Project description

Project: Deep Learning Model Development for Mental Health Condition Classification

Objective: Train a deep learning model to classify mental health conditions based on the features in the dataset.

Approach:

Target Variable: Diagnosis

Features: Use all or a selected subset of features to classify the diagnosis into various mental health conditions.

Steps:

Perform feature importance analysis to identify the most relevant features.

Train a fastai tabular learner.

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

Dataset

Dataset is taken from kaggle: Local Mental health Dataset (pakistan).
This dataset contains sample data from questionnaire responses, obtained from The Fountain House Mental Health Institute Pakistan.
https://www.kaggle.com/datasets/mariatamoor/local-mental-health-dataset-pakistan

The dataset contains variable Diagnosis which is used as the output variable for the mental health classification problem and all other variables (110) are features of mental health.

Project phases and methodology

Data Preprocessing and Exploration

Used wolta.data_tools library for data preprocessing.
Initially the dataset contained 764 rows and 111 columns.

Handling null values, single values, unique values:

- Deleted some columns from the dataset with null values by considering the maximum tolerated null value amount as 152.
- Deleted one column because it has a single value.
- Dropped some columns from the dataset with unique values by considering the maximum tolerated unique value amount is 76 in string data.
- Handled missing data in remaining columns by filling the gaps with a placeholder value 'unknown', ensuring there are no NaN values in columns.

Handling imbalanced classes:

- If one category in the target variable dominates, the model may become biased toward predicting that category, ignoring minority classes.
- So, used the function expand_df() from the wolta.data_tools library which balances the dataset by oversampling the minority classes (mood_disorder, substance_use_disorder, and neurosis) to match the size of the majority class (psychosis).

Feature Engineering

After preprocessing the dataset contains the target variable ‘Diagnosis’ and 83 features of mental health.

The 83 features are then categorized into 14 categories as follows.

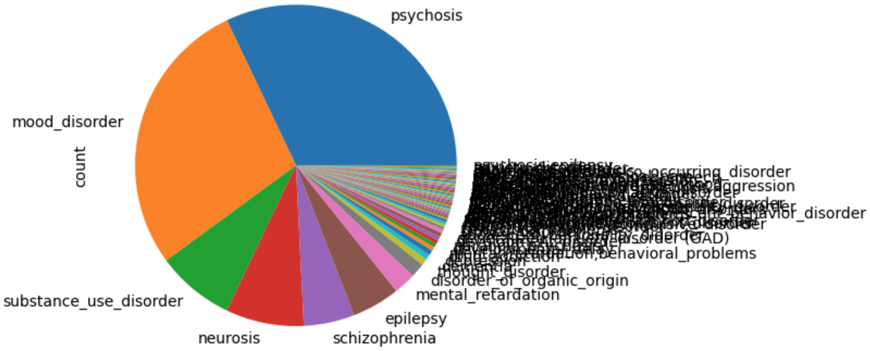
- Clinical and Vital Signs
- Demographics and Personal Information
- Symptoms and History
- Substance Use Indicators
- Treatment and Response
- Behavioral Observations
- Speech and Thought Patterns
- Mood and Emotional State
- Thought Processes
- Content of Thought
- Perceptual Abnormalities
- Cognitive Functions
- Judgment and Insight
- Recommendations and Follow-Up

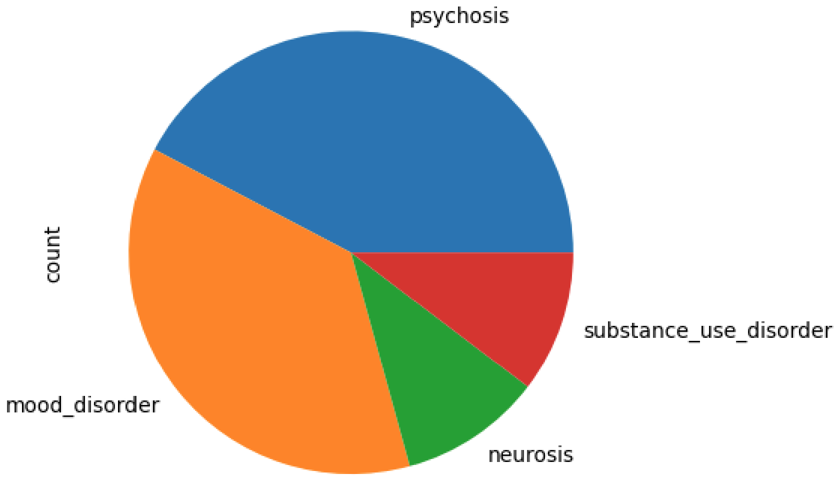
All features in the following categories were removed to avoid unnecessary noise that can degrade the model's performance.

- Demographics and Personal Information
- Treatment and Response
- Judgment and Insight
- Recommendations and Follow-Up

As this dataset contains user inputs from a survey, some features contain string values which cannot be considered as categorical variables. These features were removed by assuming they are not necessary to use in this stage of the model. In future, these values can be used for further development of the project considering Natural Language Processing techniques.

Only 4 categories [psychosis, mood_disorder, neurosis, substance_use_disorder] out of 63 categories in the target variable Diagnosis were selected by considering the frequency after analyzing these pie charts.

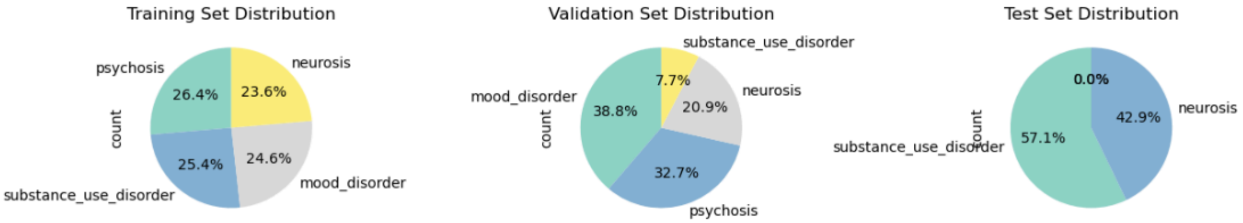




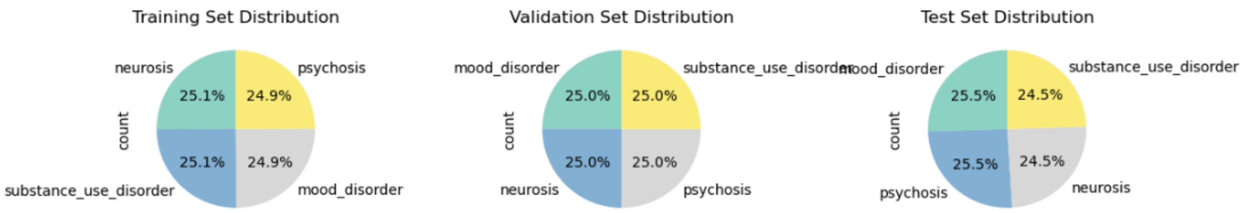
After preprocessing and feature analysing, the final dataset used to train the model contains 980 rows and 67 columns.

Splitting Dataset

Splitting data set into training, validation and test set.



After using stratified splitting for splitting.



Model Selection and Training the model

Set up and train a deep learning model for tabular data using the fastai library.

```
to = TabularPandas(
    df,
    procs=[Categorify, FillMissing, Normalize],
    cat_names=categorical_cols,
    cont_names=continuous_cols,
    y_names=target_col,
    splits=splits
)
dls = to.dataloaders(bs=64)
learn = tabular_learner(
    dls,
    layers=[200, 100],
    metrics=accuracy,
    loss_func=CrossEntropyLossFlat()
)
learn.fit_one_cycle(5, lr_max=1e-2)
```

Architecture of the neural network, deep neural network (DNN) with multiple layers.

The neural network consists of 2 layers:
    The first layer will have 200 neurons.
    The second layer will have 100 neurons.
These layers form a fully connected feedforward neural network.

Loss function used for training:

The model is trained using the 1-cycle learning rate policy, which is a training strategy that adjusts the learning rate during training.
The number of epochs is 5, or full passes over the training data. The model will go through the training data 5 times.

Model evaluation

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

Metric used to evaluate the performance of the model.
accuracy: measures the proportion of correctly classified samples in the dataset.

< —--------------->
< —--------------->

Feature Importance Analysis

Combined permutation-based feature importance and correlation matrix techniques to analyse feature importance.

1. Correlation Matrix as Preprocessing:

   First, use the correlation matrix to remove redundant features (e.g., features highly correlated with each other). This reduces the feature set size and ensures no strong multicollinearity exists.

   A correlation matrix evaluates the linear relationships between features (or between features and the target variable). Features with high inter-correlation are considered redundant, and one of them is removed.

   Example output:

2. Permutation Importance to Refine:

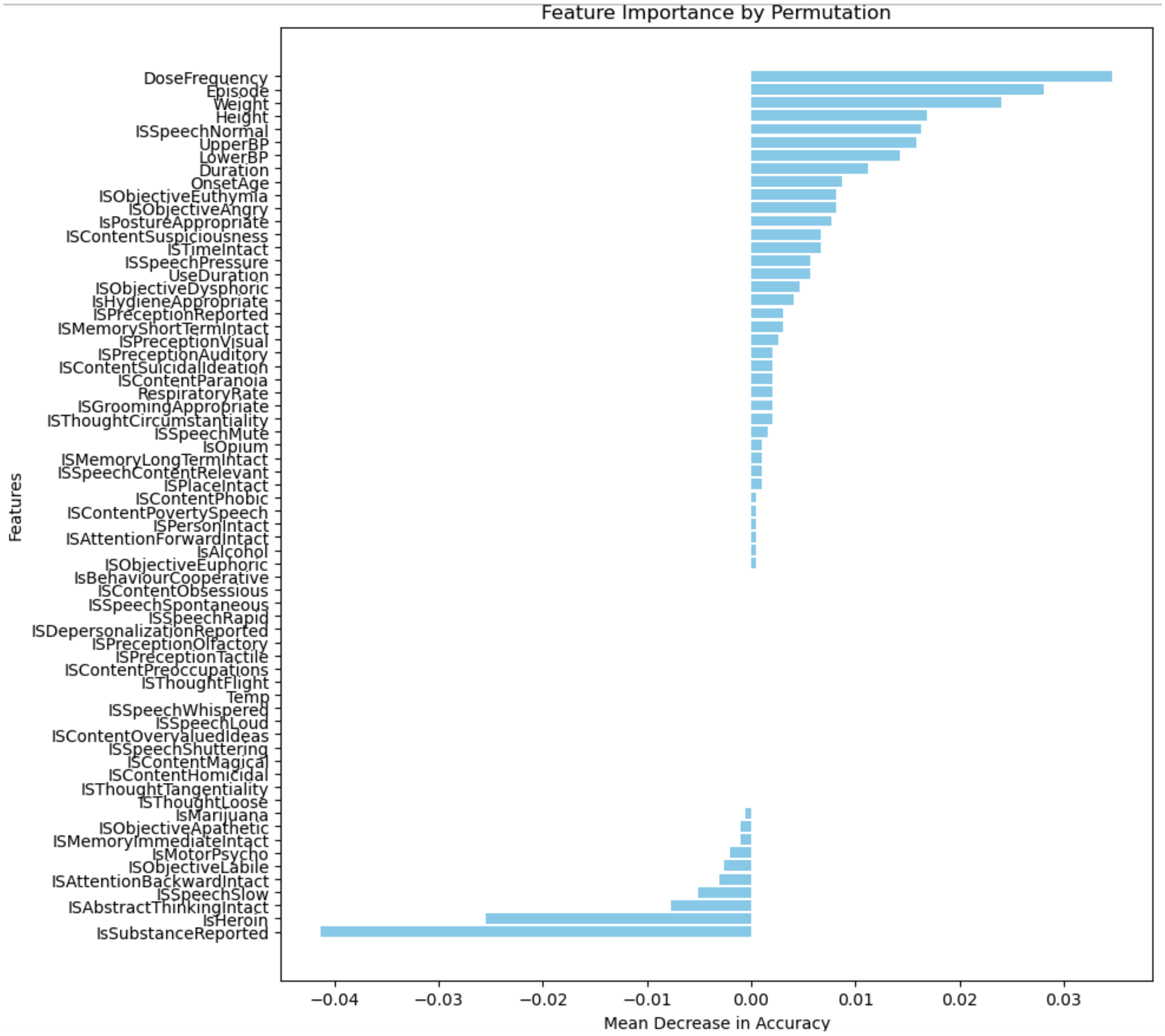
   Next, use permutation-based feature importance to evaluate the impact of the remaining features on the model's performance which helps identify features that have a direct influence on the predictions.

   Permutation-based feature importance is the technique used to evaluate the importance of features in this model by observing how the model's performance changes when the values of a feature are randomly shuffled (permuted).

   The manual implementation of permutation importance is used because the FastAI learner does not directly support standard libraries like sklearn.inspection.permutation_importance.

   Considering the output from feature importance analysis, selected threshold as 0.0 for feature set reduction.

Example output:



Benefits of Combining Both:

- Improved Efficiency: Reducing the feature set with a correlation matrix speeds up permutation importance calculations.
- Better Feature Set: Removes redundant features and focuses on those that are truly impactful for the model.
- Balanced Approach: Leverages the strengths of both techniques—correlation for redundancy and permutation for impact.

Re-Train model after Feature Importance Analysis

< —----------------->

< —----------------->

Deployment and Application

GitHub Link: https://github.com/vijanipiyawardana/MentalHealthAnalysisPlatform

References

https://wolta-docs.readthedocs.io/en/latest/datatools.html#expand-df
https://docs.fast.ai/tabular.learner.html