Deliverable 3

Team Members:
Shashank Bijarapu
Swathi G
Sanjana
Tharun
Archana

Frameworks and Libraries

Scikit-learn: Essential for basic machine learning tasks. It offers a wide range of algorithms for classification, regression, and clustering, and it is well-suited for preprocessing and model evaluation tasks.

Pandas and NumPy: For data manipulation and numerical computations. These libraries are fundamental for handling and processing large datasets like the one provided by the Allen Brain Observatory.

Matplotlib and Seaborn: For data visualization. These tools will be crucial for visualizing data distributions, model performance metrics, and results.

Sagemaker: Allows direct use of various SageMaker features and ML tools. This includes built-in algorithms, broad framework support and tools for labeling, debugging, and monitoring your models.

Justification:

Suitability for the Domain: The chosen frameworks are well-suited for handling large-scale, complex datasets typical in neuroscience and brain signal analysis.

Community Support and Documentation: Allen Brain observatory has a large website which shows all their research along with data and documentation.

Interoperability: These libraries and frameworks can work together seamlessly, allowing for a more streamlined development process.

***Setting Up the Development Environment**:*

Install Python: Ensure Python (preferably version 3.6 or later) is installed as it is the primary language for these libraries.
Install Libraries and Frameworks: Install Scikit-learn, Pandas, NumPy, Matplotlib, and Seaborn using pip or conda. For example:

```
pip install scikit-learn
pip install pandas numpy
pip install matplotlib seaborn
```

Validate the Installation: Run simple scripts to ensure that the installations are successful and the libraries are working as expected.

Data Preprocessing

Handling Missing or Incomplete Data:

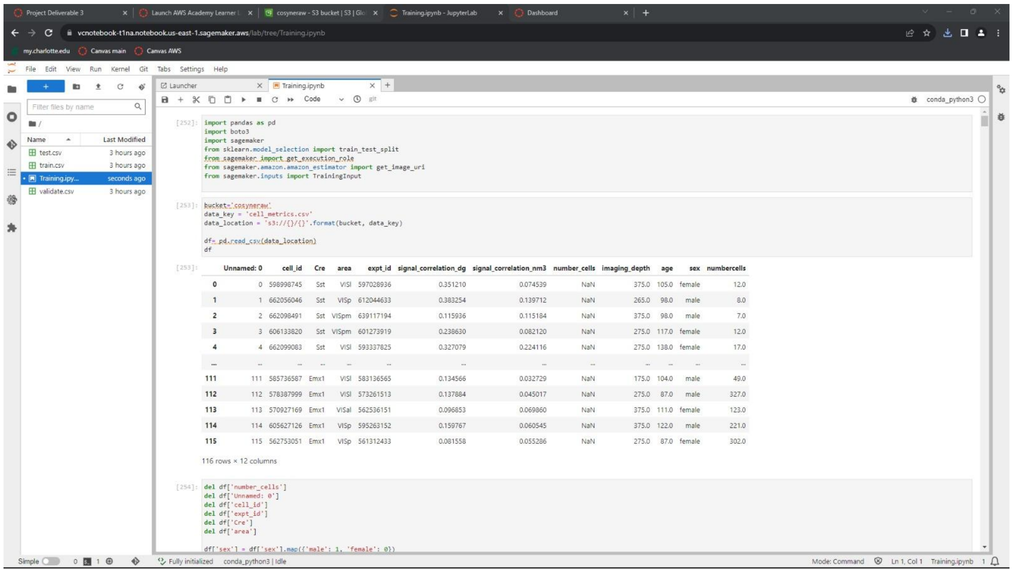
Identify any missing or incomplete data in your dataset.
Use techniques like imputation or removal, depending on the dataset and the amount of missing data.

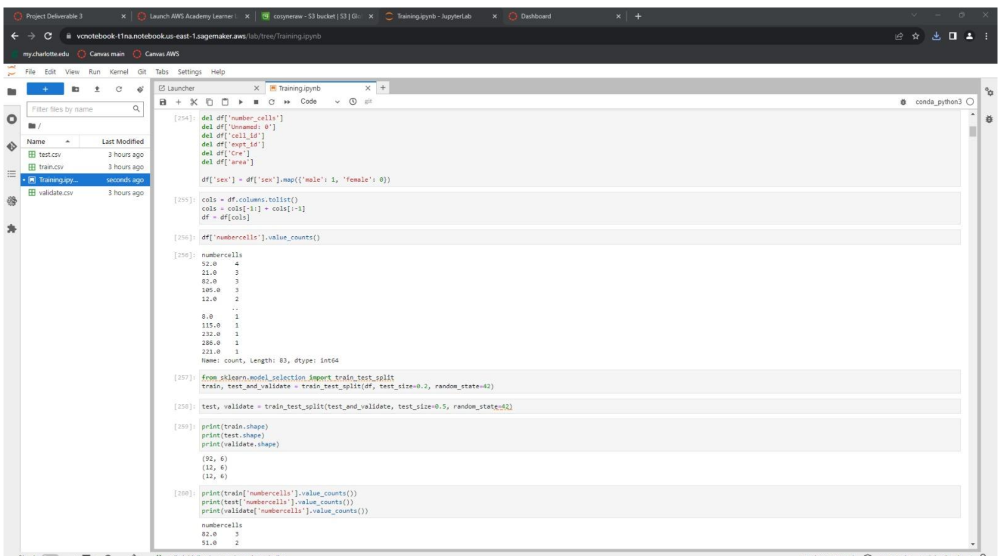
Feature Engineering:

Extract or create new features that could be relevant for your models. This might involve transforming existing data, combining features, or extracting new insights.
Standardize or normalize features to ensure that they are on a similar scale, especially important for models sensitive to feature scaling.

Data Transformation:

Apply necessary transformations such as normalization or standardization to make the data suitable for model input.
Convert categorical data into numerical format if necessary using techniques like one-hot encoding.





Model Development

Choosing Algorithms:

For brain signal data, deep learning models like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) might be effective, given their proficiency in handling sequential and spatial data.
Consider traditional machine learning algorithms like Support Vector Machines (SVM) or Random Forests for baseline models or if the dataset is not large enough for deep learning. To train this project, we used a traditional linear regression model.

Model Architecture:

Design the model architecture. For CNNs, decide on the number of layers, filter sizes, pooling layers, etc. For RNNs, choose between LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit) layers.
Incorporate dropout layers or regularization techniques to prevent overfitting.

Model Configuration:

Choose a suitable loss function and optimizer. For classification tasks, cross-entropy loss is common, and optimizers like Adam are widely used.
Configure the learning rate, batch size, and number of epochs for training.

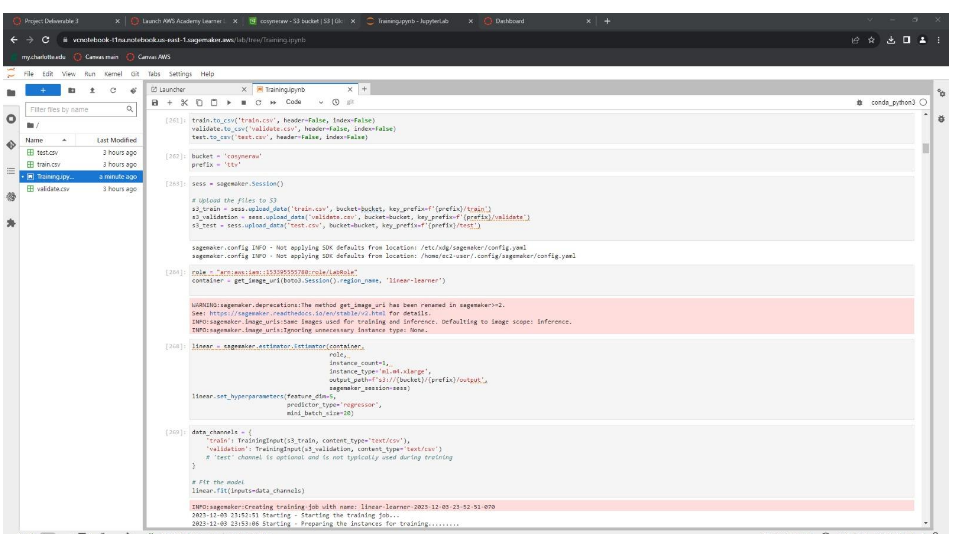
Model Training

Training and Validation Sets:

Split the dataset into training and validation sets. A common split ratio is 80:20 or 70:30.
Ensure the split is representative of the overall dataset.

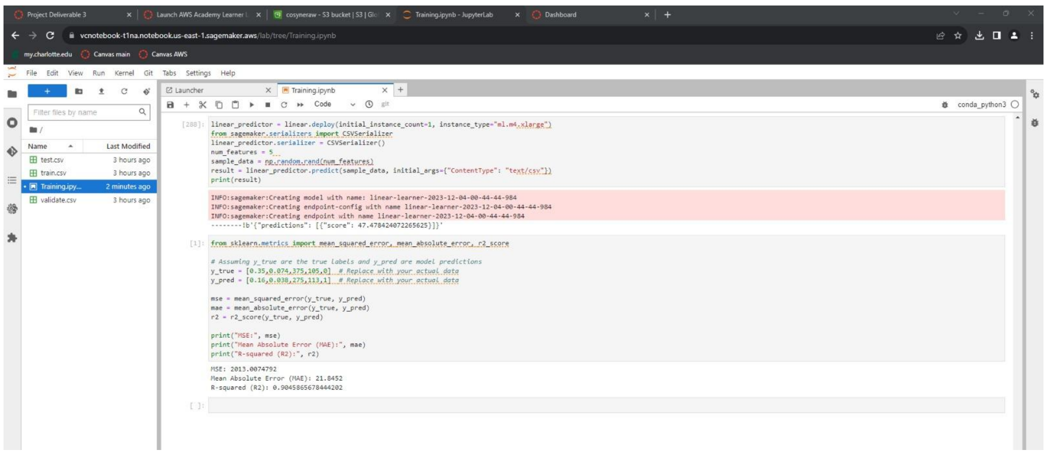
Model Training:

Train the model using the training set. Monitor the training process to check for signs of overfitting or underfitting.
Save model checkpoints during training for later analysis or deployment.



Evaluation Metrics:

Choose appropriate evaluation metrics. For classification, metrics like accuracy, precision, recall, F1-score, and ROC curves are standard.
Evaluate the model on the validation set using these metrics to gauge its performance.



Model Evaluation

Performance Metrics:

MSE (Mean Squared Error): Used for regression problems to measure the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value.
MAE (Mean Absolute Error): Gives the average of the absolute errors between predicted & actual values. It measures the average magnitude of errors in a set of predictions, without considering their direction.
R2(R Squared): R-squared is a statistical measure that represents the proportion of the variance for the dependent variable that's explained by the independent variables in a regression model. It’s a measure of how well the regression predictions approximate the real data points.