A presentation for Syllabus implementation workshop on Artificial Intelligence & Machine Learning Lab sessions: Practical no. 3, 4, 6

Presentation · February 2022

DOI: 10.13140/RG.2.2.23842.91845

2 authors:

Abhishek D. Patange
ABB

Jegadeeshwaran R
VIT University

A presentation for
Syllabus implementation workshop
on
Artificial Intelligence & Machine Learning

Course Code: 302049

Lab sessions: Practical no. 3, 4, 6

Third Year Bachelor of Engineering (Choice Based Credit System)
Mechanical Engineering (2019 Course)
Board of Studies – Mechanical and Automobile Engineering, SPPU, Pune
(With Effect from Academic Year 2021-22)

by

Abhishek D. Patange, Ph.D.
Department of Mechanical Engineering
College of Engineering Pune (COEP)

Practical no. 3, 4, 6:

3. To extract features from given data set and establish training data
4. To select relevant features using suitable technique
6. To classify features/To develop classification model and evaluate its performance (any one classifier).

Practical no. 3:

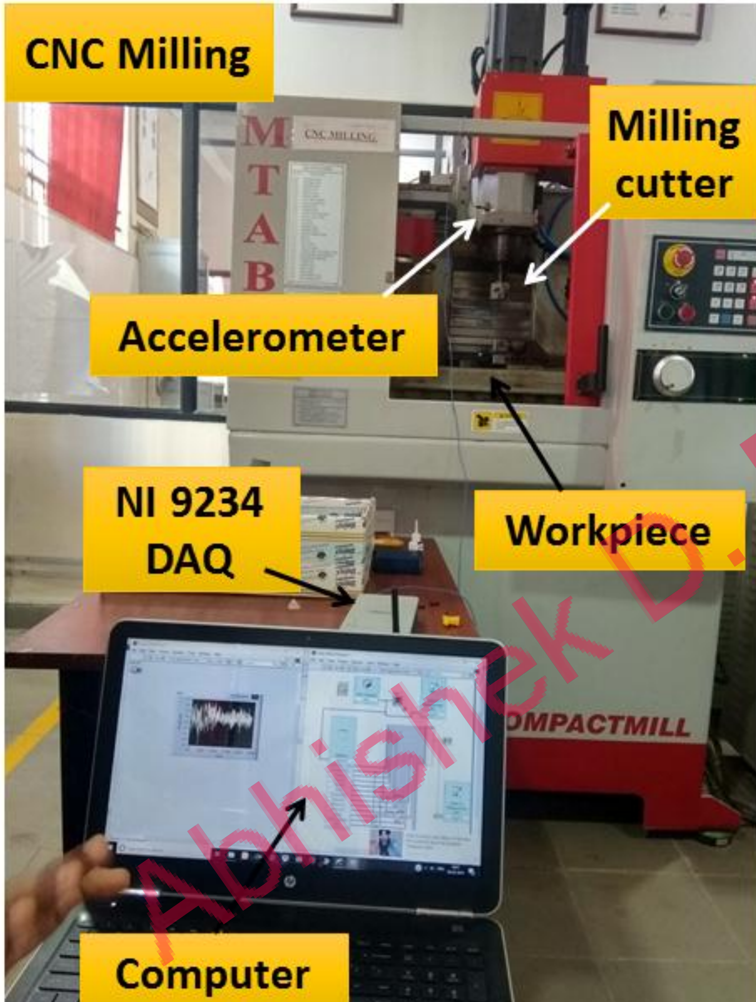
3. To extract features from given data set and establish training data

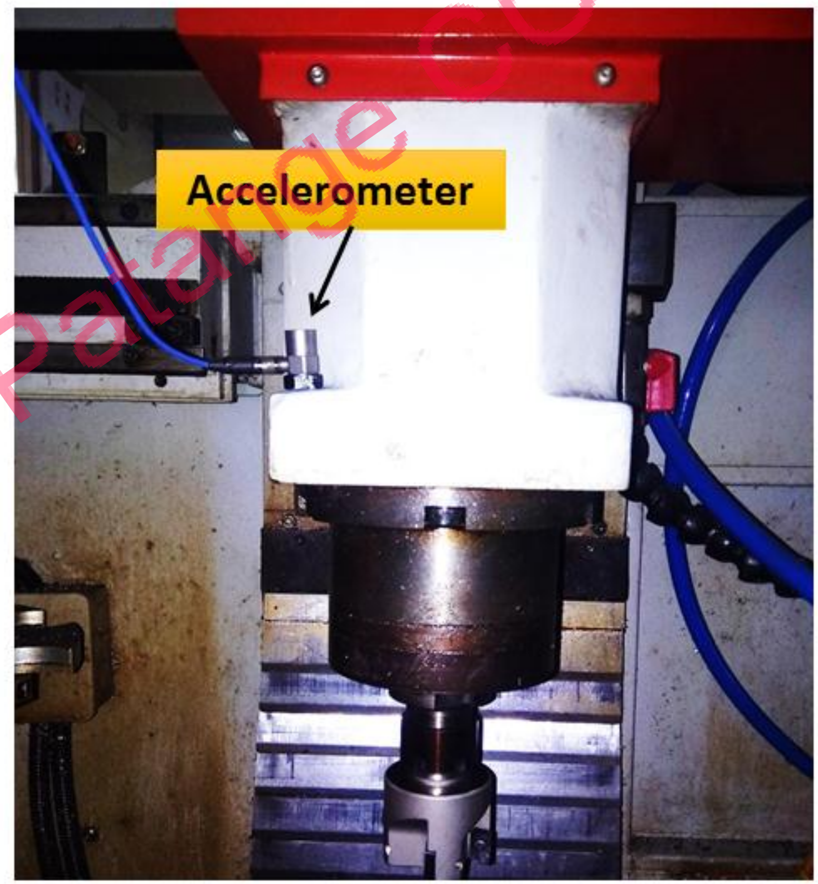
Prerequisites

- **Sensors, DAQs**
- **Development of experimental setup or simulation**
- **Design of experiment**
- **Data collection through experiment or simulation**
- **Save .csv or .xlsx files**

Data collection, preparation, pre-processing:

Collection of data: vibration signals for monitoring milling tool health





Data collection, preparation, pre-processing:

Collection of data: vibration signals for monitoring milling tool health

- Machine: CNC Milling Make: MTAB Compact mill
- Cutting tool: Face milling cutter diameter 16 mm with 4 inserts
- Workpiece: Cast Iron, Machining operation: Face milling, Name of institute: VIT CC
- Data acquisition system: NI 9234 DAQ, Accelerometer: Piezoelectric sensitivity 10.26 mV/g
- Machining parameters: Speed: 900 rpm, Feed: 2000 mm/min, DOC: 0.25 mm

Op. No.	1	2	3	4	Class label	File name
1	Normal	Normal	Normal	Normal	**ADF**	**111**
2	Normal	Normal	Normal	Flank wear	**FL**	**112**
3	Normal	Normal	Normal	Nose wear	**NS**	**113**
4	Normal	Normal	Normal	Notch wear	**NT**	**114**
5	Normal	Normal	Normal	Crater wear	**CT**	**115**
6	Normal	Normal	Normal	Edge fracture	**TB**	**116**
7	Normal	Normal	Normal	Built-up edge	**BE**	**117**
8	Flank wear	Nose wear	Notch wear	Crater wear	**AD**	**118**

Feature extraction: Descriptive statistics

Mean	$\bar{x} = \frac{x_1 + x_2 + x_3 + \ldots\ldots + x_n}{n}$
Median	$M = \left(\frac{n+1}{2}\right) th\ value$ (for ungrouped data) $M = L + \left(\frac{\frac{n}{2} - cf}{f}\right) \times h_m$ (for grouped data)
Mode	$M_o = l + \left(\frac{f_1 - f_0}{2f_1 - f_0 - f_2}\right) \times h_{m_o}$
Skewness	$\frac{n}{(n-1)(n-2)} \sum \left(\frac{x_i - \bar{x}}{\sigma}\right)^3$
Kurtosis	$\left\{\frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum \left(\frac{x_i - \bar{x}}{\sigma}\right)^4\right\} - \frac{3(n-1)^2}{(n-2)(n-3)}$

Standard Deviation	$\sigma = \sqrt{\frac{n\sum x^2 - (\sum x)^2}{n(n-1)}}$
Standard Error	$SE = \sqrt{\frac{1}{n-2} \Big(\sum (y_i - \bar{y})^2 - \frac{\sum[(x-\bar{x})(y_i - \ldots}{\sum (x-\bar{x})^2}}$
Variance	$\sigma^2 = \frac{\mathrm{n}\sum \mathrm{x}^2 - (\sum \mathrm{x})^2}{\mathrm{n}(\mathrm{n}-1)}$
Maximum value	$x_{max} = x > all\ other\ x\ in\ dataset$
Minimum value	$x_{min} = x < all\ other\ x\ in\ dataset$
Range	$R = x_{max} - x_{min}$

Feature extraction: Descriptive statistics

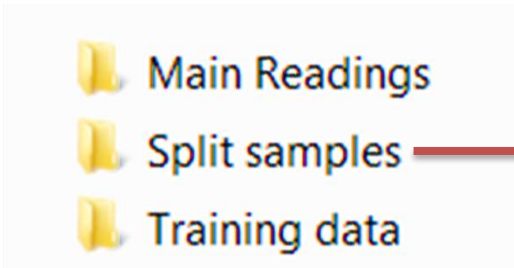
- Main Readings
- Split samples
- Training data

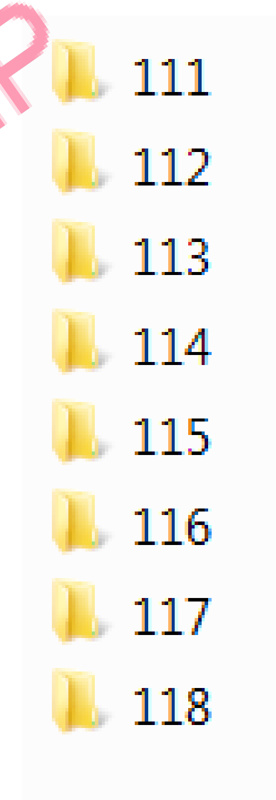
Insert 1	Insert 2	Insert 3	Insert 4	Class Label	File name
Normal	Normal	Normal	Normal	ADF	111
Normal	Normal	Normal	Flank wear	FL	112
Normal	Normal	Normal	Nose wear	NS	113
Normal	Normal	Normal	Notch wear	NT	114
Normal	Normal	Normal	Crater wear	CT	115
Normal	Normal	Normal	Edge fracture	TB	116
Normal	Normal	Normal	Built-up edge	BE	117
Flank wear	Nose wear	Notch wear	Crater wear	AD	118

- 111
- 112
- 113
- 114
- 115
- 116
- 117
- 118

Each file represents vibration data collected during face milling for 8 distinct tool configurations (class labels)
Each file contains near about 204800 data points

Feature extraction: Descriptive statistics





Instead of using 204800 data points in a single file and supplying it to algorithm input let's divide the file into samples.

Create 100 excel sheet by splitting 204800 data points so that we get 2048 data points in a single file.

So now 111 excel file will be represented in 100 excel sheets (Samples) such as 111 (1), 111 (2), 111 (3), ….. 111 (100)

Follow same procedure for all other files such as 112, …., 118

At the end 100 files for each class i.e. 800 files for 8 classes are created.

Feature extraction: Descriptive statistics

At the end 100 files for each class i.e. 800 files for 8 classes are created.

Development of training data

Statistical features

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Mean	Std. Error	Median	Mode	SD	Variance	Kurtosis	Skewness	Range	Min	Max	Sum	Count	Condition
2	0.00465	0.007406	0.00938	0.018399	0.335173	0.112341	8.152301	-0.21278	4.091093	-2.19478	1.896309	9.523034	2048	ADF
3	-0.00674	0.005898	-0.00463	0.197339	0.266926	0.07125	8.998373	-0.15533	3.684388	-1.9414	1.742983	-13.8091	2048	ADF
4	0.000963	0.007838	-0.00078	0.008899	0.354707	0.125817	7.06615	-0.04872	3.960375	-2.03701	1.923367	1.972552	2048	ADF
5	-0.00347	0.005994	-0.00283	0.061451	0.271262	0.073583	6.916217	0.0796	3.540082	-1.83787	1.702217	-7.10662	2048	ADF
6	0.003122	0.007775	0.000722	0.102698	0.351843	0.123794	7.673492	-0.09862	4.194873	-2.21078	1.984096	6.393745	2048	ADF
7	-0.00048	0.006331	-0.00204	-0.12939	0.28649	0.082076	6.820299	0.032983	3.573152	-1.76499	1.808162	-0.9897	2048	ADF
8	0.003948	0.007139	-0.00084	0.28717	0.323076	0.104378	7.722194	-0.28266	3.84517	-2.01284	1.832333	8.08562	2048	ADF
9	0.001029	0.007344	-0.00818	0.04221	0.332348	0.110455	8.297398	-0.18851	3.845651	-2.04338	1.802269	2.107357	2048	ADF
10	-0.00024	0.006475	-0.00667	0.102337	0.293034	0.085869	9.689591	0.010611	4.07077	-2.06839	2.002375	-0.48704	2048	ADF
11	0.000812	0.007772	-0.00331	-0.10498	0.351705	0.123696	7.028287	-0.09257	4.112498	-2.12203	1.990469	1.663374	2048	ADF
12	-0.00428	0.006181	-0.00625	-0.0956	0.279728	0.078248	9.936532	-0.37748	3.893272	-2.12347	1.7698	-8.76819	2048	ADF
13	-0.00023	0.006958	-0.00499	0.003848	0.314879	0.099149	6.661946	-0.20623	3.52529	-1.82812	1.697166	-0.46491	2048	ADF
14	-0.00193	0.006181	-0.00475	0.005291	0.279742	0.078256	10.77115	-0.10854	3.915399	-1.97868	1.936715	-3.95965	2048	ADF
15	0.005196	0.007919	0.002104	0.003968	0.358387	0.128441	6.989804	-0.13671	4.10865	-2.08198	2.026666	10.64177	2048	ADF
16	-0.00419	0.006	-0.00265	-0.06927	0.271548	0.073739	8.179651	-0.2281	3.701104	-1.96498	1.736129	-8.58312	2048	ADF
17	0.002671	0.007682	0.002706	-0.03848	0.347651	0.120861	7.707398	-0.14783	4.16505	-2.14139	2.02366	5.470785	2048	ADF
95	-0.001	0.006644	-0.00144	0.013469	0.300666	0.0904	7.572041	-0.17727	3.911311	-2.09665	1.814656	-2.04927	2048	ADF
96	-0.00039	0.007211	-0.00625	-0.01491	0.326325	0.106488	9.253425	-0.27132	3.994648	-2.20332	1.791326	-0.79789	2048	ADF
97	-0.00022	0.007227	-0.00319	0.068425	0.327045	0.106959	8.009849	-0.32182	3.998376	-2.19394	1.804434	-0.44302	2048	ADF
98	0.001287	0.00623	0.002826	-0.0362	0.281937	0.079489	8.490321	0.066596	3.641698	-1.89475	1.746952	2.636361	2048	ADF
99	0.003174	0.007566	0.001125	0.03752	0.342385	0.117227	7.739694	-0.10095	3.998857	-2.0251	1.973754	6.500051	2048	ADF
100	-0.00239	0.006079	-0.00547	-0.05616	0.275106	0.075683	8.918347	-0.06302	3.724433	-2.02991	1.694521	-4.88574	2048	ADF
101	0.003232	0.007285	0.00463	-0.01696	0.329677	0.108687	6.95185	-0.35186	3.725636	-2.0441	1.681533	6.619105	2048	ADF

Labels

No. of samples

Development of training data

Statistical features

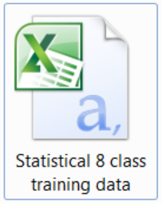
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Mean	Std. Error	Median	Mode	SD	Variance	Kurtosis	Skewness	Range	Min	Max	Sum	Count	Condition
2	-0.00102	0.006691	-0.01455	0.068546	0.3028	0.091688	3.696966	-0.04329	2.854385	-1.56657	1.287816	-2.08739	2048	FL
3	0.000937	0.005917	-0.00746	-0.09139	0.267754	0.071692	3.193106	0.064787	2.834783	-1.40627	1.428515	1.919518	2048	FL
4	0.001619	0.006561	-0.00691	0.015994	0.296926	0.088165	2.541217	-0.06182	2.633476	-1.47301	1.160466	3.315926	2048	FL
5	0.000549	0.006493	-0.01605	-0.16992	0.293824	0.086332	2.538501	0.028963	2.575753	-1.29623	1.279519	1.125231	2048	FL
6	-0.00151	0.005893	-0.00673	0.161864	0.266674	0.071115	4.126883	-0.04113	3.33673	-1.81574	1.520992	-3.08503	2048	FL
7	-0.00234	0.006869	-0.02225	-0.04017	0.310858	0.096633	2.565312	0.051909	2.910905	-1.6374	1.273506	-4.78617	2048	FL
8	-0.00147	0.005753	-0.011	0.022849	0.260369	0.067792	3.063727	-0.00306	2.660292	-1.50055	1.159744	-3.00795	2048	FL
9	0.002394	0.006902	0.001323	0.036678	0.312349	0.097562	3.324407	-0.07507	3.152498	-1.69127	1.461225	4.903298	2048	FL
10	-0.00209	0.005704	-0.00228	-0.04522	0.25813	0.066631	4.769916	0.053784	2.942893	-1.5531	1.389793	-4.27484	2048	FL
11	0.001268	0.006521	-0.00998	0.08466	0.295126	0.087099	2.054966	-0.00408	2.568898	-1.41673	1.152168	2.597399	2048	FL
12	0.000455	0.005699	-0.01022	-0.05387	0.257888	0.066506	4.457679	-0.09546	2.923412	-1.57823	1.345178	0.930898	2048	FL
13	0.004538	0.00667	-0.00234	-0.03776	0.301829	0.091101	2.889033	-0.14177	2.698774	-1.43164	1.267132	9.294187	2048	FL
14	0.002458	0.005679	-0.00265	-0.07011	0.256994	0.066046	2.658223	0.095891	2.575392	-1.29022	1.285171	5.033415	2048	FL
15	-0.00063	0.006797	-0.01148	-0.15693	0.307614	0.094627	2.828274	-0.09126	2.744832	-1.53085	1.213979	-1.28265	2048	FL
16	0.000732	0.005901	-0.00812	-0.10113	0.26707	0.071326	3.175302	-0.06747	2.682299	-1.44271	1.239594	1.498263	2048	FL
17	-0.00223	0.00636	-0.01407	-0.07239	0.287812	0.082836	2.483984	0.025308	2.522119	-1.30393	1.218188	-4.56225	2048	FL
94	-0.002	0.005831	-0.00631	-0.33563	0.263878	0.069632	1.669465	0.065524	2.297241	-1.11152	1.185719	-4.09626	2048	FL
95	-0.00087	0.00663	-0.0086	-0.23702	0.300061	0.090036	2.351601	-0.09988	2.59319	-1.3382	1.254987	-1.77473	2048	FL
96	-0.00091	0.005452	-0.0077	-0.05844	0.24675	0.060886	3.832097	0.011169	2.795941	-1.38222	1.413724	-1.86336	2048	FL
97	0.00326	0.006494	-0.00523	-0.10402	0.293895	0.086374	2.560839	-0.0194	2.531018	-1.36045	1.170567	6.677307	2048	FL
98	-0.00293	0.00566	-0.0071	0.024893	0.256124	0.0656	4.799522	-0.01271	2.876752	-1.53073	1.34602	-6.00772	2048	FL
99	0.003315	0.006918	-0.00078	-0.03151	0.313067	0.098011	1.722431	-0.03827	2.551461	-1.36839	1.183074	6.789867	2048	FL
100	-0.00246	0.005374	-0.00319	-0.0279	0.243189	0.059141	2.236136	0.067845	2.408237	-1.22336	1.184878	-5.03798	2048	FL
101	0.001577	0.006652	-0.00968	0.055678	0.301041	0.090626	2.109241	-0.02574	2.461991	-1.2824	1.179586	3.229221	2048	FL

Labels

No. of samples

Development of training data

- In this examples **there are 8 classes (labels)**, **one is all defect free** i.e. healthy class and **seven are faulty classes**.
- **100 samples and 13 statistical features** are considered for each class.
- As shown in previous slide, first, **100 samples and 13 statistical features** for all defect free i.e. **healthy class are placed**, then **100 samples and 13 statistical features** for **flank wear are placed**.
- **In similar way 100 samples of each tool class are placed** one after the other.
- Thus, **training data is consisted of 800 samples and 13 statistical features** representing all 8 classes (labels).
- **Last column** of the training data is dedicated for **class label** with respect to which classification is desired.
- Usually data is saved in **.csv (comma delimited)** format.



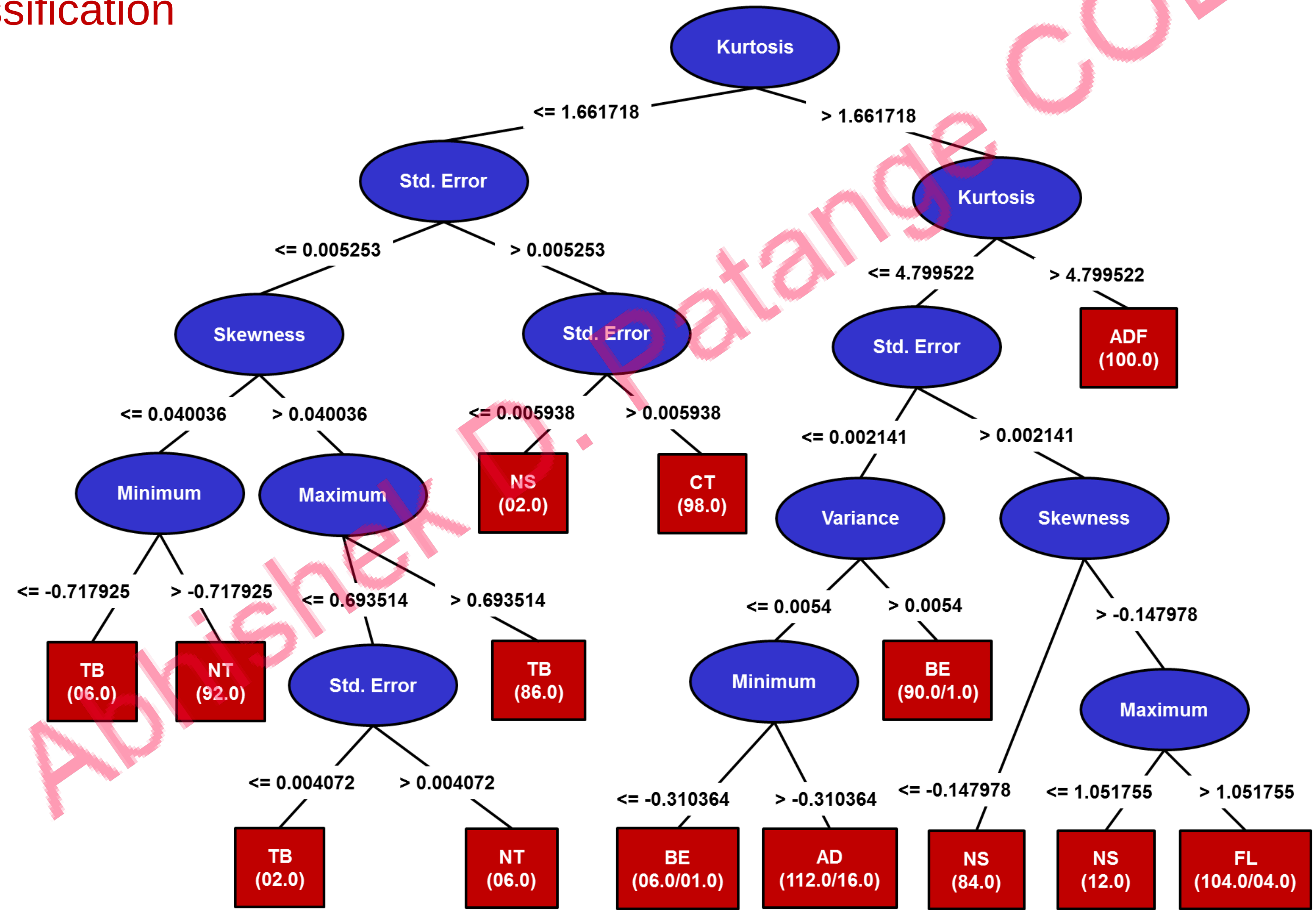
Practical no. 4:

4. To select relevant features using suitable technique

- After feature extraction, it is essential to **select the features which reflect dissimilarity between the tool classes**
- In other words, features that reflect the **similarity between the tool classes** are to be **eliminated** to achieve desirable classification accuracy
- This process is generally regarded as **'feature selection'**
- Several methods (**Filter, Wrapper, Embedded, Hybrid methods**) are available for selecting feature; amongst them techniques of **Attribute Evaluator (AE), and J48 Decision Tree (DT)** are commonly used
- The technique of **decision tree** usually used as it **reduces entropy** within the database, consumes **negligible computation time** and **represents visual illustration**

Application of decision tree for feature selection

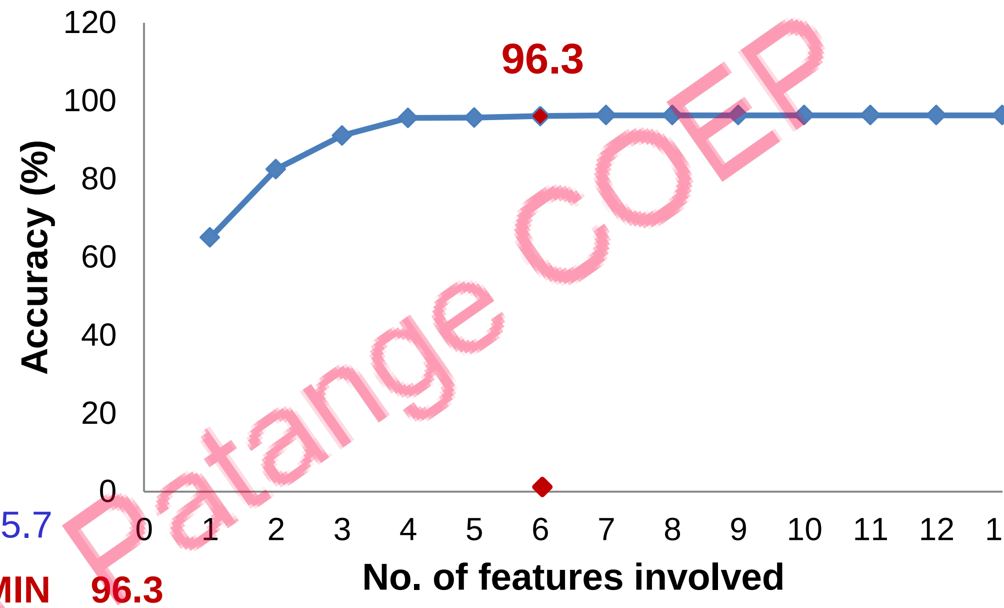
- The features exhibited by DT are **kurtosis, standard error, skewness, maximum, minimum, and variance;** thus considered for further classification



Effect of number of features

Where, Kurtosis (KUR), Standard Error (SDE), Skewness (SKE), Variance (VAR), Maximum (MAX), Minimum (MIN), Range (RNG), Mean (MN), Mode (MD), Median (MDN), Standard Deviation (SDV), Summation (SUM), Count (CNT)

1	KUR	65.0												
2	KUR	SDE	82.5											
3	KUR	SDE	SKE	91.1										
4	KUR	SDE	SKE	VAR	95.6									
5	KUR	SDE	SKE	VAR	MAX	95.7								
6	**KUR**	**SDE**	**SKE**	**VAR**	**MAX**	**MIN**	**96.3**							
7	KUR	SDE	SKE	VAR	MAX	MIN	RNG	96.3						
8	KUR	SDE	SKE	VAR	MAX	MIN	RNG	MN	96.3					
9	KUR	SDE	SKE	VAR	MAX	MIN	RNG	MN	MD	96.3				
10	KUR	SDE	SKE	VAR	MAX	MIN	RNG	MN	MD	MDN	96.3			
11	KUR	SDE	SKE	VAR	MAX	MIN	RNG	MN	MD	MDN	SDV	96.3		
12	KUR	SDE	SKE	VAR	MAX	MIN	RNG	MN	MD	MDN	SDV	SUM	96.3	
13	KUR	SDE	SKE	VAR	MAX	MIN	RNG	MN	MD	MDN	SDV	SUM	CNT	96.3



Practical no. 6:

6. To classify features/To develop classification model and evaluate its performance (any one classifier).

- Feature classification is the **task of training of algorithm (classifier)**
- The classifier **maps a set of selected features (input)** to the **normal or faulty configuration of any machine component (output)** which is evaluated using the classification accuracy
- In general, numerous classifiers are available, among these, **family of Trees, Bayes, Functions, Rules-based, Lazy, Meta, SVM** are really popular

Feature classification

Install WEKA software and its role

- **Waikato Environment for Knowledge Analysis (Weka),** developed at the University of **Waikato, New Zealand,** is **free software** licensed under the GNU General Public License, and the companion software to the book **"Data Mining: Practical Machine Learning Tools and Techniques".**
- Weka contains a **collection of visualization tools and algorithms** for data analysis and predictive modeling, together with graphical user interfaces for easy access to these functions.

Developer(s)	University of Waikato
Stable release	3.8.5 (stable) / December 21, 2020; 11 months ago
Preview release	3.9.5 / December 21, 2020; 11 months ago
Repository	svn.cms.waikato.ac.nz/svn/weka/
Written in	Java
Operating system	Windows, OS X, Linux
Platform	IA-32, x86-64; Java SE
Type	Machine learning
License	GNU General Public License
Website	www.cs.waikato.ac.nz/~ml/weka

Feature classification

Install WEKA software and its role

Downloading WEKA

https://waikato.github.io/weka-wiki/downloading_weka/

https://sourceforge.net/projects/weka/

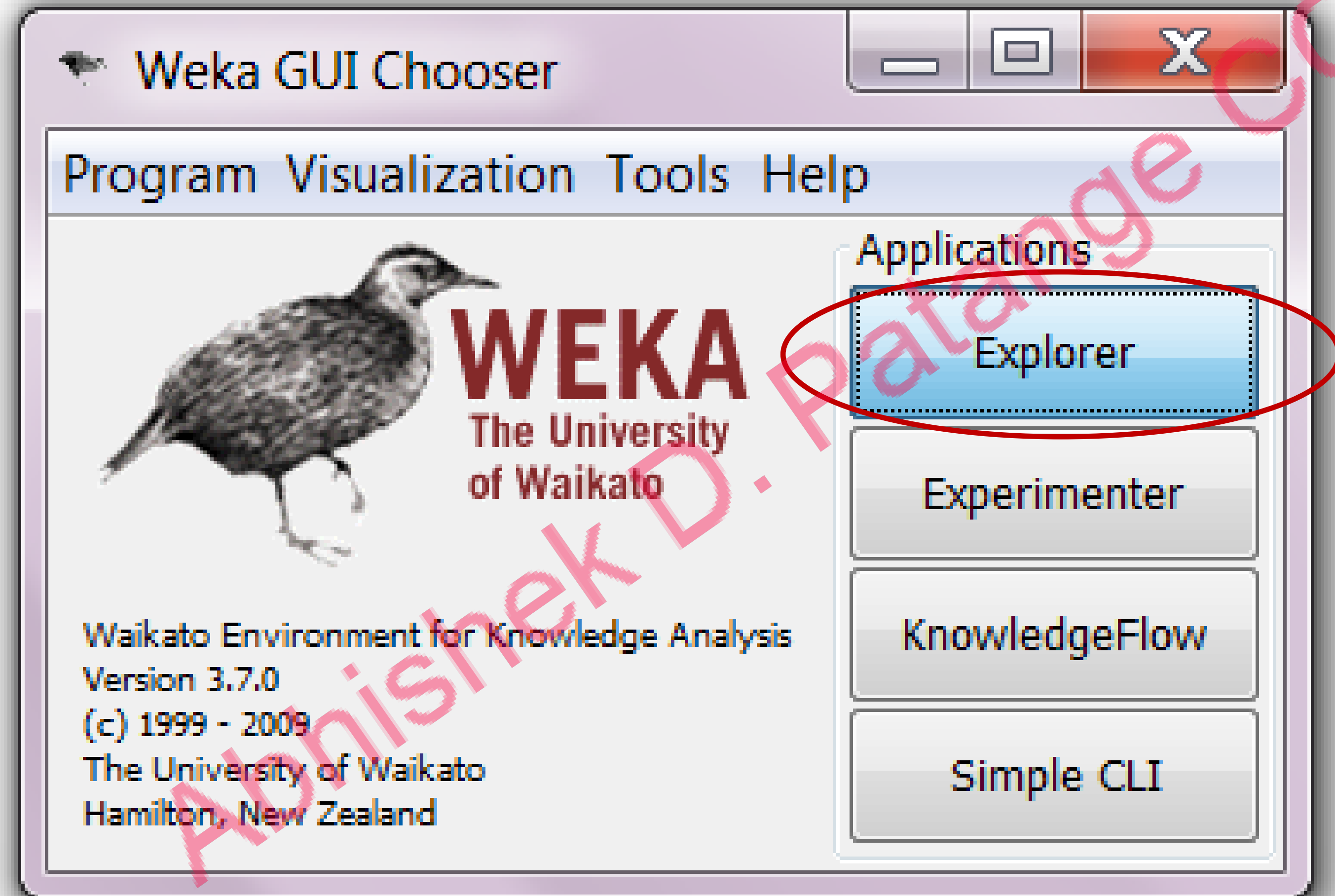
Installing WEKA

https://machinelearningmastery.com/download-install-weka-machine-learning-workbench/

I use **WEKA 3.7**

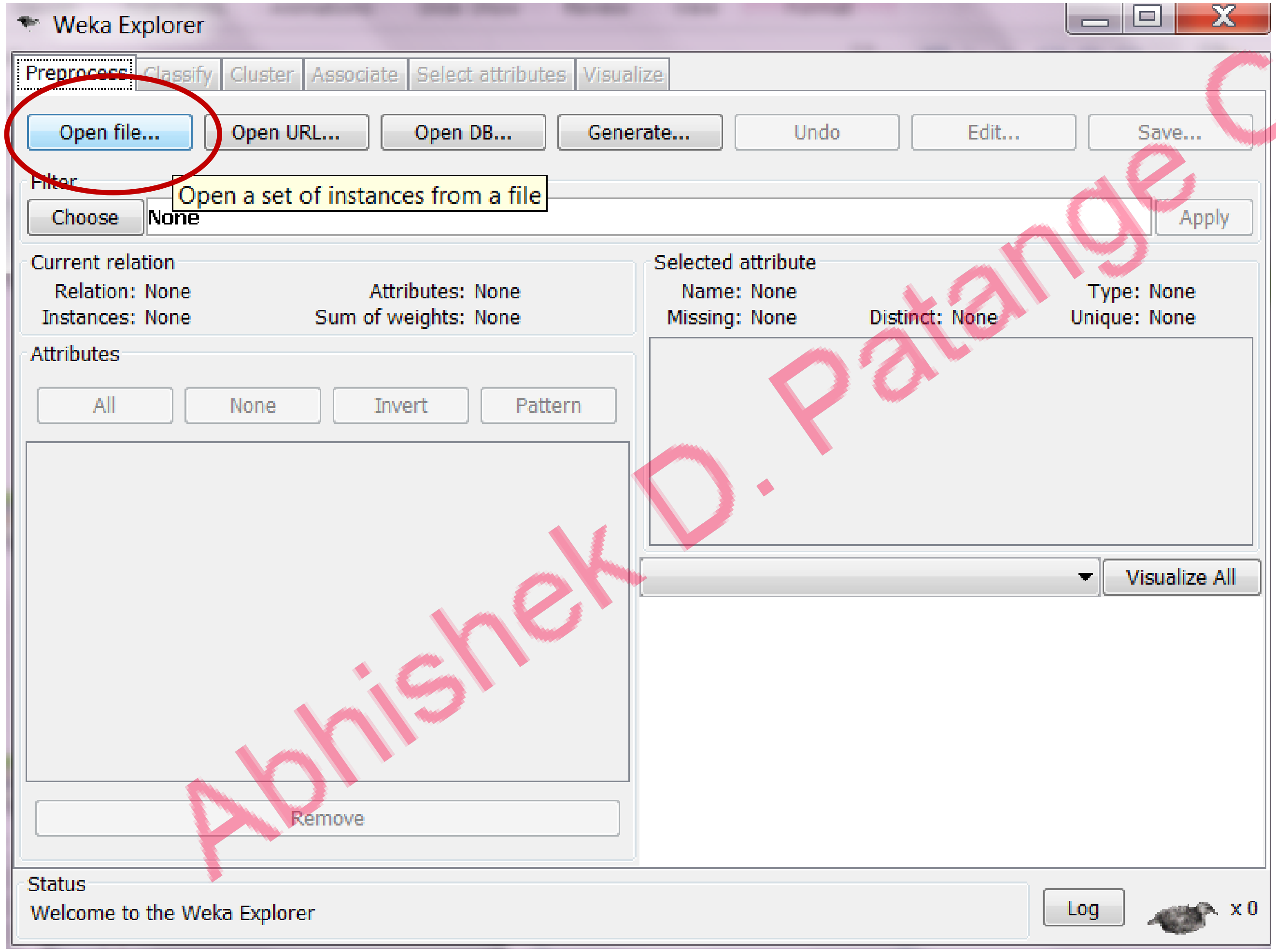
Feature classification

Open WEKA and click on Explorer



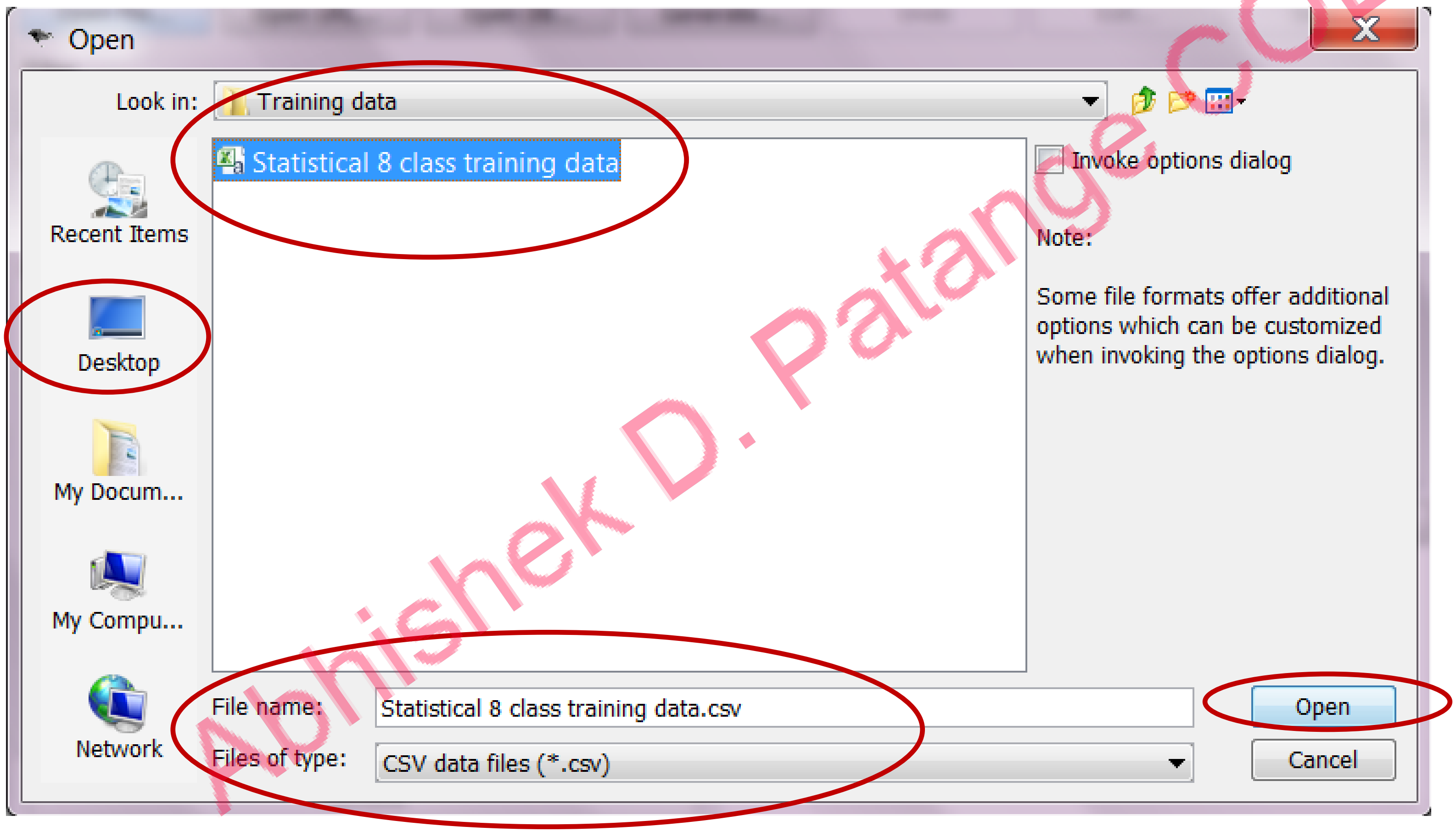
Feature classification

Click on open file in pre-process



Feature classification

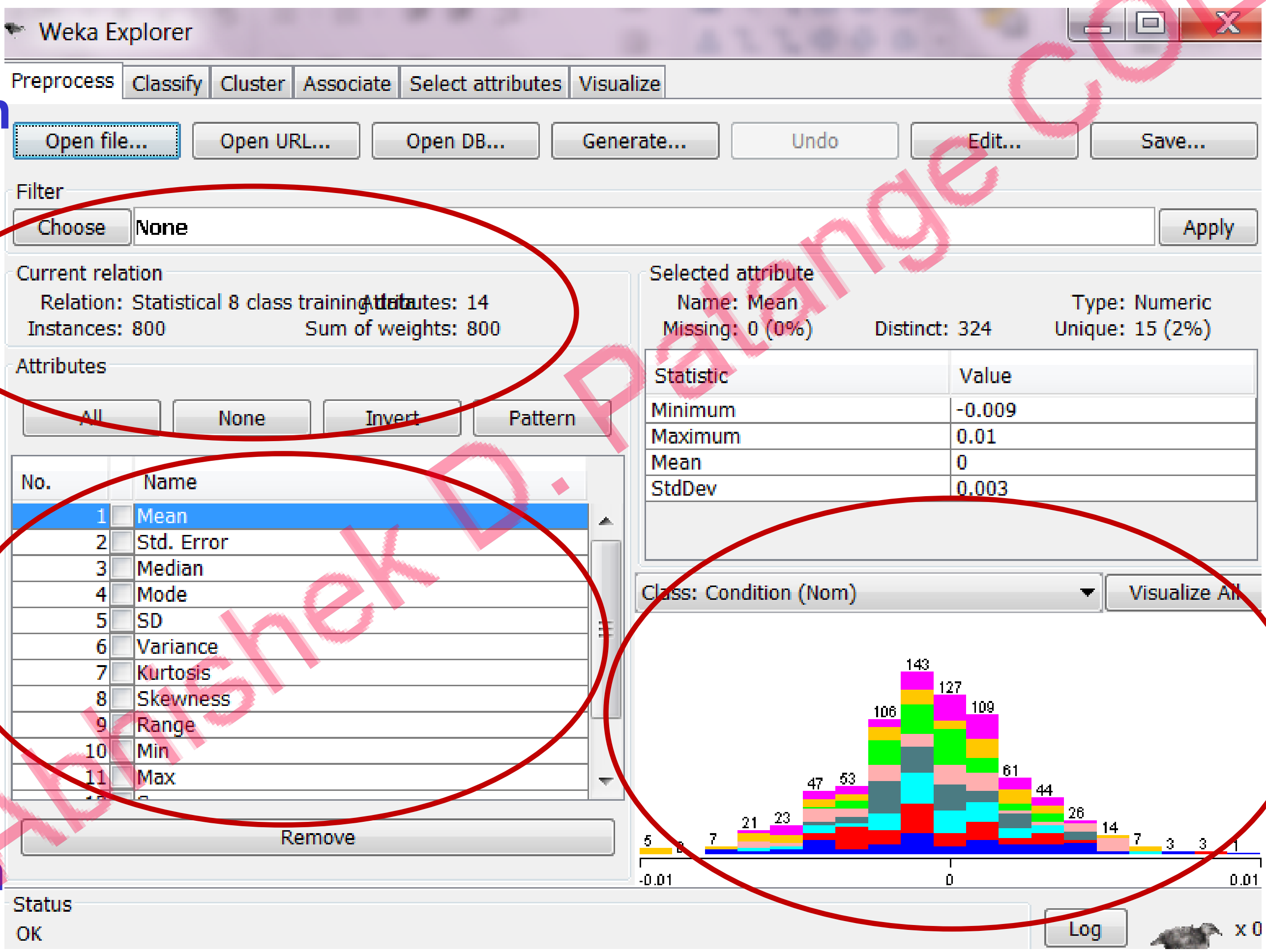
Open .csv file (training data excel sheet)



Feature classification

After opening the file you will see following GUI

File information

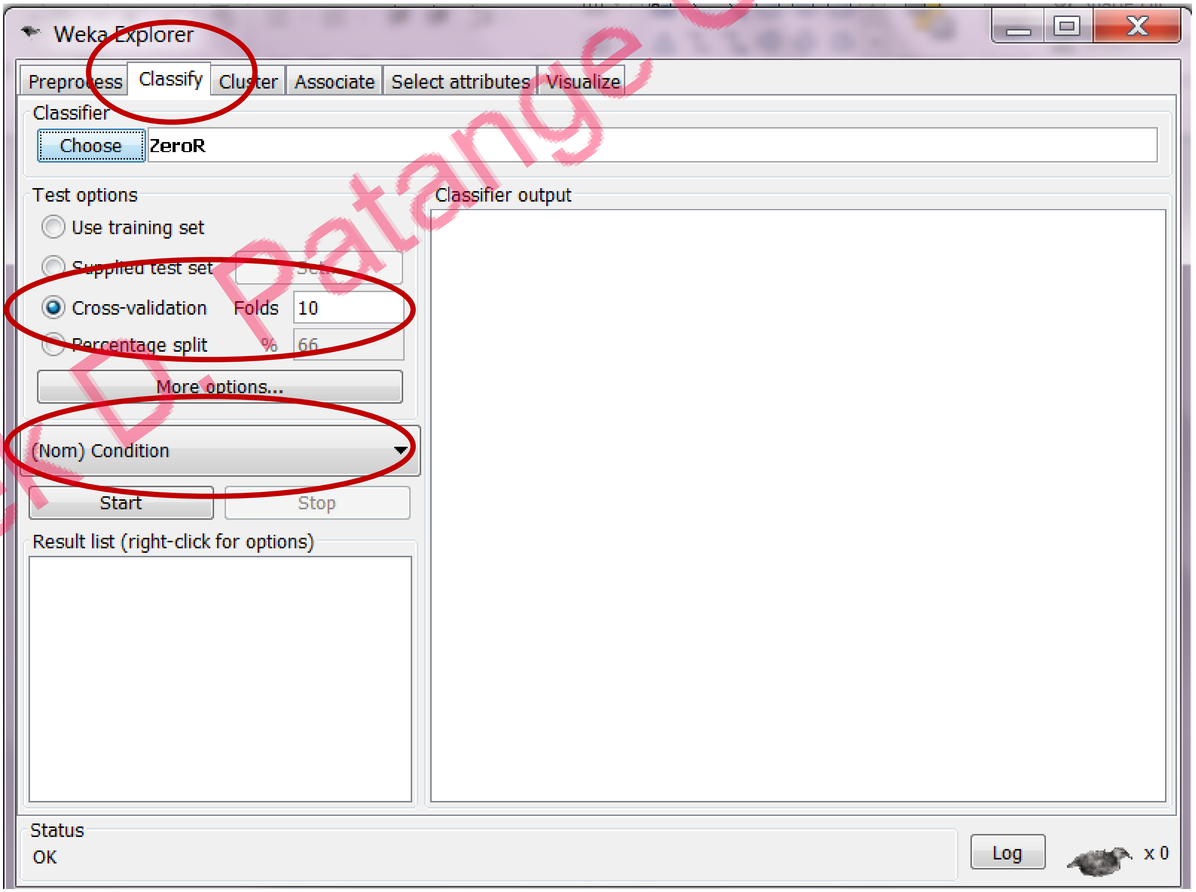


Visualization of feature distribution

Feature information

Feature classification

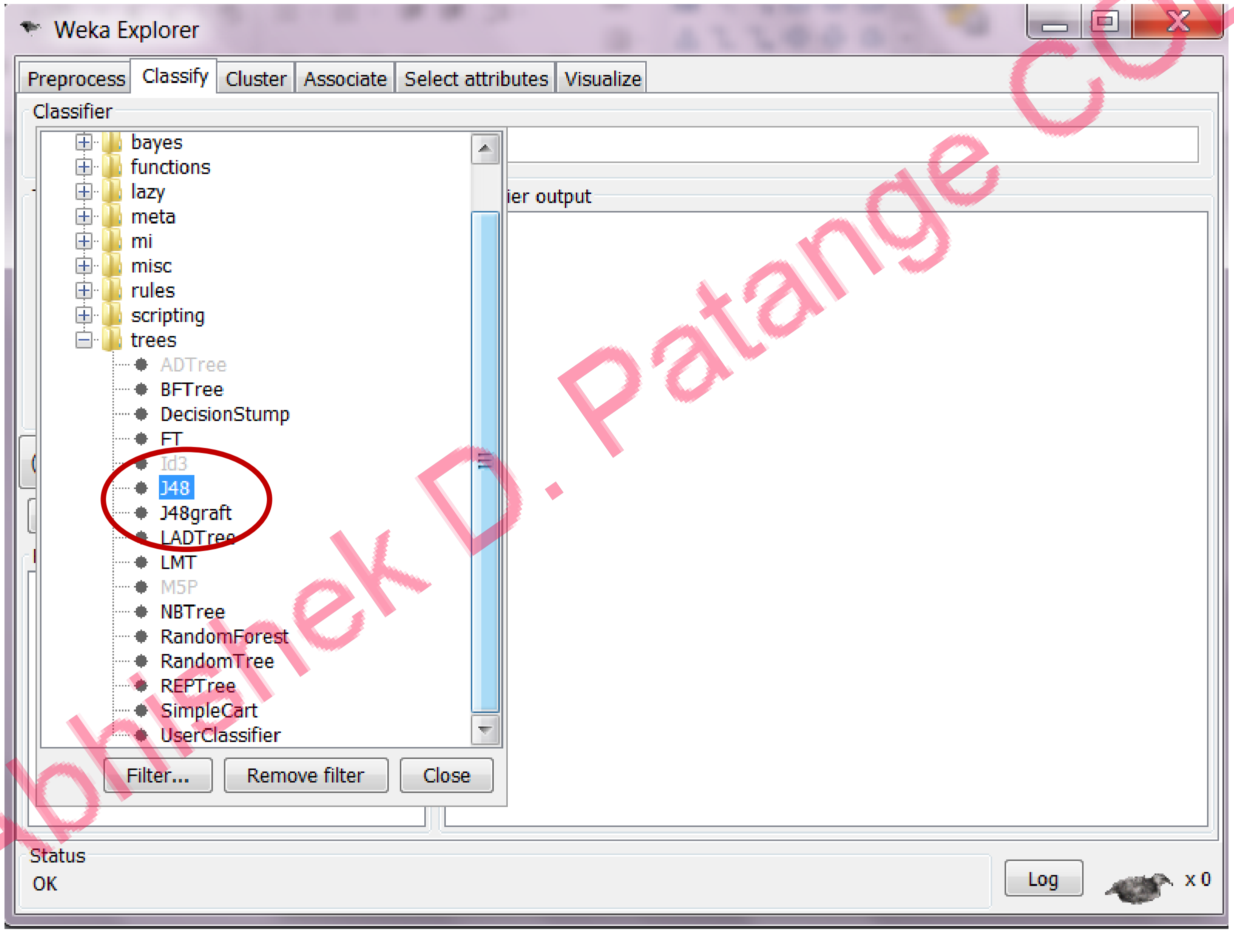
Now after opening file, click on classify, and then select cross validation, keep number of folds = 10, and select classifier.



Ensure that you have selected "condition" for classification (last column in .csv file) with respect to which we need classification

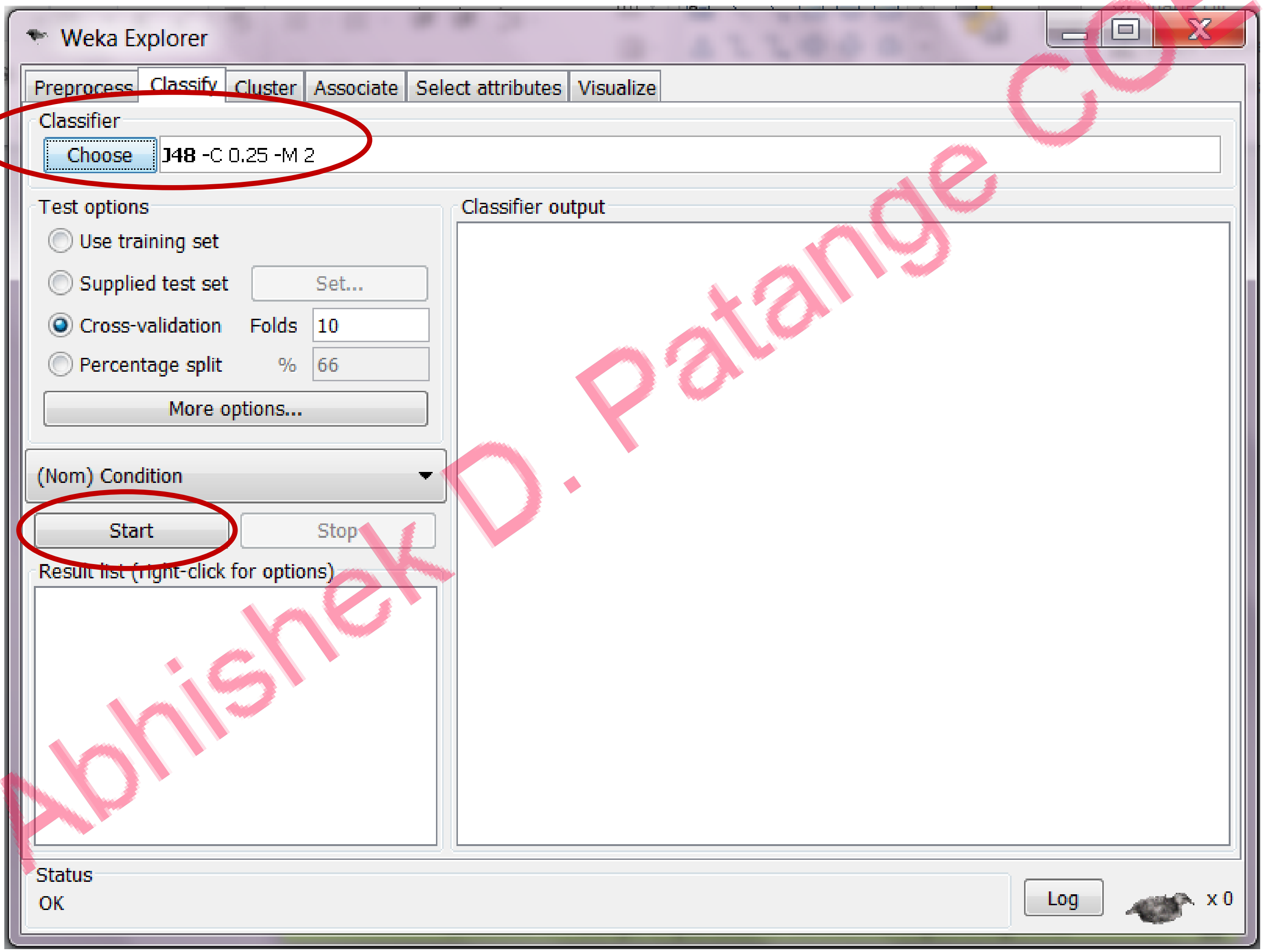
Feature classification

Click on choose, go to trees and select J48 (decision tree)



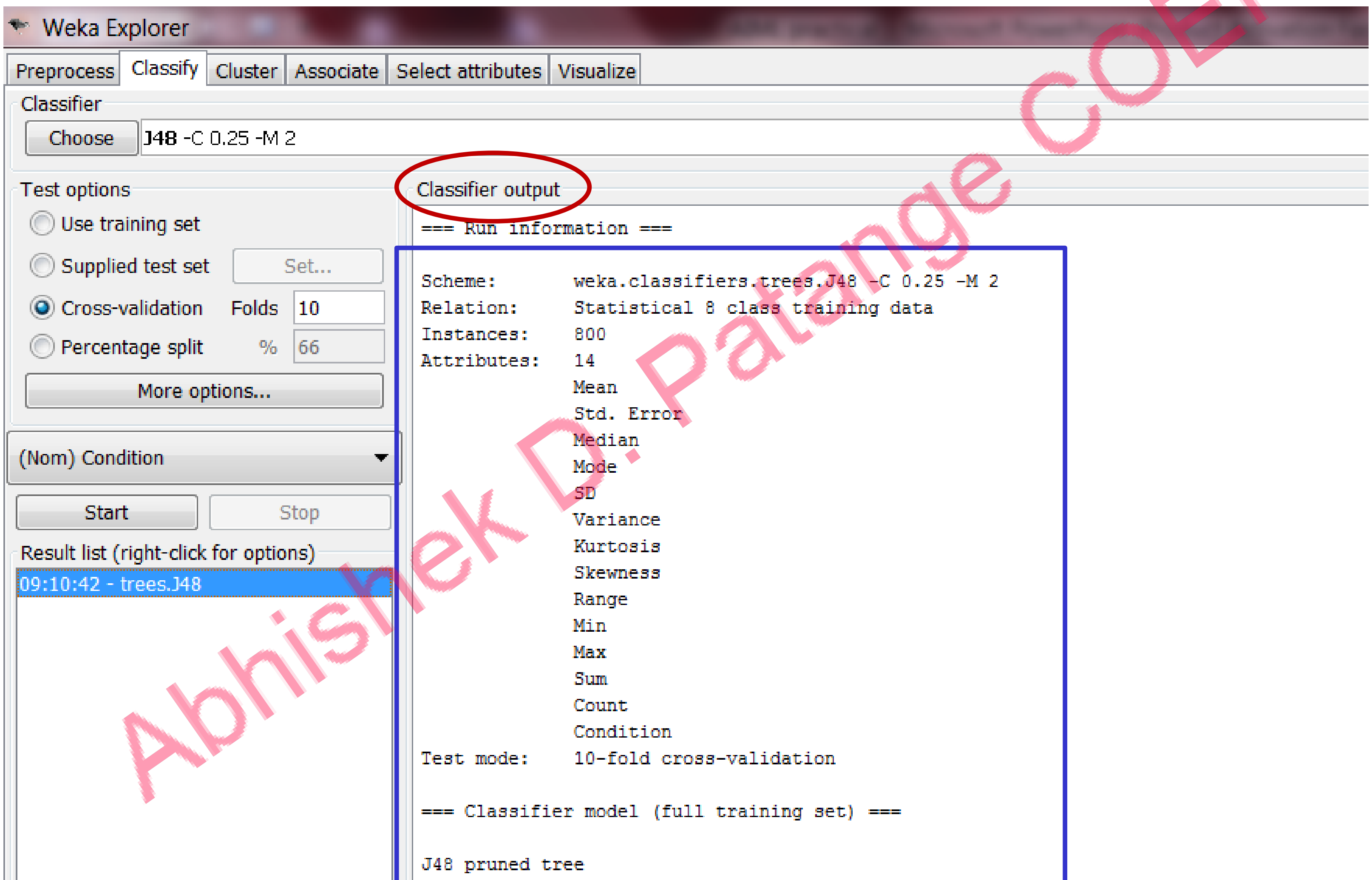
Feature classification

Once you choose the classifier, click on start



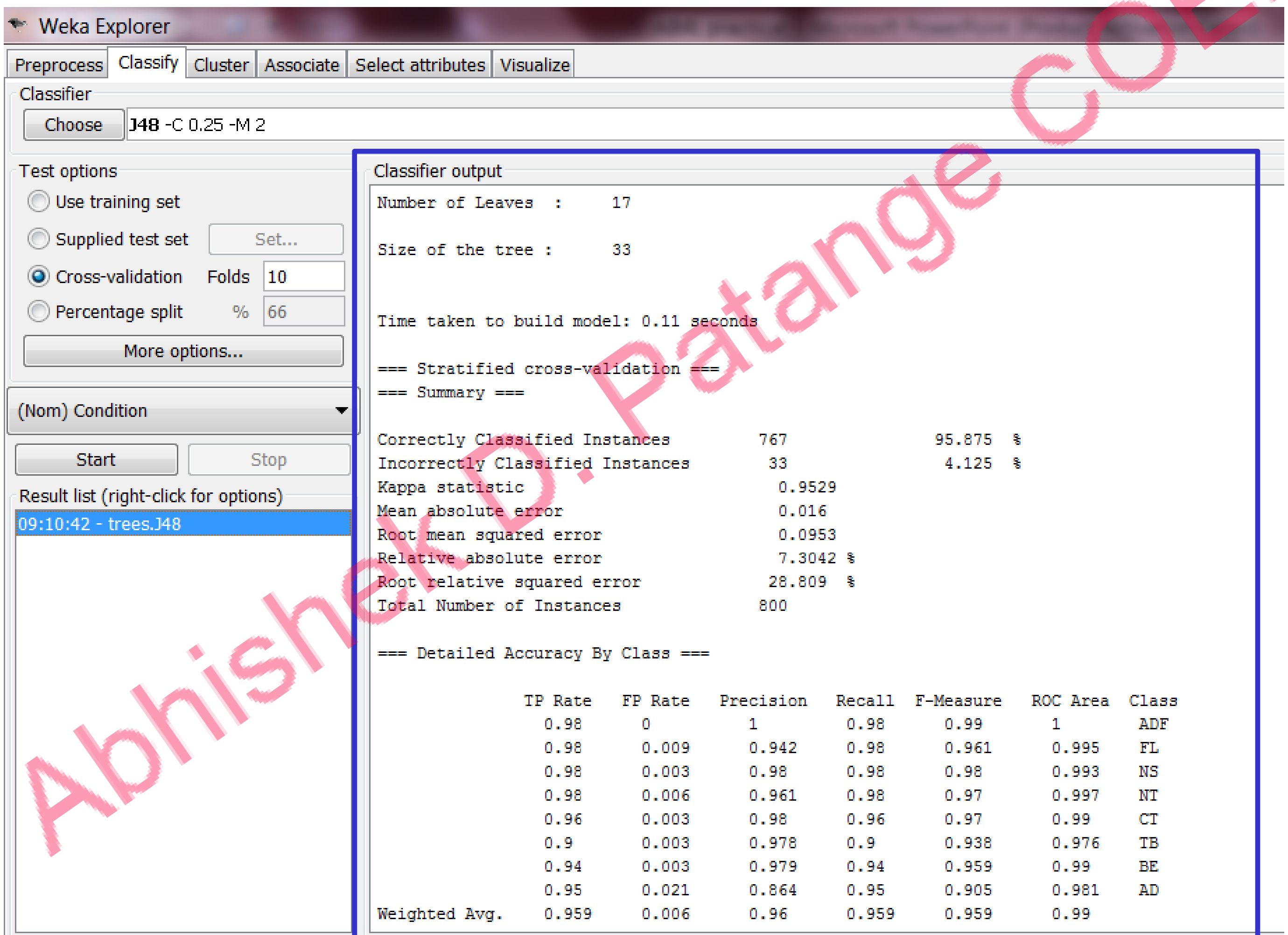
Feature classification

Once you run the classifier saying start, you will see the output as



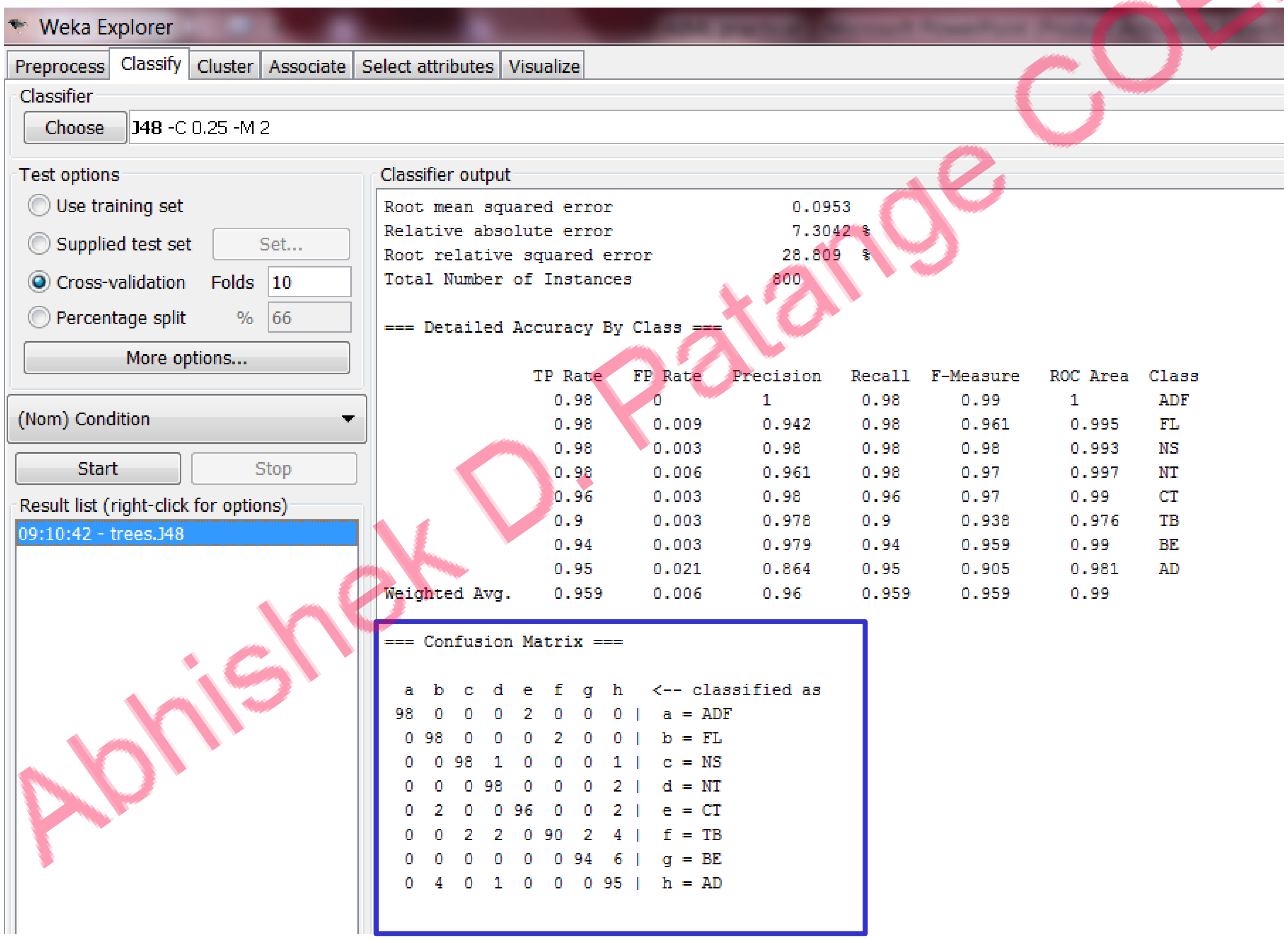
Feature classification

Once you run the classifier saying start, you will see the output as



Feature classification

Once you run the classifier saying start, you will see the output as

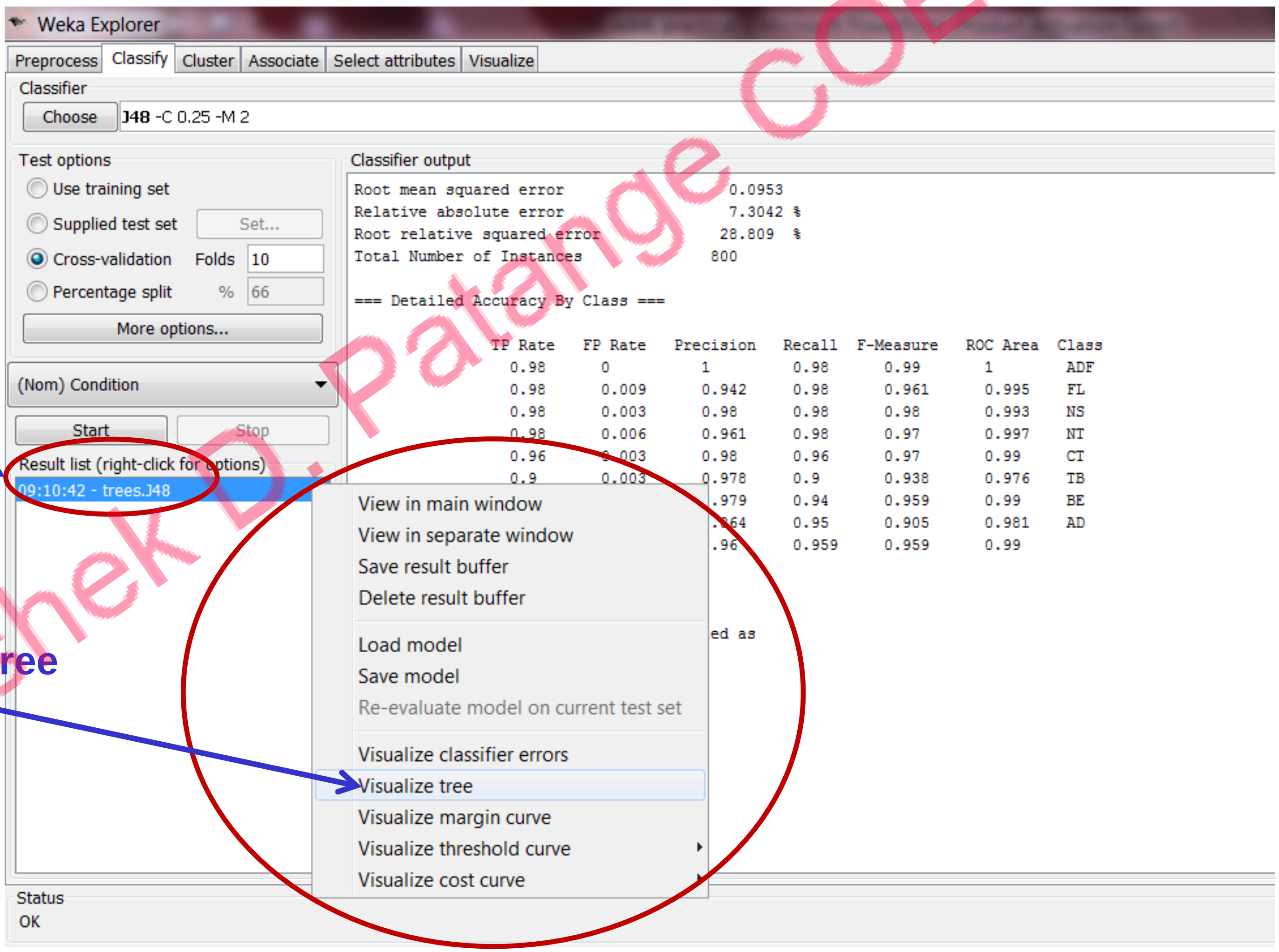


Feature classification

Visualize tree

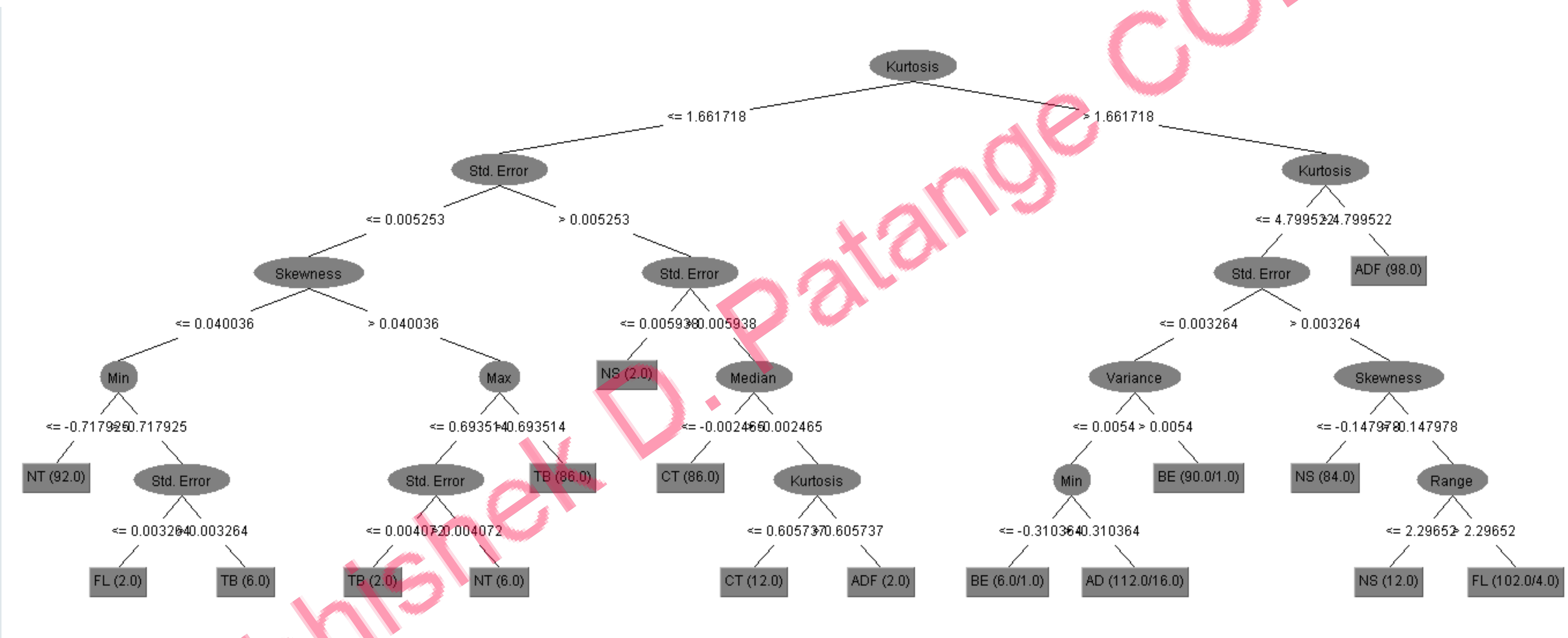
1. Right click here

2. Click on visualize tree



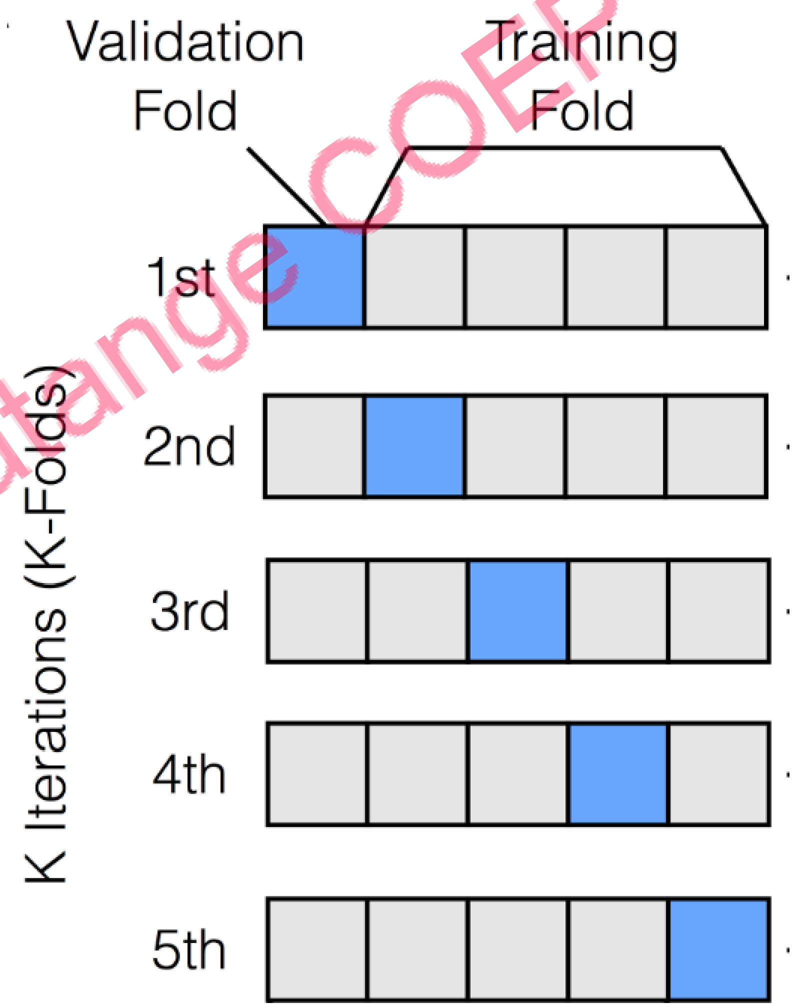
Feature classification

Once you click visualize tree, a tree structure appears as follows



Training of model: K-fold cross-validation mode

- The classifier model can be designed/trained and performance can be evaluated based on **K-fold cross-validation mode, training mode and test mode.**
- The main idea behind **K-Fold cross-validation** is that **each sample** in our dataset has the **opportunity of being tested**. It is a special case of cross-validation where we **iterate over a dataset set k times**. In each round, we **split the dataset into k parts**: **one part is used for validation**, and the **remaining k-1 parts are merged into a training subset** for model evaluation



Training of model: K-fold cross-validation mode

Advantages of K-fold cross-validation mode

- **Computation time is reduced** as we repeated the process only 10 times when the value of k is 10.
- **Reduced bias**
- **Every data points get to be tested exactly once** and is used in training k-1 times
- The **variance of the resulting estimate is reduced as k increases**

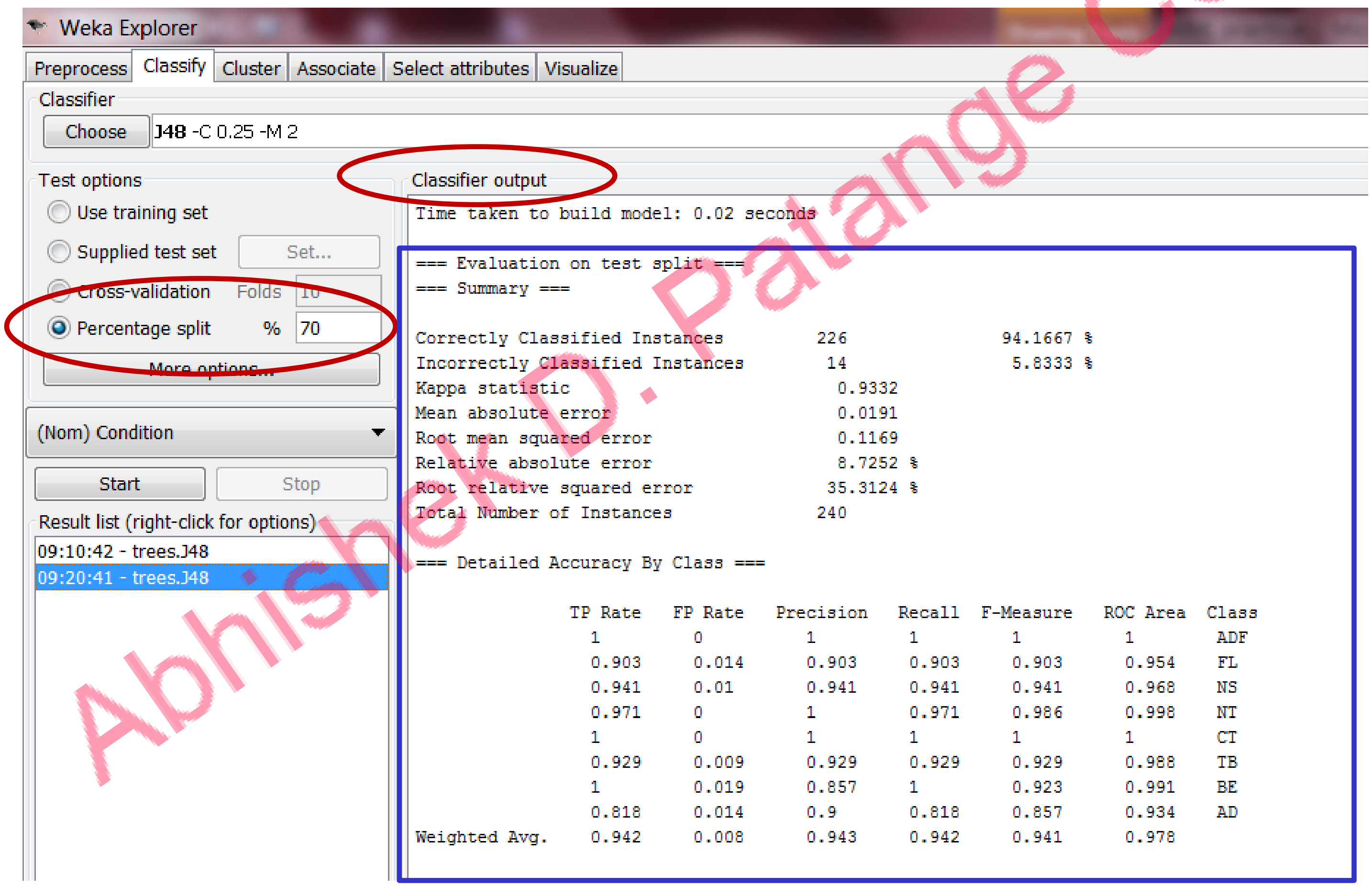
Training of model: Train / Test split

Train and test data

- **Training set —** a subset to train a model.
- **Test set —** a subset to test the trained model.
- Make sure that your test set meets the following two conditions:
- **Is large enough to yield statistically meaningful results.**
- **Is representative of the data set as a whole. In other words, don't pick a test set with different characteristics than the training set.**

Feature classification

Now after opening file, click on classify, and then select percentage spilt, keep spilt = 70%, and select classifier.

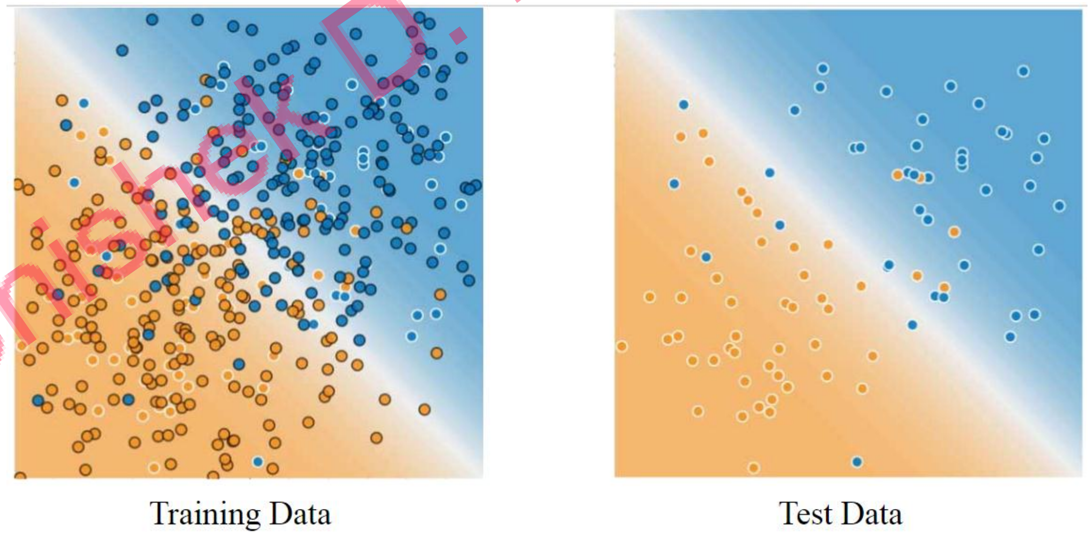


Training of model: Train / Test split

Train and test data

- Assuming that your test set meets the preceding two conditions, your goal is to create a model that generalizes well to new data.

- For example see the figure. Notice that the **model learned for the training data is very simple**. This model doesn't do a perfect **job—a few predictions are wrong**. However, this model does about as well on the test data as it does on the training data. **In other words, this simple model does not overfit the training data.**



Training of model: Train / Test split

Train and test data

- **Never train on test data.** If you are seeing surprisingly good results on your evaluation metrics, it might be a sign that you are accidentally training on the test set. For example, **high accuracy might indicate that test data has leaked into the training set.**

- For example, consider a model that **predicts whether an email is spam, using the subject line, email body, and sender's email address as features.** We apportion the data into training and test sets, with an **80-20 split.** After training, the model **achieves 99% precision on both the training set and the test set.** We'd expect a lower precision on the test set, so we take another look at the data and **discover that many of the examples in the test set are duplicates of examples in the training set** (we neglected to scrub duplicate entries for the same spam email from our input database before splitting the data). We've inadvertently trained on some of our test data, and as a result, we're no longer accurately measuring how well our model generalizes to new data.

Evaluation of model

Confusion matrix

- A Confusion matrix is an **N x N matrix** used for evaluating the performance of a classification model, where **N is the number of target classes**. The matrix compares the **actual target values with those predicted by the machine learning model**
- Using random forest 10 fold cross validation, correctly classified instances are 777 out of 800 exhibiting accuracy of 97.1%.

Actual class

Predicted class

Class	ADF	FL	NS	NT	CT	EF	BE	AD
ADF	100	0	0	0	0	0	0	0
FL	0	100	0	0	0	0	0	0
NS	0	0	98	0	0	0	0	2
NT	0	0	0	96	0	2	0	2
CT	0	0	0	0	98	0	0	2
EF	0	0	0	1	0	93	2	4
BE	0	0	0	0	0	0	96	4
AD	0	4	0	0	0	0	0	96

Evaluation of model

Performance evaluators

What can we learn from this matrix?

- There are **two possible predicted classes: "yes" and "no".** If we were predicting the presence of a disease, for example, **"yes" would mean they have the disease,** and **"no" would mean they don't have the disease.**
- The classifier made a total of **165 predictions** (e.g., **165 patients were being tested** for the presence of that disease).
- **Out of those 165 cases,** the classifier **predicted "yes" 110 times,** and **"no" 55 times.**
- **In reality, 105 patients in the sample have the disease,** and **60 patients do not.**

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Evaluation of model

Performance evaluators

- **True positives (TP):** these are cases in which we predicted yes (they have the disease), and they do have the disease.
- **True negatives (tn):** we predicted no, and they don't have the disease.
- **False positives (fp):** we predicted yes, but they don't actually have the disease. (Also known as a "type I error.")
- **False negatives (fn):** we predicted no, but they actually do have the disease. (Also known as a "type II error.")

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Evaluation of model

Performance evaluators

- **Accuracy:** Overall, how often is the classifier correct?
  (TP+TN)/total = (100+50)/165 = 0.91
- **Misclassification Rate:** Overall, how often is it wrong?
  (FP+FN)/total = (10+5)/165 = 0.09 which is equivalent to 1 minus Accuracy
- **True Positive Rate:** When it's actually yes, how often does it predict yes?
  TP/actual yes = 100/105 = 0.95 also known as "Sensitivity" or "Recall"
- **False Positive Rate:** When it's actually no, how often does it predict yes?
  FP/actual no = 10/60 = 0.17
- **True Negative Rate:** When it's actually no, how often does it predict no?
  TN/actual no = 50/60 = 0.83 which is equal to 1 minus False Positive Rate
- **Precision:** When it predicts yes, how often is it correct?
  TP/predicted yes = 100/110 = 0.91

Evaluation of model

Performance evaluators

- **Cohen's Kappa:** This is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance. In other words, a model will have a high Kappa score if there is a big difference between the accuracy and the null error rate. (More details about Cohen's Kappa.)

- **F Score:** This is a weighted average of the true positive rate (recall) and precision. (More details about the F Score.)

- **ROC Curve:** This is a commonly used graph that summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as you vary the threshold for assigning observations to a given class. (More details about ROC Curves.)

Evaluation of model

Performance evaluators

- From an AI perspective, evaluation includes model metric evaluation, confusion matrix calculations, KPIs, model performance metrics, model quality measurements and a final determination of whether the model can meet the established business goals. During the model evaluation process, you should do the following:

- **Evaluate** the models using **a validation data set.**
- **Determine confusion matrix values** for classification problems.
- Identify methods **for k-fold cross-validation** if that approach is used.
- Further tune **hyperparameters** for optimal performance.
- **Compare the machine learning model to the baseline model or heuristic.**

- Model evaluation can be considered the quality assurance of machine learning. Adequately evaluating model performance against metrics and requirements determines how the model will work in the real world.

Research team:

Dr. S. S. Pardeshi
Associate Professor
College of Engg Pune

Dr. R. Jegadeeshwaran
Associate Professor
VIT Vellore

Dr. Suhas Deshmukh
Associate Professor
Govt Col. of Engg. Karad

Abhishek Patange
Asst. Professor
College of Engg Pune

- Naman Bajaj
- Kaushal Kulkarni
- Rohan Ghatpande
- Rugved Wakchaure
- Aditya Patil

- Apoorva Khairnar
- Rushikesh Khade
- Nikhil Pradhan
- Aditya Medhi
- Narayan Gavade

Feel free to contact @ 8329347107 adp.mech@coep.ac.in


Thank you