This is the Text Model Applied over the whole dataset. Done by : Poojal Katiyar
DataSet Size Used: All 275 Participants.
There are three goals for this part.

1. Sleep Disorder(Binary)
2. Sleep Disorder(Multiclass)
3. Depression(Binary)

Colab Link: Text_final_modify.py

**Feature Extraction:**

**What was given:** Transcript for each participant which was pre-processed. For each participant following numerical features were calculated using Transcript.

| Feature | Meaning |
|---|---|
| `Total_Duration` | Total speaking time |
| `Total_Utterances` | Number of utterances |
| `Avg_Confidence` | Average ASR confidence |
| `Total_Words` | Total words spoken |
| `Avg_Words_Per_Utterance` | Average words per utterance |
| `Avg_Pause_Duration` | Mean pause between utterances |
| `Max_Pause_Duration` | Maximum pause |
| `Lexical_Diversity` | Ratio of unique words to total words |
| `Avg_Word_Length` | Mean word length (characters) |
| `Sleep_Word_Count` | Count of sleep-related words |
| `First_Person_Pronouns` | Count of "I", "me", "my" |
| `Negation_Count` | Count of negations like "not", "can't" |

After this, :

1. **Combining** each participant's full transcript into a single text string.

2. **Building a binary Bag-of-Words** (BoW) matrix showing which words (from top 1000) appear in their speech.
3. **Merging** these word features with the earlier computed timing and linguistic features using `Participant_ID`.

Dimensions were large :275*1001 columns, so **Applying PCA** to reduce dimensionality while keeping **95% of the original variance**.

Sleep_Disorder binary label created:

```
train_df['Sleep_Disorder'] = np.where((train_df['PHQ8_3_Sleep'] >= 2) & (train_df['PCL-C_13_Sleep'] >= 3), 1, 0)
train_df.loc[(train_df['PHQ8_3_Sleep'] == 3) | (train_df['PCL-C_13_Sleep'] >= 4), 'Sleep_Disorder'] = 1
```

This is to predict whether a particular person has sleep related issues or not.

## 1. Sleep Related Binary Prediction(Using label as Sleep_Disorder):

Splitting the dataset according to the split column: into test and train.

**Training and evaluating multiple models** — We try different classifiers (XGBoost, Random Forest, LightGBM) to predict sleep disorders and evaluate their accuracy and performance.

- **XGBoost Classifier**: predicts sleep disorder and evaluates using classification report and confusion matrix.

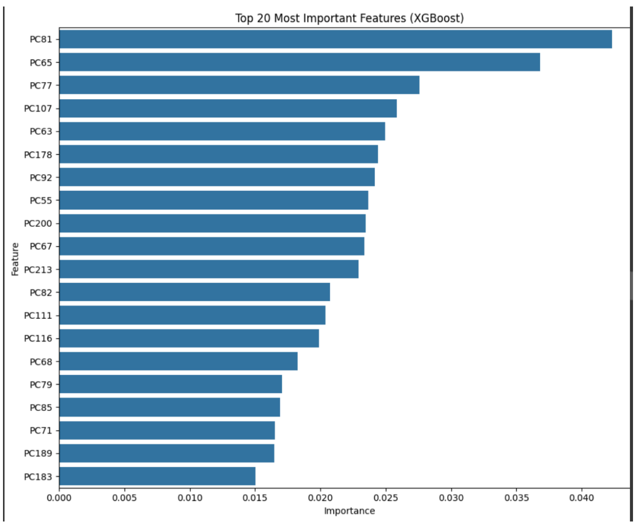
```
              precision    recall  f1-score   support

           0       0.60      0.84      0.70        31
           1       0.62      0.32      0.42        25

    accuracy                           0.61        56
   macro avg       0.61      0.58      0.56        56
weighted avg       0.61      0.61      0.58        56

[[26  5]
 [17  8]]
```

- I plotted the top 20 important features which were important for sleep disorders:



Since I had applied for PCA, it is in the form of PC Components. We can see which of the words were important for detecting sleep issues from these PCA Components.

**PC65:**

- **Top features**: `word_hurt`, `word_natural`, `word_late`, `word_cold`, `word_studies`, `word_woman`
- **Core theme**: Strong association with **negative emotions**, **unpleasant physical or emotional states**, and **social/gender references**.
- **Interpretation**: High PC65 values suggest emotional discomfort or stress (e.g., **pain, delay, disconnection**), possibly contributing to **poorer sleep**.

**PC81 features:**

- Top features: `word_members`, `word_put`, `word_hand`, `word_fall`, `word_process`, `word_think`, `word_kind`
- Core theme: This PC reflects cognitive effort, agency, and social belonging.
- Interpretation: High values suggest mental engagement, possibly overthinking, responsibility, or group dynamics. Could indicate mental busyness that might interfere with restful sleep.

```
Top features for PC81:
          Feature   Loading
547  word_members  0.117236
680      word_put  0.113843
292     word_fall  0.103153
381     word_hand  0.098350
752      word_she  0.091451
672  word_process -0.088249
859    word_think -0.088183
468     word_kind -0.086964
738   word_seeing  0.080395
556     word_mine  0.078652

Top features for PC65:
          Feature   Loading
436     word_hurt  0.102006
902      word_two  0.101330
584  word_natural -0.096295
480     word_late -0.096284
159     word_cold -0.093279
100  word_between  0.089843
969    word_woman  0.087859
338     word_full -0.086391
842    word_terms  0.086034
818  word_studies  0.084521
```

## PC77:

- **Top features**: `word_aren`, `word_watch`, `word_death`, `word_tv`, `word_happen`, `word_transportation`

- **Core theme**: Reflects **existential concerns**, **uncertainty**, and possibly **disrupted routines** (like transportation or events happening unexpectedly).
- **Interpretation**: This PC likely captures **anxiety about life events or meaning**, often associated with **restlessness or disturbed sleep**.

## PC107:

- **Top features**: `word_less`, `word_exactly`, `word_opportunity`, `word_get`, `word_comfortable`, `word_vegas`
- **Core theme**: Mix of **deprivation or striving**, **rigid thinking**, and some **aspirational or comfort-related** language.

```
Top features for PC77:
                 Feature   Loading
58             word_aren  0.111725
877            word_took -0.095515
793           word_space -0.086498
827            word_sure -0.084398
292            word_fall  0.081172
386           word_happen  0.079670
884  word_transportation  0.079564
934           word_watch -0.079520
197           word_death  0.079170
900              word_tv -0.079095

Top features for PC107:
              Feature   Loading
493         word_less  0.098746
276      word_exactly  0.094567
621  word_opportunity  0.092203
347          word_get -0.087470
823       word_summer  0.087009
781      word_someone  0.085634
964         word_will  0.085558
164  word_comfortable  0.085344
918        word_vegas -0.083053
657         word_pick -0.083035
```

- **Random Forest Classifier**:.Accuracy: 0.56 (without cross validation).
  Also applied cross validation which got **Cross-validation accuracy: 0.6945 ± 0.0506**

- **DecisionTreeClassifier(basic tree-based model)**

```
              precision    recall  f1-score   support

           0       0.66      0.61      0.63        31
           1       0.56      0.60      0.58        25

    accuracy                           0.61        56
   macro avg       0.61      0.61      0.61        56
weighted avg       0.61      0.61      0.61        56

[[19 12]
 [10 15]]
```

**The best accuracy of 0.6945 ± 0.0506 was observed using Random Forest Classifier with cross validation.**

2. **Sleep Related Prediction(Using label as `PHQ8_3_Sleep`):**

Label to predict is multi class label with numclasses=4.

**Models used:**

**1.Voting Classifier** (soft voting) combining:

- `DecisionTreeClassifier`
- `XGBClassifier` (XGBoost)
- `LogisticRegression` (with `max_iter=1000`)

```
Accuracy: 0.35714285714285715

Classification Report:
              precision    recall  f1-score   support

           0       0.39      0.72      0.51        18
           1       0.29      0.29      0.29        14
           2       0.33      0.09      0.14        11
           3       0.33      0.15      0.21        13

    accuracy                           0.36        56
   macro avg       0.34      0.31      0.29        56
weighted avg       0.34      0.36      0.31        56


Confusion Matrix:
 [[13  2  2  1]
 [ 7  4  0  3]
 [ 6  4  1  0]
 [ 7  4  0  2]]
```

**2.XGBClassifier (XGBoost)** separately trained again:

- With **sample weights** to handle **class imbalance** (`class_weight='balanced'`)
- For **multiclass classification** (`objective='multi:softprob'`, `num_class=4`)
- Accuracy: 0.39

**The accuracy best observed was: 0.39, its accuracy is less due to the fact it was multiclass classification with numclasses=4.**

3. **Depression Prediction(with label used as `Depression_label`) :**

1.DecisionTreeClassifier

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.71 | 0.82 | 0.76 | 39 |
| 1 | 0.36 | 0.24 | 0.29 | 17 |
| accuracy | | | 0.64 | 56 |
| macro avg | 0.54 | 0.53 | 0.52 | 56 |
| weighted avg | 0.61 | 0.64 | 0.62 | 56 |

```
[[32  7]
 [13  4]]
```

2.XGBClassifier

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.70 | 1.00 | 0.82 | 39 |
| 1 | 0.00 | 0.00 | 0.00 | 17 |
| accuracy | | | 0.70 | 56 |
| macro avg | 0.35 | 0.50 | 0.41 | 56 |
| weighted avg | 0.49 | 0.70 | 0.57 | 56 |

```
[[39  0]
 [17  0]]
```

Using XGB Classifier was inefficient as it could not predict depressed=1.

### The best performance is seen using DecisionTreeClassifier with accuracy 0.64.

Moreover this depression was calculated using sleep features in mind as feature extraction was done using that. This kind of helps us correlate that sleep features affect depression.

Same thing done as for sleep to find the most important features and then printed the top words for these PC

PC163 was associated with a mix of reflective states and emotional arousal. It captures both calm and anxious thoughts, with key words like word_was, word_up, and word_supposed, reflecting how individuals process emotional or mental states.

PC199 was linked to anxiety and ruminative thinking. Words like word_anxious, word_that, and word_later suggest a tendency to dwell on anxious thoughts and future concerns, indicating cognitive patterns tied to anxiety.

PC55 focused on social and motivational language, emphasizing themes of personal growth and group dynamics. Words such as word_age, word_san, and word_motivated reflect motivational and social aspects of behavior in group contexts.

PC73 was associated with emotional and social experiences, highlighting intense feelings and interactions. Words like word_very, word_party, and word_fell suggest a focus on emotional engagement and social situations that evoke strong emotions.

The best model accuracy were:

- **Sleep(binary): 0.69**
- **Sleep(multi class):0.39**
- **Depression(Binary):0.64**

**Another Thing done by me as part of Assignment was to convert Audio into Text to check if the Transcript given was apt or not and also to enable converting raw Audio when given as input to generate Transcript.**

**Colab Link: Audio_to_transcript.ipynb**

Here's a summary of what the code does:

1. Load Whisper Model: The code loads the Whisper model (`base` version) to transcribe audio to text.
2. Transcribe Audio: The `.wav` audio file is transcribed into text, and segments of speech are identified with their start and end times.
3. Process Transcription Data:
    - For each transcription segment, the start and end times are rounded.
    - The text content of each segment is extracted.
    - The confidence level (indicating how certain the model is about the transcription) is retrieved and normalized to a 0-1 scale.
4. Store Results: The transcription data (start time, end time, text, and confidence) is stored in a structured table (Pandas DataFrame).

**This is an example of Transcript produced by using example Audio as input using Whisper model.**

| Start_Time | End_Time | Text | Confidence |
|---|---|---|---|
| 0.0 | 5.1 | Alright, looks good. So let's move around just a little bit because we have an Xbox | 0.874728 |
| 5.1 | 8.3 | Connect that's not moving the motion. There we go. | 0.874728 |
| 8.3 | 16.3 | Taking you up to the spine. Now we're going to do something that things are quick minutes. | 0.874728 |
| 16.3 | 17.3 | Oh. | 0.874728 |
| 17.3 | 20.3 | Think you're equipped with that. | 0.874728 |
| 20.3 | 21.3 | Yeah. | 0.874728 |
| 21.3 | 22.3 | Which is how many? | 0.874728 |
| 22.3 | 25.3 | These are the audio and the video can get together. | 0.874728 |
| 25.3 | 26.3 | Oh, so they'll be in harmony. | 0.874728 |
| 26.3 | 27.3 | In harmony. | 0.874728 |
| 27.3 | 29.7 | Okay. All right. It sounds good. | 0.874728 |
| 30.5 | 34.3 | So I'm going to pull up the virtual human. She's going to chat with you and then when she is done, | 0.868804 |
| 34.3 | 37.1 | she'll let you know and you can go and ring that doorbell. | 0.868804 |
| 37.1 | 38.1 | Oh, okay. | 0.868804 |
| 40.1 | 41.1 | I agree. | 0.868804 |
| 48.6 | 49.6 | Hi. | 0.868804 |
| 49.6 | 51.6 | I'm out of the room. | 0.868804 |
| 51.6 | 53.6 | Thanks for being with us. | 0.868804 |
| 53.6 | 55.6 | I'm going to go in the room. | 0.868804 |
| 55.6 | 57.6 | I'm not here. | 0.868804 |
| 57.6 | 59.6 | I'm going to go in the room. | 0.891977 |
| 59.6 | 61.6 | I'm going to go in the room. | 0.891977 |
| 61.6 | 63.6 | I'm going to go in the room. | 0.891977 |
| 63.6 | 65.6 | I'm going to go in the room. | 0.891977 |
| 69.6 | 71.6 | Sure. | 0.891977 |
| 75.6 | 77.6 | Okay. | 0.891977 |
| 81.6 | 85.6 | Now I'll turn it into this. | 0.891977 |
| 87.6 | 95.6 | Well, that's a good question. | 0.959801 |
| 95.6 | 98.6 | I like to familiarity with everything. | 0.959801 |
| 98.6 | 101.6 | I know where everything is in the city. | 0.959801 |

For testing I converted the audio into transcript using the above code, and tested it using the model parameters that was the prediction.

Code for prediction taking input as transcript and predicting : Prediction.ipynb

Real Life Applications:

**1. Clinical Screening Tools in Mental Health & Sleep Clinics**

- **Context**: When patients visit a clinic, clinicians can ask open-ended questions or conduct brief interviews.
- **Application**: Audio is recorded and transcribed automatically using the Whisper model. The transcript is passed through the best-performing model (e.g., Decision Tree, HistGradientBoosting).
- **Interpretation**:
  - Binary prediction (e.g., has sleep disorder or not) helps **flag high-risk individuals**.
  - Multiclass prediction helps **triage** based on severity (e.g., mild, moderate, severe).
  - Helps doctors prioritize and tailor interventions early, reducing the diagnostic burden.

Sometimes individuals themselves don't know they are depressed. So for that they could be made to talk with such chatbot applications which can use the text to find the extent of depression, sleep related issues, etc.

Text modality has the lowest accuracy in depression prediction because language is often subtle and ambiguous, making it difficult to detect depression through words alone. The lack of direct emotional or physiological cues, combined with challenges in feature extraction and context understanding, makes text-based models less effective compared to other modalities like  speech.