CZ1015 PROJECT PRESENTATION

By Ernest, Jia Yuan, Yow Lim

EPILEPSY CLASSIFICATION

Taking one step further in tackling this mental health issue

PROBLEM BACKGROUND

DATASET SELECTED

Epilepsy Dataset

PROBLEM DEFINITON

To identify if a person is having a seizure or not based on a sample of his EEG readings (binary classification)

Epilepsy is recognised as a major medical and social problem in Singapore, there is no epidemiological survey to realise the size of the problem. There is a life-time prevalence of 3.8 per 1000

According to epilepsy.sg:

- 10% of people with epilepsy expressed strained family relationships, citing embarrassment, financial strain and being a burden to spouse and family members as chief reasons.
- 20% also admitted to difficulty making friends or maintaining a relationship at work or in social gatherings. Low self-esteem, fear of avoidance and embarrassment were among the common reasons.
- Almost 42% also chose not to divulge their medical condition to their friends. 49 - 53% of responders cite resentment, depression and anxiety as their main psychological barriers.

Epilepsy Dataset Analysis:

	Unnamed: 0	X1	X2	X3	X4	X5	X6	X7	X8	X9	...	X170	X171	X172	X173	X174	X175	X176	X177	X178	y
0	X21.V1.791	135	190	229	223	192	125	55	-9	-33	...	-17	-15	-31	-77	-103	-127	-116	-83	-51	4
1	X15.V1.924	386	382	356	331	320	315	307	272	244	...	164	150	146	152	157	156	154	143	129	1
2	X8.V1.1	-32	-39	-47	-37	-32	-36	-57	-73	-85	...	57	64	48	19	-12	-30	-35	-35	-36	5
3	X16.V1.60	-105	-101	-96	-92	-89	-95	-102	-100	-87	...	-82	-81	-80	-77	-85	-77	-72	-69	-65	5
4	X20.V1.54	-9	-65	-98	-102	-78	-48	-16	0	-21	...	4	2	-12	-32	-41	-65	-83	-89	-73	5

5 rows × 180 columns

Description of dataset: This EEG dataset includes 4097 electroencephalograms (EEG) readings per patient over 23.5 seconds, with 500 patients in total from one point. The 4097 data points were then divided equally into 23 chunks per patient; each chunk is translated into one row in the dataset. Each row contains 178 readings that are turned into columns; in other words, there are 178 columns that make up one second of EEG readings. There are 11,500 rows and 180 columns with the first being patient ID and the last column containing the status of the patient, whether the patient is having a seizure or not.

In [2]:
```
data['Seizure Status'] = ""

for i in range(11500):
    if data.iloc[i ,179] == 1:
        data.iloc[i, 180] = 1
    else:
        data.iloc[i, 180] = 0

## df.rename({'y':'Seizure Status'}, axis = 1, inplace = True)
## df.drop(df.columns[0], axis = 1, inplace = True)
data.head()
```

Out[2]:

	Unnamed: 0	X1	X2	X3	X4	X5	X6	X7	X8	X9	...	X171	X172	X173	X174	X175	X176	X177	X178	y	Seizure Status
0	X21.V1.791	135	190	229	223	192	125	55	-9	-33	...	-15	-31	-77	-103	-127	-116	-83	-51	4	0
1	X15.V1.924	386	382	356	331	320	315	307	272	244	...	150	146	152	157	156	154	143	129	1	1
2	X8.V1.1	-32	-39	-47	-37	-32	-36	-57	-73	-85	...	64	48	19	-12	-30	-35	-35	-36	5	0
3	X16.V1.60	-105	-101	-96	-92	-89	-95	-102	-100	-87	...	-81	-80	-77	-85	-77	-72	-69	-65	5	0
4	X20.V1.54	-9	-65	-98	-102	-78	-48	-16	0	-21	...	2	-12	-32	-41	-65	-83	-89	-73	5	0

5 rows × 181 columns

Epilepsy Dataset Analysis (Uni-Variate):

```
In [3]: y = pd.DataFrame(data["Seizure Status"])
        x = pd.DataFrame(data[["X1"]])
```

```
In [4]: from sklearn.model_selection import train_test_split
        x_train, x_test, y_train, y_test = train_test_split(x, y,random_state=0, test_size = 0.25)
        print("Train Set :", y_train.shape, x_train.shape)
        print("Test Set  :", y_test.shape, x_test.shape)
```

```
Train Set : (8625, 1) (8625, 1)
Test Set  : (2875, 1) (2875, 1)
```

```
In [5]: y_train["Seizure Status"].value_counts()
```

```
Out[5]: 0    6888
        1    1737
        Name: Seizure Status, dtype: int64
```

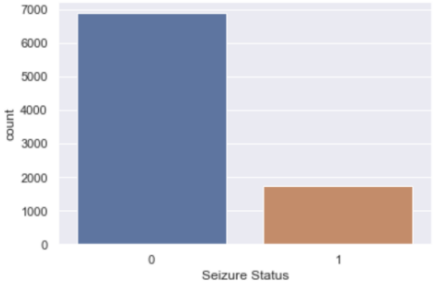
```
In [6]: x_train.describe()
```

Out[6]:

	X1
count	8625.000000
mean	-11.793159
std	169.216176
min	-1839.000000
25%	-54.000000
50%	-7.000000
75%	35.000000
max	1726.000000

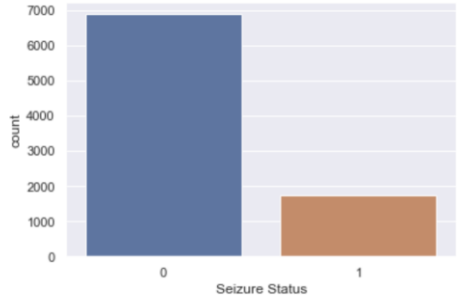
```
In [7]: sb.countplot(y_train["Seizure Status"])
```

```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x1a28db44d0>
```

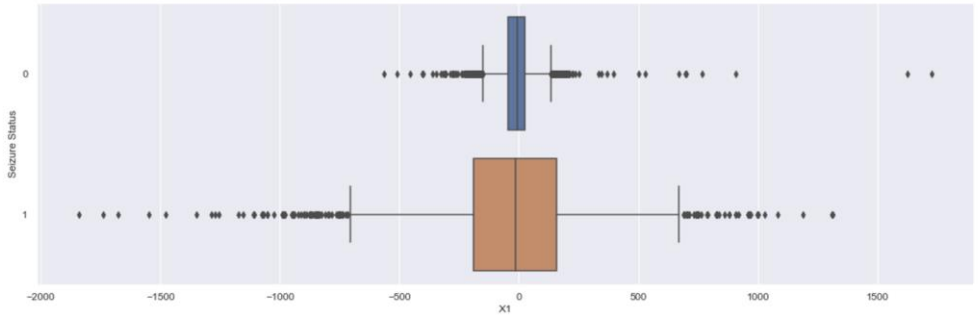


Epilepsy Dataset Analysis (Uni-Variate):

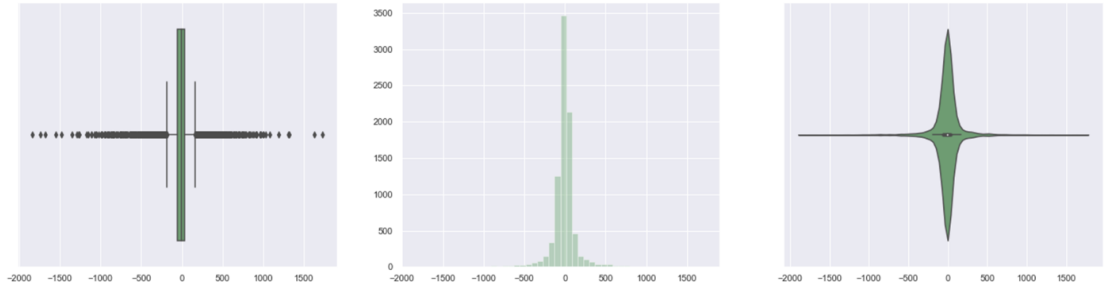
```
In [7]: sb.countplot(y_train["Seizure Status"])
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x1a28db44d0>
```




```
In [9]: jointDF = pd.concat([x_train, y_train.reindex(index=x_train.index)], sort = False, axis = 1)

        f, axes = plt.subplots(1, 1, figsize=(18, 6))
        sb.boxplot(x = "X1", y = "Seizure Status", data = jointDF, orient = "h")
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x1a2943bbd0>
```




```
In [8]: f, axes = plt.subplots(1, 3, figsize=(24, 6))
        sb.boxplot(x_train, orient = "h", ax = axes[0], color = "g")
        sb.distplot(x_train, kde = False, ax = axes[1], color = "g")
        sb.violinplot(x_train, ax = axes[2], color = "g")
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x1a28eb4cd0>
```



Epilepsy Dataset Analysis (Multi-Variate):

```
[ ] ym = pd.DataFrame(data['Seizure Status'])
    xm = pd.DataFrame(data.iloc[:,1:179])
    xm.head()
```

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	...	X169	X170	X171	X172	X173	X174	X175	X176	X177	X178
0	135	190	229	223	192	125	55	-9	-33	-38	...	8	-17	-15	-31	-77	-103	-127	-116	-83	-51
1	386	382	356	331	320	315	307	272	244	232	...	168	164	150	146	152	157	156	154	143	129
2	-32	-39	-47	-37	-32	-36	-57	-73	-85	-94	...	29	57	64	48	19	-12	-30	-35	-35	-36
3	-105	-101	-96	-92	-89	-95	-102	-100	-87	-79	...	-80	-82	-81	-80	-77	-85	-77	-72	-69	-65
4	-9	-65	-98	-102	-78	-48	-16	0	-21	-59	...	10	4	2	-12	-32	-41	-65	-83	-89	-73

5 rows × 178 columns

```
[ ] xm_train, xm_test, ym_train, ym_test = train_test_split(xm, ym, test_size = 0.25)
```

```
[ ] ym_train["Seizure Status"].value_counts()
```

```
0    6912
1    1713
Name: Seizure Status, dtype: int64
```

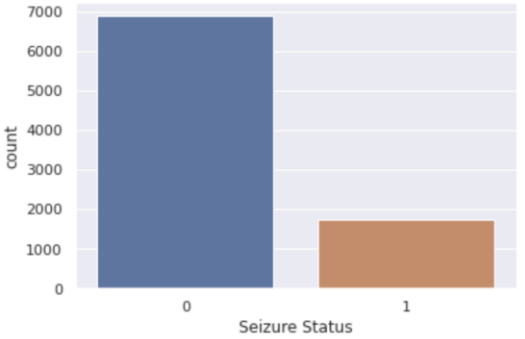
```
[ ] xm_train.describe()
```

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	...
count	8625.000000	8625.000000	8625.000000	8625.000000	8625.000000	8625.000000	8625.000000	8625.000000	8625.000000	8625.000000	...
mean	-11.023768	-9.896000	-8.774725	-7.647536	-6.613565	-5.761739	-5.580406	-6.266551	-6.899362	-7.240000	...
std	162.579408	163.116876	160.541535	158.373799	158.716967	159.663864	160.349972	161.295425	161.579309	159.523013	...
min	-1741.000000	-1587.000000	-1741.000000	-1845.000000	-1791.000000	-1743.000000	-1832.000000	-1778.000000	-1840.000000	-1867.000000	...
25%	-54.000000	-54.000000	-53.000000	-54.000000	-53.000000	-54.000000	-53.000000	-55.000000	-55.000000	-54.000000	...
50%	-8.000000	-7.000000	-7.000000	-7.000000	-8.000000	-8.000000	-8.000000	-8.000000	-7.000000	-7.000000	...
75%	34.000000	35.000000	35.000000	36.000000	35.000000	36.000000	36.000000	36.000000	37.000000	35.000000	...
max	1726.000000	1713.000000	1697.000000	1612.000000	1518.000000	1816.000000	2047.000000	2047.000000	2047.000000	2047.000000	...

8 rows × 178 columns

```
sb.countplot(ym_train["Seizure Status"])
```

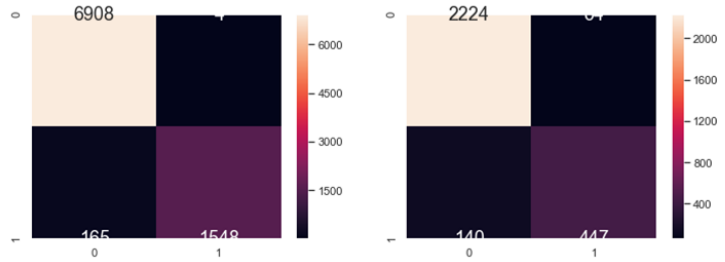
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8cec909c88>
```



Classification Model: Multivariate Decision Tree

```
In [0]: dectreexm = DecisionTreeClassifier(max_depth = 10, random_state =0)
        dectreexm.fit(xm_train, ym_train)

        ym_train_pred = dectreexm.predict(xm_train)
        ym_test_pred = dectreexm.predict(xm_test)

        treedot = export_graphviz(dectreexm, feature_names = xm_train.columns, class_names = True, out_file = None,
                                  filled = True, rounded = True, special_characters = True)


        graphviz.Source(treedot)
```

```
Goodness of Fit of Model        Train Dataset
Classification Accuracy         : 0.9804057971014493

Goodness of Fit of Model        Test Dataset
Classification Accuracy         : 0.9290434782608695

Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0x2c2b4aefbc8>
```




```
from sklearn.metrics import f1_score
print("F1 Score of Train Data \t:", f1_score(ym_train, ym_train_pred))
print("F1 Score of Test Data \t:", f1_score(ym_test, ym_test_pred))
```

```
F1 Score of Train Data   : 0.9439366240097501
F1 Score of Test Data    : 0.8287292817679558
```

- F1 Score is another metric to evaluate the model's predictions
- F1 Score is a weighted average of two other metrics: precision and recall
- Useful for uneven classification and when true negatives are not as important
- Precision: false positives
- Recall: false negatives

Classification Model: Multivariate Decision Tree

In [0]:
```
data_pred = data[data.iloc[:,0].isin(["X21.V1.791", "X15.V1.924", "X8.V1.1"])]
data_pred
```

Out[22]:

	Unnamed: 0	X1	X2	X3	X4	X5	X6	X7	X8	X9	...	X171	X172	X173	X174	X175	X176	X177	X178	y	Seizure Status
0	X21.V1.791	135	190	229	223	192	125	55	-9	-33	...	-15	-31	-77	-103	-127	-116	-83	-51	4	0
1	X15.V1.924	386	382	356	331	320	315	307	272	244	...	150	146	152	157	156	154	143	129	1	1
2	X8.V1.1	-32	-39	-47	-37	-32	-36	-57	-73	-85	...	64	48	19	-12	-30	-35	-35	-36	5	0

3 rows × 181 columns

In [0]:
```
xm_pred = pd.DataFrame(data_pred.iloc[:,1:179])
ym_pred = dectreexm.predict(xm_pred)
ym_pred
```

Out[23]:
```
array([0, 1, 0], dtype=int64)
```

In [0]:
```
ym_pred = pd.DataFrame(ym_pred, columns = ["Predict Seizure Status"], index = data_pred.index)
ym_combined = pd.concat([data_pred.iloc[:,0], data_pred[["Seizure Status"]], ym_pred], axis = 1)

ym_combined
```

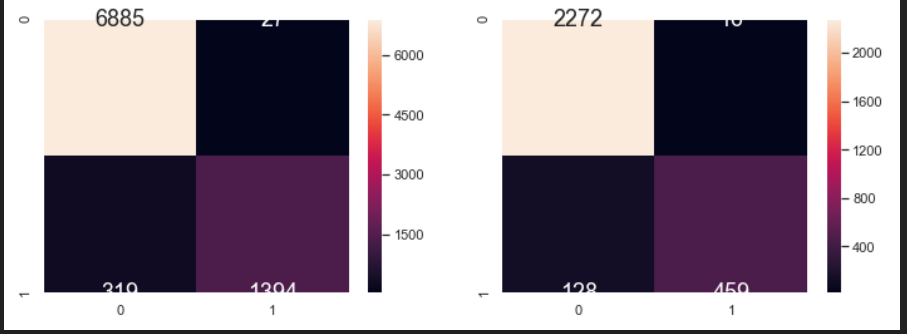
Out[24]:

	Unnamed: 0	Seizure Status	Predict Seizure Status
0	X21.V1.791	0	0
1	X15.V1.924	1	1
2	X8.V1.1	0	0

Classification Model: Random Forest

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(max_depth=6, random_state=0)
clf.fit(xm_train, ym_train)
```

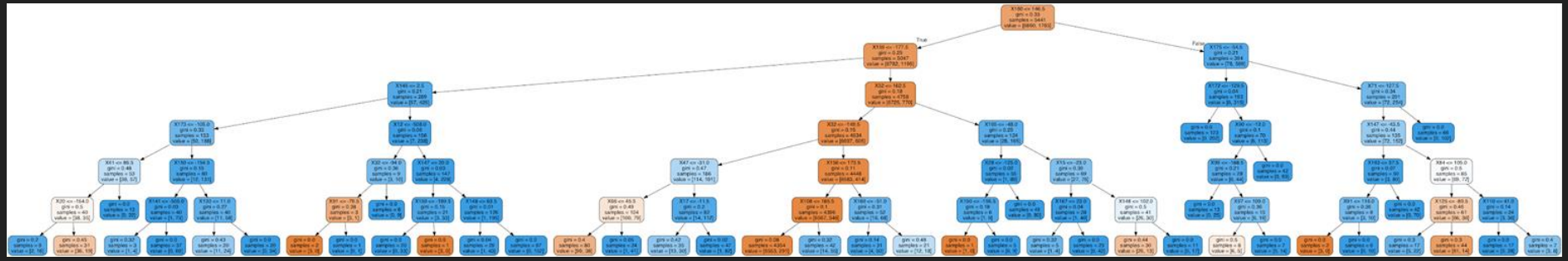
```
Goodness of Fit of Model        Train Dataset
Classification Accuracy         : 0.9598840579710145

Goodness of Fit of Model        Test Dataset
Classification Accuracy         : 0.9499130434782609

<matplotlib.axes._subplots.AxesSubplot at 0x2c2b8473c88>
```

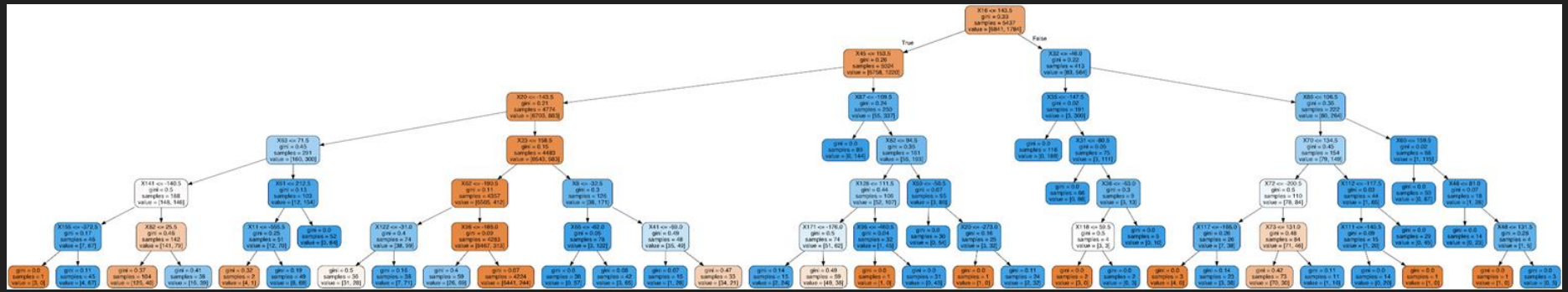



```
F1 Score of Train Data   : 0.8888888888888888
F1 Score of Test Data    : 0.8587677725118483
```

- Random forests use multiple decision trees that have low correlation with each other
- Bootstrapping
- Each tree splits nodes by the “best” feature from a random subset of all features
- More variation between trees, reduced overfitting
- Higher accuracy and F1 score for test data than simple decision tree

Classification Model: Random Forest





Classification Model: Extra Trees Classifier (ETC)

```
In [38]: from sklearn.ensemble import ExtraTreesClassifier
         etc = ExtraTreesClassifier(n_estimators=100, random_state=0)
         etc.fit(xm_train, ym_train)
```

```
Goodness of Fit of Model        Train Dataset
Classification Accuracy         : 1.0

Goodness of Fit of Model        Test Dataset
Classification Accuracy         : 0.9784347826086957
```

```
F1 Score of Train Data  : 1.0
F1 Score of Test Data   : 0.9256938227394808
```

- ETC is like random forests in that it uses multiple decision trees
- Each tree splits nodes using a random feature from a random subset of all features
- No bootstrapping
- Further reduced overfitting
- Even higher accuracy and F1 Score

Thank You!