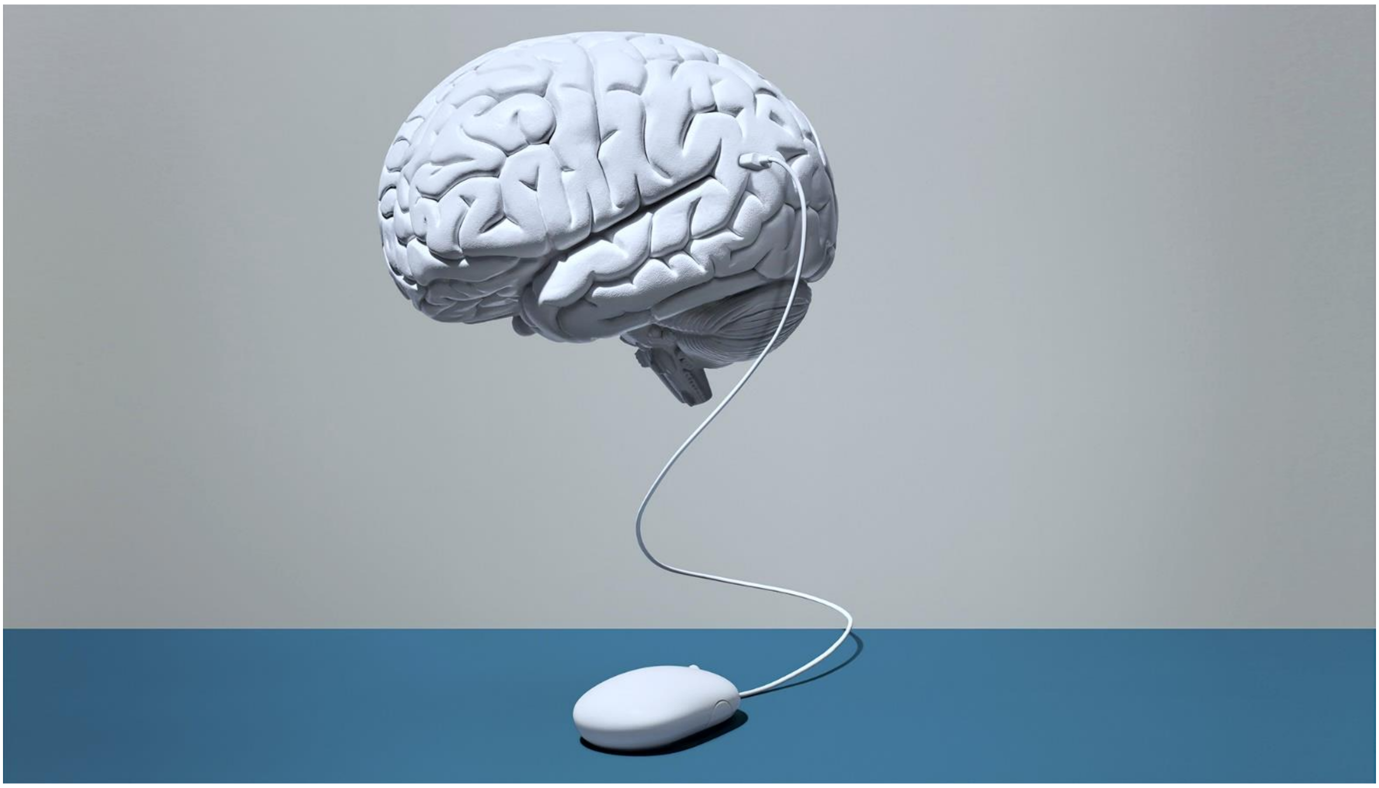
BUSINESS ANALYSIS & INSIGHTS OF NEUROTECH INDUSTRY

BY-Raj Sahu



Background

The neurotech industry refers to the field of technology that is focused on understanding and enhancing brain function through the use of neuroscience, engineering, and computer science. This industry is rapidly growing and has the potential to revolutionize many aspects of healthcare, education, and entertainment.

Neurotechnology and machine learning are rapidly advancing fields that have the potential to transform many aspects of our lives in the coming years. Here are some potential ways in which these two fields could intersect and shape the future:

- BCIs are devices that can read signals from the brain and translate them into actions or commands. This could have applications in healthcare, entertainment, and even transportation.
- Combining Machine learning algorithms with neurotechnology, doctors could have access to even more data about patients, allowing for earlier diagnosis and better treatment options.
- By using machine learning algorithms to analyse brain activity, educators could better understand how each student learns and tailor teaching methods accordingly. This could lead to more effective and personalized education.

Fermi Estimation (Breakdown of Problem Statement)

Question:

Task is to analyse the Neurotech industry business insight and come up with feasibly strategy to enter the market.

Wild Guess:

- Use Cases of Neuroscience?
- Analyse EEG, MRI and PET?
- Customer Experience?
- Personalized learning?
- Diagnosed and Treatment of neurological disorder?

Education:

Neurotech industry is one of the most promising and prospective sectors of modern biotech in particular and exponential markets in general.

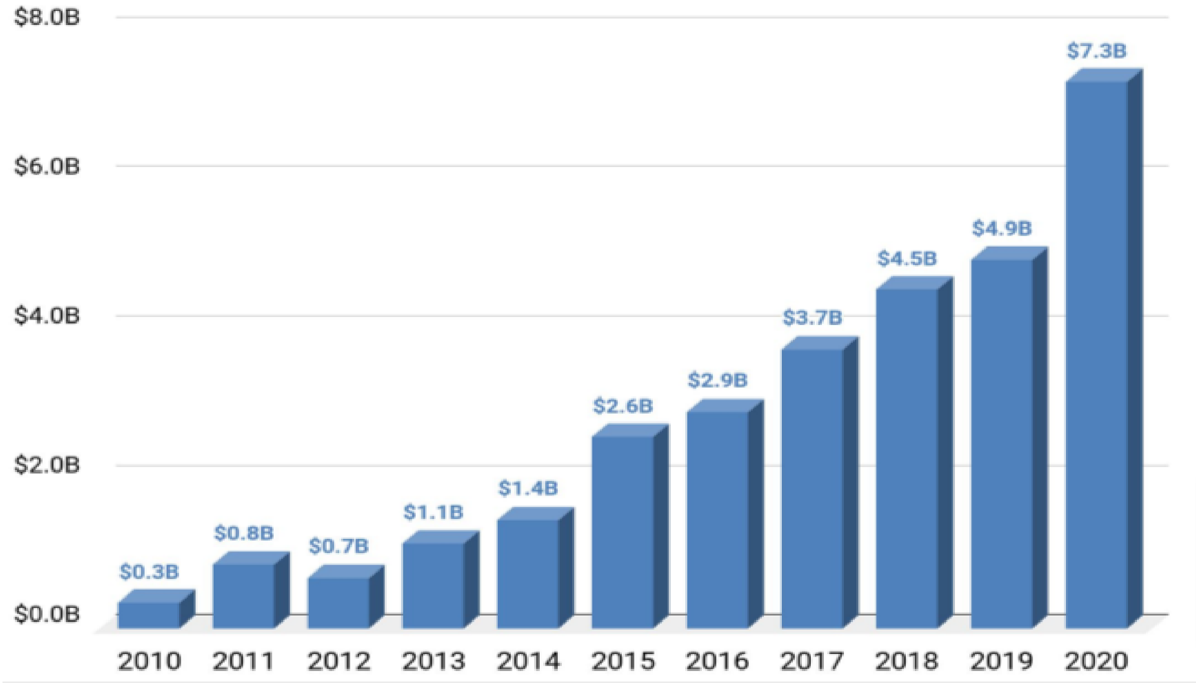
A significant increase in the number of articles published regarding deep brain stimulation and neuroimaging in a ten-year period.

Mathematical/Statistics:

That competition is presented by the amounts of investments in new biotech companies. During the last 10 years, amount of investments in NeuroTech companies increased in 21 times from $331 million to $7.1 billion. 647 US-based companies have raised more than $24 billion total funding by the beginning of 2021.

One significant business insight of the neurotech industry is the potential for significant market growth. The global market for neurotechnology is projected to reach $13.3 billion by 2025, growing at a CAGR of 15.8% from 2020 to 2025.

This growth is driven by factors such as increasing demand for neurology-based diagnostics, treatments, and therapeutics, rising incidences of neurological disorders, and advancements in technology.



Different types of analysis for various type of business insights:

Diagnostic analysis

Diagnostic analytics also focus on the past. However, these types of analyses look for cause and effect to illustrate ***why*** something happened.

Example-Diagnosis & treatment of neurological disorders

Descriptive analysis

Descriptive analytics juggles raw data from multiple data sources to give valuable insights into the past.

Example-Improved customer experience: Neurotech could enable e-commerce retailers to better understand their customers' preferences and needs.

Predictive analysis

It uses the findings of descriptive and diagnostic analytics to detect clusters and exceptions and to predict future trends, which makes it a valuable tool for forecasting.

Example- Predictive analytics in healthcare

Prescriptive analysis

Prescriptive analysis is a type of advanced analytics that goes beyond descriptive and predictive analytics to provide recommendations for optimal decision-making.

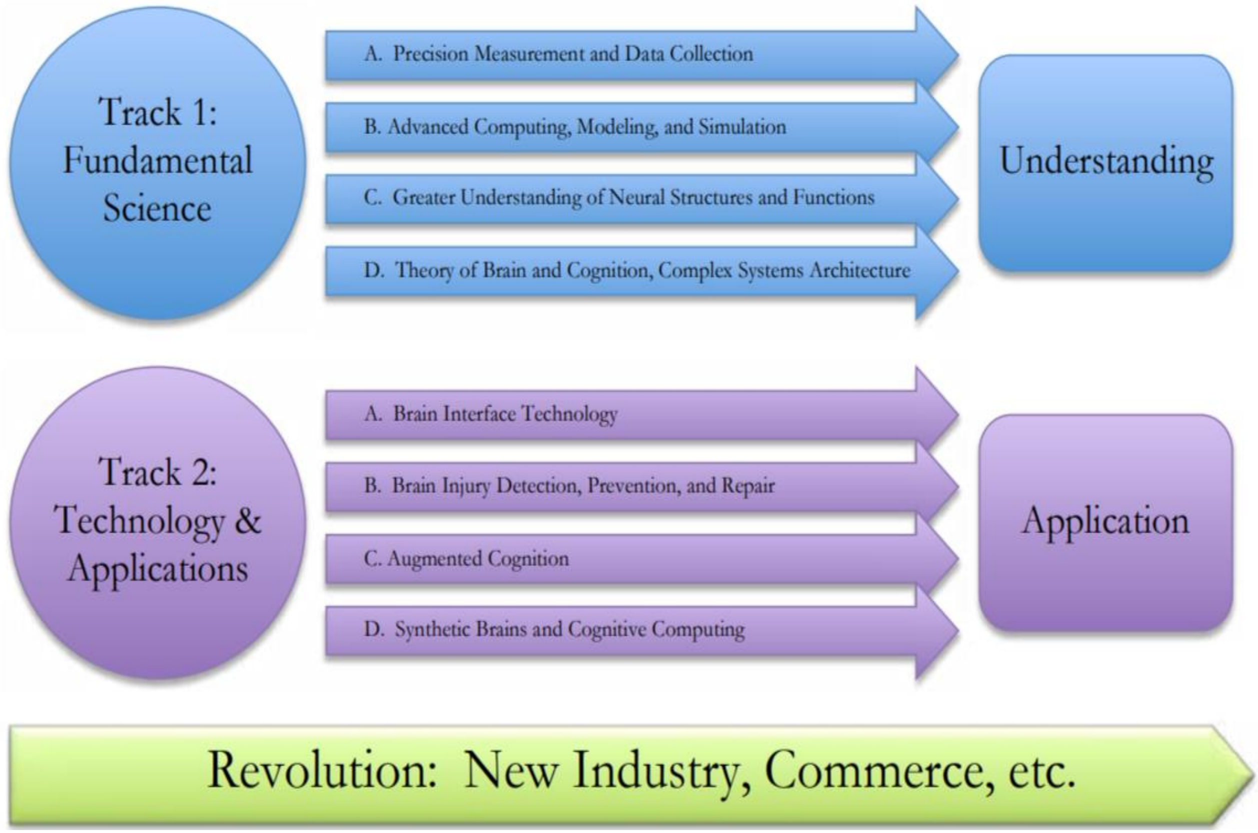
Example: personalized treatment recommendations

Business data analytics activities,

1. Defining the business problem or opportunity

A business problem related to neuroscience could be the challenge of improving employee productivity and performance.

- Neuroscience research has shown that factors such as stress, distractions, and fatigue can significantly impact cognitive function and overall performance. Therefore, a company may face the challenge of finding ways to reduce these factors and improve the cognitive function of their employees.
- Another challenge could be improving customer engagement and satisfaction. Companies may need to develop tools and techniques to measure and analyse customer emotions, such as using facial recognition software or electroencephalography (EEG) to measure brain activity.



2. Assessing the current state

The field of neurotech, which refers to the development and application of technologies that interact with the brain and nervous system, has been growing rapidly in recent years. Some of the key areas of progress include:

1. **Brain-computer interfaces** (BCIs): BCIs allow individuals to interact with computers and other devices using their brainwaves, bypassing traditional input methods such as a keyboard or mouse. Recent advances in BCI technology have shown promise for treating neurological disorders, as well as for enhancing human performance.
2. **Neuroimaging**: Neuroimaging techniques, such as magnetic resonance imaging (MRI) and positron emission tomography (PET), allow researchers to visualize the structure and function of the brain in real-time. These techniques have been instrumental in advancing our understanding of brain function and are increasingly being used in clinical settings for diagnosis and treatment planning.
3. **Neuromodulation**: Neuromodulation refers to the use of electrical or magnetic stimulation to modulate the activity of neural circuits in the brain. This approach has been used to treat a range of neurological disorders, including Parkinson's disease and epilepsy, and is being explored for other applications such as depression and chronic pain.
4. **Artificial intelligence (AI) and machine learning**: AI and machine learning algorithms are increasingly being used to analyse large datasets of brain activity and make predictions about behaviour and cognitive states. This approach has the potential to improve our understanding of the brain and could lead to new diagnostic and treatment approaches.

5. **Wearable devices**: Wearable neurotech devices, such as EEG headsets and sleep monitoring devices, are becoming increasingly popular. These devices can track brain activity, sleep patterns, and other physiological measures, providing valuable insights into health and wellness.

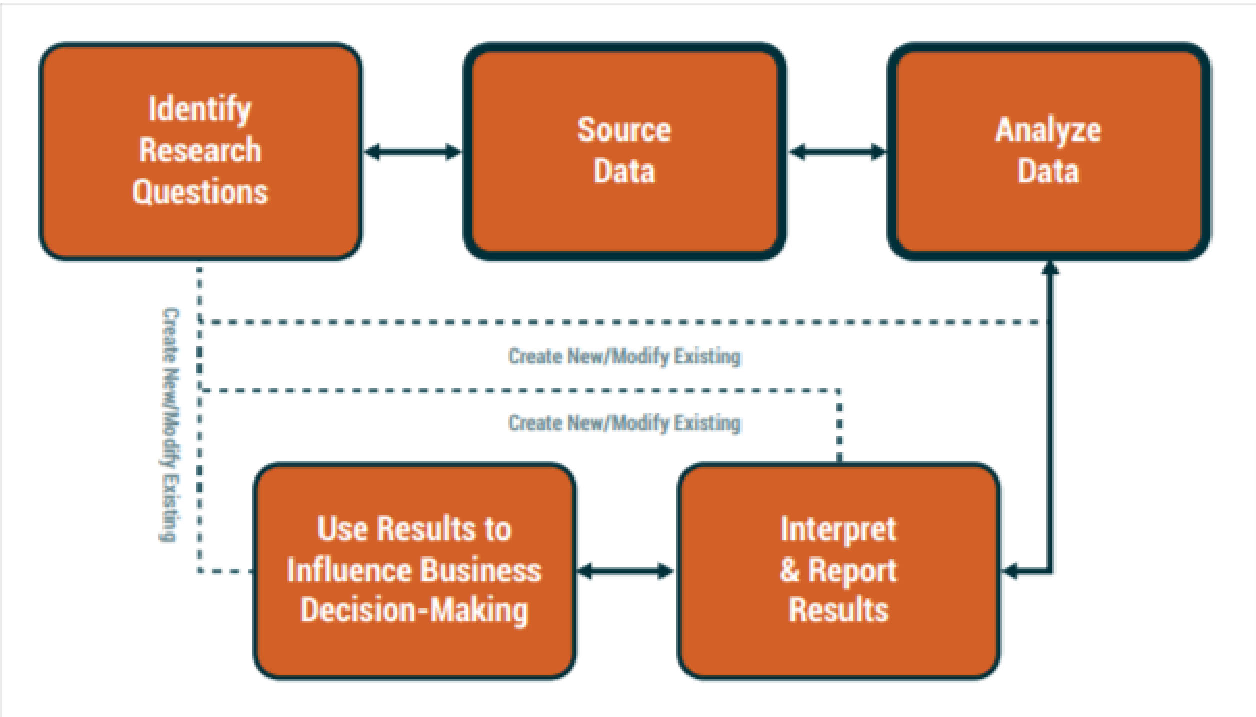
3. Defining the future state

The future of neurotech is both exciting and challenging, with numerous possibilities for new innovations and breakthroughs that could transform healthcare, education, and other fields. Here are some potential developments that could shape the future of neurotech:

1. **Brain-inspired computing**: The development of new computing architectures and algorithms inspired by the structure and function of the brain could lead to more powerful and efficient computing systems. These systems could be used for a wide range of applications, from drug discovery to autonomous vehicles.
2. **Personalized neurotherapeutics**: Advances in neuroimaging and other diagnostic tools could enable doctors to develop more personalized treatments for neurological disorders. This could lead to better outcomes for patients and a more efficient use of healthcare resources.
3. **Neuroscience-based education**: Insights from neuroscience research could inform the development of more effective educational strategies and technologies. For example, neurofeedback training could help students learn more effectively by providing real-time feedback on their brain activity.

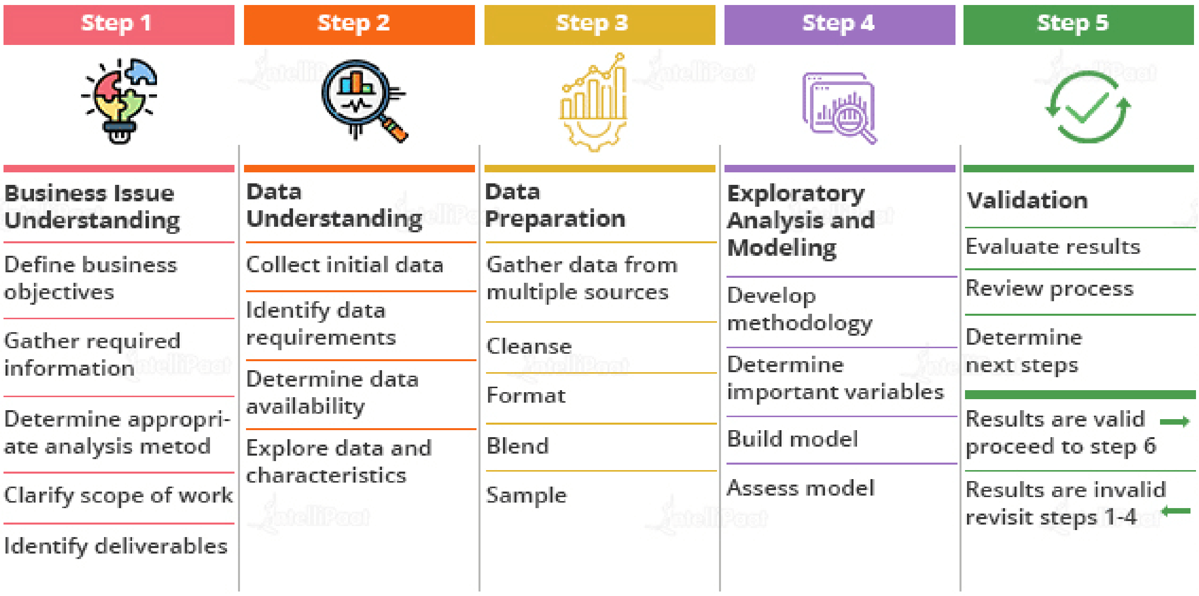
4. **Brain-computer networks**: The development of brain-computer networks could enable individuals to communicate directly with one another using their thoughts. This could have profound implications for social interaction, education, and even entertainment.

Data Analysis for Neurotech Industry



1. Problem Statement/Research Question Business Insights

"What patterns and insights can be derived from analysing large-scale neuroimaging data to better understand the underlying mechanisms of neurological disorders?"



Alzheimer's Disease

- Alzheimer's disease (AD) is a neurodegenerative disorder of uncertain cause and pathogenesis that primarily affects older adults and is the most common cause of dementia.
- However, in order to reach that stage clinicians and researchers will have to make use of machine learning techniques that can accurately predict progress of a patient from mild cognitive impairment to dementia.
- We propose to develop that can help clinicians do that and predict early Alzheimer's.

2. Source Data

The team has found MRI related data that was generated by the Open Access Series of Imaging Studies (OASIS) project that is available both, on their website and Kaggle that can be utilized for the purpose of training various machine learning models to identify patients with mild to moderate dementia.

2.1 Data Description

- We will be using the longitudinal MRI data.
- The dataset consists of a longitudinal MRI data of 150 subjects aged 60 to 96.
- Each subject was scanned at least once.

3. Exploratory Data Analysis (Eda)

we have focused on exploring the relationship between each feature of MRI tests and dementia of the patient. The reason we conducted this Exploratory Data Analysis process is to state the relationship of data explicitly through a graph so that we could assume the correlations before data extraction or data analysis.

The minimum, maximum, and average values of each feature for graph implementation are as follows:

	Min	Max	Mean
Educ	6	23	14.6
SES	1	5	2.34
MMSE	17	30	27.2
CDR	0	1	0.29
eTIV	1123	1989	1490
nWBV	0.66	0.837	0.73
ASF	0.883	1.563	1.2

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

sns.set()

df = pd.read_csv('../input/oasis_longitudinal.csv')
df.head()
```

Out[1]:

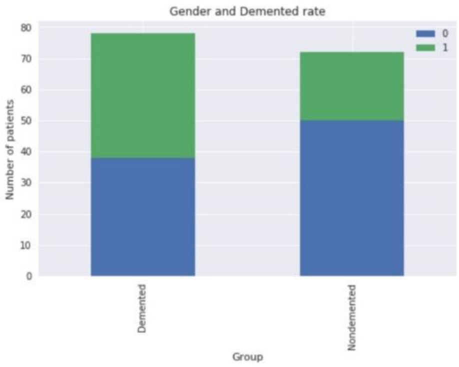
Subject ID	MRI ID	Group	Visit	MR Delay	M/F	Hand	Age	EDUC	SES	MMSE	CDR	eTIV	nWBV	ASF
OAS2_0001	OAS2_0001_MR1	Nondemented	1	0	M	R	87	14	2.0	27.0	0.0	1987	0.696	0.883
OAS2_0001	OAS2_0001_MR2	Nondemented	2	457	M	R	88	14	2.0	30.0	0.0	2004	0.681	0.876
OAS2_0002	OAS2_0002_MR1	Demented	1	0	M	R	75	12	NaN	23.0	0.5	1678	0.736	1.046
OAS2_0002	OAS2_0002_MR2	Demented	2	560	M	R	76	12	NaN	28.0	0.5	1738	0.713	1.010
OAS2_0002	OAS2_0002_MR3	Demented	3	1895	M	R	80	12	NaN	22.0	0.5	1698	0.701	1.034

In [2]:
```
df = df.loc[df['Visit']==1] # use first visit data only because of the analysis we're doing
df = df.reset_index(drop=True) # reset index after filtering first visit data
df['M/F'] = df['M/F'].replace(['F','M'], [0,1]) # M/F column
df['Group'] = df['Group'].replace(['Converted'], ['Demented']) # Target variable
df['Group'] = df['Group'].replace(['Demented', 'Nondemented'], [1,0]) # Target variable
df = df.drop(['MRI ID', 'Visit', 'Hand'], axis=1) # Drop unnecessary columns
```

In [3]:
```
# bar drawing function
def bar_chart(feature):
    Demented = df[df['Group']==1][feature].value_counts()
    Nondemented = df[df['Group']==0][feature].value_counts()
    df_bar = pd.DataFrame([Demented,Nondemented])
    df_bar.index = ['Demented','Nondemented']
    df_bar.plot(kind='bar',stacked=True, figsize=(8,5))
```

In [4]:
```
# Gender  and  Group ( Femal=0, Male=1)
bar_chart('M/F')
plt.xlabel('Group')
plt.ylabel('Number of patients')
plt.legend()
plt.title('Gender and Demented rate')
```

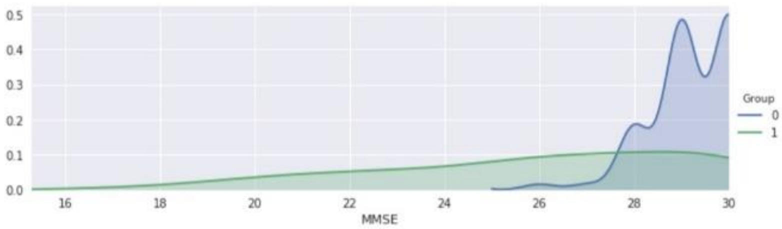
Out[4]:
```
Text(0.5,1,'Gender and Demented rate')
```



The above graph indicates that men are more likely with dementia than women.

In [5]:
```
#MMSE : Mini Mental State Examination
# Nondemented = 0, Demented =1
# Nondemented has higher test result ranging from 25 to 30.
#Min 17 ,MAX 30
facet= sns.FacetGrid(df,hue="Group", aspect=3)
facet.map(sns.kdeplot,'MMSE',shade= True)
facet.set(xlim=(0, df['MMSE'].max()))
facet.add_legend()
plt.xlim(15.30)
```

Out[5]:
```
(15.3, 30.0)
```

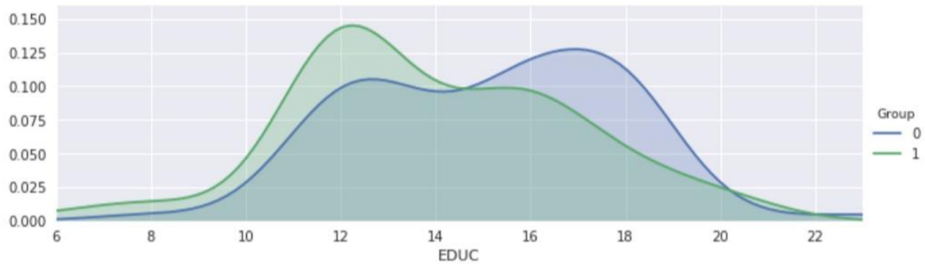


In [8]:

```
#'EDUC' = Years of Education
# Nondemented = 0, Demented =1
facet= sns.FacetGrid(df,hue="Group", aspect=3)
facet.map(sns.kdeplot,'EDUC',shade= True)
facet.set(xlim=(df['EDUC'].min(), df['EDUC'].max()))
facet.add_legend()
plt.ylim(0, 0.16)
```

Out[8]:

```
(0, 0.16)
```



In [6]:

```
#bar_chart('ASF') = Atlas Scaling Factor
facet= sns.FacetGrid(df,hue="Group", aspect=3)
facet.map(sns.kdeplot,'ASF',shade= True)
facet.set(xlim=(0, df['ASF'].max()))
facet.add_legend()
plt.xlim(0.5, 2)

#eTIV = Estimated Total Intracranial Volume
facet= sns.FacetGrid(df,hue="Group", aspect=3)
facet.map(sns.kdeplot,'eTIV',shade= True)
facet.set(xlim=(0, df['eTIV'].max()))
facet.add_legend()
plt.xlim(900, 2100)

#'nWBV' = Normalized Whole Brain Volume
# Nondemented = 0, Demented =1
facet= sns.FacetGrid(df,hue="Group", aspect=3)
facet.map(sns.kdeplot,'nWBV',shade= True)
facet.set(xlim=(0, df['nWBV'].max()))
facet.add_legend()
plt.xlim(0.6,0.9)
```

Out[6]:

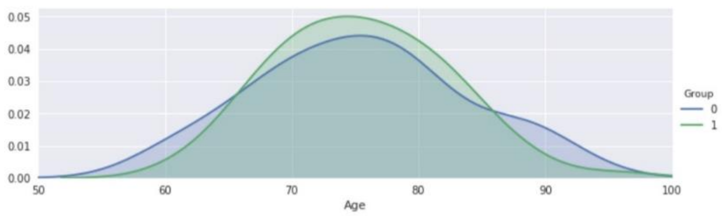
```
(0.6, 0.9)
```

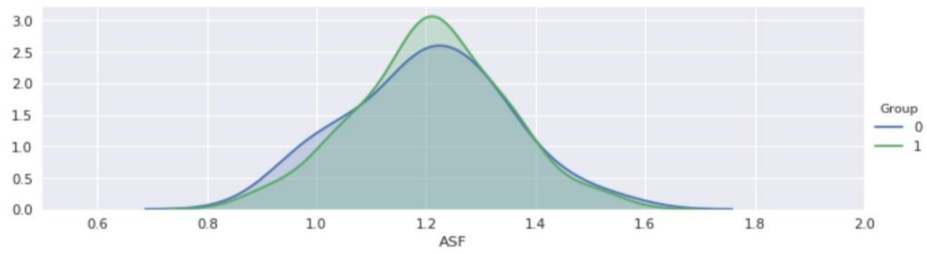
In [7]:

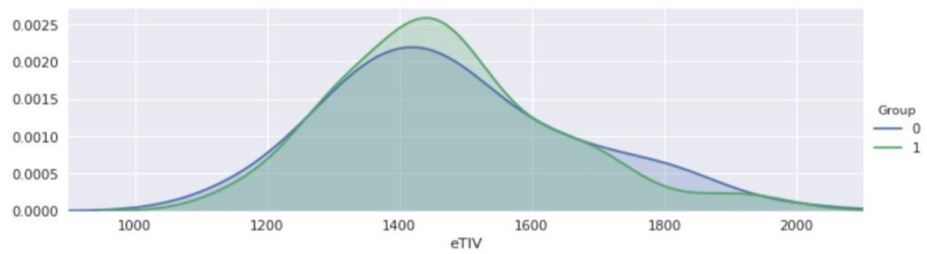
```
#AGE. Nondemented =0, Demented =0
facet= sns.FacetGrid(df,hue="Group", aspect=3)
facet.map(sns.kdeplot,'Age',shade= True)
facet.set(xlim=(0, df['Age'].max()))
facet.add_legend()
plt.xlim(50,100)
```

Out[7]:

```
(50, 100)
```







Intermediate Result Summary

1. Men are more likely with demented, an Alzheimer's Disease, than Women.
2. Demented patients were less educated in terms of years of education.
3. Nondemented group has higher brain volume than Demented group.

In [11]:
```
df_dropna['Group'].value_counts()
```

Out[11]:
```
0    72
1    70
Name: Group, dtype: int64
```

3.1 Splitting Train/Validation/Test Sets

In [16]:
```
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import cross_val_score
```

In [17]:
```
# Dataset with imputation
Y = df['Group'].values # Target for the model
X = df[['M/F', 'Age', 'EDUC', 'SES', 'MMSE', 'eTIV', 'nWBV', 'ASF']] # Features we use

# splitting into three sets
X_trainval, X_test, Y_trainval, Y_test = train_test_split(
    X, Y, random_state=0)

# Feature scaling
scaler = MinMaxScaler().fit(X_trainval)
X_trainval_scaled = scaler.transform(X_trainval)
X_test_scaled = scaler.transform(X_test)
```

```
In [18]:
# Dataset after dropping missing value rows
Y = df_dropna['Group'].values # Target for the model
X = df_dropna[['M/F', 'Age', 'EDUC', 'SES', 'MMSE', 'eTIV', 'nWBV', 'ASF']] # Features we use

# splitting into three sets
X_trainval_dna, X_test_dna, Y_trainval_dna, Y_test_dna = train_test_split(
    X, Y, random_state=0)

# Feature scaling
scaler = MinMaxScaler().fit(X_trainval_dna)
X_trainval_scaled_dna = scaler.transform(X_trainval_dna)
X_test_scaled_dna = scaler.transform(X_test_dna)
```

Cross-validation

We conduct 5-fold cross-validation to figure out the best parameters for each model, Logistic Regression, SVM, Decision Tree, Random Forests, and AdaBoost. Since our performance metric is accuracy, we find the best tuning parameters by *accuracy*. In the end, we compare the accuracy, recall and AUC for each model.

4.Model

4.A Logistic Regression

The parameter C, inverse of regularization strength. Tuning range: [0.001, 0.1, 1, 10, 100]

```
In [19]:
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, recall_score, roc_curve, auc
```

```
In [20]:
acc = [] # list to store all performance metric
```

```
In [21]:
# Dataset with imputation
best_score=0
kfolds=5 # set the number of folds

for c in [0.001, 0.1, 1, 10, 100]:
    logRegModel = LogisticRegression(C=c)
    # perform cross-validation
    scores = cross_val_score(logRegModel, X_trainval, Y_trainval, cv=kfolds, scoring='accuracy') # Get
recall for each parameter setting
```

```
    # compute mean cross-validation accuracy
    score = np.mean(scores)

    # Find the best parameters and score
    if score > best_score:
        best_score = score
        best_parameters = c

# rebuild a model on the combined training and validation set
SelectedLogRegModel = LogisticRegression(C=best_parameters).fit(X_trainval_scaled, Y_trainval)

test_score = SelectedLogRegModel.score(X_test_scaled, Y_test)
PredictedOutput = SelectedLogRegModel.predict(X_test_scaled)
test_recall = recall_score(Y_test, PredictedOutput, pos_label=1)
fpr, tpr, thresholds = roc_curve(Y_test, PredictedOutput, pos_label=1)
test_auc = auc(fpr, tpr)
print("Best accuracy on validation set is:", best_score)
print("Best parameter for regularization (C) is: ", best_parameters)
print("Test accuracy with best C parameter is", test_score)
print("Test recall with the best C parameter is", test_recall)
print("Test AUC with the best C parameter is", test_auc)
m = 'Logistic Regression (w/ imputation)'
acc.append([m, test_score, test_recall, test_auc, fpr, tpr, thresholds])
```

Best accuracy on validation set is: 0.754112554113
Best parameter for regularization (C) is: 10
Test accuracy with best C parameter is 0.789473684211
Test recall with the best C parameter is 0.75
Test AUC with the best C parameter is 0.791666666667

4.B SVM

In [23]:

```
best_score = 0

for c_paramter in [0.001, 0.01, 0.1, 1, 10, 100, 1000]: #iterate over the values we need to try for the
parameter C
    for gamma_paramter in [0.001, 0.01, 0.1, 1, 10, 100, 1000]: #iterate over the values we need to tr
y for the parameter gamma
        for k_parameter in ['rbf', 'linear', 'poly', 'sigmoid']: # iterate over the values we need to
try for the kernel parameter
            svmModel = SVC(kernel=k_parameter, C=c_paramter, gamma=gamma_paramter) #define the model
            # perform cross-validation
            scores = cross_val_score(svmModel, X_trainval_scaled, Y_trainval, cv=kfolds, scoring='accu
racy')
            # the training set will be split internally into training and cross validation
```

```python
            # compute mean cross-validation accuracy
            score = np.mean(scores)
            # if we got a better score, store the score and parameters
            if score > best_score:
                best_score = score #store the score
                best_parameter_c = c_paramter #store the parameter c
                best_parameter_gamma = gamma_paramter #store the parameter gamma
                best_parameter_k = k_parameter


# rebuild a model with best parameters to get score
SelectedSVMmodel = SVC(C=best_parameter_c, gamma=best_parameter_gamma, kernel=best_parameter_k).fit(X_
trainval_scaled, Y_trainval)

test_score = SelectedSVMmodel.score(X_test_scaled, Y_test)
PredictedOutput = SelectedSVMmodel.predict(X_test_scaled)
test_recall = recall_score(Y_test, PredictedOutput, pos_label=1)
fpr, tpr, thresholds = roc_curve(Y_test, PredictedOutput, pos_label=1)
test_auc = auc(fpr, tpr)
print("Best accuracy on cross validation set is:", best_score)
print("Best parameter for c is: ", best_parameter_c)
print("Best parameter for gamma is: ", best_parameter_gamma)
print("Best parameter for kernel is: ", best_parameter_k)
print("Test accuracy with the best parameters is", test_score)
print("Test recall with the best parameters is", test_recall)
print("Test recall with the best parameter is", test_auc)

m = 'SVM'
acc.append([m, test_score, test_recall, test_auc, fpr, tpr, thresholds])
```

Best accuracy on cross validation set is: 0.77150385846
Best parameter for c is: 100
Best parameter for gamma is: 0.1
Best parameter for kernel is: rbf
Test accuracy with the best parameters is 0.815789473684
Test recall with the best parameters is 0.7
Test recall with the best parameter is 0.822222222222

4.C Random Forest Classifier

n_estimators(M),max_features(d), max_depth(m)

```python
In [27]:
best_score = 0

for M in range(2, 15, 2): # combines M trees
    for d in range(1, 9): # maximum number of features considered at each split
        for m in range(1, 9): # maximum depth of the tree
            # train the model
            # n_jobs(4) is the number of parallel computing
            forestModel = RandomForestClassifier(n_estimators=M, max_features=d, n_jobs=4,
                                                 max_depth=m, random_state=0)

            # perform cross-validation
            scores = cross_val_score(forestModel, X_trainval_scaled, Y_trainval, cv=kfolds, scori
ng='accuracy')
```

Best accuracy on validation set is: 0.796329757199
Best parameters of M, d, m are: 14 5 7
Test accuracy with the best parameters is 0.842105263158
Test recall with the best parameters is: 0.8
Test AUC with the best parameters is: 0.844444444444

4.D AdaBoost

In [29]:

```
best_score = 0

for M in range(2, 15, 2): # combines M trees
    for lr in [0.0001, 0.001, 0.01, 0.1, 1]:
        # train the model
        boostModel = AdaBoostClassifier(n_estimators=M, learning_rate=lr, random_state=0)

        # perform cross-validation
        scores = cross_val_score(boostModel, X_trainval_scaled, Y_trainval, cv=kfolds, scoring='accur
acy')

        # compute mean cross-validation accuracy
        score = np.mean(scores)

        # if we got a better score, store the score and parameters
        if score > best_score:                                                                          inv
            best_score = score
            best_M = M
            best_lr = lr

# Rebuild a model on the combined training and validation set
SelectedBoostModel = AdaBoostClassifier(n_estimators=M, learning_rate=lr, random_state=0).fit(X_train
val_scaled, Y_trainval )

PredictedOutput = SelectedBoostModel.predict(X_test_scaled)
test_score = SelectedRFModel.score(X_test_scaled, Y_test)
test_recall = recall_score(Y_test, PredictedOutput, pos_label=1)
fpr, tpr, thresholds = roc_curve(Y_test, PredictedOutput, pos_label=1)
test_auc = auc(fpr, tpr)
  m = 'Random Forest'
  acc.append([m, test_score, test_recall, test_auc, fpr, tpr, thresholds])
```

```
print("Best accuracy on validation set is:", best_score)
print("Best parameter of M is: ", best_M)
print("best parameter of LR is: ", best_lr)
print("Test accuracy with the best parameter is", test_score)
print("Test recall with the best parameters is:", test_recall)
print("Test AUC with the best parameters is:", test_auc)

m = 'AdaBoost'
acc.append([m, test_score, test_recall, test_auc, fpr, tpr, thresholds])
```

Best accuracy on validation set is: 0.778543195935
Best parameter of M is: 2
best parameter of LR is: 0.0001
Test accuracy with the best parameter is 0.842105263158
Test recall with the best parameters is: 0.65
Test AUC with the best parameters is: 0.825

5. Results

In [31]:

```
# Performance Metric for each model
result = pd.DataFrame(acc, columns=['Model', 'Accuracy', 'Recall', 'AUC', 'FPR', 'TPR', 'TH'])
result[['Model', 'Accuracy', 'Recall', 'AUC']]
```

	Regression			
2	SVM	0.815789	0.70	0.822222
4	Random Forest	0.842105	0.80	0.844444
5	AdaBoost	0.842105	0.65	0.825000

Limitation

There are limitations in implementing a complex model because of the quantity of the dataset. Even though the nature of each feature is evident, the ranges of each group's test value are not classified well. In other words, we should have identified more clearly the differences in the variables which might have played a role in the result. The predicted value using the random forest model is higher than the other models. It implies there is a potential for higher prediction rate if we pay more attention to develop the data cleaning and analysis process.

Financial Yield

One way that businesses can profit from the neurotech industry is by developing and commercializing new medical devices and treatments. These devices and treatments can help patients with neurological conditions such as epilepsy, Parkinson's disease, and traumatic brain injuries, among others. Companies that develop effective and innovative treatments for these conditions can generate significant revenue from sales and licensing fees.

Another way that businesses can profit from the neurotech industry is by developing and selling brain-computer interfaces (BCIs) and other neural devices. BCIs allow individuals to control devices such as prosthetic limbs or computer programs using their thoughts. These devices have a wide range of potential applications, including assistive technology, gaming, and virtual reality. As the technology continues to advance, the market for BCIs and other neural devices is expected to grow significantly.

In addition to medical devices and BCIs, businesses in the neurotech industry can also profit from developing and selling software and data analytics tools. These tools can be used to analyse brain activity and data, providing valuable insights into neurological conditions and how they can be treated.

Overall, the neurotech industry presents a significant opportunity for businesses to develop innovative technologies and treatments that can improve the lives of individuals with neurological conditions while also generating profits for their companies.

Futures Prospects for Neurotech Startups

There is a significant need for neurotech startups in the industry as they can bring innovation, creativity, and agility to the development of new neurotech products and solutions. Here are some reasons why:

1. Address Unmet Needs: Neurological disorders and conditions can be highly complex, and existing solutions may not be effective or accessible for all patients.
2. Flexibility and Agility: Startups are more agile and can adapt more quickly to market changes and technological advancements, allowing them to stay ahead of the competition and provide better solutions.
3. Attracting Investment: Neurotech startups often attract significant investment from venture capital firms, angel investors, and strategic partners, providing them with the necessary resources to develop their solutions and bring them to market.
4. Collaboration and Partnerships: Startups can collaborate with other companies, research institutions, and medical centers to develop and test their products and solutions, which can lead to more significant breakthroughs and advancements.
5. Disrupting Existing Markets: Neurotech startups can disrupt existing markets by offering more affordable, accessible, and effective solutions for neurological disorders and conditions.
6. Advancing Research: Neurotech startups can play a significant role in advancing research in the field by developing new technologies, collecting and analysing data, and collaborating with researchers and clinicians to better understand neurological disorders and conditions.

References

https://www.iiba.org/globalassets/standards-and-resources/business-data-analytics/files/introduction-to-bda-practitioner-2.pdf

https://github.com/NeuroTech-UCSD/Neural-Data-Analysis-Notebooks