Record Analysis

Michael Caiola (Michael.Caiola@fda.hhs.gov) and Meijun Ye (Meijun.Ye@fda.hhs.gov)

Step by step curation of dataset subset followed by prediction by BERT transformer on the remiander of the data.

Table of Contents

Load Datasets

```
load("PreRecordDatabase.mat");
Avaneesh = CohortFiles(A);
AllFiles = CohortFiles("PreData202012221319.xlsx");
y = load("PrePaper_Tables.mat");
names = fieldnames(y);
categ = [repmat(categorical("Normal"),[height(y.(names{1})),1]);...
    repmat(categorical("Stroke"),[height(y.(names{2})),1]);...
    repmat(categorical("TBI"),[height(y.(names{3})),1])];
Original = [y.(names{1});y.(names{2});y.(names{3})];
Original.Category = categ;
try
    Lopez = load("PreLopezData.mat");
    Lopez = Lopez.T;
catch
    Lopez = GetLopezData("D:\Lopez");
```

```
end
disp("Total Dataset")
```

```
Total Dataset
```

```
NumSubjects(AllFiles);
```

```
Number of Total subjects: 15001
Number of Normal subjects: 0
Number of TBI subjects: 0
Number of Stroke subjects: 0
```

```
disp("Avaneesh Dataset")
```

```
Avaneesh Dataset
```

```
NumSubjects(Avaneesh);
```

```
Number of Total subjects: 3938
Number of Normal subjects: 1054
Number of TBI subjects: 326
Number of Stroke subjects: 488
```

```
disp("Original Dataset")
```

```
Original Dataset
```

```
disp("Number of Total subjects: " + numel(unique(Original.Subject)))
```

```
Number of Total subjects: 487
```

```
disp("Number of Normal subjects: " +
numel(unique(Original.Subject(Original.Category=="Normal"))))
```

```
Number of Normal subjects: 114
```

```
disp("Number of TBI subjects: " +
numel(unique(Original.Subject(Original.Category=="TBI"))))
```

```
Number of TBI subjects: 142
```

```
disp("Number of Stroke subjects: " +
numel(unique(Original.Subject(Original.Category=="Stroke"))))
```

```
Number of Stroke subjects: 232
```

```
disp("Lopez Dataset")
```

```
Lopez Dataset
```

```
disp("Number of Total subjects: " + numel(unique(Lopez.Subject)))
```

```
Number of Total subjects: 2329
```

```
disp("Number of Normal subjects: " +
numel(unique(Lopez.Subject(Lopez.Category=="Normal"))))
```

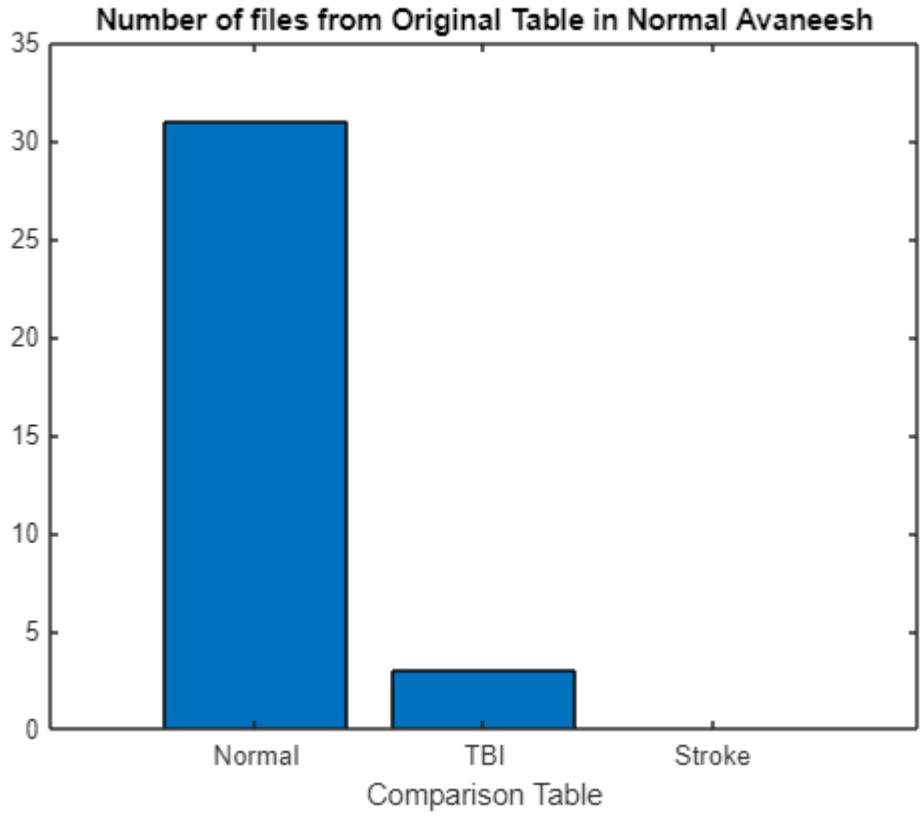
```
Number of Normal subjects: 1385
```

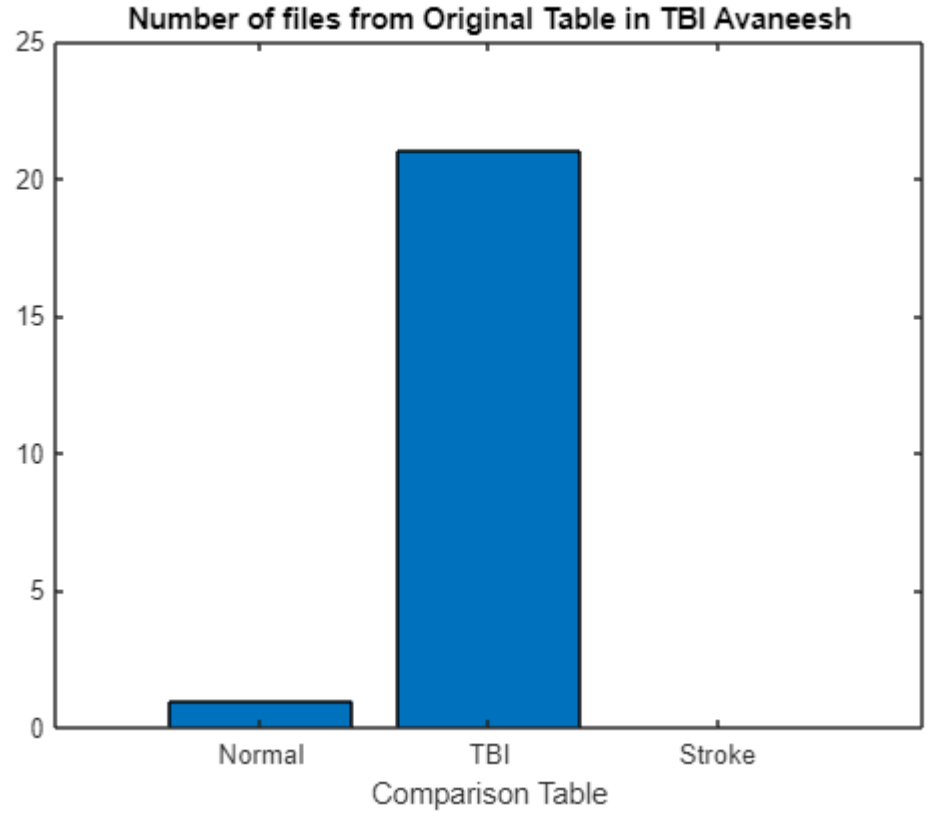
```
disp("Number of Abnormal subjects: " +
numel(unique(Lopez.Subject(Lopez.Category=="Abnormal"))))
```

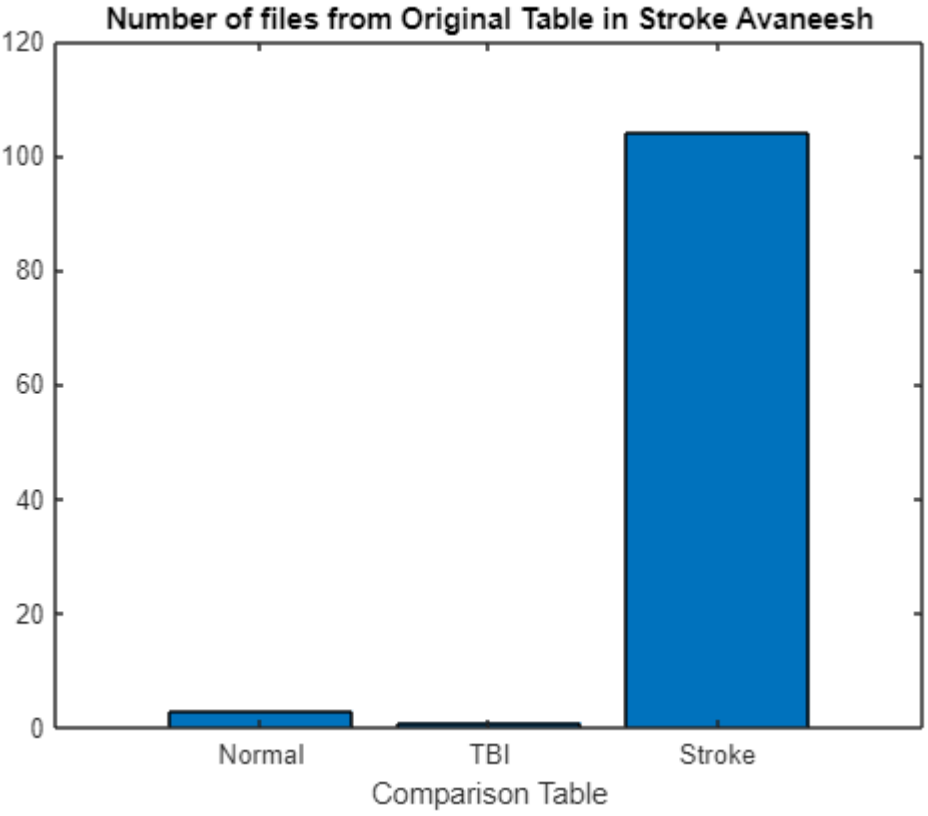
```
Number of Abnormal subjects: 998
```

Compare

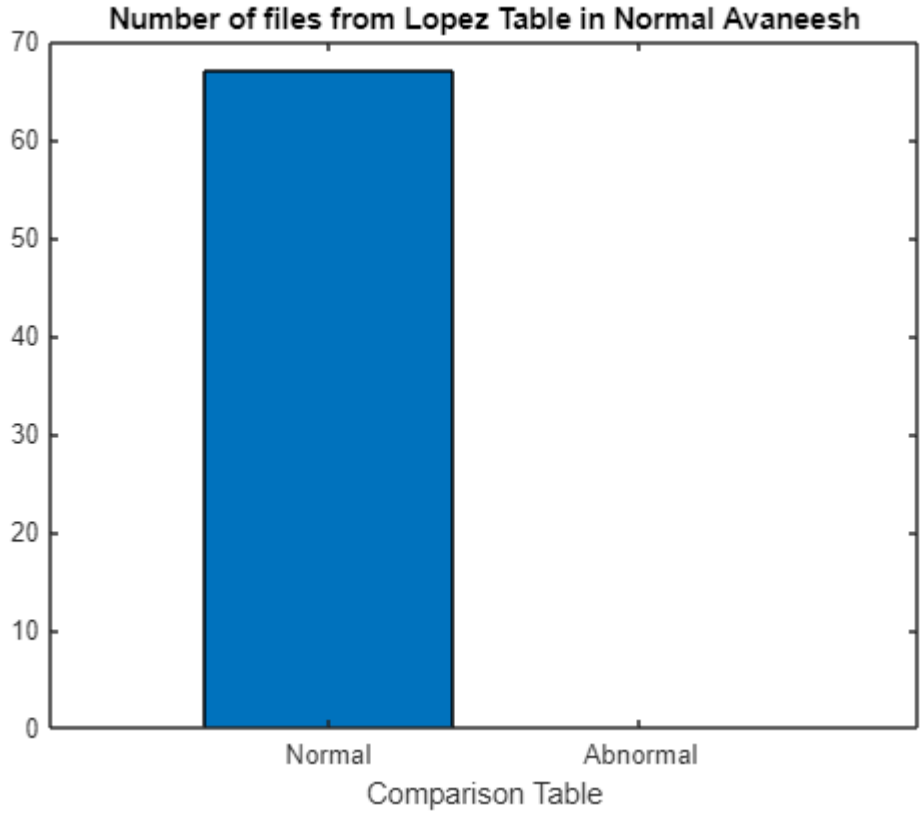
```
filesO = Compare(Avaneesh,Original,"Normal","TBI","Stroke");
```

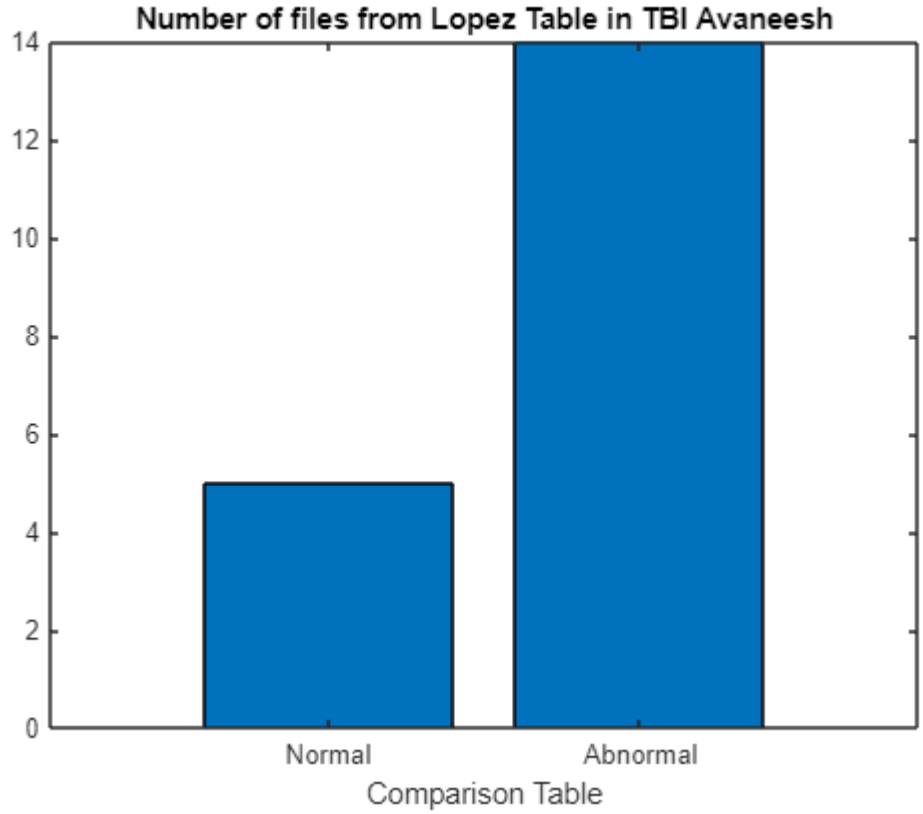


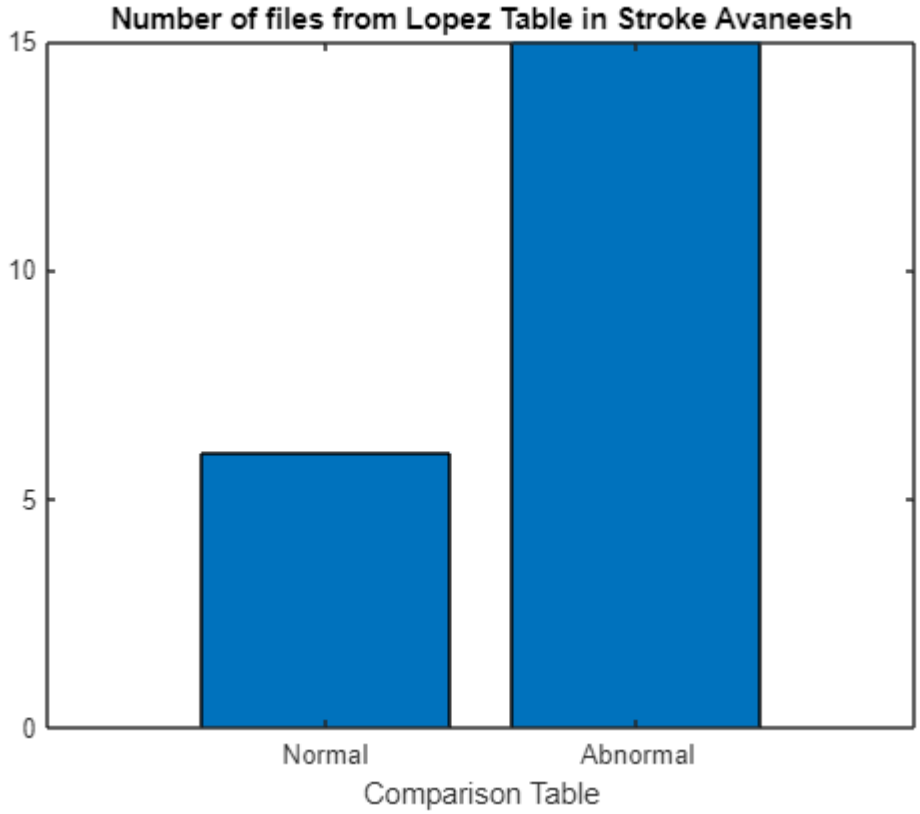





```
files1 = Compare(Avaneesh,Lopez,"Normal","Abnormal");
```


Number of files from Lopez Table in Normal Avaneesh
70
60
50
40
30
20
10
0
Normal
Abnormal
Comparison Table


Number of files from Lopez Table in TBI Avaneesh
14
12
10
8
6
4
2
0
Normal
Abnormal
Comparison Table



1. Remove Original TBI from normal
2. Remove abnormal from normal and normal from TBI/Stroke.
3. Add remaning Lopez Healthy? No
4. Other way? Yep
5. Do a subject match. Mixed Cohort - > Unknown. Cohort + unknown -> cohort? after BERT
6. Word Clouds
7. BERT

Original Data Comparison

Check out Original TBI classified as normal.

```
Data = Avaneesh;
[inT,inA] = Index(Data,Original);
disp(Data.A.("Full Note")(inA(logical(filesO{1}(:,2)))));
```

```
"CLINICAL HISTORY: This is a 44-year-old male who fell from a ladder with loss of consciousness and concussion t
"CLINICAL HISTORY: This is a 15-year-old male with leg numbness and unsteadiness for three to five weeks, migrai
"CLINICAL HISTORY: This is a 38-year-old, right-handed male with recent concussion followed by an episode of sev
```

Convert these two to TBI

```
Data.A.Category(inA(logical(filesO{1}(:,2)))) = "TBI";
```

Check out Original normal classified as TBI.

```
disp(Data.A.("Full Note")(inA(logical(filesO{2}(:,1)))));
```

```
CLINICAL HISTORY: This is a 17-year-old male with a skull defect with a gun shot wound to the head, right ear pain,
MEDICATIONS: Dilantin, others.
INTRODUCTION: Digital video EEG is performed at the bedside using standard 10-20 system of electrode placement with
DESCRIPTION OF THE RECORD: Drowsiness is characterized by slow rolling eye movements with rhythmic background theta
HR:  60 to 116 BPM.
IMPRESSION: EEG within normal limits in sleep.
CLINICAL CORRELATION: The higher amplitude POSTS on the right compared to the left may be related to the patient's p
```

```
Data.A.Category(inA(logical(filesO{2}(:,1)))) = "TBI";
```

Check out Original normal classified as Stroke.

```
disp(Data.A.("Full Note")(inA(logical(filesO{3}(:,1)))));
```

```
    "CLINICAL HISTORY: This is a 56-year-old male with congestive heart failure, CVA, and pacemaker.↵MEDICATIONS: D
    "CLINICAL HISTORY: This is a 37-year-old woman with headache versus TIA versus stroke, right-sided weakness, and
    "CLINICAL HISTORY: This is a 58-year-old woman status-post transplant with syncope, disequilibrium, and cerebell
```

These look like Stroke.

Check out Original TBI classified as Stroke.

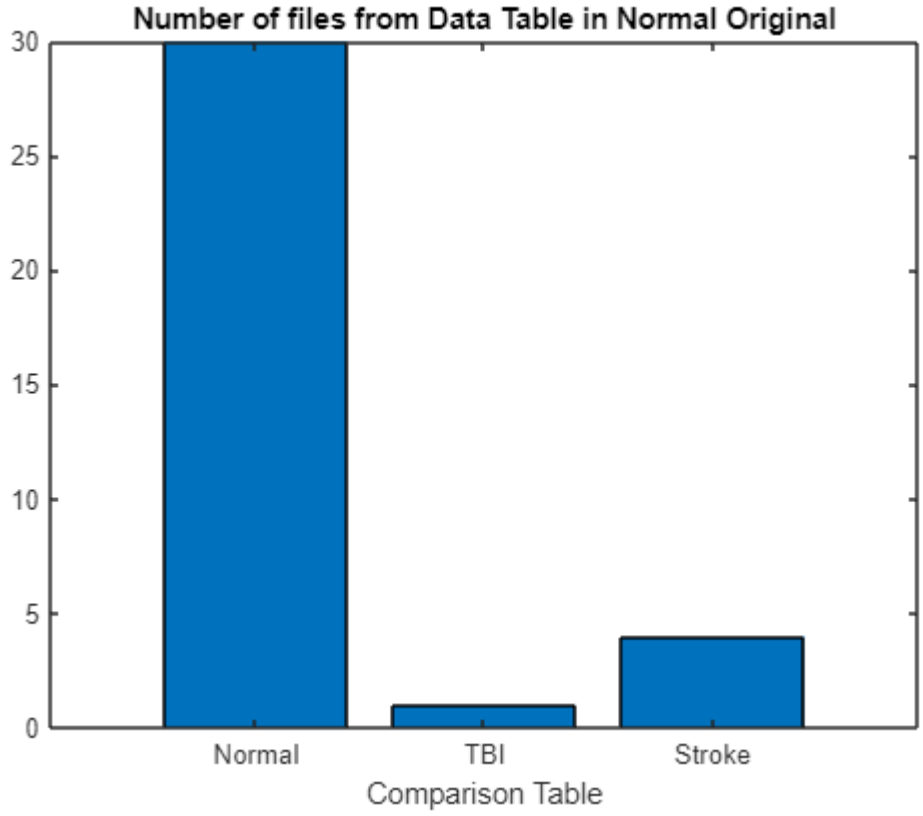
```
disp(Data.A.("Full Note")(inA(logical(filesO{3}(:,2)))));
```

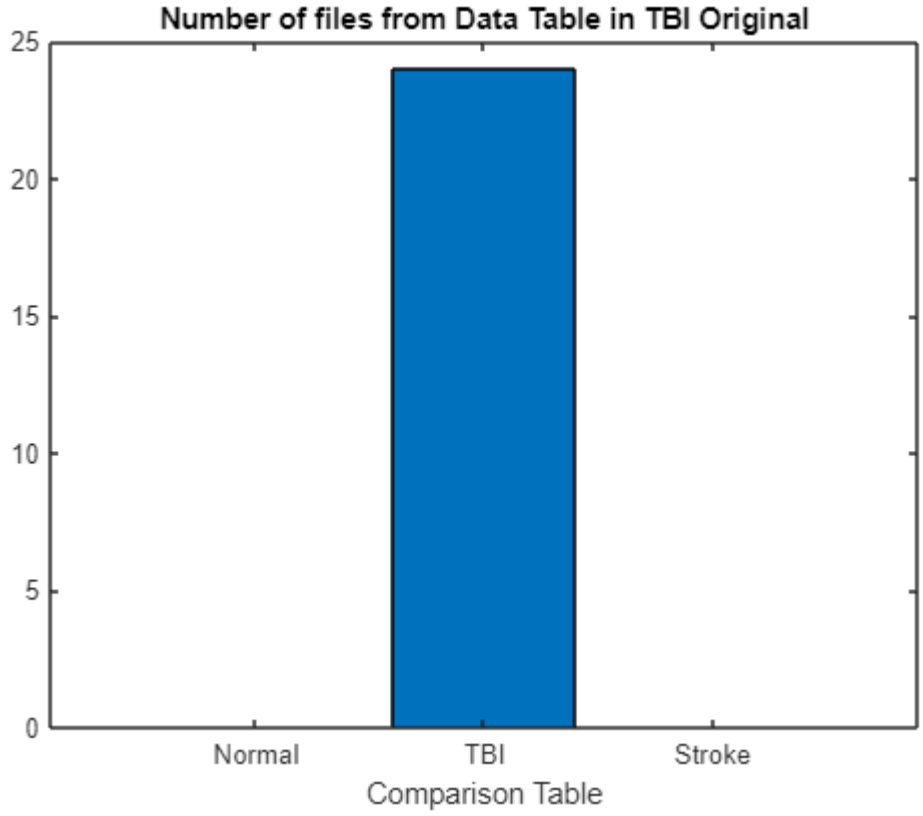
```
CLINICAL HISTORY:  53 year old left handed male with prior EEGs, presents with status in the past and hypoxic injury

MEDICATIONS:  Keppra, Ceftriaxone, Vancomycin, Zosyn, Heparin

REASON FOR STUDY:  Rule out seizures; patient having myoclonic jerks.

INTRODUCTION:  Digital video EEG was performed in lab using the standard 10-20 electrode placement system with addit

TECHNICAL DIFFICULTIES:   None

DESCRIPTION OF THE RECORD:  The record opens to a diffusely slow background with a frequency of 2 to 4 Hz and an amp

ABNORMAL DISCHARGES:  Generalized slow waves at 2 to 4 Hz and an amplitude of 20 to 30 microvolts.

SEIZURES:   None

IMPRESSION:  Abnormal EEG due to:
1. Generalized slow waves.

CLINICAL CORRELATION:  This EEG reveals evidence of diffuse cerebral dysfunction, which is nonspecific in regard to
```

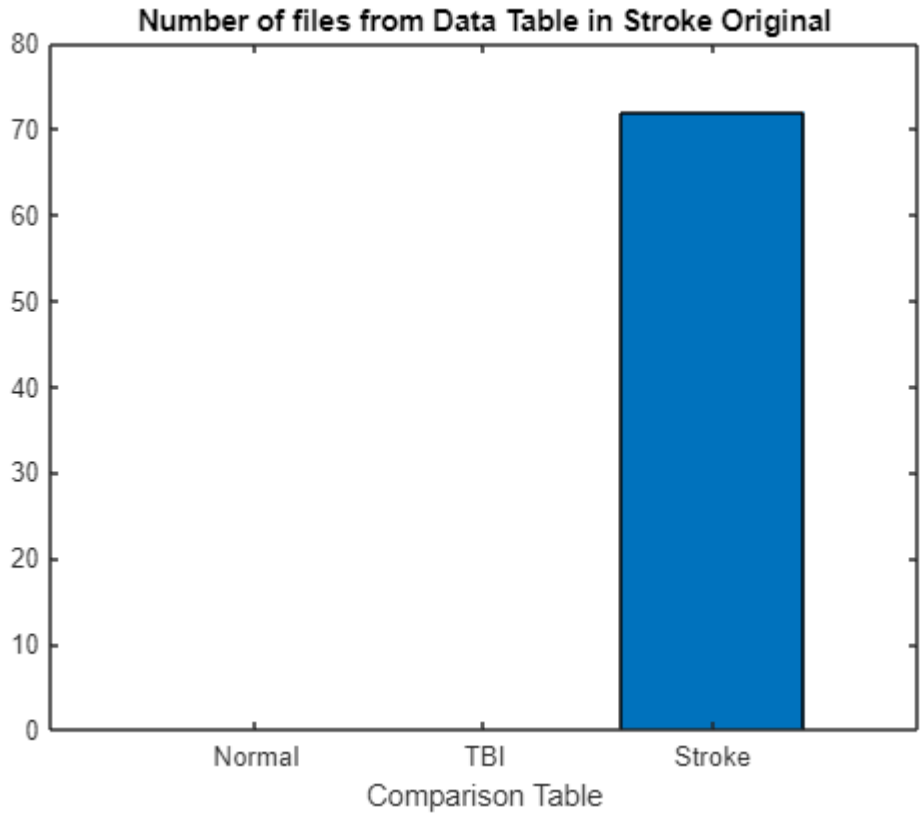
Stroke and TBI - > Unknown

```
Data.A.Category(inA(logical(filesO{3}(:,2)))) = "Unknown";
```

```
filesOr = Compare1(Data,Original,"Normal","TBI","Stroke");
```


Number of files from Data Table in Normal Original
30
25
20
15
10
5
0
Normal
TBI
Stroke
Comparison Table


Number of files from Data Table in TBI Original
25
20
15
10
5
0
Normal
TBI
Stroke
Comparison Table



Check out TBI in Normal Original.

```
disp(Data.A.("Full Note")(logical(filesOr{1}(:,2))));
```

```
CLINICAL HISTORY: This is a 17-year-old male with a skull defect with a gun shot wound to the head, right ear pain,
MEDICATIONS: Dilantin, others.
INTRODUCTION: Digital video EEG is performed at the bedside using standard 10-20 system of electrode placement with
DESCRIPTION OF THE RECORD: Drowsiness is characterized by slow rolling eye movements with rhythmic background theta
HR:  60 to 116 BPM.
IMPRESSION: EEG within normal limits in sleep.
CLINICAL CORRELATION: The higher amplitude POSTS on the right compared to the left may be related to the patient's p
```

These are fine.

Check out Stroke in Normal Original.

```
disp(Data.A.("Full Note")(logical(filesOr{1}(:,3))));
```

```
    "CLINICAL HISTORY: This is a 56-year-old male with congestive heart failure, CVA, and pacemaker.↵MEDICATIONS: D
    "CLINICAL HISTORY: This is a 37-year-old woman with headache versus TIA versus stroke, right-sided weakness, and
    "CLINICAL HISTORY: This is a 58-year-old woman status-post transplant with syncope, disequilibrium, and cerebell
    "CLINICAL HISTORY: This is a 58-year-old woman status-post transplant with syncope, disequilibrium, and cerebell
```

These are fine.

Check out Normal in TBI Original.

```
disp(Data.A.("Full Note")(logical(filesOr{2}(:,1))));
%Data.A.Category(logical(filesOr{2}(:,1))) = categorical(["TBI";"TBI"]);
```

Check out Stroke in TBI Original.

```
disp(Data.A.("Full Note")(logical(filesOr{2}(:,3))));
%Data.A.Category(logical(filesOr{2}(:,3))) = "Unknown";
```

Lopez Data Comparison

Check out Lopez Normal classified as TBI.

Lopez, S. (2017). Automated Identification of Abnormal EEGs. Temple University.

```
[inT,inA] = Index(Data,Lopez);
disp(Data.A.("Full Note")(inA(logical(files1{2}(:,1)))));
```

```
"↵CLINICAL HISTORY: →80 year old woman with recent trauma status post fall due to unsteady gait. Had episode of
"CLINICAL HISTORY:  51 year old right handed woman with syncope and headaches, past history of closed head injur
"LENGTH OF THE RECORDING:  24 minutes.↵ACTIVATION PROCEDURES:  Hyperventilation and photic stimulation.↵CONDIT
"CLINICAL HISTORY: 54 year old right handed man with seizures and history of traumatic brain injury, diabetes ar
"CLINICAL HISTORY:  54 year old right handed man with history of epilepsy and prior traumatic head injury.  He i
```

```
Data.A.Category(inA(logical(files1{2}(:,1)))) =
categorical(["Unknown";"TBI";"TBI";"Unknown";"Unknown"]) ;
```

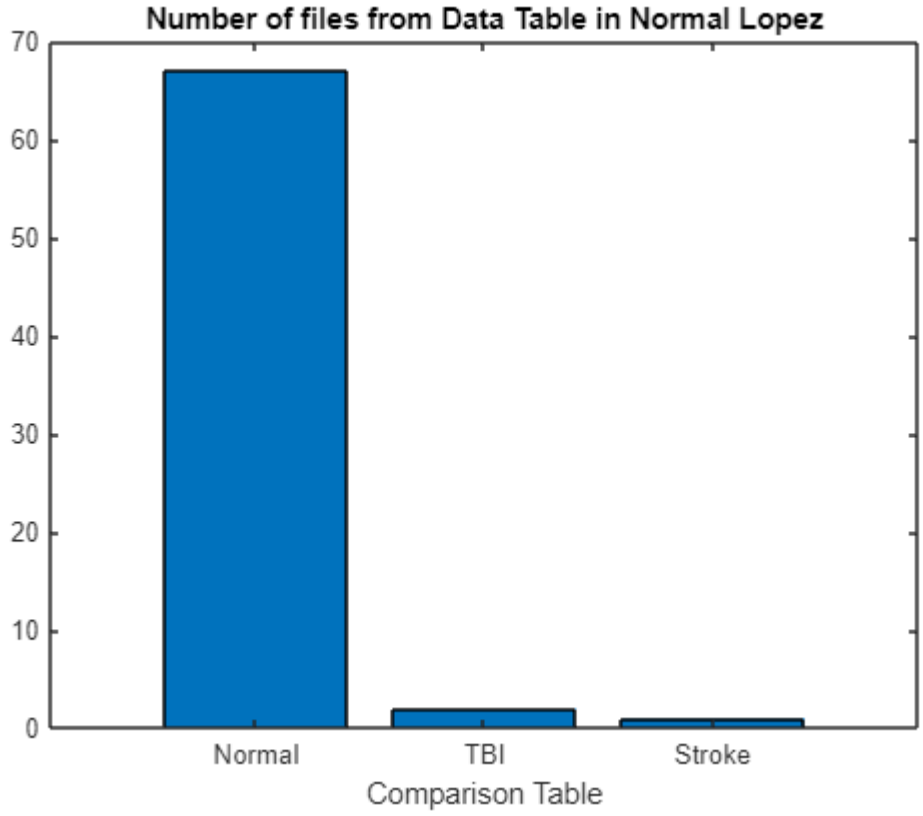
Check out Lopez Normal classified as Stroke.

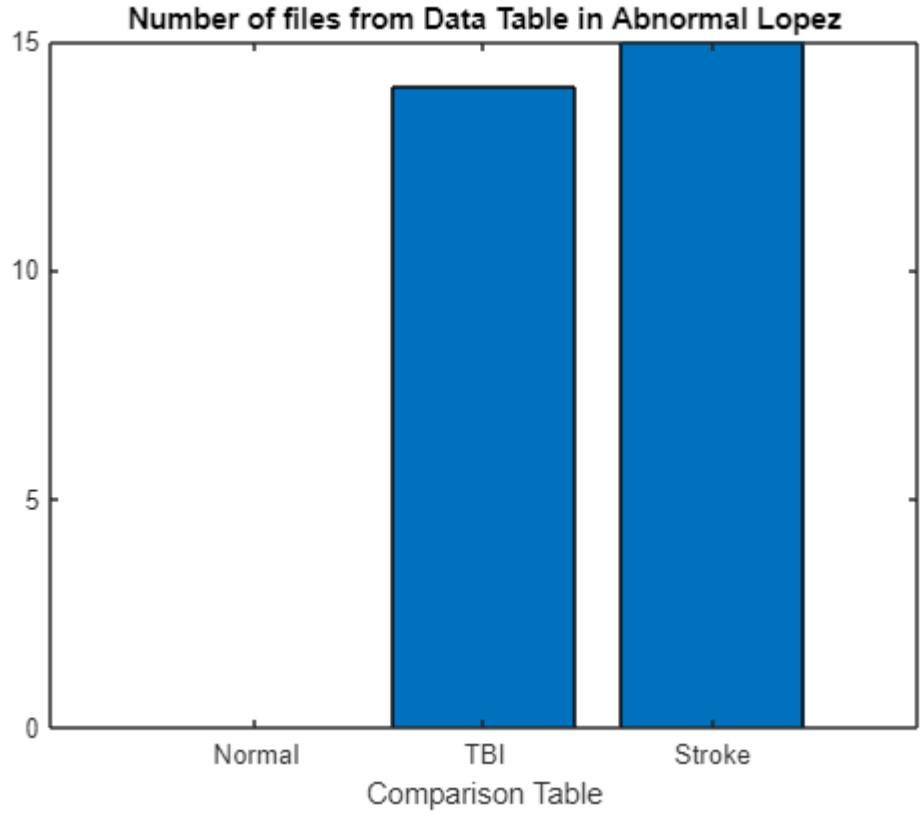
```
disp(Data.A.("Full Note")(inA(logical(files1{3}(:,1)))));
```

```
"CLINICAL HISTORY:  78 year old right handed female who has a history of strokes in↵2007 with subsequent seizur
"CLINICAL HISTORY: 62 year old woman with a stroke and intermittent episodes of left facial spasms and twitching
"REASON FOR STUDY:  History of seizures.↵CLINICAL HISTORY:  A 54-year-old woman with a history of seizures, hac
"↵CLINICAL HISTORY: 62 year old right handed male with epilepsy, stroke and MVA.↵MEDICATIONS: Keppra, Diovan,
"CLINICAL HISTORY: 32 year old right handed female with a history of multiple strokes and a seizure presents wit
"CLINICAL HISTORY:  57 year old right handed woman with a history of schizophrenia, memory disorder, hypertensio
```

```
Data.A.Category(inA(logical(files1{3}(:,1)))) =
categorical(["Unknown";"Stroke";"Unknown";"Unknown";"Unknown";"Unknown"]) ;
```

```
files1r = Compare1(Data,Lopez,"Normal","Abnormal");
```





Check out TBI in Normal Lopez.

```
disp(Data.A.("Full Note")(logical(files1r{1}(:,2))));
```

```
    "CLINICAL HISTORY:  51 year old right handed woman with syncope and headaches, past history of closed head injur
    "LENGTH OF THE RECORDING:  24 minutes.↵ACTIVATION PROCEDURES:  Hyperventilation and photic stimulation.↵CONDIT
```

```
Data.A.Category(logical(files1r{1}(:,2))) = categorical(["TBI";"TBI"]);
```

Check out Stroke in Normal Lopez.

```
disp(Data.A.("Full Note")(logical(files1r{1}(:,3))));
```

```
CLINICAL HISTORY: 62 year old woman with a stroke and intermittent episodes of left facial spasms and twitching. Pas
MEDICATIONS: Aricept, Plavix, Metoprolol, Pravachol
INTRODUCTION: Digital video EEG was performed in lab using standard 10-20 system of electrode placement with 1 chann
DESCRIPTION OF THE RECORD: In wakefulness, there is a 10.5 Hz posterior dominant rhythm of 30 mcv with a generous am
HR: 90 bpm

IMPRESSION: EEG within normal limits.
CLINICAL CORRELATION: No significant focal features nor epileptiform features are observed. There is a single burst
```

```
Data.A.Category(logical(files1r{1}(:,3))) = categorical(["Stroke"]);
```

Remove subjects with multiple classes

```
in = unique(Data.A.Subject,'rows','stable');
for i = 1: length(in)
    in2 = Data.A.Category(Data.A.Subject == in(i));
    h = histcounts(in2);
    if logical(h(2:end)>1)
        Data.A.Category(Data.A.Subject == in(i)) = "Unknown";
    end
end
```

New Database

```
NumSubjects(Avaneesh);
```

```
Number of Total subjects: 3938
Number of Normal subjects: 1054
Number of TBI subjects: 326
Number of Stroke subjects: 488
```

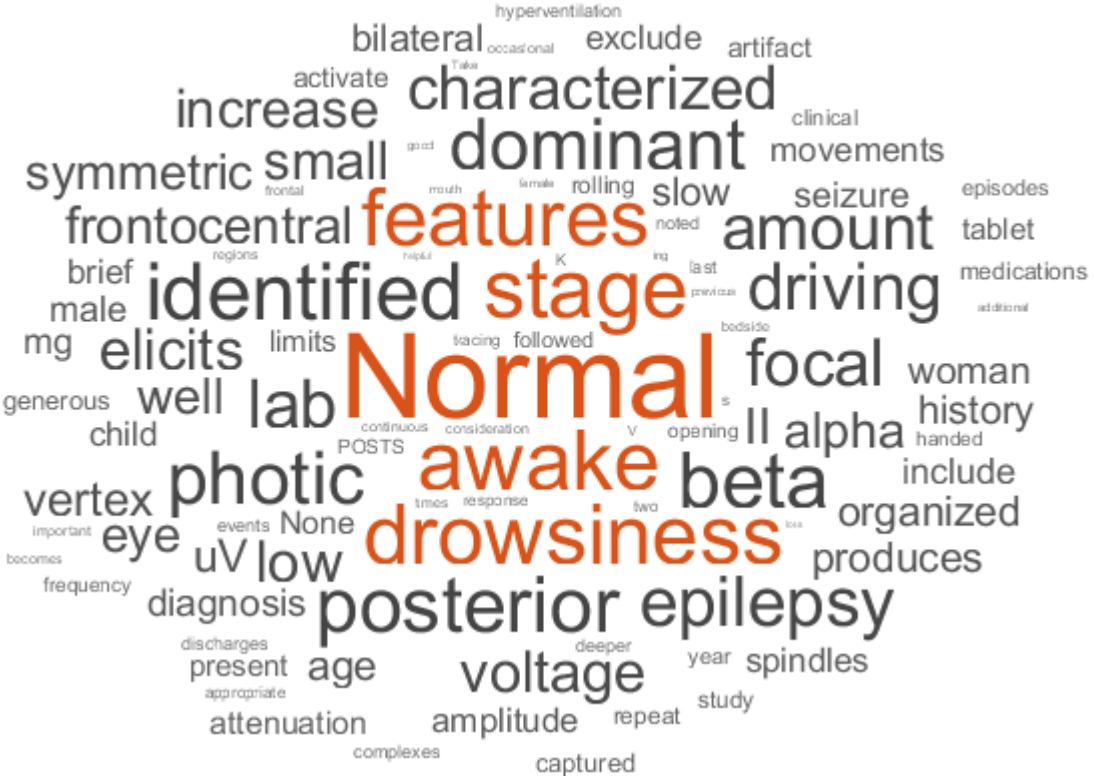
```
NumSubjects(Data);
```

```
Number of Total subjects: 3938
Number of Normal subjects: 1051
Number of TBI subjects: 328
Number of Stroke subjects: 487
```

Word Clouds

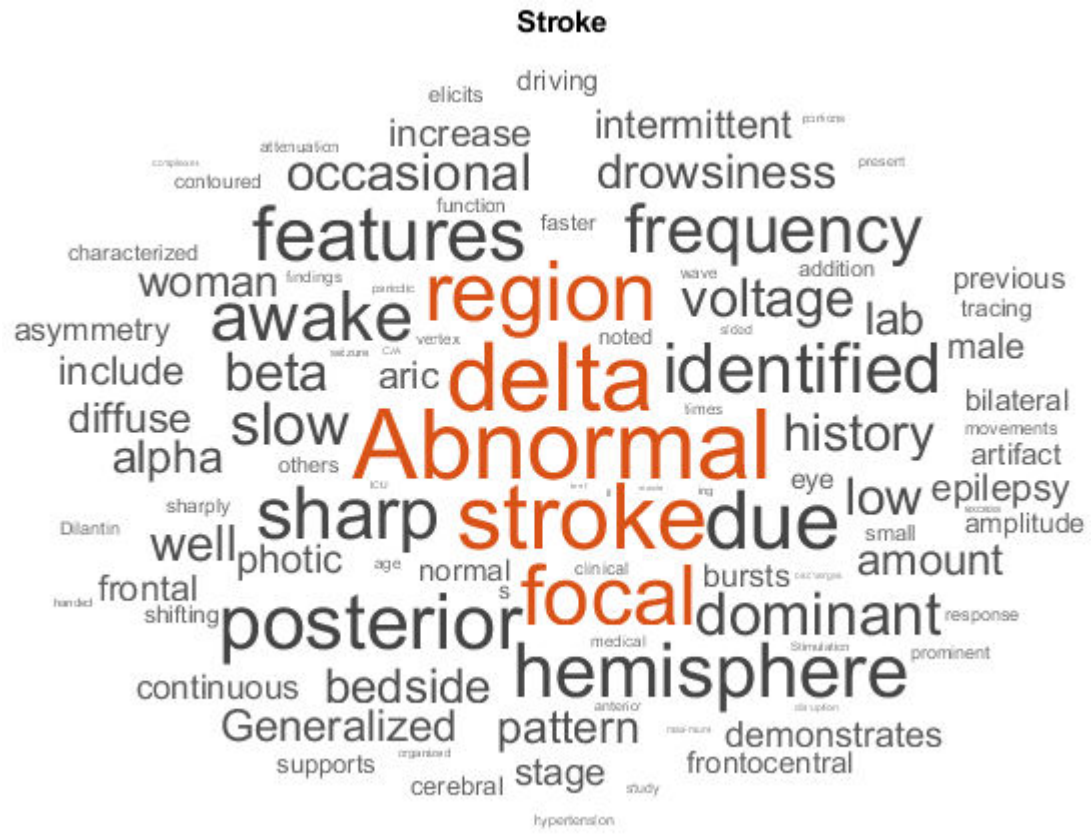
```
PlotWordClouds(Data,36);
```

Normal


hyperventilation
bilateral
occasional
exclude
artifact
activate
characterized
increase
clinical
dominant
movements
symmetric
small
rolling
slow
seizure
episodes
frontocentral
features
noted
amount
tablet
brief
identified
stage
last
driving
medications
male
mg
elicits
limits
tracing
followed
Normal
focal
woman
generous
well
lab
history
child
continuous
consideration
opening
ll
alpha
handed
POSTS
awake
include
vertex
photic
beta
organized
eye
events
None
drowsiness
produces
uV
low
frequency
diagnosis
posterior
epilepsy
discharges
deeper
present
age
voltage
year
spindles
appropriate
study
attenuation
amplitude
repeat
complexes
captured

TBI


produces
generalized
male
dominant
bedside
characterized
elicits
trauma
demonstrates
movements
frequency
features
increase
vertex
organized
sharp
Abnormal
slow
lab
Dilantin
bursts
excess
delta
voltage
amount
due
focal
addition
cerebral
wave
diffuse
seizure
well
include
region
awake
stage
previous
small
beta
head
artifact
occasional
frontal
posterior
photic
continuous
tracing
eye
central
age
frontocentral
time
normal
identified
epilepsy
medications
pattern
low
drowsiness
alpha
driving
study
bilateral
history
amplitude
followed
symmetric
hemisphere



BERT

```
path = pwd;
bertfol = uigetdir([],"BERT Directory"); %Replace with directory to MATLAB
Transformer models
% Can be found at https://github.com/matlab-deep-learning/transformer-models
cd(bertfol)
mdl = bert
```

```
Downloading bert.RIGHTS to: C:\Users\nonun\OneDrive\Documents\MATLAB\Examples\R2022a\supportfiles\nnet\data\networks
Downloading parameters.mat to: C:\Users\nonun\OneDrive\Documents\MATLAB\Examples\R2022a\supportfiles\nnet\data\netwo
Downloading vocab.txt to: C:\Users\nonun\OneDrive\Documents\MATLAB\Examples\R2022a\supportfiles\nnet\data\networks\b
mdl = struct with fields:
     Tokenizer: [1×1 bert.tokenizer.BERTTokenizer]
    Parameters: [1×1 struct]
```

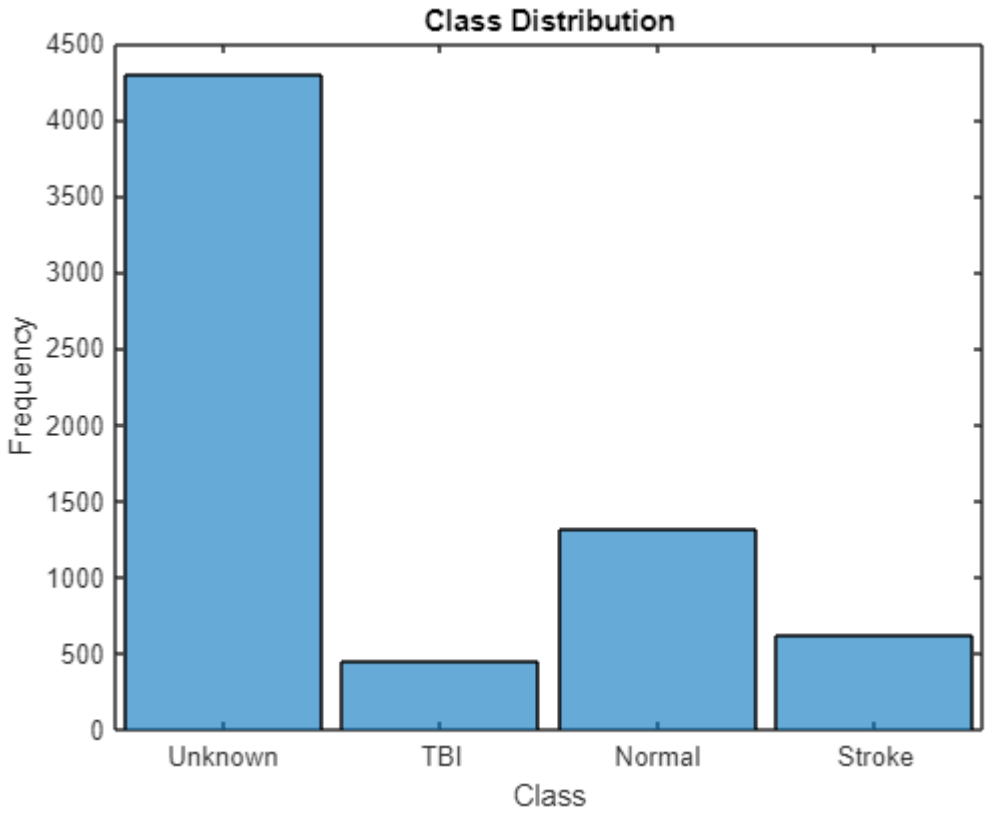
View the BERT model tokenizer. The tokenizer encodes text as sequences of integers and holds the details of padding, start, separator and mask tokens.

```
tokenizer = mdl.Tokenizer
```

```
tokenizer =
  BERTTokenizer with properties:

      PaddingToken: "[PAD]"
        StartToken: "[CLS]"
    SeparatorToken: "[SEP]"
         MaskToken: "[MASK]"
```

```
FullTokenizer: [1×1 bert.tokenizer.internal.FullTokenizer]
  PaddingCode: 1
SeparatorCode: 103
    StartCode: 102
     MaskCode: 104
```

```
figure
histogram(Data.A.Category);
xlabel("Class")
ylabel("Frequency")
title("Class Distribution")
```



Encode the text data using the BERT model tokenizer using the `encode` function and add the tokens to the training data table.

```
data = Data.A;
data.Tokens = encode(tokenizer, data.("Full Note"));
```

```
classes = categories(data.Category);
numClasses = numel(classes)
```

```
numClasses = 4
```

Balance large unknown group

```
cd(path);
smallergroup = ceil(sum(data.Category~="Unknown")/(numClasses-1)); %make average
size
in = data.Category== "Unknown";
in = find(in);
in = in(randperm(length(in)));
in = in(1:smallergroup);
data = [data(in,:);data(data.Category~="Unknown",:)];
figure
histogram(data.Category);
xlabel("Class")
ylabel("Frequency")
title("Class Distribution")
```

The next step is to partition it into sets for training and validation. Partition the data into a training partition and a held-out partition for validation and testing. Specify the holdout percentage to be 20%.

```
cvp = cvpartition(data.Category,"Holdout",0.2);
dataTrain = data(training(cvp),:);
dataValidation = data(test(cvp),:);
```

View the number of training and validation observations.

```
numObservationsTrain = size(dataTrain,1)
```

```
numObservationsTrain = 2556
```

```
numObservationsValidation = size(dataValidation,1)
```

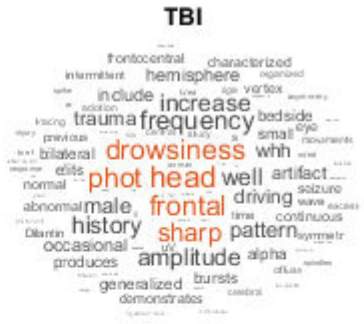
```
numObservationsValidation = 639
```

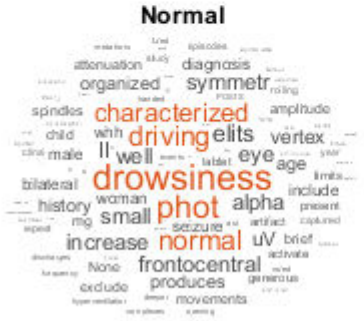
Extract the text data, labels, and encoded BERT tokens from the partitioned tables.

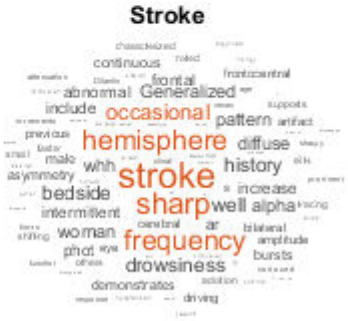
```
textDataTrain = dataTrain.("Full Note");
textDataValidation = dataValidation.("Full Note");

TTrain = dataTrain.Category;
TValidation = dataValidation.Category;

tokensTrain = dataTrain.Tokens;
tokensValidation = dataValidation.Tokens;
```

To check that you have imported the data correctly, visualize the training text data using a word cloud.

```
c = categories(data.Category);
pat = regexpPattern('[A-Z\s]*[A-Z]+:');
x = data.("Full Note");
x = erase(x,pat); % Remove capital headings
wc = wordCloudCounts(x);
wc = wc{1:50,1};
x = erase(x,wc);
x = erase(x,"ic");
x = replace(x,regexpPattern('\s[a-z][\s\*\)\.\,]')," ");
x = replace(x,"  "," ");
wc = wordCloudCounts(x);
figure
for i = 1: 4
    nexttile();
    in = data.Category == c(i);
    y = x(in);
    wordcloud(y)
    title(c(i))
end
```







View the BERT token codes of the first few training documents.

```
tokensTrain{1:5}
```

```
ans = 1×211
         102        6613        2382        1025        2024        2004 ···
ans = 1×297
         102        6613        2382        1025        2024        2004 ···
ans = 1×335
         102        6613        2382        1025        2024        2004 ···
ans = 1×316
         102        6613        2382        1025        4230        2096 ···
ans = 1×237
         102        6613        2382        1025        2024        2004 ···
```

Prepare Data for Training

Convert the documents to feature vectors using the BERT model as a feature extractor.

```
% To extract the features of the training data by iterating over
% mini-batches, create a |minibatchqueue| object.

% Mini-batch queues require a single datastore that outputs both the
% predictors and responses. Create array datastores containing the training
% BERT tokens and labels and combine them using the |combine| function.
dsXTrain = arrayDatastore(tokensTrain,"OutputType","same");
dsTTrain = arrayDatastore(TTrain);
cdsTrain = combine(dsXTrain,dsTTrain);
```

```
% Create a combined datastore for the validation data using the same steps.
dsXValidation = arrayDatastore(tokensValidation,"OutputType","same");
dsTValidation = arrayDatastore(TValidation);
cdsValidation = combine(dsXValidation,dsTValidation);
```

Create a mini-batch queue for the training data. Specify a mini-batch size of 32 and preprocess the mini-batches using the `preprocessPredictors` function, listed at the end of the example.

```
miniBatchSize = 32;
paddingValue = mdl.Tokenizer.PaddingCode;
maxSequenceLength = mdl.Parameters.Hyperparameters.NumContext;

mbqTrain = minibatchqueue(cdsTrain,1,...
    "MiniBatchSize",miniBatchSize, ...
    "MiniBatchFcn",@(X) preprocessPredictors(X,paddingValue,maxSequenceLength));
```

Create a mini-batch queue for the validation data using the same steps.

```
mbqValidation = minibatchqueue(cdsValidation,1,...
    "MiniBatchSize",miniBatchSize, ...
    "MiniBatchFcn",@(X) preprocessPredictors(X,paddingValue,maxSequenceLength));
```

To speed up feature extraction. Convert the BERT model weights to gpuArray if a GPU is available.

```
if canUseGPU
    mdl.Parameters.Weights = dlupdate(@gpuArray,mdl.Parameters.Weights);
end
```

Convert the training sequences of BERT model tokens to a `N`-by-|embeddingDimension| array of feature vectors, where `N` is the number of training observations and `embeddingDimension` is the dimension of the BERT embedding.

```
cd(bertfol)
featuresTrain = [];
reset(mbqTrain);
while hasdata(mbqTrain)
    X = next(mbqTrain);
    features = bertEmbed(X,mdl.Parameters);
    featuresTrain = [featuresTrain gather(extractdata(features))];
end
```

Transpose the training data to have size `N`-by-|embeddingDimension|.

```
featuresTrain = featuresTrain.';
```

Convert the validation data to feature vectors using the same steps.

```
featuresValidation = [];

reset(mbqValidation);
while hasdata(mbqValidation)
    X = next(mbqValidation);
    features = bertEmbed(X,mdl.Parameters);
    featuresValidation = cat(2,featuresValidation,gather(extractdata(features)));
end
featuresValidation = featuresValidation.';
cd(path);
```

Define Deep Learning Network

Define a deep learning network that classifies the feature vectors.

```
numFeatures = mdl.Parameters.Hyperparameters.HiddenSize;
layers = [
    featureInputLayer(numFeatures)
    dropoutLayer(.05)
    %fullyConnectedLayer(100)
    reluLayer()
    dropoutLayer(.2)
    fullyConnectedLayer(4)
    softmaxLayer
    classificationLayer];
```

Specify Training Options

Specify the training options using the `trainingOptions` function. * Train with a mini-batch size of 64. * Shuffle the data every epoch. * Validate the network using the validation data. * Display the training progress in a plot and suppress the verbose output.

```
opts = trainingOptions('adam',...
    "MiniBatchSize",64,...
    "ValidationData",{featuresValidation,dataValidation.Category},...
    "Shuffle","every-epoch", ...
    "Plots","training-progress","MaxEpochs",400,
"InitialLearnRate",.01,"LearnRateSchedule","piecewise","LearnRateDropPeriod",100,...
    "Verbose",0);
```

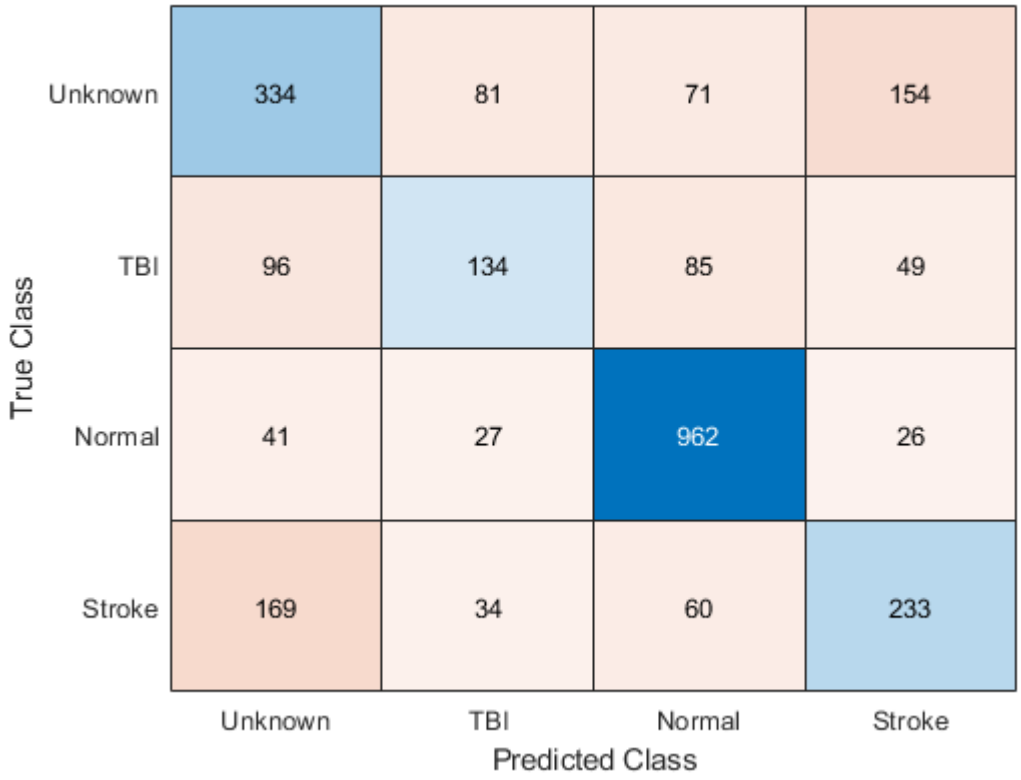
```
classificationSVM = fitcecoc(...
    featuresTrain, ...
    dataTrain.Category);
kfoldmodel = crossval(classificationSVM, 'KFold', 5);
predLabels = kfoldPredict(kfoldmodel);
loss = kfoldLoss(kfoldmodel)*100;
fprintf('Loss is %2.2f percent\n',loss);
```

```
Loss is 34.94 percent
```

```
accuracy = 100-loss;
fprintf('Accuracy is %2.2f percent\n',accuracy);
```

```
Accuracy is 65.06 percent
```

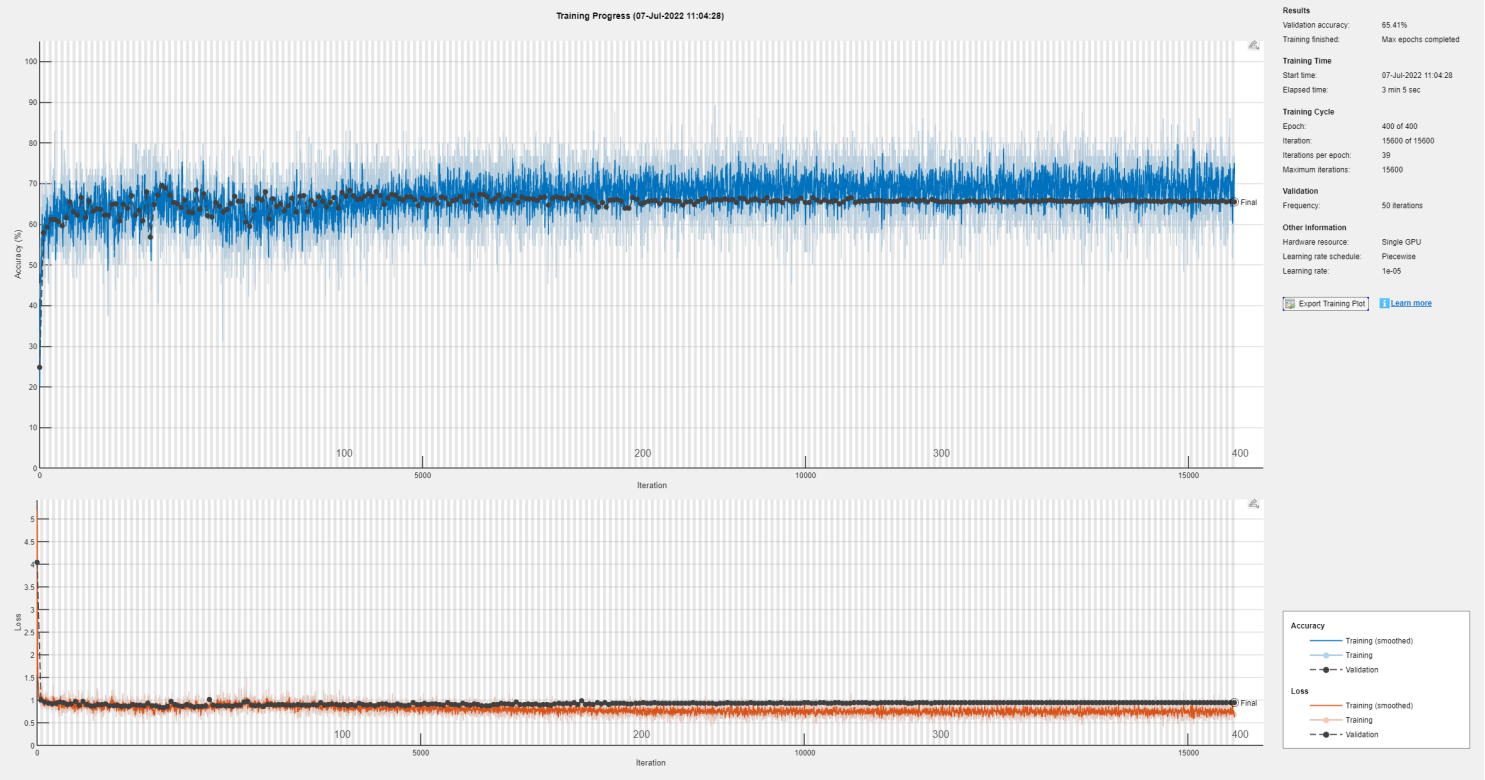
```
figure;
confmatCV = confusionchart(dataTrain.Category,predLabels);
```



Train Network

Train the network using the `trainNetwork` function.

```
%net = trainNetwork(featuresTrain,dataTrain.Category,layers,opts);
```

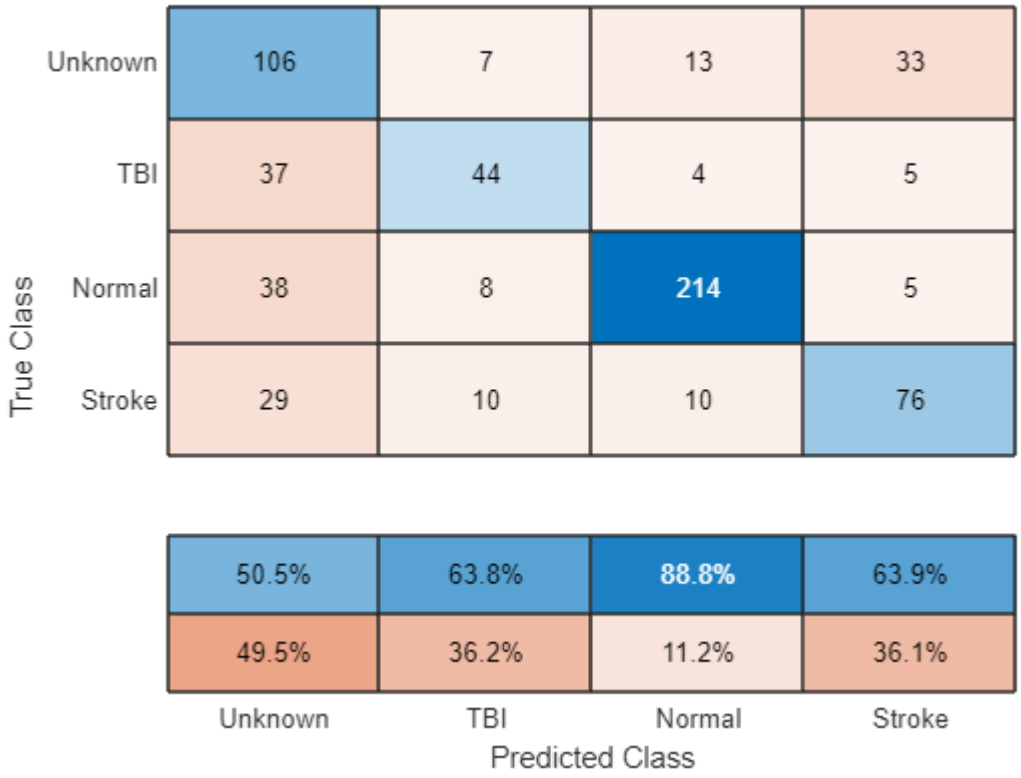



```
load("Pretextnet.mat")
```

Test Network

Make predictions using the validation data and display the results in a confusion matrix.

```
[YPredValidation,score] = classify(net,featuresValidation);
figure
cm = confusionchart(TValidation,YPredValidation);
cm.ColumnSummary = 'column-normalized';
```



Calculate the validation accuracy.

```
accuracy = mean(dataValidation.Category == YPredValidation)
```

```
accuracy = 0.6886
```

Calculate calidation accuracy

```
precision = (sum(diag(cm.NormalizedValues))-cm.NormalizedValues(1,1))/
sum(cm.NormalizedValues(:,2:end),'all')
```

```
precision = 0.7786
```

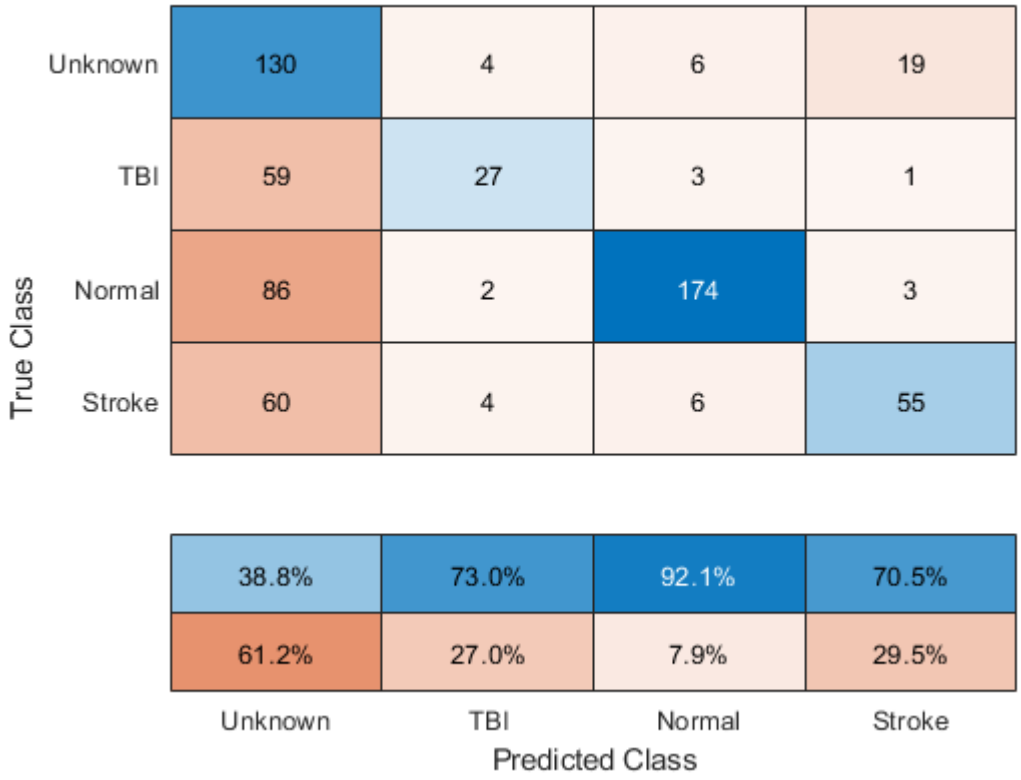
```
meanprecision = sum(([0, ones(1,4-1)] .* diag(cm.NormalizedValues)')./[1
sum(cm.NormalizedValues(:,2:end))])/(4 - 1)
```

```
meanprecision = 0.7214
```

Using confident answers

Taking only confident answers (example: score >.7) or unknown

```
V
= table(YPredValidation,score(:,1),score(:,2),score(:,3),score(:,4),'VariableNames',
{'Prediction',classes{1},classes{2},classes{3},classes{4}});
V = [dataValidation(:,4) V];
[p, pm] = precisionChart(.9,score,classes,V);
```




```
disp("Precision is " + p)
```

```
Precision is 0.84211
```

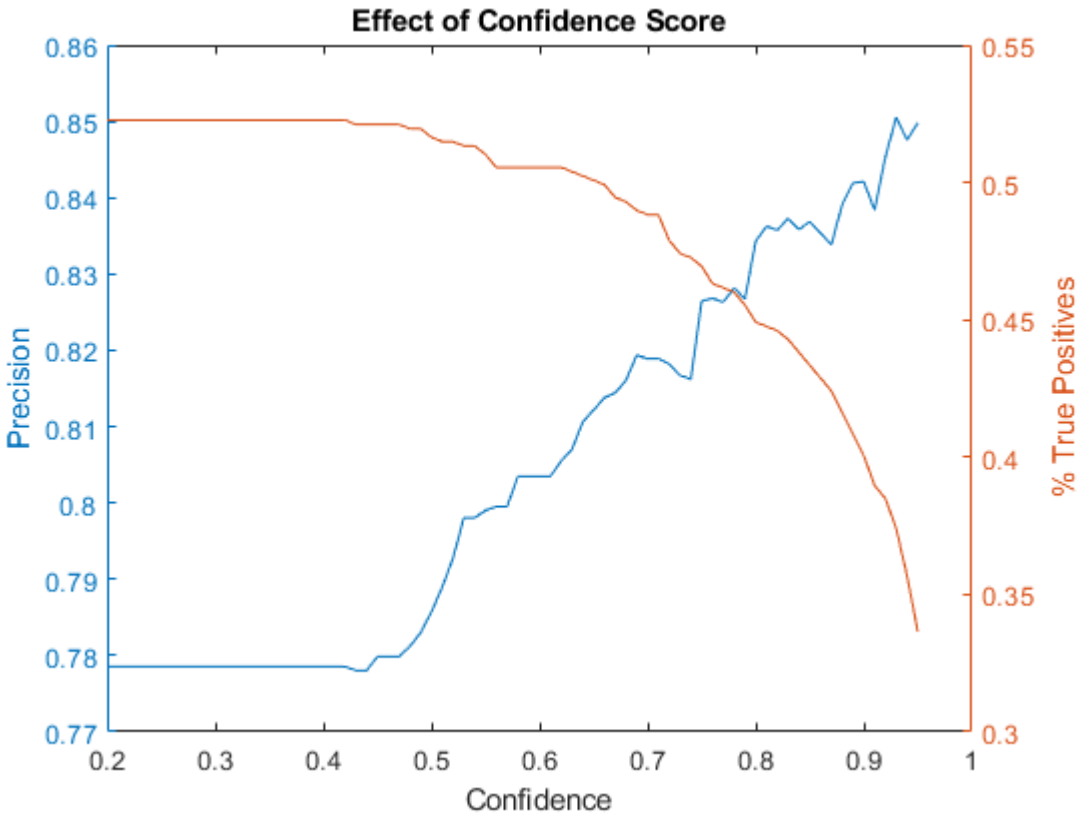
```
disp("Mean precision is " + pm)
```

```
Mean precision is 0.78516
```

What is the relation of the confidence to precision and correctly labled predictions?

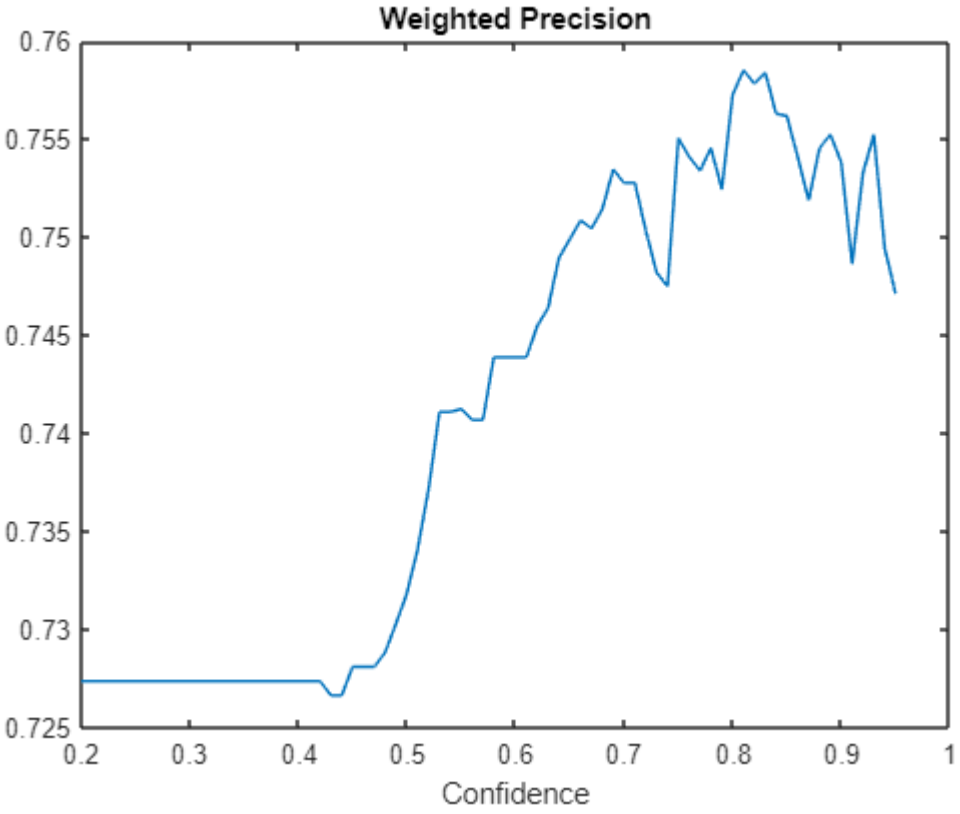
```
p = [];
r = [];
k = 1;
ran = .2:.01:.95;
for i = ran
    confidx = i;
    V
= table(YPredValidation,score(:,1),score(:,2),score(:,3),score(:,4),'VariableNames',
{'Prediction',classes{1},classes{2},classes{3},classes{4}});
    V = [dataValidation(:,4) V];
    [max_score, pred_in] = max(score,[],2);
    in = max_score > confidx;
    pred_in(~in) = find(string(classes) == "Unknown");
    V.Confident = categorical(string(classes(pred_in)));
    figure
    cm = confusionmat(V.Category,V.Confident);
    p(k) = (sum(diag(cm))-cm(1,1))/sum(cm(:,2:end),'all');
    r(k) = (sum(diag(cm))-cm(1,1))/sum(cm,'all');
```

```
    k = k+1;
end
figure;
yyaxis left
plot(ran,p)
ylabel("Precision")
hold on
yyaxis right
plot(ran,r)
xlabel("Confidence")
ylabel("% True Positives")
title("Effect of Confidence Score")
```



With a large enough database we can value higher precsion over loss of datapoints.

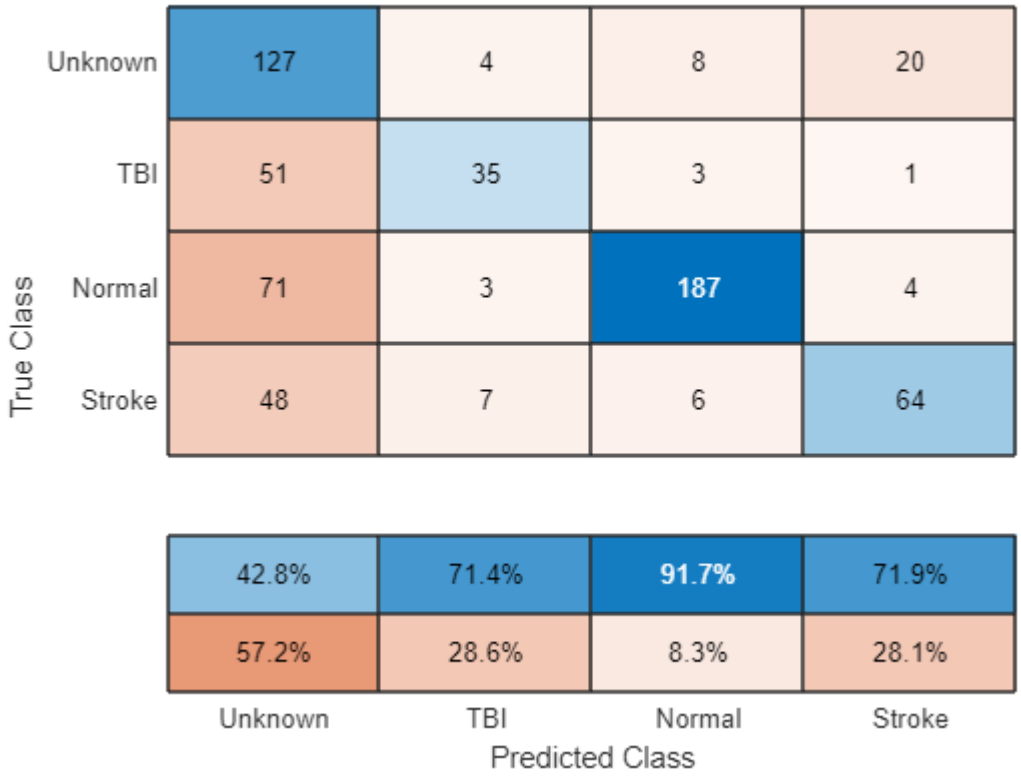
```
w = 4; % weighted term
x = (w*p + r)/(w+1);
figure
plot(ran, x)
xlabel("Confidence")
title("Weighted Precision")
```




```
maxPrescison = ran(x == max(x));
disp("Use a confidence score of " + maxPrescison + " to maximize the weighted
precision score")
```

```
Use a confidence score of 0.81 to maximize the weighted precision score
```

```
[p, pm] = precisionChart(ran(x == max(x)),score,classes,V);
```




```
disp("Precision is " + p)
```

```
Precision is 0.83626
```

Predict New Data

Find Data in AllFiles and remove it. Then predict NewFiles category and put back together.

```
ind = [];
for i = 1:height(Data.A)
    ind = [ind; find(and(AllFiles.A.Subject==Data.A.Subject(i),
AllFiles.A.Session==Data.A.Session(i)))];
end
newin = zeros(1,height(AllFiles.A));
newin(ind) = 1;
newin = logical(newin);
```

```
newtext = string(AllFiles.A.("Full Note")(~newin));
```

Tokenize the text data using the same steps as the training documents.

```
cd(bertfol)
tokensNew = encode(tokenizer,newtext);
```

Create minibatch

```
dsXNew = arrayDatastore(tokensNew,"OutputType","same");

mbqNew = minibatchqueue(dsXNew,1,...
    "MiniBatchSize",miniBatchSize, ...
    "MiniBatchFcn",@(X) preprocessPredictors(X,paddingValue,maxSequenceLength));
```

Make new predictions

```
pred = [];
score = [];
reset(mbqNew);
while hasdata(mbqNew)
    X = next(mbqNew);
    y = bertEmbed(X,mdl.Parameters)';
    y = gather(extractdata(y));
    [a,b] = classify(net,y);
    pred = [pred; a];
    score = [score; max(b,[],2)];
end
cd(path)
```

Add confident predictions

```
pin = score > maxPrescison;
disp("Number of Normal files predicted: " + sum(pred(pin)=="Normal"));
```

```
Number of Normal files predicted: 2907
```

```
disp("Number of TBI files predicted: " + sum(pred(pin)=="TBI"));
```

```
Number of TBI files predicted: 1166
```

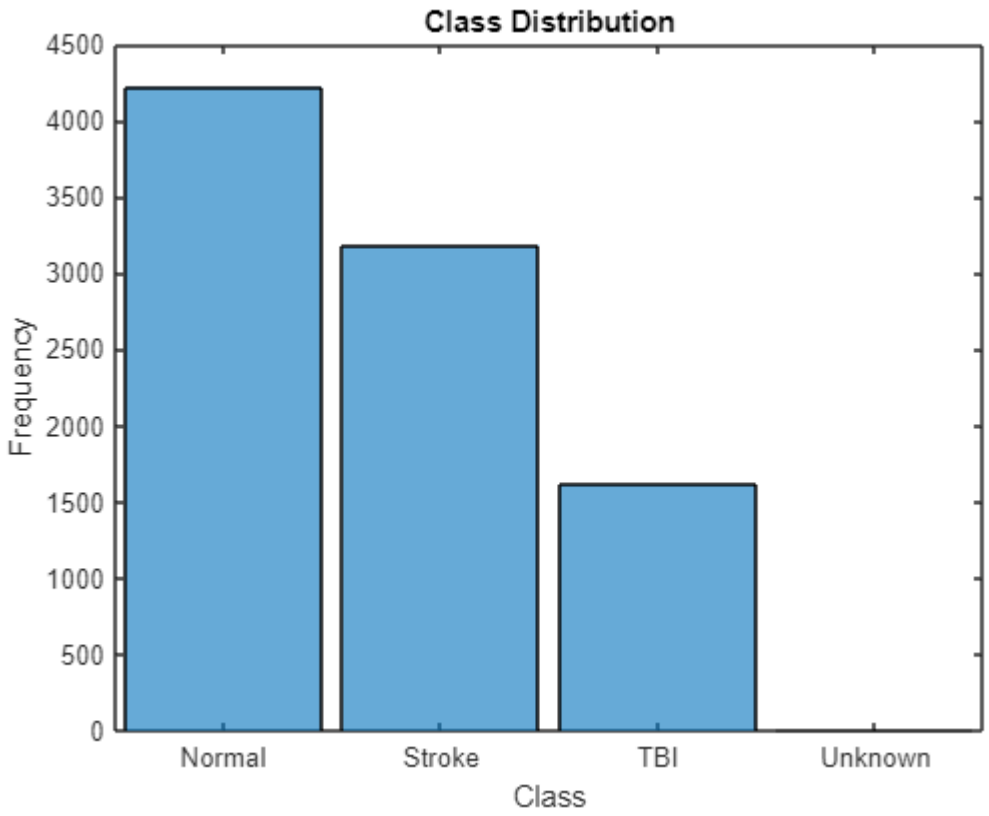
```
disp("Number of Stroke files predicted: " + sum(pred(pin)=="Stroke"));
```

```
Number of Stroke files predicted: 2571
```

```
T = AllFiles.A;
in = find(~newin);
T.Score(in) = score;
T.Category(in(pin)) = pred(pin);
T.Category(ind) = Data.A.Category;
T.Score(ind) = 1;
T.Category = categorical(T.Category);
```

```
figure
histogram(T.Category(T.Category~="Unknown"));
xlabel("Class")
ylabel("Frequency")
```

```
title("Class Distribution")
```




```
disp("TBI records:" + sum(T.Category=="TBI"))
```

```
TBI records:1614
```

```
disp("STR records:" + sum(T.Category=="Stroke"))
```

```
STR records:3185
```

```
disp("HEA records:" + sum(T.Category=="Normal"))
```

```
HEA records:4223
```

Final Curation

Remove subjects with multiple classes

```
in = unique(T.Subject,'rows','stable');
for i = 1: length(in)
    in2 = T.Category(T.Subject == in(i));
    h = histcounts(in2);
    if logical(h(2:end)>1)
        T.Category(T.Subject == in(i)) = "Unknown";
    end
end
```

Check for seizure and epilepsy

Seizure Check 1

```
%Checks for obvious seizures
sin2 = contains(T.("Full Note"),"with " + optionalPattern("recurrent ") +
"seizure","IgnoreCase",true);
cin = T.Category ~= "Unknown";
in = sin2 & cin;
S = T(in,:);
disp("Check table S with " + height(S) + " entries")
```

```
Check table S with 368 entries
```

Update

```
S.Category(:) = "Unknown";
T(in,:) = S;
```

Seizure Check 2

```
%Checks for seizure but excludes "seizure free" records
sin = contains(T.("Full Note"),"seizure","IgnoreCase",true);
sinf1 = contains(T.("Full Note"),"seizure-free","IgnoreCase",true);
sinf2 = contains(T.("Full Note"),"seizure free","IgnoreCase",true);
sinf3 = contains(T.("Full Note"),"no seizure","IgnoreCase",true);
sinf4 = contains(T.("Full Note"),"If seizures are an important
consideration","IgnoreCase",true);
sinf = sinf1 | sinf2 | sinf3 | sinf4;
cin = T.Category ~= "Unknown";
in = sin & cin & ~sinf;
S = T(in,:);
disp("Check table S with " + height(S) + " entries")
```

```
Check table S with 3049 entries
```

Update

```
S.Category(:) = "Unknown";
T(in,:) = S;
```

```
in = and(contains(T.("Full Note"),"Evaluate for seizure","IgnoreCase",true),
contains(T.("Full Note"),"brain injury","IgnoreCase",true));
S = T(in,:);
disp("Check table S with " + height(S) + " entries")
```

```
Check table S with 1 entries
```

Manual Update

```
T(in,:) = S;
```

```
in = or(contains(T.("Full Note"),"Evaluate for seizure versus
stroke","IgnoreCase",true),contains(T.("Full Note"),"Evaluate for stroke versus
seizure","IgnoreCase",true));
S = T(in,:);
disp("Check table S with " + height(S) + " entries")
```

```
Check table S with 7 entries
```

Manual Update

```
T(in,:) = S;
```

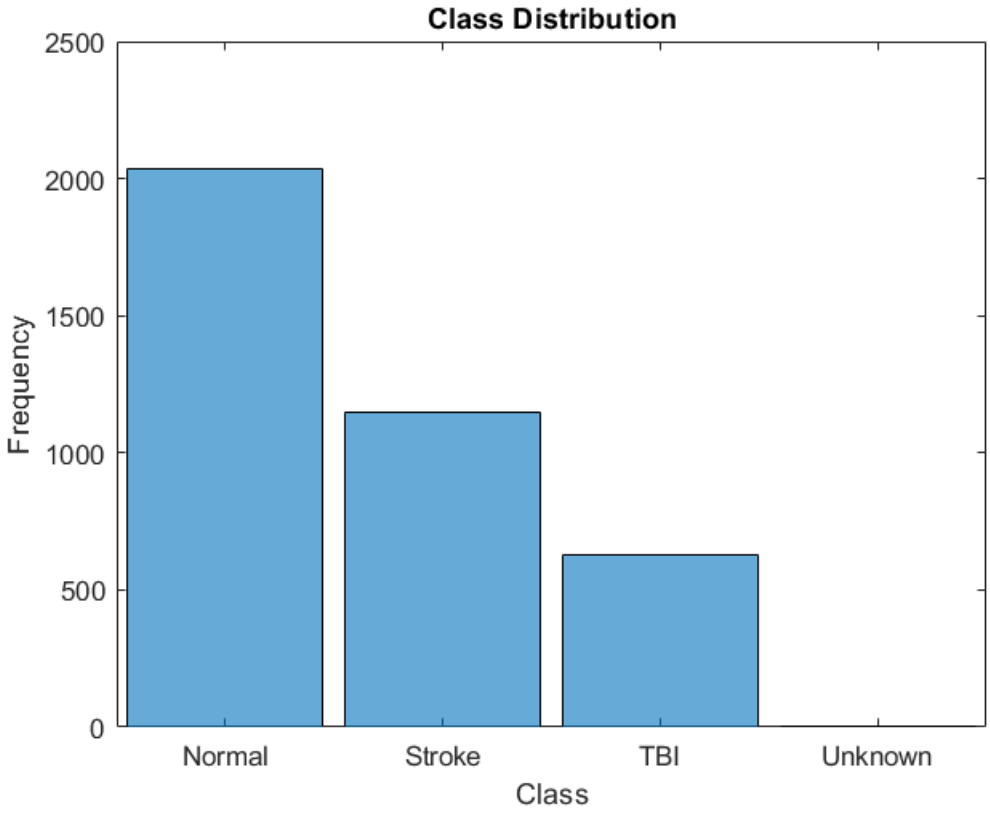
Age

```
in = and(T.Age>=18,T.Age<=65);
T = T(in,:);
```

Alzheimer/Dementia Keywords

```
in = and(or(contains(T.("Full Note"),"Alzheim","IgnoreCase",true),contains(T.("Full
Note"),"dement","IgnoreCase",true)),T.Category~="Unknown");
S = T(in,:);
subs = unique(S.Subject);
sin = ismember(T.Subject,subs);
T.Category(sin) = "Unknown";
```

```
PredictedFiles = CohortFiles(T);
figure
histogram(T.Category(T.Category~="Unknown"));
xlabel("Class")
ylabel("Frequency")
title("Class Distribution")
```




```
disp("TBI records:" + sum(T.Category=="TBI"))
```

```
TBI records:629
```

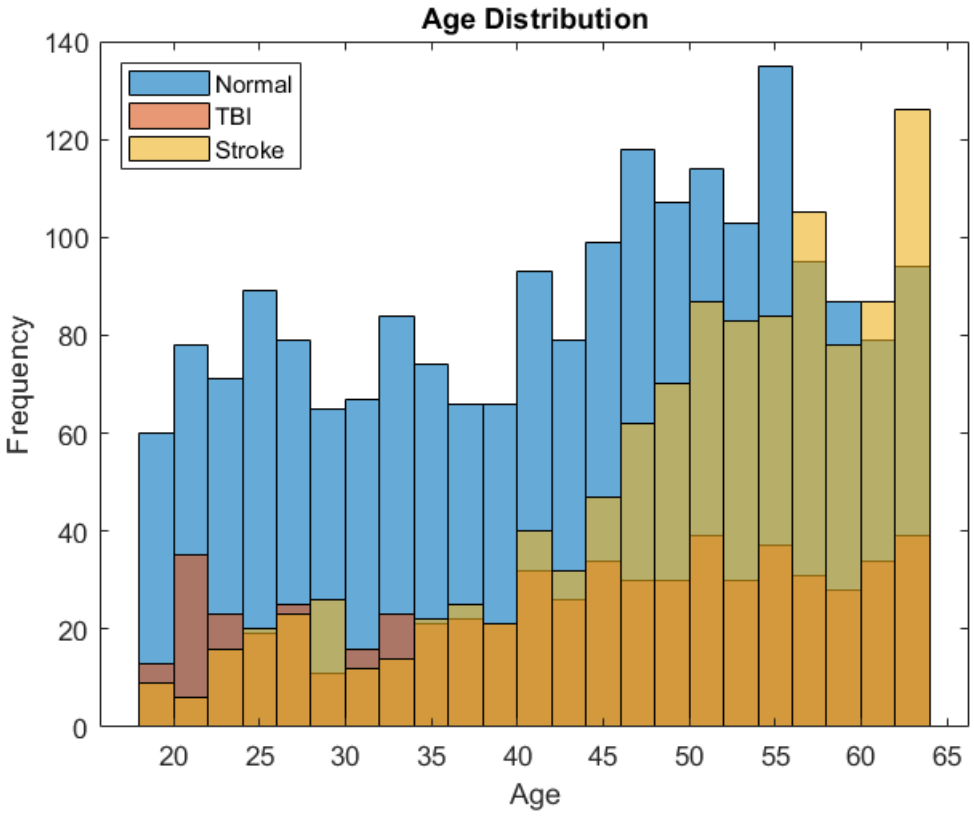
```
disp("STR records:" + sum(T.Category=="Stroke"))
```

```
STR records:1146
```

```
disp("HEA records:" + sum(T.Category=="Normal"))
```

```
HEA records:2033
```

```
figure;
histogram(T.Age(T.Category=="Normal"),18:2:65);
hold on
histogram(T.Age(T.Category=="TBI"),18:2:65);
histogram(T.Age(T.Category=="Stroke"),18:2:65);
xlabel Age
ylabel Frequency
title("Age Distribution")
legend(["Normal","TBI","Stroke"],'Location',"northwest");
```



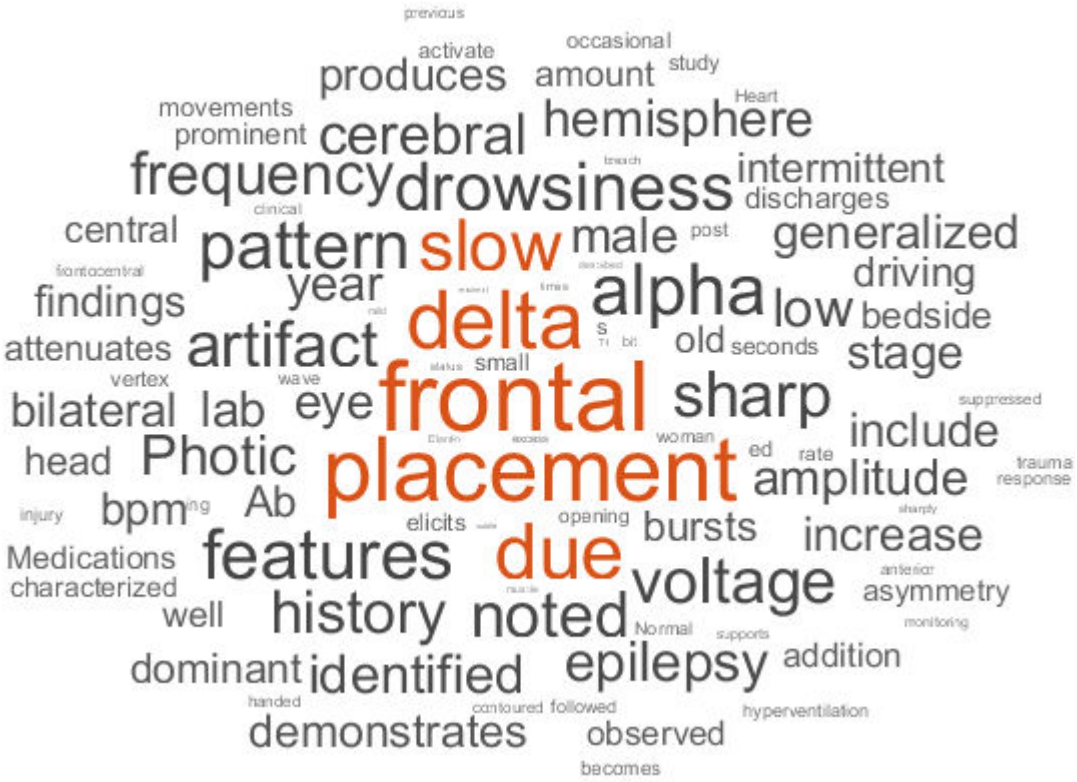
Final World Clouds

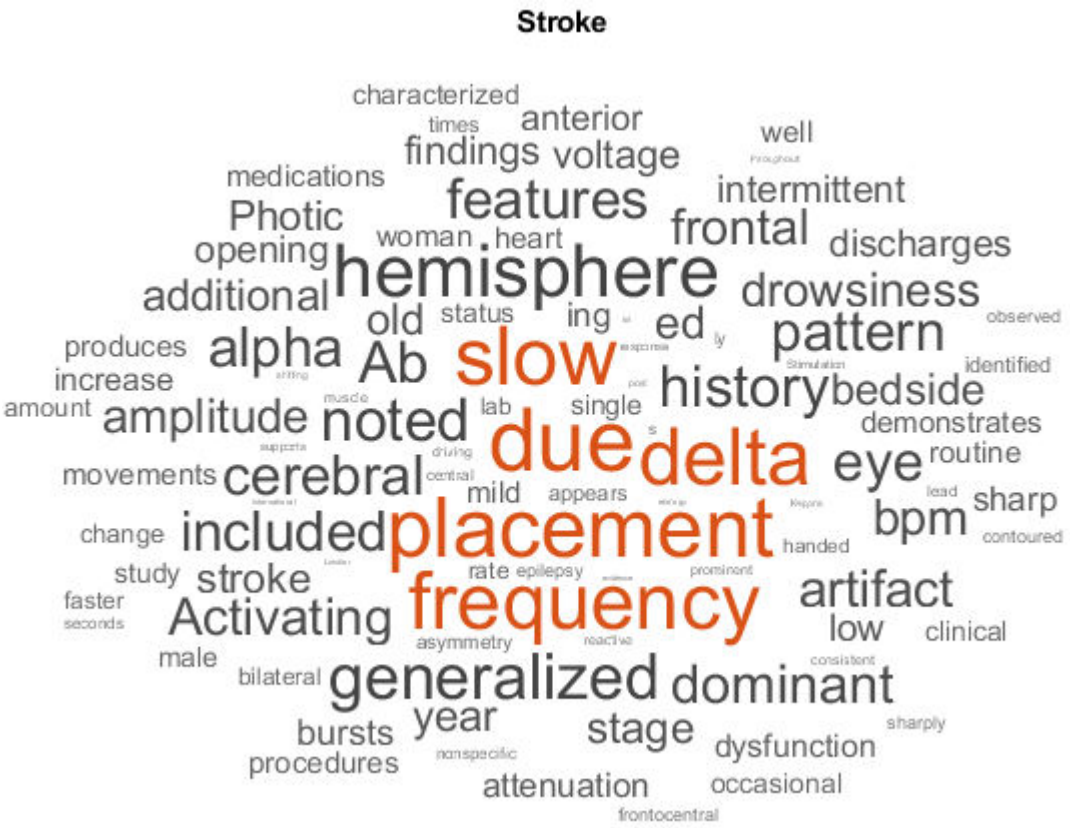
```
PlotWordClouds(PredictedFiles);
```

Normal


episodes
Findings
becomes
mouth
second
reactive
diagnosis
characterized
single
noted
dominant
bpm
handed
times
produces
Normal
Heart
Medications
increase
epilepsy
Take
generalized
lead
per
driving
drowsiness
procedures
organized
history
routine
woman
rate
daily
old
eye
slow
vertex
identified
Photic
stage
intermittent
low
voltage
tablet
male
T1
features
Waking
alpha
amplitude
ed
central
study
elicits
International
Activating
seizure
clinical
lab
year
spindles
placement
amount
well
bilateral
exclude
DAY
additional
small
frontal
discharges
included
followed
hyperventilation
attenuates
opening
frontocentral
injection
movements

TBI


previous
activate
occasional
produces
amount
study
movements
Heart
prominent
cerebral
hemisphere
frequency
drowsiness
intermittent
discharges
central
pattern
slow
male
post
generalized
year
driving
findings
alpha
low
bedside
delta
attenuates
artifact
old
seconds
stage
small
vertex
wave
bilateral
lab
eye
frontal
sharp
suppressed
include
head
Photic
placement
ed
rate
amplitude
trauma
response
injury
bpm
Ab
elicits
opening
bursts
increase
Medications
features
due
voltage
characterized
asymmetry
well
history
noted
Normal
supports
monitoring
dominant
identified
epilepsy
addition
handed
contoured
followed
hyperventilation
demonstrates
observed
becomes



Save T

```
save("PredictionDatabase.mat",'T','net','PredictedFiles')
```

Disclaimer

This software and documentation (the "Software") were developed at the Food and Drug Administration (FDA) by employees of the Federal Government in the course of their official duties. Pursuant to Title 17, Section 105 of the United States Code, this work is not subject to copyright protection and is in the public domain. Permission is hereby granted, free of charge, to any person obtaining a copy of the Software, to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, or sell copies of the Software or derivatives, and to permit persons to whom the Software is furnished to do so. FDA assumes no responsibility whatsoever for use by other parties of the Software, its source code, documentation or compiled executables, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. Further, use of this code in no way implies endorsement by the FDA or confers any advantage in regulatory decisions. Although this software can be redistributed and/or modified freely, we ask that any derivative works bear some notice that they are derived from it, and any modified versions bear some notice that they have been modified. The Software is not intended to make clinical diagnoses or to be used in any way to diagnose or treat subjects for whom the EEG is taken.

Supporting Functions

```
function T = GetLopezData(file)
```

```
filename = "";
categ = "";
k = 1;
fol1 = ["eval", "train"];
fol2 = ["abnormal","normal"];
for f = fol1
    for f2 = fol2
        path = fullfile(file,f,f2,"01_tcp_ar");
        d = dir(path);
        for i = 3:length(d)
            d1 = dir(fullfile(d(i).folder,d(i).name));
            for j1 = 3:length(d1)
                d2 = dir(fullfile(d1(j1).folder,d1(j1).name));
                for j2 = 3:length(d2)
                    d3 = dir(fullfile(d2(j2).folder,d2(j2).name,"*.txt"));
                    filename(k) = d3.name;
                    if f2=="abnormal"
                        categ(k) = "Abnormal";
                    else
                        categ(k) = "Normal";
                    end
                    k = k+1;
                end
            end
        end
    end
end
T = table(filename',categorical(categ'),'VariableNames',["Filename","Category"]);
sub=[];
ses=[];
for i = height(T):-1:1
    s = split(T.Filename(i),"_");
    sub(i) = str2double(s(1));
    s = split(s(2),".");
    s = split(s(1),"s");
    ses(i) = str2double(s(2));
end
T.Subject = sub';
T.Session = ses';
save("LopezData.mat","T");
end
function X = preprocessPredictors(X,paddingValue,maxSeqLen)

X = truncateSequences(X,maxSeqLen);
X = padsequences(X,2,"PaddingValue",paddingValue);

end
```

BERT Embedding Function

The `bertEmbed` function maps input data to embedding vectors and optionally applies dropout using the "DropoutProbability" name-value pair.

```
function Y = bertEmbed(X,parameters,args)

arguments
    X
    parameters
    args.DropoutProbability = 0
end

dropoutProbabilitiy = args.DropoutProbability;

Y = bert.model(X,parameters, ...
    "DropoutProb",dropoutProbabilitiy, ...
    "AttentionDropoutProb",dropoutProbabilitiy);

% To return single feature vectors, return the first element.
Y = Y(:,1,:);
Y = squeeze(Y);

end
```

Confusion Chart Function

```
function [p,pm] = precisionChart(confidx,score,classes,V)
numClasses = length(classes);
[max_score, pred_in] = max(score,[],2);
in = max_score > confidx;
pred_in(~in) = find(string(classes) == "Unknown");
V.Confident = categorical(string(classes(pred_in)));
figure
cm = confusionchart(V.Category,V.Confident);
cm.ColumnSummary = 'column-normalized';
p = (sum(diag(cm.NormalizedValues))-cm.NormalizedValues(1,1))/
sum(cm.NormalizedValues(:,2:end),'all');
pm = sum(([0, ones(1,numClasses-1)] .* diag(cm.NormalizedValues)')./[1
sum(cm.NormalizedValues(:,2:end))])/(numClasses - 1);
end
```