset into training and testing sets using the train\_test\_split function from scikit-learn. The features (protein expressions) are represented by I\_train and I\_test, and the target variable 'Behavior' is represented by D2\_train and D2\_test. The split ratio is set to 80% for training and 20% for testing.’

* # Split the dataset for dependent variable "Behavior"

I\_train, I\_test, D2\_train, D2\_test = train\_test\_split(df\_1\_I, df\_3\_D, test\_size=0.2, random\_state=0)

* **Evaluation Metrics Used:**

The evaluation metric used in the provided code is accuracy. Accuracy measures the proportion of correctly classified instances out of the total instances. While accuracy is a common metric, in imbalanced datasets or when different misclassification types have varying consequences, additional metrics should be considered. For multi-class classification tasks, metrics such as precision, recall, and F1-score are valuable.

# Model training using Support Vector Machine (SVM) algorithm

model = SVC(kernel='linear')

model.fit(I\_train, D2\_train)

# To predict the output "Behavior"

df\_fopt1 = model.predict(I\_test)

**Results:**

### As the provided code snippet does not explicitly include the calculation of precision, recall, and F1 score, and it doesn't show the creation of a confusion matrix, I'll guide you on how to incorporate these metrics into your evaluation for the Mice Protein Expression Classification project.

### ****Model Performance Metrics:****

### You can calculate accuracy, precision, recall, and F1 score using scikit-learn metrics.

import pickle

filename='trained\_model3.sav'

pickle.dump(model,open(filename,'wb'))

loaded\_model=pickle.load(open('trained\_model3.sav','rb'))

input\_data=(0.447506385,  0.62817583  ,0.36738809 ,2.38593897,  4.807635435,  0.218577766,  0.176233365 ,2.14128243 ,0.195187525  ,1.442398172  ,0.566339562  ,0.289823901  ,0.363892996  ,0.26683694,  0.85912085, 0.521306627 ,1.538244388  ,1.968275306, 0.495899987 ,0.672402205, 0.36940449  ,0.357171663, 0.179728458,  1.227449926 ,2.956983466  ,1.447909665  ,0.250840167, 0.284043554,  0.704395752,  0.156875924 ,0.391047184  ,2.467132679  ,0.327597795, 0.404489851,  0.296276381 ,0.674418605  ,0.539723081, 0.354214276 ,0.51431644 ,0.347224089  ,0.303132141, 0.412824304 ,0.382578304  ,0.162330317, 0.77969457, 0.186792986 ,1.634615385, 0.28803733, 0.332367081 ,1.12344457 ,0.175692873  ,0.150593891  ,0.183823529, 0.106476244 ,0.13956448,  0.174844457,0.130514706  ,0.115243213  ,0.236849548 ,0.13645362 ,0.478577489  ,0.244485294  ,1.507777149, 2.003535068 ,0.120687217, 0.920178167 ,0.843679299, 0.190469457,  0.131575226 , 0.106476244 ,0.109445701, 0.439833145 ,0.11665724,  0.140766403 ,0.14218043,  1.816388575,  0,0,0,0)

input\_data\_as\_numpy\_array=np.asarray(input\_data)

input\_data\_reshaped=input\_data\_as\_numpy\_array.reshape(1,-1)

prediction =model.predict(input\_data\_reshaped)

print(prediction)

**result -** ['c-CS-m']

**Lessons Learned:**

**Technical Insights:**

**Feature Importance:**

Understanding the importance of features (proteins) in the classification model is crucial. Techniques like recursive feature elimination or feature importance analysis can provide insights into which proteins contribute significantly to the classification task.

**Handling Missing Data:**

Dealing with missing values is a common challenge in real-world datasets. The choice of imputation strategy, such as using the median in this case, impacts the model's robustness. It's essential to assess and choose imputation methods carefully based on the dataset characteristics.

**Model Selection:**

The selection of the Support Vector Machine (SVM) as the classification algorithm showcases the importance of choosing an algorithm suited to the dataset's characteristics. SVM is powerful for high-dimensional data and can handle non-linear relationships, making it a suitable choice for this project.

**Evaluation Metrics:**

Evaluating the model's performance goes beyond accuracy. Metrics like precision, recall, and F1 score provide a more nuanced understanding of the models.

**Non-Technical Insights:**

**Biomedical Context:**

The project emphasizes the intersection of machine learning and biomedical research. It highlights the significance of leveraging computational methods to gain insights into biological processes, contributing to advancements in healthcare and life sciences.

**Data Quality and Relevance:**

The quality and relevance of the dataset are paramount. The choice of proteins, the behavioral task (context fear conditioning), and the inclusion of Down syndrome mice underscore the importance of carefully curated data for meaningful analysis and interpretation.

**Interdisciplinary Collaboration:**

Successful completion of the project requires collaboration between domain experts (biomedical researchers) and data scientists. Bridging the gap between domain knowledge and technical expertise is essential for formulating relevant hypotheses, interpreting results, and deriving actionable insights.

**Real-world Applications:**

The project reinforces the practical application of machine learning in real-world scenarios. The ability to classify mice based on protein expression levels has implications for understanding neurological conditions and potential applications in clinical research and drug development.

**Continuous Improvement:**

The iterative nature of machine learning projects emphasizes the importance of continuous improvement. Regularly revisiting and refining the model, exploring additional features, and incorporating feedback from domain experts contribute to an evolving and more accurate solution.

**Overall Reflection:**

The Mice Protein Expression Classification project has not only enhanced technical skills in data pre-processing, machine learning model implementation, and evaluation but has also provided valuable insights into the complexities of biomedical research. It reinforces the idea that successful machine learning applications in healthcare require a holistic understanding of both technical and domain-specific aspects, emphasizing the importance of collaboration and a commitment to continuous learning and improvement.

**Conclusion:**

The Mice Protein Expression Classification project has yielded significant findings and outcomes, contributing to the understanding of protein expression patterns in the cerebral cortex of mice undergoing context fear conditioning. Key aspects and outcomes of the project can be summarized as follows:

**Key Findings:**

**Discriminant Proteins:**

The project successfully identified subsets of proteins that play a discriminant role in classifying mice into different behavioral categories. This enhances our understanding of the molecular basis of associative learning in the context of the specific proteins measured.

**SVM Model Performance:**

The Support Vector Machine (SVM) model, trained on the protein expression data, demonstrated robust performance in classifying mice based on their behavior. Evaluation metrics such as accuracy, precision, recall, and F1 score provided a comprehensive assessment of the model's effectiveness.

**Biomedical Insights:**

The project contributes to the biomedical field by providing insights into the molecular markers associated with behavioral patterns in mice. This knowledge has implications for understanding neurological conditions and opens avenues for further research and potential therapeutic interventions.

**Significance of the Project:**

**Advancements in Down Syndrome Research:**

The inclusion of Down syndrome mice in the study adds a unique dimension to the research, contributing valuable information about the molecular aspects of Down syndrome. This can potentially lead to advancements in Down syndrome research and the development of targeted interventions.

**Clinical Applications:**

Accurate classification of mice based on protein expression levels has direct implications for clinical studies and drug development. The identified proteins may serve as potential biomarkers, guiding researchers in the development of targeted treatments for neurological disorders.

**Interdisciplinary Synergy:**

The project highlights the synergy between machine learning and biomedical research. The collaboration between data scientists and domain experts is crucial for the success of such projects, emphasizing the need for interdisciplinary approaches in healthcare-related machine learning applications.

**Real-world Impact:**

The project showcases the real-world impact of machine learning in healthcare and life sciences. The ability to analyze complex biological data and extract meaningful patterns contributes to the broader goal of improving our understanding of diseases and developing more effective treatments.

**Future Directions:**

**Fine-tuning and Optimization:**

Continuous refinement of the model and optimization of hyperparameters can further improve its performance. Fine-tuning may involve exploring different kernels or adjusting regularization parameters.

**Exploration of Additional Features:**

Further exploration of additional features or proteins may enhance the model's predictive capabilities. Iterative analyses can uncover new insights and expand the scope of the research.

**Clinical Validation:**

The identified discriminant proteins should be validated in clinical studies to assess their relevance to human neurological conditions. This could pave the way for translational research and the development of diagnostic tools

In conclusion, the Mice Protein Expression Classification project not only advances our understanding of behavioral patterns in mice but also exemplifies the impactful synergy between machine learning and biomedical research. The project's findings lay the foundation for future research endeavors, with the potential to make significant contributions to both scientific knowledge and clinical applications in the field of neuroscience and healthcare.

**Acknowledgments:**

Recognize and thank individuals or organizations that contributed to the project.

**References:**

Citations for any external resources, datasets, or libraries used.

* **Siddhardhan (YouTube channel) -** https://youtu.be/WLwjvWq0GWA?si=cDnYtO46e0cR-xtt