**INVESTIGATING SPEECH SIGNALS USING BRAIN COMPUTER INTERFACE**

Bonafide record of work done by

|  |  |
| --- | --- |
| **NAVANEETHA KRISHNAN K S** | **(21Z233)** |
| **SUREYA NARAYANAN K** | **(21Z261)** |
| **SUSHANTH S** | **(21Z262)** |
| **ASWIN SAILESH V S** | **(21Z265)** |
| **ABAI KUMAR I** | **(22Z434)** |

**19Z610 – MACHINE LEARNING LABORATORY**

Dissertation submitted in the fulfillment of the requirements for the

degree of

**BACHELOR OF ENGINEERING**

**BRANCH: COMPUTER SCIENCE AND ENGINEERING**



APRIL 2024

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**PSG COLLEGE OF TECHNOLOGY**

**(Autonomous Institution)**

**COIMBATORE – 641 004**

**Problem Statement:**

Brain-Computer Interfaces (BCIs) offer a promising avenue for direct communication by bridging the gap between the brain and external devices. This project investigates the role of the TP9 region in speech generation using Electroencephalography (EEG) signals. The TP9 region remains relatively unexplored in speech production, but we hypothesize it plays a crucial role. By utilizing EEG, a non-invasive neuroimaging technique, this study aims to pinpoint the TP9 region's contribution to speech signal generation. This research has potential implications for advancing our understanding of neuroscience, as well as for developing clinical applications and interventions for speech disorders

**Dataset Description:**

The dataset involves utilizing a Muse headband, a wearable EEG device designed to measure brain activity. The Muse headband records EEG signals, capturing brainwaves such as alpha, beta, theta, and gamma waves. These brainwave patterns are indicative of different mental states, offering valuable insights into cognitive processes. The collected dataset comprises EEG signals recorded while participants engaged in tasks involving imagination of speaking specific words and periods of remaining silent. This is a real time dataset where the task design allows for the examination of neural activity associated with speech production, offering rich data for investigating the role of the TP9 region in speech generation.

Link: <https://drive.google.com/drive/folders/1e6fXKMCwLqs7R3SpSnNq7z_K541MpsCI?usp=sharing>

**Models used:**

**Random Forest:**

Random Forest is a versatile machine learning algorithm used for both classification and regression tasks. It belongs to the ensemble learning family, combining multiple decision trees to make predictions.

**Extra tree classifier:**

The Extra Trees Classifier is a machine learning algorithm that belongs to the ensemble learning family, similar to Random Forest and Gradient Boosting.

**Tools and Libraries used:**

**Hardware Requirements:**

* Muse Brain Sensing headband

**Software Requirements:**

* Visual studio code editor
* Python 3.11.4
* Mind monitor Application

**Challenges faced:**

* **Artifact Removal:** EEG signals are susceptible to various artifacts, including muscle activity, eye movements, and environmental noise. Removing these artifacts from the recorded data is crucial to ensure the accuracy of the analysis.
* **Participant Variability:** Neural activity varies between individuals, making it important to recruit a diverse sample of participants. However, this variability can also introduce challenges in data analysis and interpretation.
* **Equipment Limitations:** EEG recording equipment may have limitations in terms of spatial resolution, signal quality, and sampling rate. Researchers must carefully select and calibrate equipment to ensure optimal data quality.
* **Task Compliance:** Ensuring that participants accurately follow task instructions during data collection is essential for obtaining reliable results. Researchers must carefully monitor participants and provide clear instructions to minimize errors and inconsistencies.

**Contribution of Team Members:**

|  |  |  |
| --- | --- | --- |
| **Roll No** | **Name** | **Contributions** |
| 21z233 | **Navaneetha Krishnan K S** | Exploration and implementation of data preprocessing |
| 21z261 | **Sureya Narayanan K** | Data collection |
| 21z262 | **Sushanth S** | Data collection |
| 21z265 | **V S Aswin Sailesh** | Exploration and implementation of training models |
| 22z434 | **Abaikumar I** | Exploration and implementation of training models. |

**Annexure 1:**

**Extra tree classifier:**

**import** pandas **as** pd

**from** sklearn.ensemble **import** ExtraTreesClassifier

**from** sklearn.metrics **import** accuracy\_score

**from** sklearn.model\_selection **import** train\_test\_split

**from** joblib **import** dump

output\_list **=** ['Move', 'Silent', 'Water',

'Yes', 'No', 'Toilet', 'Doctor', 'Pain']

**def** generate\_model(grp1, grp2):

df **=** pd**.**read\_csv("./Step-5 Combined CSV/Speech\_data.csv")

**for** i **in** output\_list:

**if** i **!=** grp1 **and** i **!=** grp2:

df **=** df[**~**df['Output']**.**str**.**contains(i)]

*# remove features*

**for** features **in** associations[grp1][grp2]:

**del** df[features]

*# Extract features and target variable*

X **=** df**.**drop('Output', axis**=**1)

y **=** df['Output']

*# Split the data into training and testing sets*

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(

X, y, test\_size**=**0.2, random\_state**=**42)

et\_classifier **=** ExtraTreesClassifier(n\_estimators**=**100, random\_state**=**42)

*# Train the classifier*

et\_classifier**.**fit(X\_train, y\_train)

*# Make predictions on the test set*

y\_pred **=** et\_classifier**.**predict(X\_test)

*# Evaluate the performance*

accuracy **=** accuracy\_score(y\_test, y\_pred)

print(f'Accuracy for {grp1}-{grp2}: {accuracy:.4f}')

**Random Forest:**

*# Data Processing*

**import** pandas **as** pd

**import** numpy **as** np

*# Modelling*

**from** sklearn.ensemble **import** RandomForestClassifier

**from** sklearn.metrics **import** accuracy\_score, confusion\_matrix, classification\_report

**from** sklearn.model\_selection **import** train\_test\_split, GridSearchCV

**import** pandas **as** pd

**from** scipy.stats **import** ttest\_ind

**def** performFeatureSelect(grp1\_name, grp2\_name, df\_group1, df\_group2):

*# Significance level (alpha)*

alpha **=** 0.05

*# Extract numeric column names*

numeric\_columns\_No **=** df\_group1**.**columns[1:]

numeric\_columns\_water **=** df\_group2**.**columns[1:]

*# Prepare result matrix*

result\_matrix **=** []

*# Perform t-test for each pair of numeric columns*

**for** col\_group1, col\_group2 **in** zip(numeric\_columns\_No, numeric\_columns\_water):

stat, p\_value **=** ttest\_ind(

df\_group1[col\_group1], df\_group2[col\_group2], nan\_policy**=**'omit')

*# Prepare result dictionary*

result\_dict **=** {

"Group\_1": grp1\_name,

"Group\_2": grp2\_name,

f"{grp1\_name}\_Group\_1": col\_group1,

f"{grp2\_name}Column\_Group\_2": col\_group2,

"T-Statistic": stat,

"P-Value": p\_value,

"Significant Difference": "Yes" **if** p\_value **<** alpha **else** "No"

}

result\_matrix**.**append(result\_dict)

*# Create DataFrame from result matrix*

result\_df **=** pd**.**DataFrame(result\_matrix)

*# Save DataFrame to CSV file*

result\_df**.**to\_csv(

f"./Step-6 Feature Ttest/Speech/{grp1\_name}\_{grp2\_name}\_ttest.csv", index**=False**)

*# preprocess test dataset*

new\_data **=** pd**.**read\_csv("sushant\_new\_water.csv")

*# new\_data = new\_data.iloc[15:-15]*

new\_data**.**reset\_index(drop**=True**, inplace**=True**)

new\_data**.**replace([np**.**inf, **-**np**.**inf], np**.**nan, inplace**=True**)

new\_data**.**dropna(axis**=**0, inplace**=True**)

new\_data **=** new\_data[**~**(new\_data **==** 0.0)**.**any(axis**=**1)]

new\_data**.**drop\_duplicates(inplace**=True**)

**del** new\_data['Delta\_AF7']

**del** new\_data['Delta\_TP10']

**del** new\_data['Alpha\_TP9']

**for** col **in** new\_data**.**select\_dtypes(include**=**['float64'])**.**columns:

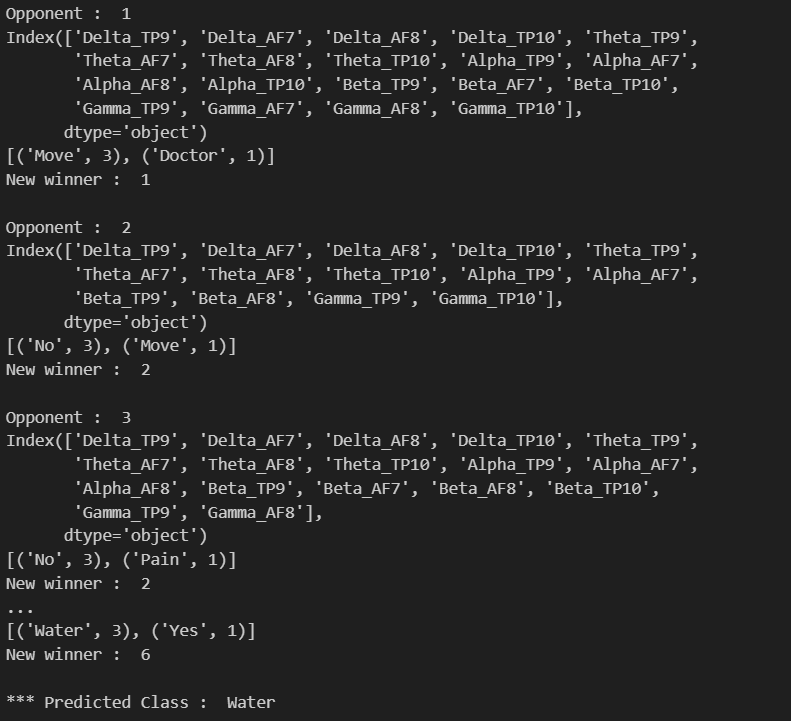
new\_data[col] **=** new\_data[col]**.**map(**lambda** x: f'{x:.6f}')

**Annexure 2:**

**Model Analysis using New datasets :**

**1.Subject 1:**

**Water:**



***Predicted correctly***

**Silent:**



***Predicted correctly***

**Reference:**

[**https://www.sciencedirect.com/science/article/pii/S1877705812022114**](https://www.sciencedirect.com/science/article/pii/S1877705812022114)

[**https://arxiv.org/pdf/2301.00948.pdf**](https://arxiv.org/pdf/2301.00948.pdf)

[**https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5150228/**](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5150228/)

[**https://www.researchgate.net/publication/331733067\_A\_Deep\_Evolutionary\_Approach\_to\_Bioinspired\_Classifier\_Optimisation\_for\_Brain-Machine\_Interaction**](https://www.researchgate.net/publication/331733067_A_Deep_Evolutionary_Approach_to_Bioinspired_Classifier_Optimisation_for_Brain-Machine_Interaction)