

# Work breakdown structure for adVAE: Alzheimer's Data Variational Autoencoder

## Activity 1: Set up GitHub repository

- ✓ Push SRS and Datasets.md to remote repository
- ✓ Push WBS to remote repository

## Activity 2: Create WBS document

- ✓ Add tasks from Problem 1
- ✓ Add tasks from Problem 2

## Activity 3: Source Training Data

- ✓ Look up databases online for gene expression, EEG and MRI data
- ✓ Identify example – training – validation – test data split
- ✓ Perform sanity checks on data sources

## Activity 4: Preprocess example data for developing the model

- ✓ Create function to read example gene expression data (in .csv format)
- ✓ Clean data by removing missing values
- ✓ Normalize data
- ✓ Scale data

## Activity 5: Perform exploratory data analysis on example dataset

- ✓ Visualize the dataset using appropriate plots
- ✓ Compute relevant statistics (mean, median, variance etc.)
- ✓ Identify data distribution

## Activity 6: Construct VAE architecture

- ✓ Define class for VAE model and build VAE network

## Activity 7: Implement auxiliary scripts

- ✓ Code utility functions such as weight initialization, hyperparameter tuning, KL loss evaluation, reconstruction loss evaluation, validation metrics etc.
- ✓ Create script for training model

## Activity 8: Create pipeline for Gene Expression-based VAE

- ✓ Integrate existing scripts into pipeline
- ✓ Make sure pipeline follows data preprocessing > data loading > defining model class > training model > evaluating performance > generating synthetic data

## Activity 9: Create pipeline for MRI-based VAE

- ✓ Identify Alzheimer's MRI database
- ✓ Sanity checks for the MRI images
- ✓ Perform data preprocessing
- ✓ Create scripts for full pipeline

## Activity 10: Assign and implement GitHub issues

- ✓ Assign Gouri specific functionality as a GitHub issue
- ✓ Implement assigned functionality and create pull request
- ✓ Review Gouri's pull request and provide feedback
- ✓ Merge approved pull requests

## Activity 11: Update design documents

- ✓ Update WBS, SRS, DDS to reflect my current progress
- ✓ Update README.md

## Activity 12: Create tutorial

- ✓ Update Datasets.md with description and justification of example data
- ✓ Create a Jupyter Notebook for the tutorial
- ✓ Update README.md with tutorial and dataset links
- ✓ Run the happy path (both pipelines) and generate results using the model

## Activity 13: Test pipelines on real data

- ✓ Run both pipelines on entire real dataset and troubleshoot if necessary
- ✓ Create a Jupyter Notebook for tutorial on real dataset and interpret results
- ✓ Update design documents and README.md
- ✓ Prepare for demo