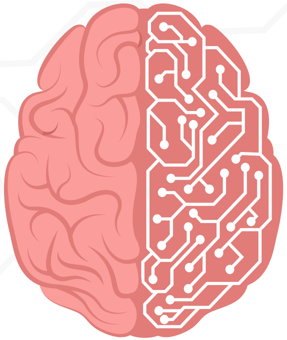


# Automation of seizure detection using EEG signals.


**Prepared by:**

Abir LAKHAL

Mehdi HMAIDI

Racha MAGHERBI

Khaled CHARAABI

Nada BELAIDI

Fares KHECHANA

# Presentation Plan :

**01.** Introduction

**02.** Business & Data Comprehension

**03.** Data Preparation

**04.** Data Modeling & Evaluation

**05.** Deployment

**06.** Product Advertising

**07.** Conclusion

# 01

# Introduction

# 01.1

# Problem Identification

# Problem :

- Epilepsy is a neurological disease that results in abnormal electrical activity in the brain.

- A sudden increase in electrical activity in the brain
- A temporary disruption of communication between neurons.

# Detection :

**Analysis of EEG (electroencephalogram) signals to detect and categorize disease trends:**

- **Long and laborious visual examination.**
- **Lots of hours to review the recording.**
- **A heavy burden on neurologists and reduces their efficiency.**

## Target audience :

Neurologists

&

Neurology Institutes.

## Target audience' needs :

An application that helps neurologists detect epileptic seizures is needed.

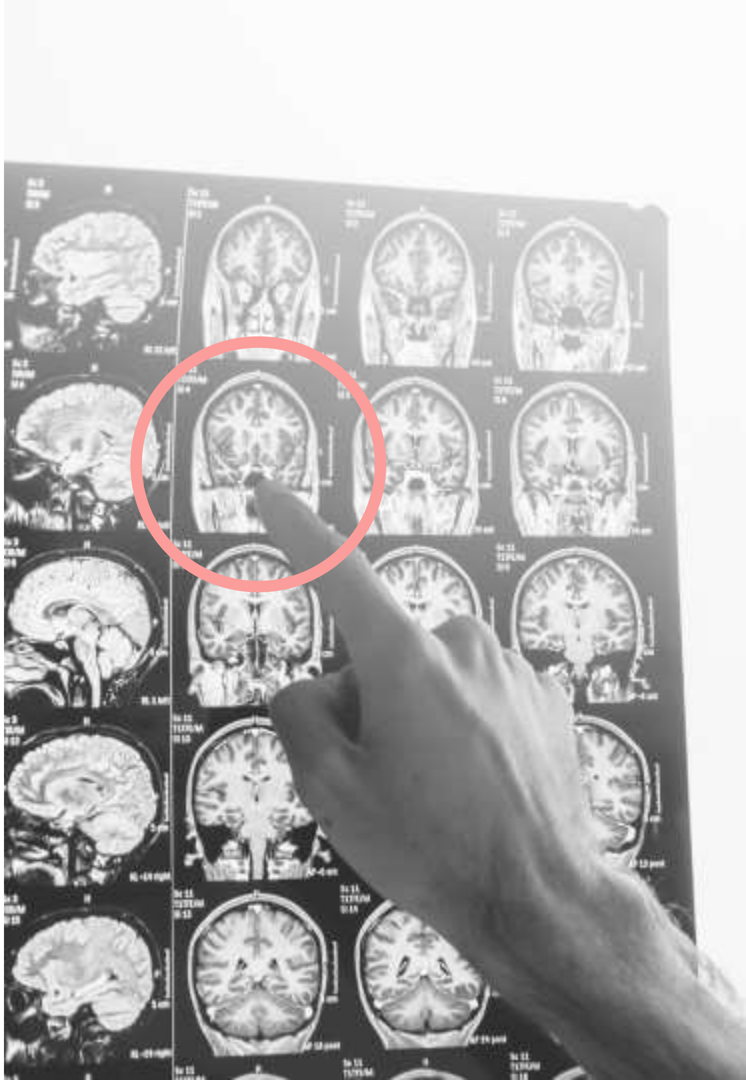
# 01.2

# Product Description

## General Description :

**Application dedicated to neurologists and neurology departments in hospitals.**

**The application is based on the detection, clinically, of epileptic seizures, i.e. scalp electroencephalogram (EEG) analysis.**



# Product-Service Description :

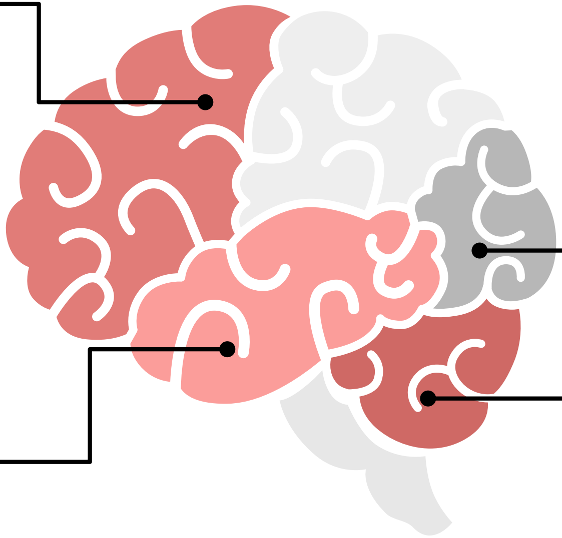
**Help the neurologist to :**

- → Properly detect and categorize the disease of epilepsy.
- → Keep the efficiency of neurologists.
- → Avoid medical errors.
- → Increase the diagnosis accuracy.
- → Cost reduction : decrease the number of experts in hospitals.
- → Save time: reduce the need of an expert.
- → Reduce the EEG training time

# Promises and potential :

**Classification of EEG signals: epileptic and non-epileptic.**

**Processing of MRI images to detect the epilptic area.**



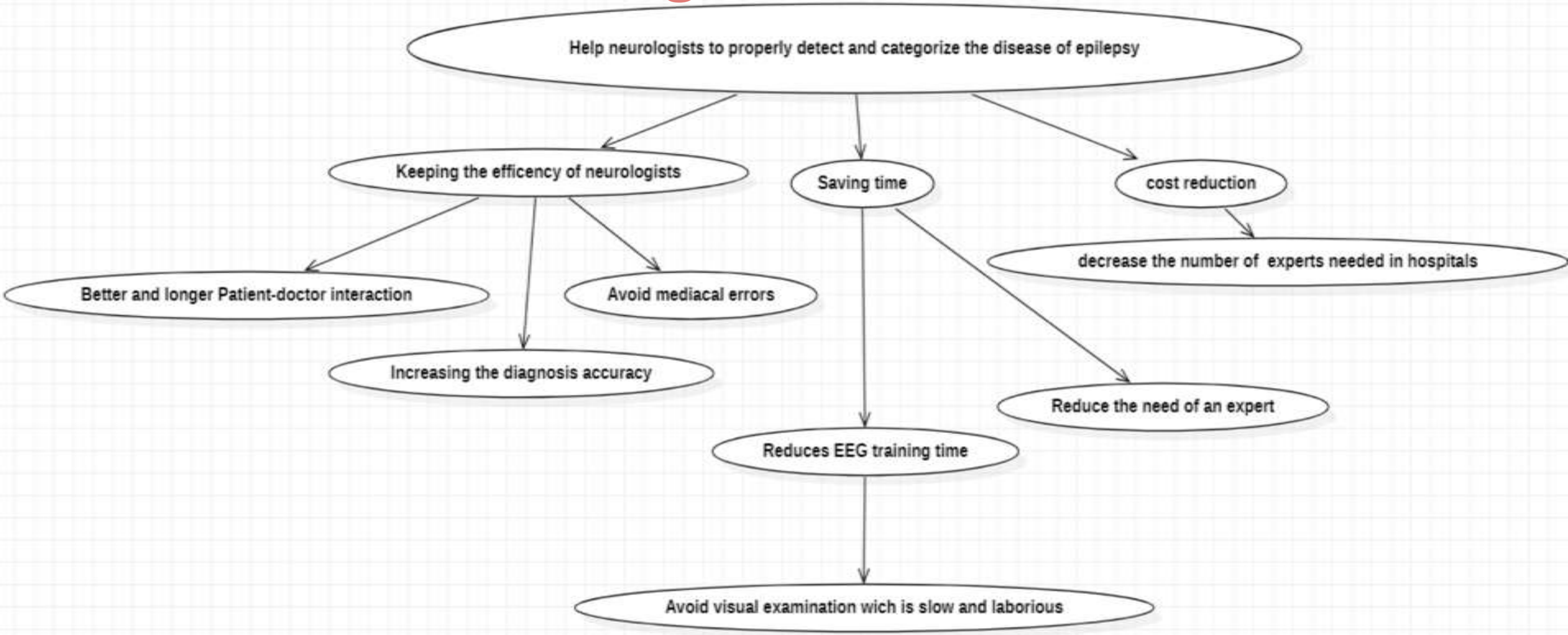
**Processing of medical reports to classify :epiletic of epilepsy.**

**Voice recognition of doctors and automation of writing medical reports (voice to text).**

# 02.

# Business & Data Comprehension

# Business Understanding :




| | |
|---|---|
| Description of the specific objective selected | Create an epilepsy detection platform dedicated to neurologists in order to help them detect and categorize the disease in a short time |

# Data Science Objectives :

- Creating an epilepsy detective system.
  → Facilitating neurologists' jobs through our system.
  → Detecting much more cases.
  → Modernize the Tunisian medical field.

- Identifying the suitable technologies for our business objectives.
- Training and deploying fast and efficient Deep Learning models.
  Key Results :
  - Using NLP for text mining.
  - Using LSTM for EEG classification.
  - Using machine learning algorithms for text classification.
  - Using R-CNN for MRI analysis.
- Implementing the API in a demo web application (Flask app).

# Data Science Project Steps :

| | |
|---|---|
| Feature engineering | Data understanding.<br>Data cleaning.<br>Creating new features for the database. |
| External data | - Add an EDF file that describes the brain signal of the patient.<br>- Use an MRI Dataset.<br>- Use medical reports to classify epilepsy type. |
| Data modeling | Classification.<br>Scoring. |
| Deployment | Creating a web platform for this epilepsy detective system (Using Flask). |

# 03.

# Data Preparation

# 03.1

# Internal Data Preparation

# EEG SIGNALS :

THE EEG signals were recorded using a standard 10 to 20 channels. The complete data consists of five sets (A to E), each containing 100 instances .

Each file is a recording of brain activity for 23.6 seconds. The time series corresponding is sampled at 4097 data points. Each data point is the value of the EEG recording at a different time. So we have a total of 500 individuals each with 4097 data points.

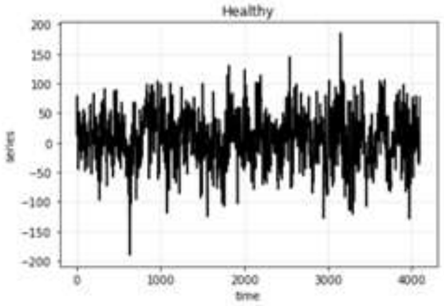
1- Loading files from folders then we extracting the data into 5 small tables, each for every set, then we assembled all of data into a big dataframe.
2- Checking for missing data.
3- Changing the labels to binary (0 for healthy, 1 for epileptic).
4- Normalizing the data.
5- CHI2 statistical test (for feature selection).

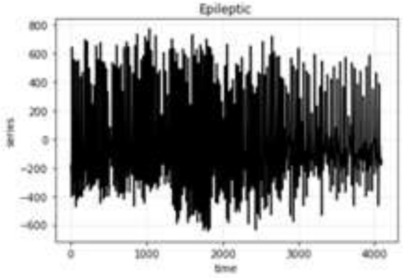
```
###Statistical Test to determine whether input features are relevant to the outcome to be predicted.

###P-value <= 0.05 significant result
###P-value > 0.05 not significant result
```

Out[45]:

| | file_name | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 4089 | 4090 | 4091 | 4092 | 4093 | 4094 | 4095 | 4096 | 4097 | label |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Z001.txt | 12.0 | 22.0 | 35.0 | 45.0 | 69.0 | 74.0 | 79.0 | 78.0 | 66.0 | ... | -28.0 | -21.0 | -14.0 | -14.0 | -25.0 | -28.0 | -11.0 | 8.0 | 77.0 | A |
| 1 | Z002.txt | -56.0 | -50.0 | -64.0 | -91.0 | -135.0 | -140.0 | -134.0 | -114.0 | -115.0 | ... | -82.0 | -114.0 | -138.0 | -159.0 | -172.0 | -180.0 | -173.0 | -162.0 | -82.0 | A |
| 2 | Z003.txt | -37.0 | -22.0 | -17.0 | -24.0 | -31.0 | -20.0 | -5.0 | 14.0 | 31.0 | ... | -52.0 | -23.0 | -14.0 | -5.0 | -3.0 | 7.0 | 3.0 | 4.0 | 82.0 | A |
| 3 | Z004.txt | -31.0 | -43.0 | -39.0 | -39.0 | -9.0 | -5.0 | 18.0 | 7.0 | -12.0 | ... | -32.0 | -40.0 | -23.0 | -1.0 | 11.0 | 12.0 | -6.0 | 10.0 | 33.0 | A |
| 4 | Z005.txt | 14.0 | 26.0 | 32.0 | 25.0 | 16.0 | 8.0 | 8.0 | 12.0 | 11.0 | ... | -19.0 | -29.0 | -35.0 | -51.0 | -55.0 | -58.0 | -32.0 | -6.0 | -17.0 | A |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 95 | S096.txt | -40.0 | -58.0 | -75.0 | -88.0 | -89.0 | -81.0 | -67.0 | -52.0 | -28.0 | ... | 32.0 | 32.0 | 18.0 | 6.0 | -3.0 | -10.0 | -13.0 | -16.0 | -151.0 | E |
| 96 | S097.txt | 187.0 | 44.0 | -147.0 | -368.0 | -550.0 | -657.0 | -665.0 | -581.0 | -442.0 | ... | 510.0 | 562.0 | 607.0 | 667.0 | 748.0 | 763.0 | 703.0 | 446.0 | -537.0 | E |
| 97 | S098.txt | -438.0 | -561.0 | -622.0 | -581.0 | -460.0 | -295.0 | -164.0 | -70.0 | 3.0 | ... | 443.0 | 399.0 | 319.0 | 196.0 | 40.0 | -47.0 | -118.0 | -163.0 | -56.0 | E |
| 98 | S099.txt | -476.0 | -518.0 | -521.0 | -362.0 | -68.0 | 175.0 | 289.0 | 184.0 | 15.0 | ... | -261.0 | -248.0 | -147.0 | 36.0 | 224.0 | 299.0 | 246.0 | 556.0 | 276.0 | E |
| 99 | S100.txt | 23.0 | 144.0 | 228.0 | 260.0 | 255.0 | 218.0 | 178.0 | 126.0 | 60.0 | ... | -127.0 | -123.0 | -152.0 | -231.0 | -272.0 | -272.0 | -155.0 | 6.0 | -221.0 | E |

500 rows × 4099 columns



## Clinical reports :

The dataset contains 522 clinical report texts, containing information about each patient and his treatment, divided in epileptic and none epileptic.

## Data Preparation Steps :

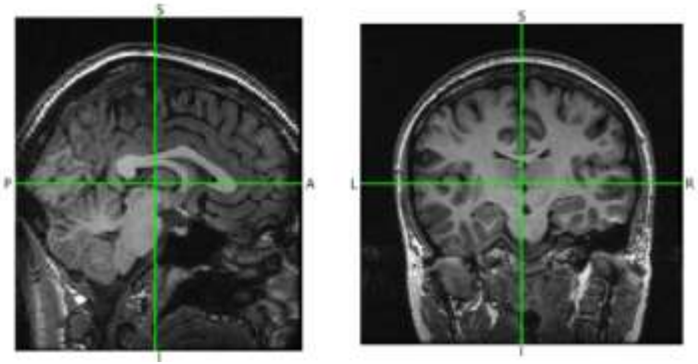
-Collecting of clinical text reports.
-Appending of all readlines in one matrix.
-Applying the natural language processing.
-Converting text to lowercase.
-Removing punctuation.
-Removing whitespaces.
-Sentence segmentation.
-Stopwords removal.
-Count vectorizing.
-word level tf-idf

# 03.2

# External Data Preparation

# MRI :

The dataset contains 50 brain T1-weighted MRI volumes with hippocampus labels (13 males / 27 females).
Age range : 15-64.

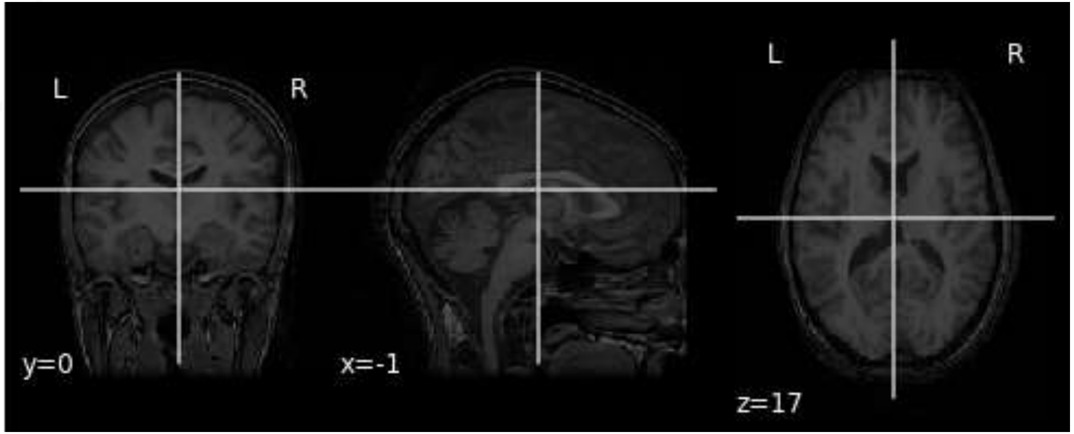


We have divided the dataset into two groups of training and testing (50% each).
We used the libraries nilearn and nibabel to visualize and understand our data.
Our data was composed of .HDR files so we had to extract a 3D matrix from each file and flattened them into one array and then normalized it.

The preparation of our data was based on :

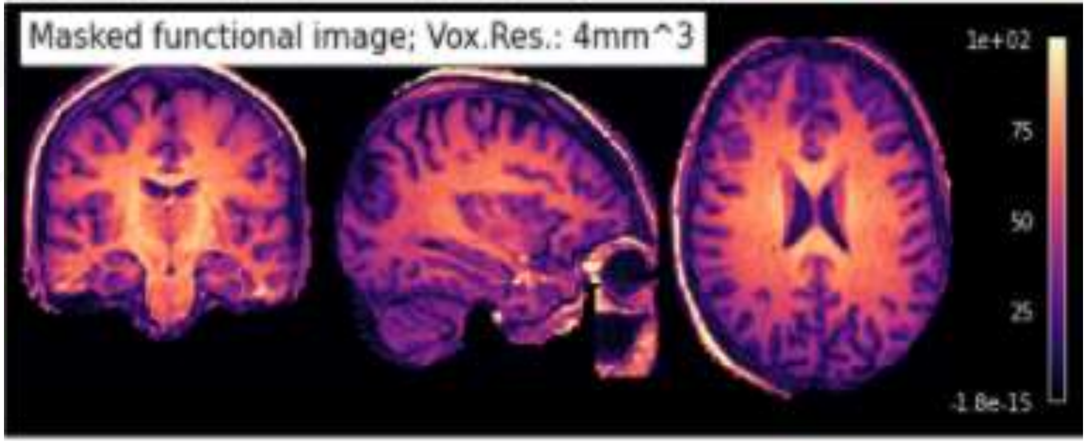
1-Putting all our MRI files into the same shape :
Since our dataset files have different shapes,sizes and dimensions , we will be reducing all of them to one main size.

2- Smoothing them to become more clear



**3- Cropping them to get rid of all the unnecessary voxels (voxels outside the brain) using masks .**

**4- Creating a mask for the hippocampus MRI labels and then cropping them too.**



# VOICE RECOGNITION API :

**Speech recognition helps us save time by speaking instead of typing.**
**We used google speech to text API to help medical stuff and doctors record their clinical reports without needing a pen.**

# 04.

# Data Modeling & Evaluation

# 04.1

# Internal Data Modeling & Evaluation

# EEG SIGNALS :

**To find out which is the most suitable model for our internal data we've started by applying SIMPLE RNN model.**
**We came out with 0.75 as a training accuracy for this model.**

```
13/13 [==============================] - 0s 3ms/step - loss: 0.6755 - val_loss: 0.7820
Epoch 93/100
13/13 [==============================] - 0s 4ms/step - loss: 0.7126 - val_loss: 0.7821
Epoch 94/100
13/13 [==============================] - 0s 4ms/step - loss: 0.7266 - val_loss: 0.8117
Epoch 95/100
13/13 [==============================] - 0s 4ms/step - loss: 0.7459 - val_loss: 0.8153
Epoch 96/100
13/13 [==============================] - 0s 3ms/step - loss: 0.6796 - val_loss: 0.8428
Epoch 97/100
13/13 [==============================] - 0s 3ms/step - loss: 0.6867 - val_loss: 0.8323
Epoch 98/100
13/13 [==============================] - 0s 3ms/step - loss: 0.7201 - val_loss: 0.7561
Epoch 99/100
13/13 [==============================] - 0s 3ms/step - loss: 0.7195 - val_loss: 0.7794
Epoch 100/100
13/13 [==============================] - 0s 3ms/step - loss: 0.7137 - val_loss: 0.7542
4/4 [==============================] - 0s 997us/step - loss: 0.7542

Out[19]: 0.7542365193367004
```

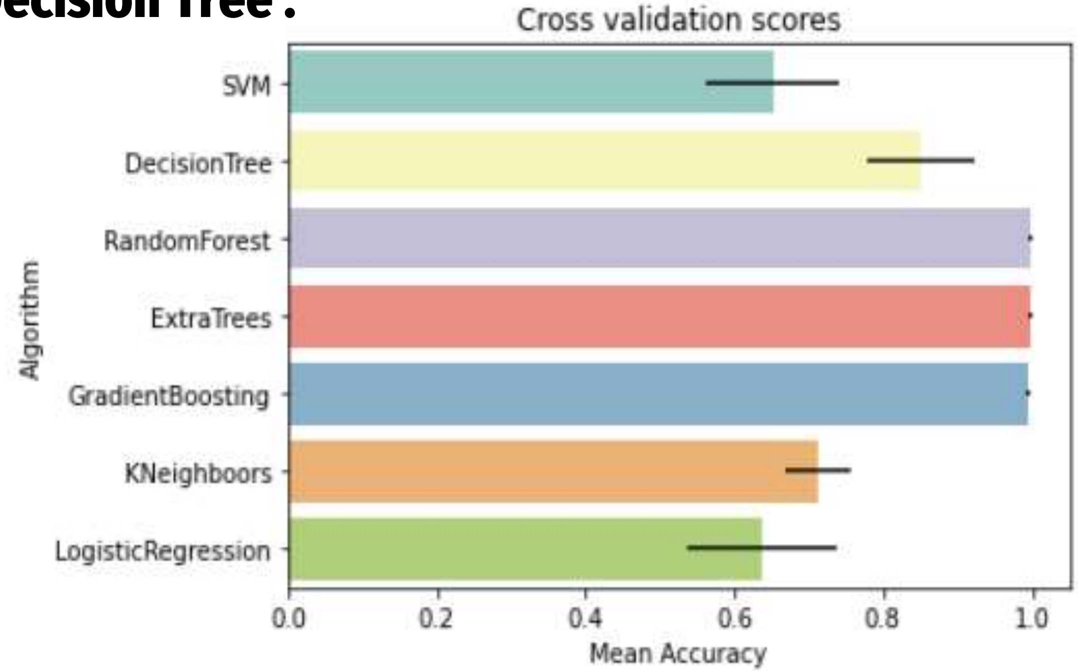
# Then we've chosen to apply LSTM model that provided us with a better training accuracy (0.95).

Entrée [24]:

```
from sklearn.metrics import accuracy_score
pred = model.predict(x_test)
predict_classes = np.argmax(pred,axis=1)
expected_classes = np.argmax(y_test,axis=1)
print(expected_classes.shape)
print(predict_classes.shape)
correct = accuracy_score(expected_classes,predict_classes)
print(f"Training Accuracy: {correct}")
```

```
(100,)
(100,)
Training Accuracy: 0.95
```

**Finally we tried some basic Machine learning models.**
**As a result, 3 models (RandomForest, ExtraTrees, GradientBoosting) have shown an overfitting and the rest (SVM, DecisionTree, KNN, LogisticRegression) came out with different accuracy with the best one is 0.85 for Decision Tree .**



# Clinical Reports :

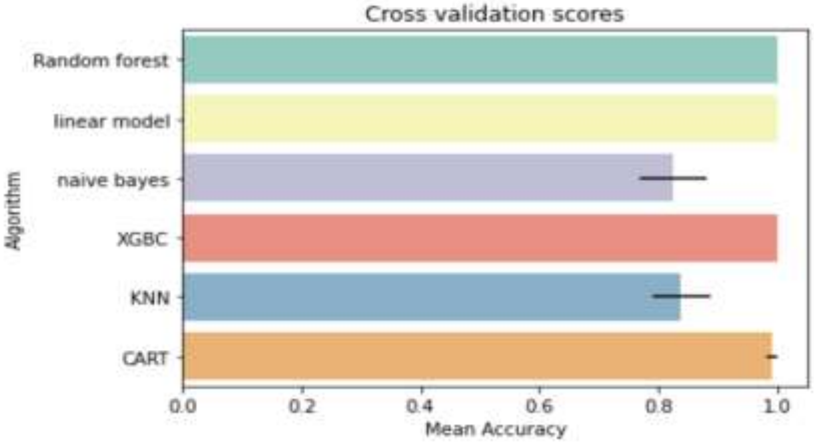
```
[87] #naive_bayes
     gnb = naive_bayes.MultinomialNB()
     gnb.fit(x_train,y_train)
     predictions = gnb.predict(x_test)
     metrics.accuracy_score(predictions, y_test)

     0.8952380952380953
```

```
knn = KNeighborsClassifier(2)
knn_model = knn.fit(x_train, y_train)
y_pred_knn =knn_model.predict(x_test)

print('Accuracy of K-NN classifier on training set: {:.2f}'
     .format(knn.score(x_train, y_train)))
print('Accuracy of K-NN classifier on test set: {:.2f}'
     .format(knn.score(x_test, y_test)))
```

```
Accuracy of K-NN classifier on training set: 0.96
Accuracy of K-NN classifier on test set: 0.91
```

## Here we used other machine learning models (CART, Linear model, Random Forest) but the result was an overfitting.
## For the rest we had a better accuracy with the KNN model.

```
   CrossValMeans  CrossValerrors      Algorithm
0       1.000000        0.000000  Random forest
1       1.000000        0.000000   linear model
2       0.825761        0.057021    naive bayes
3       1.000000        0.000000           XGBC
4       0.838549        0.048592            KNN
5       0.991414        0.010176           CART
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass t
  FutureWarning
```



# 04.2

# External Data Modeling & Evaluation

# MRI :

**Since 3D U-Net model is the most suitable for our Data ,we've chosen to apply it on our MRI files.**
**We proceeded to use MONAI and Catalyst, two PyTorch-based frameworks in order to implement our model since it wasn't available anywhere and had to be coded from scratch.**

```
# create UNet, DiceLoss and Adam optimizer
# device = torch.device("cuda:0")  # you don't need device, because Catalyst uses autoscaling
model = UNet(
    dimensions=3,
    in_channels=1,
    out_channels=1,
    channels=(16, 32, 64, 128, 256),
    strides=(2, 2, 2, 2),
    num_res_units=2,
)
```

# This was our training output, we had 0.94 as an accuracy value.

```
ice_metric=0.9410 | loss=-7.309e-01
2/2 * Epoch 2 (valid): _timer/_fps=0.2591 | _timer/batch_time=6.4243 | _timer/data_time=0.2666 | _timer/model_time=6.1577 | d
ice_metric=0.9441 | loss=-7.498e-01
[2021-04-29 00:54:11,730]
2/2 * Epoch 2 (_base): lr=0.0010 | momentum=0.9000
2/2 * Epoch 2 (train): _timer/_fps=0.1138 | _timer/batch_time=9.6863 | _timer/data_time=6.5672 | _timer/model_time=3.1191 | d
ice_metric=0.9410 | loss=-7.309e-01
2/2 * Epoch 2 (valid): _timer/_fps=0.2591 | _timer/batch_time=6.4243 | _timer/data_time=0.2666 | _timer/model_time=6.1577 | d
ice_metric=0.9441 | loss=-7.498e-01
INFO:metrics_logger:
2/2 * Epoch 2 (_base): lr=0.0010 | momentum=0.9000
2/2 * Epoch 2 (train): _timer/_fps=0.1138 | _timer/batch_time=9.6863 | _timer/data_time=6.5672 | _timer/model_time=3.1191 | d
ice_metric=0.9410 | loss=-7.309e-01
2/2 * Epoch 2 (valid): _timer/_fps=0.2591 | _timer/batch_time=6.4243 | _timer/data_time=0.2666 | _timer/model_time=6.1577 | d
ice_metric=0.9441 | loss=-7.498e-01
Top best models:
C:\Users\msi\AppData\Local\Temp\tmpgq7413gq\logs\checkpoints/train.2.pth       0.9441
=> Loading checkpoint C:\Users\msi\AppData\Local\Temp\tmpgq7413gq\logs/checkpoints/best_full.pth
loaded state checkpoint C:\Users\msi\AppData\Local\Temp\tmpgq7413gq\logs/checkpoints/best_full.pth (global epoch 2, epoch 2,
stage train)
```

# VOICE RECOGNITION API :

```
import speech_recognition as sr
```

```
r = sr.Recognizer()
```

```
harvard = sr.AudioFile('enree.wav')
with harvard as source:
    r.adjust_for_ambient_noise(source)
    audio = r.record(source)
```

```
r.recognize_google(audio)
```

```
'today I want you to ask yourself this one question why not you why not you to do something for work that you love'
```

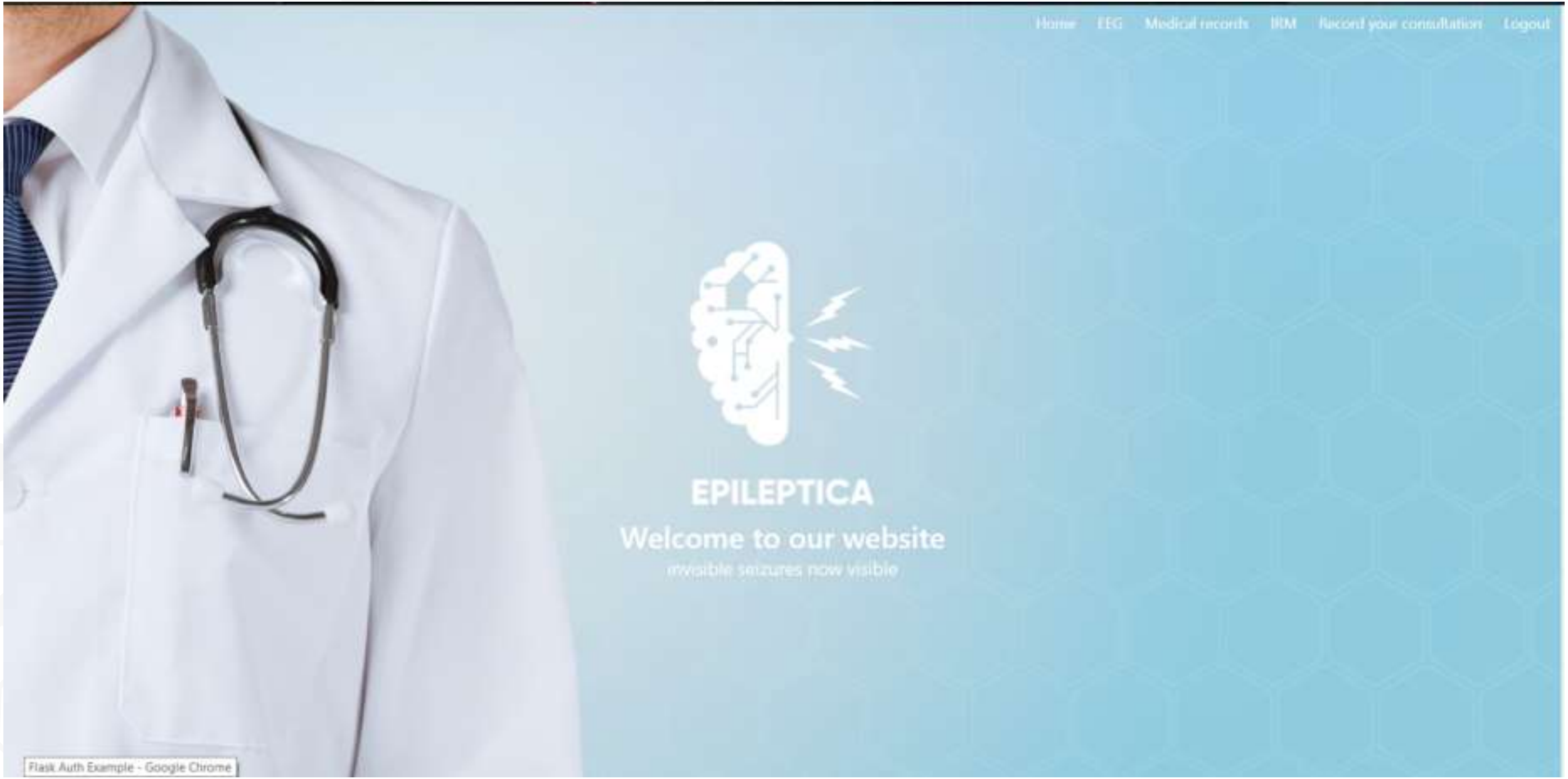
# 05.

# Deployment

We used Flask, which is an open-source micro framework for web development in Python.

Flask

web development,
one drop at a time


Home
EEG
Medical records
IRM
Record your consultation
Logout
EPILEPTICA
Welcome to our website
invisible seizures now visible
Flask Auth Example - Google Chrome

# 06.

# Product Advertising

**EPILEPTICA:**

- An application that helps neurologists detect epileptic seizures.
- Increases the diagnosis accuracy.
- Avoids medical errors.
- Cost reduction.
- Saving time.


- Online.
- Google play.
- App store.

- Advertising.
- Public relations.
- Personal sales.
- Sales promotion.
- Online Marketing.
- Social media.
- Sponsorship.

- Perception of values.

## Logo :

EPILEPTICA

EPILEPTICA

EPILEPTICA

EPILEPTICA

EPILEPTICA

# Graphic Chart :

ec7c90

df206a

ec7c90

# Marketing materials.

# Business Card :



# Poster :


WE'RE READY!
INVISIBLE SEIZURES
NOW VISIBLE.


WE'RE READY!
INVISIBLE SEIZURES
NOW VISIBLE.
Automation of seizure detection
Voice to text features
Image processing to detect epileptic area
Reports Processing
Google Play
App Store
EPILEPTICA
www.epileptica.tn
Facebook.com/Epileptica
+216 28 064 788

# Call-to-action :

# 07.

# Conclusion

**Our application will facilitate neurologists' job face to epilepsy disease, in terms of time saving and cost reduction.**

EPILEPTICA

## Contact us :

www.facebook.com/epilecpticaApp

@epilecpticaApp

epilepticaapp@gmail.com

# Thanks for your attention.