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)

**System Architecture:**

The architecture of the Mice Protein Expression Classification project involves several components that work together seamlessly to achieve the project objectives. Below is a high-level overview of the system architecture:

**1. Data Loading and Preprocessing:**

**Component: Dataset Loader**

Responsible for loading the dataset from the Cassandra database.

Utilizes Pandas and Cassandra driver for data retrieval.

Component: Data Preprocessor

Performs feature selection, handling missing values, and label encoding.

Uses scikit-learn for preprocessing tasks.

**2. Model Training and Evaluation:**

**Component: SVM Model Trainer**

Initializes and trains a Support Vector Machine (SVM) model.

Utilizes scikit-learn for SVM implementation.

**Component: Model Evaluator**

Evaluates the trained model using metrics like accuracy, precision, recall, and F1 score.

Visualizes evaluation results, including a confusion matrix.

**3. Logging and Documentation:**

**Component: Logging Module**

Implements Python logging for tracking actions and events.

Logs stored in a dedicated log file for future reference.

**Component: Documentation Generator**

Generates documentation in compliance with PEP 8 standards.

Produces high-level, architecture, wireframe, and detailed project report documents.

**4. Database Integration:**

**Component: Cassandra Connector**

Establishes a connection to the Cassandra database.

Retrieves and stores data for training and testing.

**5. Cloud Deployment and API/Interface:**

**Component: Cloud Deployment Module**

Selects a cloud platform (AWS, Azure, GCP) for deployment.

Hosts the solution for accessibility and scalability.

**Component: API or User Interface**

Exposes the model as an API or user interface for interaction.

Allows users to test the model with new data.

**6. Ops Pipeline and Continuous Integration:**

**Component: CI/CD Integration**

Utilizes CI/CD tools like Jenkins, Circle CI for automated testing and deployment.

Ensures smooth integration and delivery of project updates.

**Component: AI Ops Pipeline**

Implements AI Ops pipeline using tools like MLflow or DVC.

Manages model versions, experiments, and artifacts.

**7. Optimization Strategies:**

**Component: Optimization Module**

Implements code-level optimizations for improved efficiency.

Explores architecture-level enhancements for performance boost.

8. Testing and Test Cases:

**Component: Testing Suite**

Includes unit, integration, and system testing components.

Ensures the correctness and reliability of each module.

**9. Documentation Links and References:**

**Component: Documentation Repository**

Hosts documentation links on a public GitHub repository.

Includes high-level, architecture, wireframe, and detailed project report documents.

**10. Latency Measurement and Optimization:**

**Component: Latency Measurement Module**

Measures the response time of the model for a given input dataset.

Optimizes the system for reduced latency.

This architecture provides a comprehensive view of how the different components of the Mice Protein Expression Classification project interact and contribute to the overall functionality. Each component plays a crucial role in achieving a well-rounded and effective system.

**References:**

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

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