

HAP 780 Final Project Report in Epileptic Seizure Detection

Merin Joy – G01118158

Volgenau School of Engineering, George Mason University, 4400 University Dr,
Fairfax, VA, 22030

Abstract

The study is about the classification and detection of epileptic seizure using the Electroencephalogram (EEG) time series dataset. An EEG time series, particularly those of short duration, can carry enough information to reveal dynamic properties of the underlying system brain. This study however, compares dynamical properties of brains electrical activity from different recording regions and different physiological and pathological brain states. The dataset is collected from five different sets and merged together with the help of SQL tools in the form suitable for the machine learning tools such as Weka. Data is converged in a way such that the resultant contains all the independent and dependent variables for classification learning. In this study, models are created on the training dataset and later concluded on which models performs the best detection of the epileptic seizure on the time series dataset using some significant measures of accuracy.

Key Words: Epileptic seizure, Electroencephalogram(EEG) signal, Structured Query Language(SQL), Weka, Logistic model, Naïve Bayes, Random Forest, J48 trees.

1. Introduction

A seizure is a sudden, uncontrolled electrical disturbance in the brain. It can lead to changes in your behavior, movements or feelings and in levels of consciousness. An epileptic seizure is a brief episode of signs or symptoms due to abnormally excessive or synchronous neuronal activity in the brain.

Electroencephalogram (EEG) signal is a representative signal containing information about the condition or state of the brain. The shape of the wave corresponds to the state of the brain. These kind of signals are highly subjective, the symptoms may appear at random in the time scale. The EEG signal parameters extracted and analyzed are highly useful in diagnostics. Certain chaotic measures like correlation dimension(CD), Hurst exponent(H) and entropy can be used to characterize the signal. These measures can help distinguish between a normal, alcoholic or epileptic EEG signals.

The EEG time series data is used in this study to classify the dynamical properties of brains electrical activity from different extracranial and intracranial recording regions. In this context, the EEG readings were obtained by the implantation of electrodes to localize the seizure generating area which can be termed as the epileptogenic zone. The EEG recordings during epileptic seizures called ictal activity, is periodic and has high amplitude caused from hypersynchronous activity of large assemblies of neurons.

2. About the data

The dataset was obtained from the University of Bonn, Department of Epileptology. The data was collected for their study of indications of nonlinear deterministic and finite dimensional structures in the time series of brain electrical activity, dependence on recording region and brain state. The sampling rate of the data was 173.61Hz. The time series have a spectral bandwidth of the acquisition system, which is 0.5Hz to 85Hz.

The main aspect of this classification was based on how the data was selected and recorded. The data is divided into five different sets (Set A-E), each set representing a condition the person was in while taking the EEG reading. For Set A and Set B, the EEG recordings were carried out on healthy volunteers. Volunteers had their eyes open, while taking the EEG reading for Set A, whereas volunteers had their eyes closed for Set B, while taking the EEG reading. For Set C, D and E, patients having epilepsy were asked to volunteer. For Set C, EEG recording from the healthy part of the brain for epileptic patients was taken. For Set D, EEG was recorded from the epileptogenic zone, i.e., place where the tumor is located and for Set E the seizure activity was recorded.

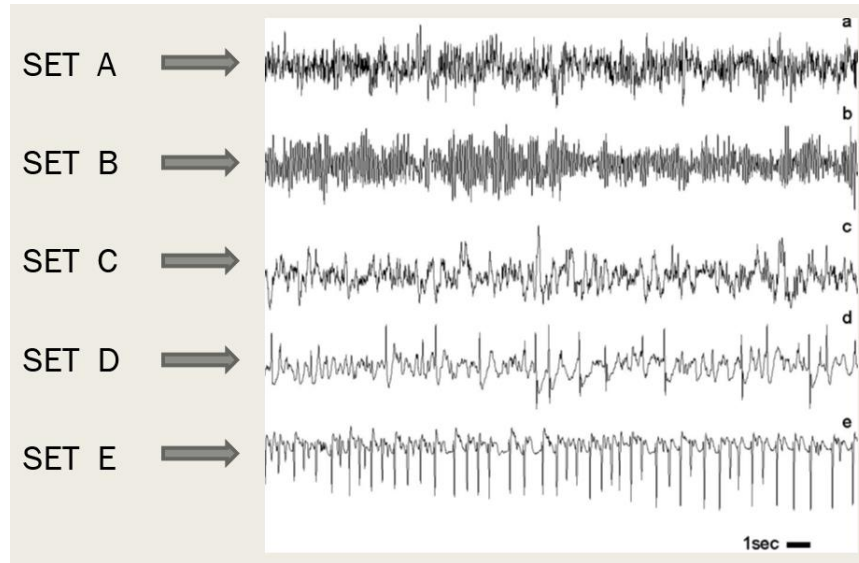


Fig. 2.1

The EEG time series dataset is available as a ZIP-file in five different sets, each containing 100 text files each. Each file representing a single person. Also, each file is a recording of brain activity of duration 23.6 seconds. Each person's corresponding time series is sampled into 4097 data points. Each data point is the value of the EEG recording at a different point in time. In total, the dataset contains 500 individuals each having 4097 recordings or data points for a duration of 23.5 seconds.

SET A Z.zip with Z000.txt - Z100.txt (564 kB)
SET B O.zip with O000.txt - O100.txt (611 kB)
SET C N.zip with N000.txt - N100.txt (560 kB)
SET D F.zip with F000.txt - F100.txt (569kB)
SET E S.zip with S000.txt - S100.txt (747kB)

Fig. 2.2

3. Data Preprocessing

The main motive of data preprocessing is to determine what is to be achieved from the dataset. For data preparation the goal is to combine the data points in such a way that the unit of analysis is per person per second. Therefore, in total 500 individuals times 23 seconds will give 11500 records and each data point is a value of EEG recording at a different point in time. Since, each set has 100 files containing 4097 data points for a duration of 23.6 seconds. In order to create the unit of analysis per person per second, 4097 data points is divided into 23 chunks (23.6 seconds) with each chunk containing 178 data points. These 178 data points represent a record for 1 second.

3.1 Independent and Dependent variable

Using the set categories, defining the dependent variable by adding the output attribute Y –{1,2,3,4,5} where Y represents each Set

Set A - 5, Set B - 4, Set C - 3, Set D - 2, Set E - 1

Since each chunk or record contains 178 data points for each second, the target table needs 178 input attributes. Therefore, independent variables or input attributes X1,X2,X3...X178 are added where each X represents a data point per chunk.

3.2 Data preprocessing in SQL

In order to arrange the data points in 23 different chunks for each set and combine them with the dependent and independent variables, SQL queries are used. Following are the steps for data preprocessing in SQL:

Step 1: Adding a column name to each file. In this case, starting with Set A.

```
--add column names  
sp_rename '[Set A].[dbo].[Z001].[Column 0]', 'SET-A1', 'COLUMN';
```

	SET-A1
1	12
2	22
3	35
4	45
5	69
6	74
7	79
8	78
9	66
10	43
11	33
12	36
13	34
14	38

Fig. 3.2.1

Step 2: Adding a unique row number to the records.

```
--add row number
SELECT "SET-A1", ROW_NUMBER() OVER(ORDER BY (SELECT NULL)) AS Rownum
into [Set A].[dbo].AZ001
FROM [Set A].[dbo].[Z001];
```

	SET-A1	Rownum
1	12	1
2	22	2
3	35	3
4	45	4
5	69	5
6	74	6
7	79	7
8	78	8
9	66	9
10	43	10
11	33	11
12	36	12
13	34	13
14	38	14
15	36	15
16	28	16
17	6	17

Fig. 3.2.2

Step 3: Adding chunk numbers for 23 chunks.

```
--add chunk numbers
SELECT Rownum, "SET-A1",ntile(23) over(ORDER BY Rownum) AS My_Chunks
into [Set A].[dbo].AZ001_chunks
FROM [Set A].[dbo].AZ001
```

	Rownum	SET-A1	My_Chunks
1	1	12	1
2	2	22	1
3	3	35	1
4	4	45	1
5	5	69	1
6	6	74	1
7	7	79	1
8	8	78	1
9	9	66	1
10	10	43	1
11	11	33	1
12	12	36	1
13	13	34	1

Fig. 3.2.3

Step 4: Adding batch number for input attributes.

```
--add batch number
select t.*, 1 + ((ROW_NUMBER() over (order by Rownum) - 1) % 178) as
new_batch_no
into [Set A].[dbo].AZ001_chunks_batch
from [Set A].[dbo].AZ001_chunks t
```

	Rownum	SET-A1	My_Chunks	new_batch_no
1	1	12	1	1
2	2	22	1	2
3	3	35	1	3
4	4	45	1	4
5	5	69	1	5
6	6	74	1	6
7	7	79	1	7
8	8	78	1	8
9	9	66	1	9
10	10	43	1	10
11	11	33	1	11
12	12	36	1	12
13	13	34	1	13
14	14	38	1	14

Fig. 3.2.4

Step 5: Converting the previous table into the format below for 500 files by adding the independent and dependent variable. Input attribute is $X1 \rightarrow X178$ and output variable is Y , in this case $Y=5$ (Set A). The query below converts the 4097 records to 23 chunks or records.

```
--Conversion to 23 records
SELECT My_Chunks,
MAX(CASE WHEN new_batch_no = 1 THEN "SET-A1" ELSE NULL END) AS X1,
MAX(CASE WHEN new_batch_no = 2 THEN "SET-A1" ELSE NULL END) AS X2,
MAX(CASE WHEN new_batch_no = 3 THEN "SET-A1" ELSE NULL END) AS X3,
MAX(CASE WHEN new_batch_no = 4 THEN "SET-A1" ELSE NULL END) AS X4,
MAX(CASE WHEN new_batch_no = 5 THEN "SET-A1" ELSE NULL END) AS X5,
MAX(CASE WHEN new_batch_no = 6 THEN "SET-A1" ELSE NULL END) AS X6,
MAX(CASE WHEN new_batch_no = 7 THEN "SET-A1" ELSE NULL END) AS X7,
MAX(CASE WHEN new_batch_no = 8 THEN "SET-A1" ELSE NULL END) AS X8,
MAX(CASE WHEN new_batch_no = 9 THEN "SET-A1" ELSE NULL END) AS X9,
MAX(CASE WHEN new_batch_no = 10 THEN "SET-A1" ELSE NULL END) AS X10,
MAX(CASE WHEN new_batch_no = 11 THEN "SET-A1" ELSE NULL END) AS X11,
MAX(CASE WHEN new_batch_no = 12 THEN "SET-A1" ELSE NULL END) AS X12,
MAX(CASE WHEN new_batch_no = 13 THEN "SET-A1" ELSE NULL END) AS X13,
MAX(CASE WHEN new_batch_no = 14 THEN "SET-A1" ELSE NULL END) AS X14,
MAX(CASE WHEN new_batch_no = 15 THEN "SET-A1" ELSE NULL END) AS X15,
MAX(CASE WHEN new_batch_no = 16 THEN "SET-A1" ELSE NULL END) AS X16,
MAX(CASE WHEN new_batch_no = 17 THEN "SET-A1" ELSE NULL END) AS X17,
MAX(CASE WHEN new_batch_no = 18 THEN "SET-A1" ELSE NULL END) AS X18,
MAX(CASE WHEN new_batch_no = 19 THEN "SET-A1" ELSE NULL END) AS X19,
MAX(CASE WHEN new_batch_no = 20 THEN "SET-A1" ELSE NULL END) AS X20,
MAX(CASE WHEN new_batch_no = 21 THEN "SET-A1" ELSE NULL END) AS X21,
MAX(CASE WHEN new_batch_no = 22 THEN "SET-A1" ELSE NULL END) AS X22,
MAX(CASE WHEN new_batch_no = 23 THEN "SET-A1" ELSE NULL END) AS X23,
MAX(CASE WHEN new_batch_no = 24 THEN "SET-A1" ELSE NULL END) AS X24,
MAX(CASE WHEN new_batch_no = 25 THEN "SET-A1" ELSE NULL END) AS X25,
MAX(CASE WHEN new_batch_no = 26 THEN "SET-A1" ELSE NULL END) AS X26,
MAX(CASE WHEN new_batch_no = 27 THEN "SET-A1" ELSE NULL END) AS X27,
MAX(CASE WHEN new_batch_no = 28 THEN "SET-A1" ELSE NULL END) AS X28,
MAX(CASE WHEN new_batch_no = 29 THEN "SET-A1" ELSE NULL END) AS X29,
MAX(CASE WHEN new_batch_no = 30 THEN "SET-A1" ELSE NULL END) AS X30,
MAX(CASE WHEN new_batch_no = 31 THEN "SET-A1" ELSE NULL END) AS X31,
MAX(CASE WHEN new_batch_no = 32 THEN "SET-A1" ELSE NULL END) AS X32,
MAX(CASE WHEN new_batch_no = 33 THEN "SET-A1" ELSE NULL END) AS X33,
MAX(CASE WHEN new_batch_no = 34 THEN "SET-A1" ELSE NULL END) AS X34,
MAX(CASE WHEN new_batch_no = 35 THEN "SET-A1" ELSE NULL END) AS X35,
```

MAX	(CASE WHEN	new_batch_no	= 36	THEN	"SET-A1"	ELSE	NULL	END)	AS	X36,
MAX	(CASE WHEN	new_batch_no	= 37	THEN	"SET-A1"	ELSE	NULL	END)	AS	X37,
MAX	(CASE WHEN	new_batch_no	= 38	THEN	"SET-A1"	ELSE	NULL	END)	AS	X38,
MAX	(CASE WHEN	new_batch_no	= 39	THEN	"SET-A1"	ELSE	NULL	END)	AS	X39,
MAX	(CASE WHEN	new_batch_no	= 40	THEN	"SET-A1"	ELSE	NULL	END)	AS	X40,
MAX	(CASE WHEN	new_batch_no	= 41	THEN	"SET-A1"	ELSE	NULL	END)	AS	X41,
MAX	(CASE WHEN	new_batch_no	= 42	THEN	"SET-A1"	ELSE	NULL	END)	AS	X42,
MAX	(CASE WHEN	new_batch_no	= 43	THEN	"SET-A1"	ELSE	NULL	END)	AS	X43,
MAX	(CASE WHEN	new_batch_no	= 44	THEN	"SET-A1"	ELSE	NULL	END)	AS	X44,
MAX	(CASE WHEN	new_batch_no	= 45	THEN	"SET-A1"	ELSE	NULL	END)	AS	X45,
MAX	(CASE WHEN	new_batch_no	= 46	THEN	"SET-A1"	ELSE	NULL	END)	AS	X46,
MAX	(CASE WHEN	new_batch_no	= 47	THEN	"SET-A1"	ELSE	NULL	END)	AS	X47,
MAX	(CASE WHEN	new_batch_no	= 48	THEN	"SET-A1"	ELSE	NULL	END)	AS	X48,
MAX	(CASE WHEN	new_batch_no	= 49	THEN	"SET-A1"	ELSE	NULL	END)	AS	X49,
MAX	(CASE WHEN	new_batch_no	= 50	THEN	"SET-A1"	ELSE	NULL	END)	AS	X50,
MAX	(CASE WHEN	new_batch_no	= 51	THEN	"SET-A1"	ELSE	NULL	END)	AS	X51,
MAX	(CASE WHEN	new_batch_no	= 52	THEN	"SET-A1"	ELSE	NULL	END)	AS	X52,
MAX	(CASE WHEN	new_batch_no	= 53	THEN	"SET-A1"	ELSE	NULL	END)	AS	X53,
MAX	(CASE WHEN	new_batch_no	= 54	THEN	"SET-A1"	ELSE	NULL	END)	AS	X54,
MAX	(CASE WHEN	new_batch_no	= 55	THEN	"SET-A1"	ELSE	NULL	END)	AS	X55,
MAX	(CASE WHEN	new_batch_no	= 56	THEN	"SET-A1"	ELSE	NULL	END)	AS	X56,
MAX	(CASE WHEN	new_batch_no	= 57	THEN	"SET-A1"	ELSE	NULL	END)	AS	X57,
MAX	(CASE WHEN	new_batch_no	= 58	THEN	"SET-A1"	ELSE	NULL	END)	AS	X58,
MAX	(CASE WHEN	new_batch_no	= 59	THEN	"SET-A1"	ELSE	NULL	END)	AS	X59,
MAX	(CASE WHEN	new_batch_no	= 60	THEN	"SET-A1"	ELSE	NULL	END)	AS	X60,
MAX	(CASE WHEN	new_batch_no	= 61	THEN	"SET-A1"	ELSE	NULL	END)	AS	X61,
MAX	(CASE WHEN	new_batch_no	= 62	THEN	"SET-A1"	ELSE	NULL	END)	AS	X62,
MAX	(CASE WHEN	new_batch_no	= 63	THEN	"SET-A1"	ELSE	NULL	END)	AS	X63,
MAX	(CASE WHEN	new_batch_no	= 64	THEN	"SET-A1"	ELSE	NULL	END)	AS	X64,
MAX	(CASE WHEN	new_batch_no	= 65	THEN	"SET-A1"	ELSE	NULL	END)	AS	X65,
MAX	(CASE WHEN	new_batch_no	= 66	THEN	"SET-A1"	ELSE	NULL	END)	AS	X66,
MAX	(CASE WHEN	new_batch_no	= 67	THEN	"SET-A1"	ELSE	NULL	END)	AS	X67,
MAX	(CASE WHEN	new_batch_no	= 68	THEN	"SET-A1"	ELSE	NULL	END)	AS	X68,
MAX	(CASE WHEN	new_batch_no	= 69	THEN	"SET-A1"					

[illegible]


```

MAX(CASE WHEN new_batch_no = 154 THEN "SET-A1" ELSE NULL END) AS X154,
MAX(CASE WHEN new_batch_no = 155 THEN "SET-A1" ELSE NULL END) AS X155,
MAX(CASE WHEN new_batch_no = 156 THEN "SET-A1" ELSE NULL END) AS X156,
MAX(CASE WHEN new_batch_no = 157 THEN "SET-A1" ELSE NULL END) AS X157,
MAX(CASE WHEN new_batch_no = 158 THEN "SET-A1" ELSE NULL END) AS X158,
MAX(CASE WHEN new_batch_no = 159 THEN "SET-A1" ELSE NULL END) AS X159,
MAX(CASE WHEN new_batch_no = 160 THEN "SET-A1" ELSE NULL END) AS X160,
MAX(CASE WHEN new_batch_no = 161 THEN "SET-A1" ELSE NULL END) AS X161,
MAX(CASE WHEN new_batch_no = 162 THEN "SET-A1" ELSE NULL END) AS X162,
MAX(CASE WHEN new_batch_no = 163 THEN "SET-A1" ELSE NULL END) AS X163,
MAX(CASE WHEN new_batch_no = 164 THEN "SET-A1" ELSE NULL END) AS X164,
MAX(CASE WHEN new_batch_no = 165 THEN "SET-A1" ELSE NULL END) AS X165,
MAX(CASE WHEN new_batch_no = 166 THEN "SET-A1" ELSE NULL END) AS X166,
MAX(CASE WHEN new_batch_no = 167 THEN "SET-A1" ELSE NULL END) AS X167,
MAX(CASE WHEN new_batch_no = 168 THEN "SET-A1" ELSE NULL END) AS X168,
MAX(CASE WHEN new_batch_no = 169 THEN "SET-A1" ELSE NULL END) AS X169,
MAX(CASE WHEN new_batch_no = 170 THEN "SET-A1" ELSE NULL END) AS X170,
MAX(CASE WHEN new_batch_no = 171 THEN "SET-A1" ELSE NULL END) AS X171,
MAX(CASE WHEN new_batch_no = 172 THEN "SET-A1" ELSE NULL END) AS X172,
MAX(CASE WHEN new_batch_no = 173 THEN "SET-A1" ELSE NULL END) AS X173,
MAX(CASE WHEN new_batch_no = 174 THEN "SET-A1" ELSE NULL END) AS X174,
MAX(CASE WHEN new_batch_no = 175 THEN "SET-A1" ELSE NULL END) AS X175,
MAX(CASE WHEN new_batch_no = 176 THEN "SET-A1" ELSE NULL END) AS X176,
MAX(CASE WHEN new_batch_no = 177 THEN "SET-A1" ELSE NULL END) AS X177,
MAX(CASE WHEN new_batch_no = 178 THEN "SET-A1" ELSE NULL END) AS X178, 5 as
Y
into [Set A].[dbo].SETA1
FROM [Set A].[dbo].AZ001_chunks_batch
GROUP BY My_Chunks
ORDER BY My_Chunks

```

82 %

Results Messages

	X163	X164	X165	X166	X167	X168	X169	X170	X171	X172	X173	X174	X175	X176	X177	X178	Y
1	43	52	65	52	30	7	-4	-18	-32	-47	-53	-48	-40	-17	-23	-32	5
2	54	62	58	59	53	52	46	34	22	4	-18	-31	-27	-26	-21	-30	5
3	-23	-16	-8	-1	13	14	15	1	-21	-38	-44	-31	-17	4	35	59	5
4	55	40	23	11	5	-5	-3	-22	-47	-68	-85	-92	-96	-83	-73	-66	5
5	83	67	36	6	-11	-2	6	4	-5	-16	-29	-35	-21	3	35	66	5
6	66	77	68	43	19	-8	-25	-25	-13	-2	-15	-27	-44	-40	-21	3	5
7	-71	-36	-15	-2	-5	0	15	41	48	39	27	17	17	21	30	43	5
8	19	6	7	27	40	53	49	42	30	16	3	-17	-24	-20	-14	-27	5
9	-35	-48	-38	-25	-10	-5	-2	-2	0	-3	-18	-36	-41	-39	-30	-17	5
10	-37	-29	-39	-29	-8	13	14	9	11	17	39	54	54	34	29	31	5
11	24	5	-14	-22	-20	-9	9	27	29	31	33	42	46	56	59	42	5

Query executed successfully. DESKTOP-PGFLDSM (14.0 RTM) DESKTOP-PGFLDSM\Merin ... master 00:00:00 23 rows

Fig. 3.2.5

Step 6: Repeating the previous steps for all 100 files in the respective sets.

Step 7: Using UNION ALL to combine all 100 tables from one Set. Each Set will return 2300 records.

--Combining the 100 tables from one Set to a single table.

```
select *
INTO [Set A].[dbo].[A]
FROM
(select * from [Set A].[dbo].[SETA1] union all
select * from [Set A].[dbo].[SETA2] union all
select * from [Set A].[dbo].[SETA3] union all
select * from [Set A].[dbo].[SETA4] union all
select * from [Set A].[dbo].[SETA5] union all
select * from [Set A].[dbo].[SETA6] union all
select * from [Set A].[dbo].[SETA7] union all
select * from [Set A].[dbo].[SETA8] union all
select * from [Set A].[dbo].[SETA9] union all
select * from [Set A].[dbo].[SETA10] union all
select * from [Set A].[dbo].[SETA11] union all
select * from [Set A].[dbo].[SETA12] union all
select * from [Set A].[dbo].[SETA13] union all
select * from [Set A].[dbo].[SETA14] union all
select * from [Set A].[dbo].[SETA15] union all
select * from [Set A].[dbo].[SETA16] union all
select * from [Set A].[dbo].[SETA17] union all
select * from [Set A].[dbo].[SETA18] union all
select * from [Set A].[dbo].[SETA19] union all
select * from [Set A].[dbo].[SETA20] union all
select * from [Set A].[dbo].[SETA21] union all
select * from [Set A].[dbo].[SETA22] union all
select * from [Set A].[dbo].[SETA23] union all
select * from [Set A].[dbo].[SETA24] union all
select * from [Set A].[dbo].[SETA25] union all
select * from [Set A].[dbo].[SETA26] union all
select * from [Set A].[dbo].[SETA27] union all
select * from [Set A].[dbo].[SETA28] union all
select * from [Set A].[dbo].[SETA29] union all
select * from [Set A].[dbo].[SETA30] union all
select * from [Set A].[dbo].[SETA31] union all
select * from [Set A].[dbo].[SETA32] union all
select * from [Set A].[dbo].[SETA33] union all
select * from [Set A].[dbo].[SETA34] union all
select * from [Set A].[dbo].[SETA35] union all
select * from [Set A].[dbo].[SETA36] union all
select * from [Set A].[dbo].[SETA37] union all
select * from [Set A].[dbo].[SETA38] union all
select * from [Set A].[dbo].[SETA39] union all
select * from [Set A].[dbo].[SETA40] union all
select * from [Set A].[dbo].[SETA41] union all
select * from [Set A].[dbo].[SETA42] union all
select * from [Set A].[dbo].[SETA43] union all
select * from [Set A].[dbo].[SETA44] union all
select * from [Set A].[dbo].[SETA45] union all
select * from [Set A].[dbo].[SETA46] union all
select * from [Set A].[dbo].[SETA47] union all
select * from [Set A].[dbo].[SETA48] union all
select * from [Set A].[dbo].[SETA49] union all
```

[illegible]

```

--select *
--INTO [Set A].[dbo].[A]
--FROM
--(select * from [Set A].[dbo].[SETA1] union all
--select * from [Set A].[dbo].[SETA2] union all
--select * from [Set A].[dbo].[SETA3] union all
--select * from [Set A].[dbo].[SETA4] union all
--select * from [Set A].[dbo].[SETA5] union all
--select * from [Set A].[dbo].[SETA6] union all
--select * from [Set A].[dbo].[SETA7] union all
--select * from [Set A].[dbo].[SETA8] union all

```

	X163	X164	X165	X166	X167	X168	X169	X170	X171	X172	X173	X174	X175	X176	X177	X178	Y
1	43	52	65	52	30	7	-4	-18	-32	-47	-53	-48	-40	-17	-23	-32	5
2	83	67	36	6	-11	-2	6	4	-5	-16	-29	-35	-21	3	35	66	5
3	-35	-48	-38	-25	-10	-5	-2	-2	0	-3	-18	-36	-41	-39	-30	-17	5
4	44	48	41	23	0	-18	-39	-46	-48	-46	-50	-38	-28	-31	-33	-29	5
5	81	88	87	97	105	102	86	67	58	46	43	47	35	5	-22	-27	5
6	14	-11	-33	-38	-25	-14	-15	-27	-50	-72	-80	-65	-53	-42	-36	-34	5
7	17	33	3	-35	-83	-94	-78	-72	-68	-85	-99	-111	-88	-84	-91	-96	5
8	-68	-42	-15	-17	-24	-29	-19	-13	-29	-53	-73	-70	-48	-32	-20	-4	5
9	-59	-56	-47	-64	-72	-102	-124	-127	-110	-94	-88	-77	-70	-48	-24	-10	5
10	17	-8	-42	-76	-87	-82	-71	-46	-47	-51	-54	-36	3	36	53	27	5
11	7	-14	-51	-107	-149	-188	-204	-180	-148	-105	-71	-21	-9	7	-10	-47	5
12	10	13	0	-29	-55	-92	-113	-115	-113	-102	-92	-89	-94	-92	-72	-32	5
13	20	20	29	45	53	59	58	59	48	36	55	65	84	86	105	126	5
14	51	45	48	48	43	30	22	17	25	58	77	85	67	45	42	45	5

Fig. 3.2.6

Step 8: Repeat all the previous steps to the remaining Sets (B,C,D,E) by adding the respective Y values.

Step 9: Using UNION ALL to combine all 5 tables/Sets. 11500 records are returned.

--Combining all 5 tables/Sets

```

select *
INTO [Set A].[dbo].[SetABCDE]
FROM
(select * from [Set A].[dbo].[A] union all
select * from [Set B].[dbo].[B] union all
select * from [Set C].[dbo].[C] union all
select * from [Set D].[dbo].[D] union all
select * from [Set E].[dbo].[E] ) a

```

```

--select *
--INTO [Set A].[dbo].[SetABCDE]
--FROM
--(select * from [Set A].[dbo].[A] union all
--select * from [Set B].[dbo].[B] union all
--select * from [Set C].[dbo].[C] union all
--select * from [Set D].[dbo].[D] union all
--select * from [Set E].[dbo].[E] ) a
--select * from [Set A].[dbo].[SetABCDE] order by NEWID()

```

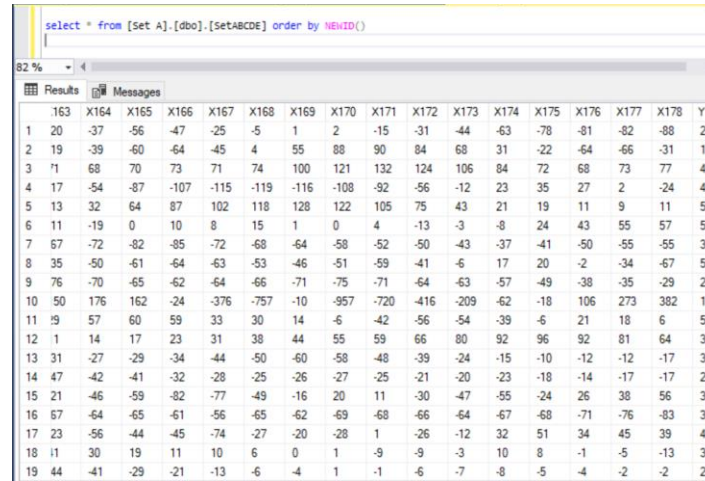
	X163	X164	X165	X166	X167	X168	X169	X170	X171	X172	X173	X174	X175	X176	X177	X178	Y
1	-62	-63	-26	32	102	167	184	159	114	80	59	44	18	-32	-94	-130	4
2	-2	-28	-38	-41	-45	-35	-10	23	64	88	44	-8	-11	7	41	50	4
3	-7	-1	-29	-16	-25	-14	-15	3	7	-10	-28	-50	-53	-53	-37	-51	3
4	-7	-12	-7	2	11	27	39	44	45	43	36	24	9	-1	-12	-25	2
5	15	7	-4	-9	-20	-29	-44	-59	-74	-75	-61	-46	-36	-22	-21	-29	3
6	-31	-34	-48	-31	-32	-18	-35	-41	-50	-57	-69	-63	-50	-38	-27	-33	2
7	-52	-61	-70	-76	-84	-86	-77	-61	-46	-26	-4	20	35	45	54	55	3
8	-2	5	56	76	83	83	55	33	-7	-24	-35	-50	-67	-68	-82	-69	3
9	-2	24	37	21	2	10	14	5	1	9	-3	-2	29	65	95	119	4
10	86	88	96	89	94	88	81	63	35	48	46	40	42	50	60	75	5
11	-185	-158	-142	-127	-110	-86	-58	-28	2	31	58	82	105	126	147	161	1
12	-1	5	-5	-8	-1	-15	-24	-25	-11	-11	1	-3	-15	-8	-11	-7	5
13	-2	-22	-49	-71	-72	-90	-95	-93	-93	-83	-78	-46	-2	13	17	-1	5
14	81	86	70	29	-3	-50	-96	-49	-44	-44	-125	-219	-265	-257	-212	-150	1

Query executed successfully. DESKTOP-PGFLDSM (14.0 RTM) DESKTOP-PGFLDSM\Merin ... master 00:00:13 11500 rows

Fig. 3.2.7

Step 10: Shuffle the 11500 records by using ORDER BY NEWID()

```
--Shuffling
select * from [Set A].[dbo].[SetABCDE] order by NEWID()
```



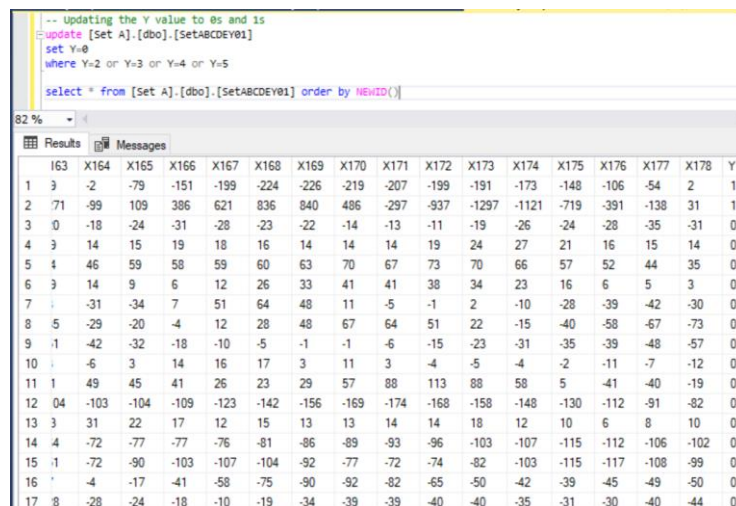
	X163	X164	X165	X166	X167	X168	X169	X170	X171	X172	X173	X174	X175	X176	X177	X178	Y
1	20	-37	-56	-47	-25	-5	1	2	-15	-31	-44	-63	-78	-81	-82	-88	2
2	19	-39	-60	-64	-45	4	55	88	90	84	68	31	-22	-64	-66	-31	1
3	1	68	70	73	71	74	100	121	132	124	106	84	72	68	73	77	4
4	17	-54	-87	-107	-115	-119	-116	-108	-92	-56	-12	23	35	27	2	-24	4
5	13	32	64	87	102	118	128	122	105	75	43	21	19	11	9	11	5
6	11	-19	0	10	8	15	1	0	4	-13	-3	-8	24	43	55	57	5
7	67	-72	-82	-85	-72	-68	-64	-58	-52	-50	-43	-37	-41	-50	-55	-55	3
8	35	-50	-61	-64	-63	-53	-46	-51	-59	-41	-6	17	20	-2	-34	-67	5
9	76	-70	-65	-62	-64	-66	-71	-75	-71	-64	-63	-57	-49	-38	-35	-29	2
10	50	176	162	-24	-376	-757	-10	-957	-720	-416	-209	-62	-18	106	273	382	1
11	9	57	60	59	33	30	14	-6	-42	-56	-54	-39	-6	21	18	6	5
12	1	14	17	23	31	38	44	55	59	66	80	92	96	92	81	64	3
13	31	-27	-29	-34	-44	-50	-60	-58	-48	-39	-24	-15	-10	-12	-12	-17	3
14	47	-42	-41	-32	-28	-25	-26	-27	-25	-21	-20	-23	-18	-14	-17	-17	2
15	21	-46	-59	-82	-77	-49	-16	20	11	-30	-47	-55	-24	26	38	56	3
16	67	-64	-65	-61	-56	-65	-62	-69	-68	-66	-64	-67	-68	-71	-76	-83	3
17	23	-56	-44	-45	-74	-27	-20	-28	1	-26	-12	32	51	34	45	39	4
18	1	30	19	11	10	6	0	1	-9	-9	-3	10	8	-1	-5	-13	3
19	44	-41	-29	-21	-13	-6	-4	1	-1	-6	-7	-8	-5	-4	-2	-2	2

Fig. 3.2.8

Classification learning is performed for two cases, one case in which all five Sets are classified and modeled, second case was taken for when the seizure activity was recorded against a seizure free interval. For the first case, we use the above obtained file for classification in 5 categories. For the second case, we update the Y value for Set E to 1, which was when the seizure was recorded and Y value for remaining Sets to 0 (seizure free intervals).

Step 11: Updating the dependent variable Y to 0s and 1s for the second type of classification.

```
-- Updating the Y value to 0s and 1s
update [Set A].[dbo].[SetABCDEY01]
set Y=0
where Y=2 or Y=3 or Y=4 or Y=5
```



	X163	X164	X165	X166	X167	X168	X169	X170	X171	X172	X173	X174	X175	X176	X177	X178	Y
1	3	-2	-79	-151	-199	-224	-226	-219	-207	-199	-191	-173	-148	-106	-54	2	1
2	71	-99	109	386	621	836	840	486	-297	-937	-1297	-1121	-719	-391	-138	31	1
3	0	-18	-24	-31	-28	-23	-22	-14	-13	-11	-19	-26	-24	-28	-35	-31	0
4	3	14	15	19	18	16	14	14	14	19	24	27	21	16	15	14	0
5	4	46	59	58	59	60	63	70	67	73	70	66	57	52	44	35	0
6	3	14	9	6	12	26	33	41	41	38	34	23	16	6	5	3	0
7	-31	-34	7	51	64	48	11	-5	-1	2	-10	-28	-39	-42	-30	0	0
8	5	-29	-20	-4	12	28	48	67	64	51	22	-15	-40	-58	-67	-73	0
9	-1	-42	-32	-18	-10	-5	-1	-1	-6	-15	-23	-31	-35	-39	-48	-57	0
10	-6	3	14	16	17	3	11	3	-4	-5	-4	-2	-11	-7	-12	0	0
11	49	45	41	26	23	29	57	88	113	88	58	5	-41	-40	-19	0	0
12	04	-103	-104	-109	-123	-142	-156	-169	-174	-168	-158	-148	-130	-112	-91	-82	0
13	3	31	22	17	12	15	13	13	14	14	18	12	10	6	8	10	0
14	4	-72	-77	-77	-76	-81	-86	-89	-93	-96	-103	-107	-115	-112	-106	-102	0
15	-1	-72	-90	-103	-107	-104	-92	-77	-72	-74	-82	-103	-115	-117	-108	-99	0
16	-4	-17	-41	-58	-75	-90	-92	-82	-65	-50	-42	-39	-45	-49	-50	0	0
17	8	-28	-24	-18	-10	-19	-34	-39	-39	-40	-40	-35	-31	-30	-40	-44	0

Fig. 3.2.9

Step 12: For classification learning dividing the dataset into training and testing.

```
--Adding a random number to sort
select [Set A].[dbo].[SetABCDEY01].* , abs(cast(cast(newid() as varbinary)
as int)) as RandomNumber
into [SET A].[dbo].splittable
from [Set A].[dbo].[SetABCDEY01]
order by RandomNumber asc;

--Extractig the top 80 percent
select top 80 percent *
into dbo.train
from [SET A].[dbo].splittable
order by RandomNumber

-- Extracting the remaining 20 percent
select *
into dbo.test
from (select * from [SET A].[dbo].splittable
except
select * from dbo.train) a
```

Saving the respective tables into a csv file suitable for classification learning and machine learning tools such as Weka. As discussed earlier, classification learning is performed in two different cases, one containing all five categories and the second kind of classification containing patients having recorded seizure versus non seizure recordings.

4. Classification Learning in Weka

4.1 Classification Learning 1

In this classification the dependent variable is divided into five different classes.

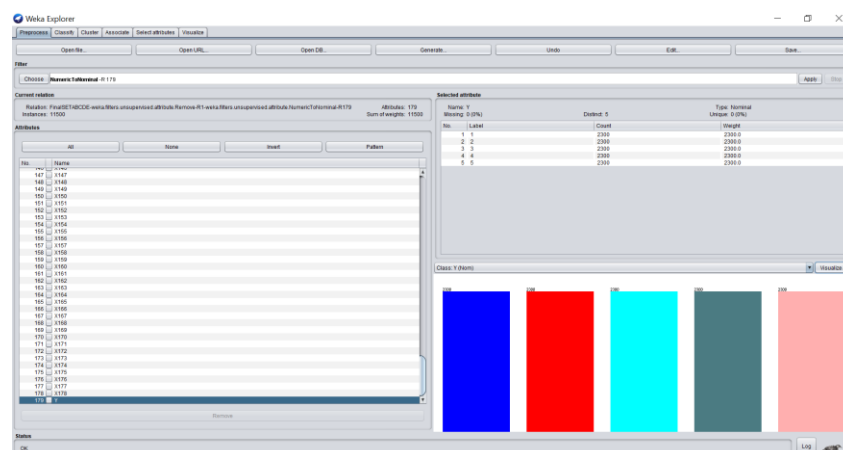


Fig. 4.1

The independent variables $X_1 \rightarrow X_{178}$ are a range of values that can be displayed with the help of a histogram below.

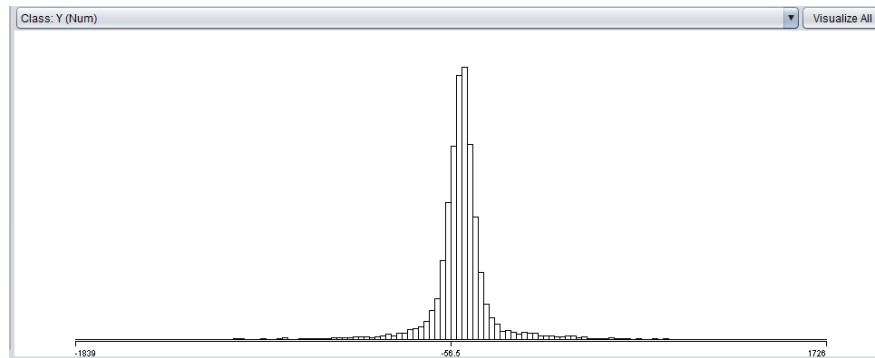


Fig. 4.2

The plot below describes beautifully that each set contains a range of values.

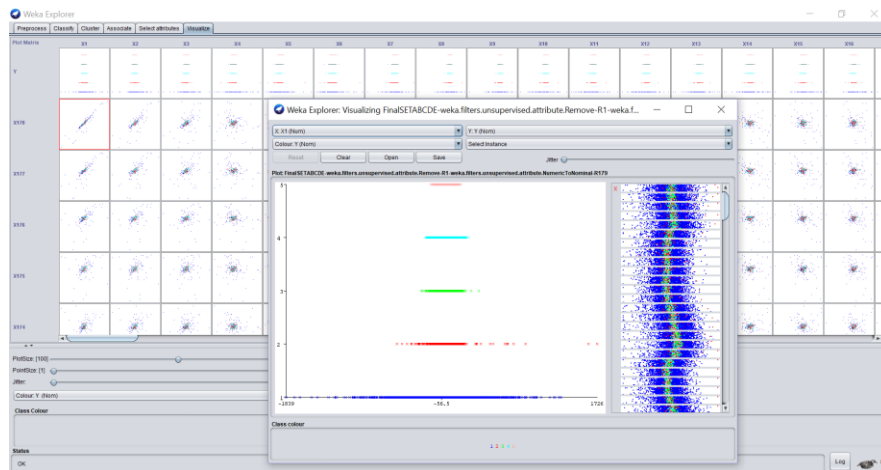


Fig. 4.3

4.1.1 Modeling

Logistic Model

Split the dataset into 75 percent training and run the logistic model. The ROC area (AUC) is obtained for different classes with the confusion matrix. The performance of the logistic model on this type of classification is fairly poor. With the ROC area values in the range of 0.50s the model does not perform well. Also, the precision and recall for the model is very less.

```

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.302    0.141    0.342     0.302    0.321     0.169    0.486    0.384    1
          0.148    0.176    0.178     0.148    0.161    -0.030    0.512    0.193    2
          0.176    0.152    0.226     0.176    0.198     0.026    0.526    0.198    3
          0.289    0.240    0.232     0.289    0.258     0.046    0.541    0.221    4
          0.281    0.243    0.221     0.281    0.247     0.035    0.536    0.200    5
Weighted Avg.    0.238    0.190    0.239     0.238    0.236     0.048    0.521    0.238

=== Confusion Matrix ===
  a  b  c  d  e  <-- classified as
170 105  62 122 103 | a = 1
 90  87  98 155 159 | b = 2
 69  98 102 148 164 | c = 3
 94  90  92 167 134 | d = 4
 74 109  97 127 159 | e = 5

```

Naïve Bayes Model

Split the dataset into 75 percent training and run the Naïve Bayes model. The ROC area (AUC) is obtained for different classes with the confusion matrix. Unlike, the logistic model, the performance of Naïve Bayes is better. The ROC area has increased when compared to logistic model and slightly increased for precision and recall.

```

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.826    0.015    0.932     0.826    0.875     0.850    0.987    0.920    1
          0.185    0.140    0.253     0.185    0.214     0.051    0.540    0.221    2
          0.194    0.122    0.288     0.194    0.232     0.085    0.626    0.281    3
          0.281    0.091    0.435     0.281    0.341     0.226    0.759    0.377    4
          0.742    0.330    0.355     0.742    0.481     0.333    0.761    0.328    5
Weighted Avg.    0.441    0.140    0.450     0.441    0.425     0.304    0.732    0.422

=== Confusion Matrix ===
  a  b  c  d  e  <-- classified as
464  93   0   5   0 | a = 1
 33 109  83  68 296 | b = 2
  1  86 113  88 293 | c = 3
  0 126 116 162 173 | d = 4
  0  16  81  49 420 | e = 5

```

J48 Tress model

Split the dataset into 75 percent training and run the J48 Tress model. The ROC area (AUC) is obtained for different classes with the confusion matrix. Compared to Naïve Bayes model ROC area for certain classes have increased whereas for certain classes it has decreased. Also, J48 trees model is giving better precision and recall than the previous models.

```

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.806    0.043    0.821     0.806    0.813     0.768    0.887    0.683    1
          0.346    0.173    0.341     0.346    0.343     0.172    0.589    0.270    2
          0.367    0.169    0.354     0.367    0.360     0.195    0.599    0.272    3
          0.484    0.131    0.481     0.484    0.482     0.352    0.682    0.336    4
          0.389    0.140    0.405     0.389    0.397     0.253    0.640    0.301    5
Weighted Avg.    0.476    0.132    0.478     0.476    0.477     0.345    0.678    0.370

=== Confusion Matrix ===
  a  b  c  d  e  <-- classified as
453  50  13  43   3 | a = 1
 40 204 173  75  97 | b = 2
 27 165 213  69 107 | c = 3
 30  72  80 279 116 | d = 4
  2 108 122 114 220 | e = 5

```


Random Forest model

Split the dataset into 75 percent training and run the Random Forest model. The ROC area (AUC) is considerably higher than the previous models. With very high precision and recall, the model performs really well for the dataset provided.

```

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      -----  -
      0.945    0.022    0.911     0.945    0.928     0.910    0.995     0.980     1
      0.514    0.078    0.629     0.514    0.566     0.471    0.871     0.631     2
      0.554    0.105    0.571     0.554    0.562     0.454    0.857     0.567     3
      0.757    0.068    0.736     0.757    0.746     0.682    0.946     0.846     4
      0.701    0.110    0.609     0.701    0.652     0.561    0.884     0.597     5
Weighted Avg. 0.692    0.077    0.690     0.692    0.689     0.613    0.910     0.723

=== Confusion Matrix ===

  a  b  c  d  e  <-- classified as
531  6  6  19  0 |  a = 1
31 303 164 30  61 |  b = 2
19 135 322 37  68 |  c = 3
 2  4  8 437 126 |  d = 4
 0 34  64 71 397 |  e = 5

```

4.2 Classification Learning 2

In this classification the dependent variable is divided into two different classes. Y=1 when the seizure is recorded, i.e. Set E and Y=0 for non-seizure recordings, i.e. Set A,B,C and D.

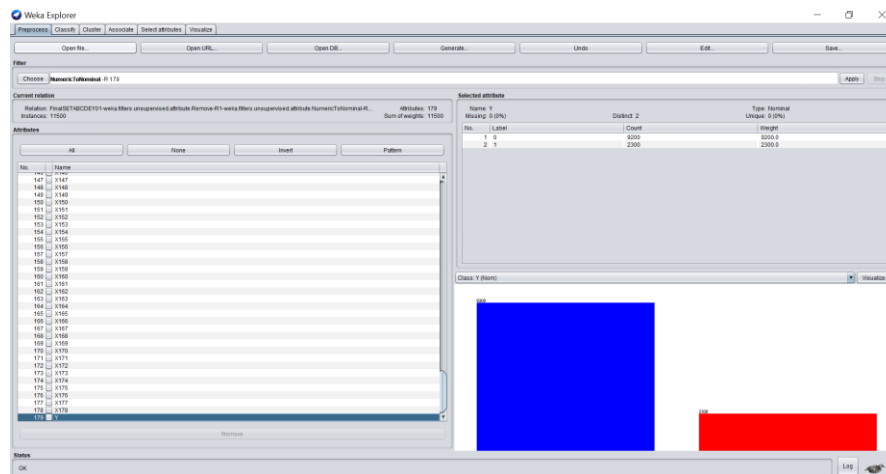


Fig. 4.2.1

The range of values the different classes in dependent variable contains is shown below.

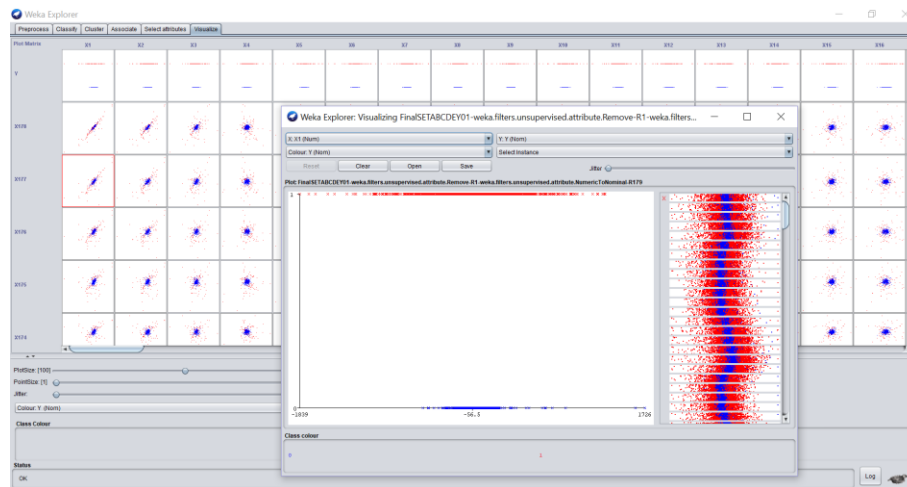


Fig 4.2.2

4.2.1 Modeling

Logistic Model

Using the data that was split in SQL into 80 percent training and 20 percent remaining for testing and run the logistic model. The ROC area (AUC) obtained is 0.544 which is fairly poor. Also, the main drawback of this model is that the recall is extremely low, 0.128. Looking at the confusion matrix we can see that 376 cases are claimed to not have recorded seizures whereas in reality they did.

```

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC    ROC Area  PRC Area  Class
          0.997   0.872   0.832    0.997   0.907    0.306   0.544   0.770    0
          0.128   0.003   0.917    0.128   0.224    0.306   0.544   0.439    1
Weighted Avg.   0.834   0.709   0.848    0.834   0.779    0.306   0.544   0.708

=== Confusion Matrix ===

   a    b  <-- classified as
1864    5 |    a = 0
 376   55 |    b = 1

```

J48 Tress model

Using the data that was split in SQL into 80 percent training and 20 percent remaining for testing and run the J48 Tress model. The ROC area (AUC) obtained is 0.888 which is good compared to the poor performance in the logistic model. Also, the models generates very good precision and recall.

```

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.973    0.160    0.963     0.973    0.968     0.826    0.888     0.956     0
          0.840    0.027    0.877     0.840    0.858     0.826    0.888     0.791     1
Weighted Avg.    0.948    0.135    0.947     0.948    0.947     0.826    0.888     0.925

=== Confusion Matrix ===
      a    b  <-- classified as
1818  51 |   a = 0
 69 362 |   b = 1

```

Random Forest model

Using the data that was split in SQL into 80 percent training and 20 percent remaining for testing and run the Random Forest model. The ROC area (AUC) is exceptionally good compared to all the other previous models. Also, the precision and recall is extremely good.

```

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.987    0.097    0.978     0.987    0.982     0.903    0.995     0.999     0
          0.903    0.013    0.940     0.903    0.921     0.903    0.995     0.979     1
Weighted Avg.    0.971    0.082    0.971     0.971    0.971     0.903    0.995     0.995

=== Confusion Matrix ===
      a    b  <-- classified as
1844  25 |   a = 0
 42 389 |   b = 1

```

Naïve Bayes model

Using the data that was split in SQL into 80 percent training and 20 percent remaining for testing and run the Naïve Bayes model. The ROC area (AUC) obtained is 0.984 which is very good. Also, the precision and recall for this model is very good, especially recall which a value of 0.910.

```

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.971    0.090    0.979     0.971    0.975     0.868    0.956     0.982     0
          0.910    0.029    0.877     0.910    0.893     0.868    0.984     0.892     1
Weighted Avg.    0.959    0.079    0.960     0.959    0.959     0.868    0.961     0.965

=== Confusion Matrix ===
      a    b  <-- classified as
1814  55 |   a = 0
 39 392 |   b = 1

```

5. Conclusion

Confusion matrix is a table that gives a performance evaluation of a classification model on a test dataset of which the true values are known. The matrix represents true positives, true negatives, false positives and false negatives.

In the Classification learning 1, Random Forest performs the best out of the remaining models. With a high precision, recall and AUC the Random Forest model can be used for the classification of Sets into five classes. In the Classification learning 2, false negatives would refer to the cases that fail to reject the null hypothesis by stating that the person does not have seizure when he/she actually does. Whereas, false positives would refer to the cases were rejecting the null hypothesis by stating that the person does not seizure when he/she actually does not. Therefore according to the severity of the situation, in this case, false negatives are worse than false positives. In terms of accuracy a better recall would determine a good model than having better AUC or precision. Hence, out of all the models Naïve Bayes model performs the best for Classification learning 2, having a maximum recall of 0.910.

	RECALL
NAÏVE BAYES	0.910
RANDOM FOREST	0.903
J48 TREES	0.840
LOGISTIC	0.128

6. Limitations

The only limitation dealt during this study was to combine files present into 500 different locations into one single table. As each table had to be converted into respective formats and converged into a single table. Hence, this procedure took up 80 percent of the study. Furthermore, an increase in data would cause data preparation issues leading in time consumption and errors.

References

- [1]. Andrzejak, Ralph G. (May 8, 2003). EEG times series download page. Retrieved from <http://epileptologie-bonn.de/cms/upload/workgroup/lehnertz/eegdata.html>
- [2]. Andrzejak RG, Lehnertz K, Rieke C, Mormann F, David P, Elger CE (2001) Indications of nonlinear deterministic and finite dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state, [Phys. Rev. E](#), 64, 061907
- [3]. Mayo Clinic. (n.d). Seizures. Retrieved from <https://www.mayoclinic.org/diseases-conditions/seizure/symptoms-causes/syc-20365711>
- [4]. Epileptic Seizures. (July 18, 2018).Retrieved from https://en.wikipedia.org/wiki/Epileptic_seizure
- [5]. Taylor, Courtney. (July 31, 2017). Type I and Type II Errors in Statistics. Retrieved from <https://www.thoughtco.com/type-i-error-vs-type-ii-error-3126410>
- [6]. Confusion Matrix. (n.d). Retrieved from http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html
- [7]. N, Kannathal. (October 2005). Characterization of EEG – a comparative study. Retrieved from <https://www.ncbi.nlm.nih.gov/pubmed/16099533>