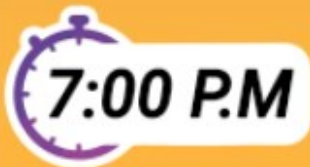


Workshop On:



Machine Learning using Python



Meeting Id: 2784159267

<https://bdren.zoom.us/j/2784159267>



জগন্নাথ বিশ্ববিদ্যালয়
Jagannath University



Session Chair

Prof. Dr. Mohammed Nasir Uddin

Chairman

Dept. of CSE, Jagannath University



Instructor

Rajib Kumar Halder

Dept. of Computer Science & Engineering,
Jagannath University



Instructor

Fahima Hossain

Dept. of Computer Science & Engineering,
Jagannath University

Organized By:

Dept. of Computer Science and Engineering
Jagannath University



Outline (1st Day)

❑ Data Preprocessing

➤ Data Cleaning

- ✓ Remove Missing Values
- ✓ Remove Duplicate Values

➤ Data Encoding

- ✓ Label Encoding
- ✓ One Hot Encoding
- ✓ Frequency Encoding

➤ Data Transformation

- ✓ Normalization
- ✓ DBSCAN

❑ Feature Selection

✓ Methods

- Filter Methods
- Wrapper Methods
- Embedded Methods

✓ Tasks

- Remove highly inter-correlated variables
- Select highly correlated features with target variable



Outline (2nd Day)

☐ Dataset Splitting

- Hold Out Validation Approach
 - ✓ Train Test Split Method
- Cross Validation Approach
 - ✓ Leave One Out
 - ✓ K Fold
 - ✓ Stratified K Fold
 - ✓ Time Series Cross Validation

☐ Classification

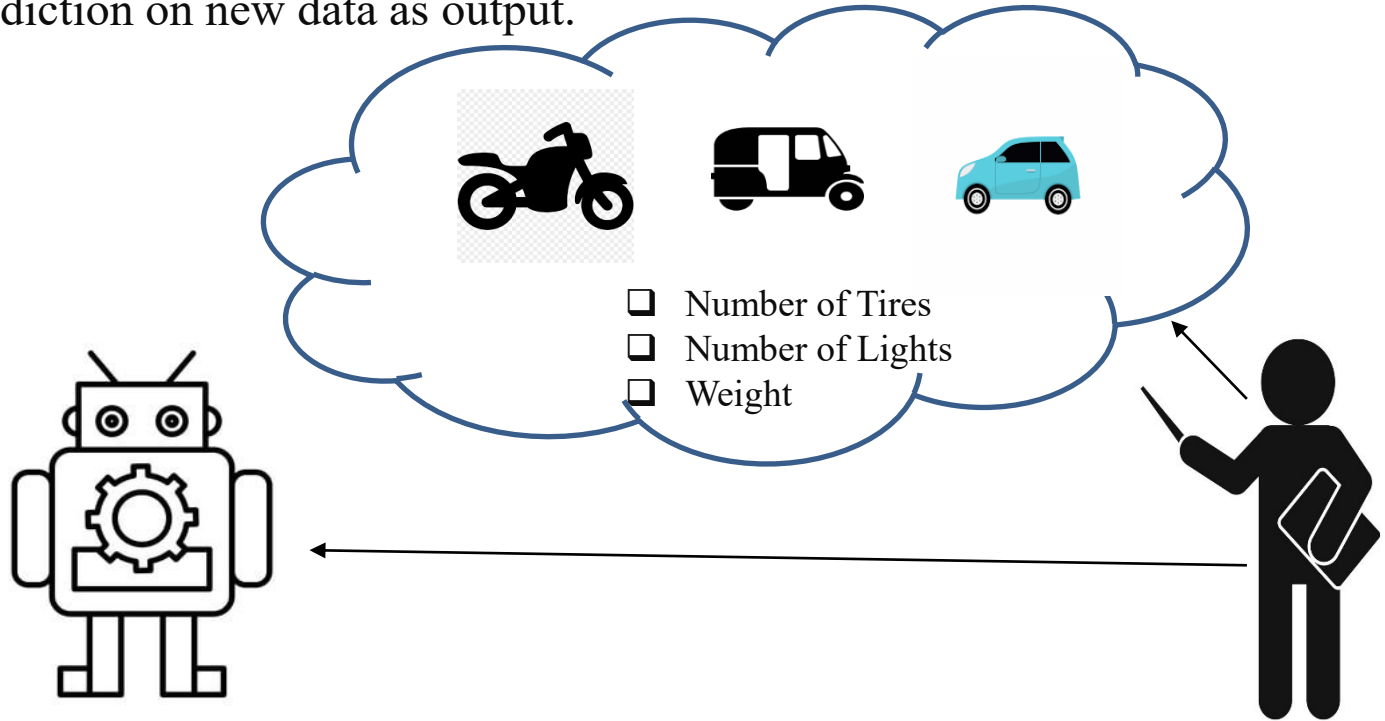
- ✓ Classification With Train Test Split
- ✓ Voting Ensemble Classification With Train Test Split
- ✓ Classification With Cross Validation
- ✓ Voting Ensemble Classification With Cross Validation
- ✓ Multiclass Classification

☐ Clustering

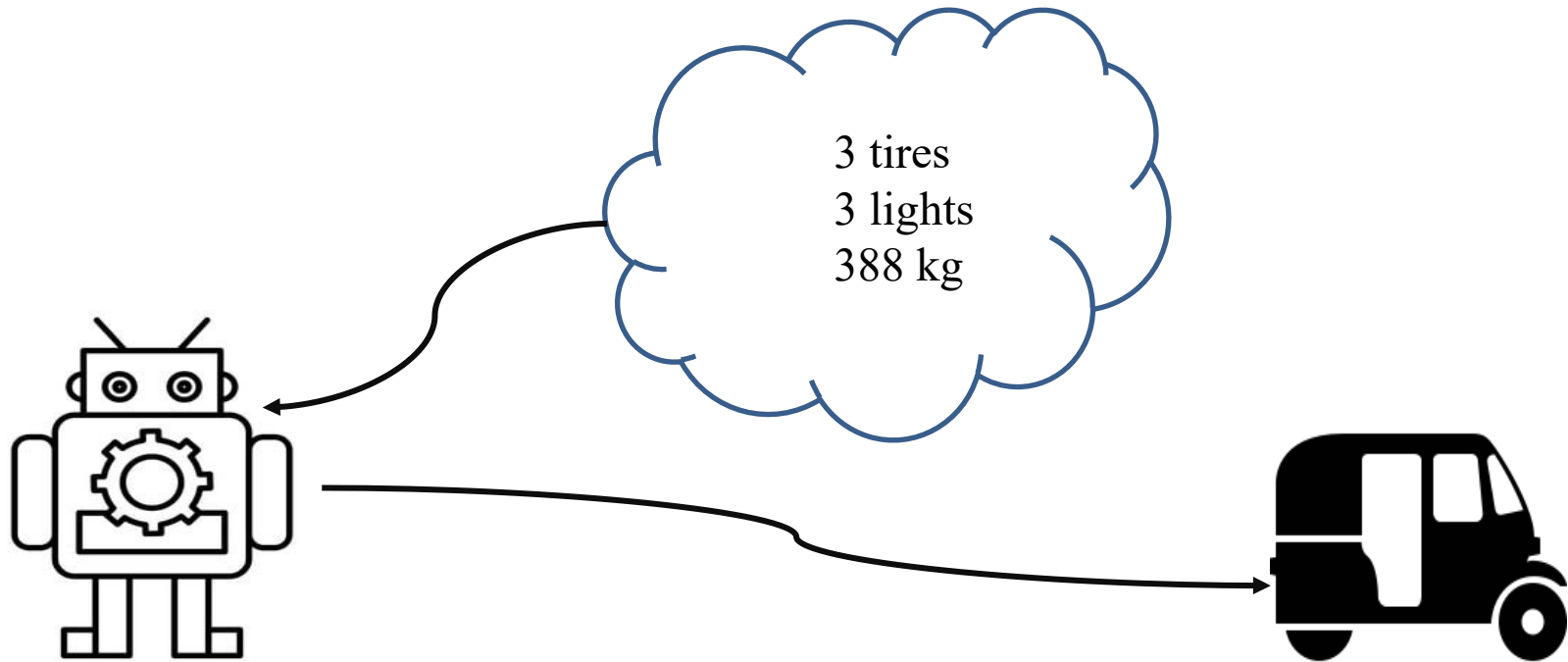
☐ Result Analysis

Machine Learning

❑ Receives input data, learn from data, identify patterns and gives prediction on new data as output.



Machine Learning



Steps of Machine Learning





Data Mining VS Machine Learning

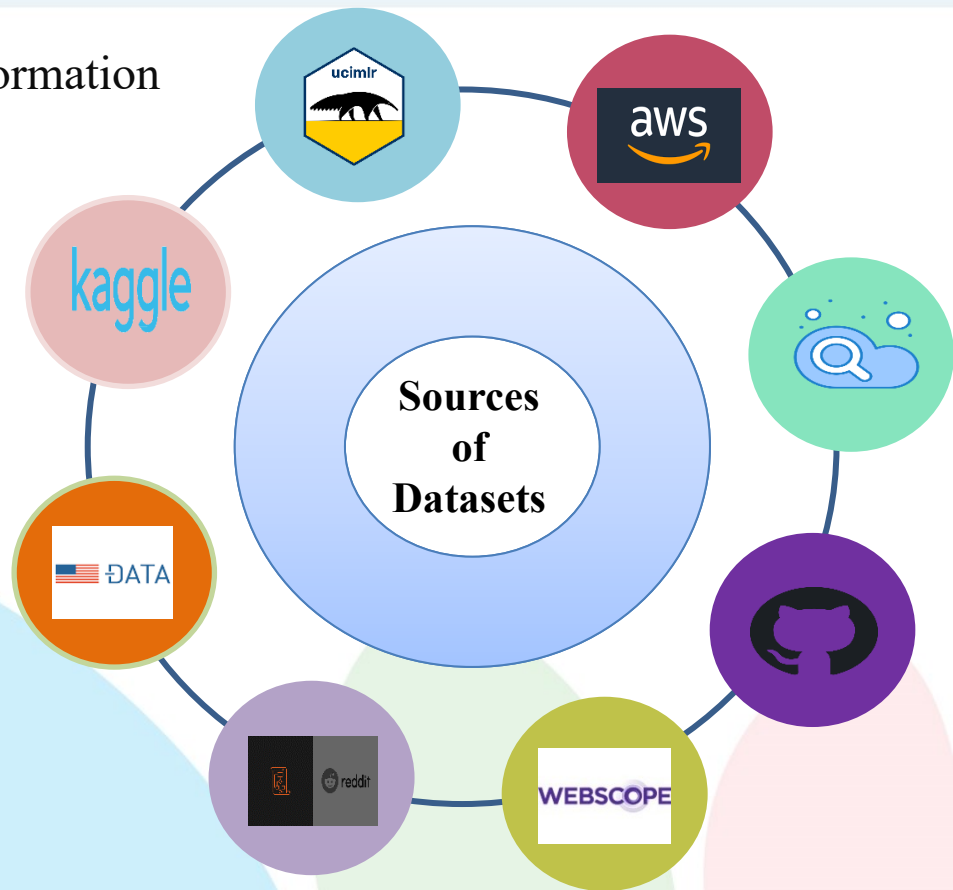
Data Mining	Machine Learning
Find patterns from large amount of raw data.	Makes computers learn new.
Need human intervention to explicitly look for patterns.	Does not need human intervention. It uses a algorithm / model to identify patterns.
Utilizes the database, data warehouse server, data mining engine, and pattern assessment techniques to obtain useful information.	Utilizes neural networks, predictive models, and automated algorithms to make the decisions.
Provides data management techniques	Provides methods for data analysis.
Limited as manual efforts is required.	Not limited as it can work with large dataset .



Data Collection

❑ Process of gathering and measuring information from countless different sources.

- ❖ Kaggle Datasets
- ❖ Google's Datasets Search Engine
- ❖ UCI Machine Learning Repository
- ❖ Yahoo WebScope
- ❖ Amazon Datasets
- ❖ Datasets subreddit
- ❖ .gov Datasets
- ❖ GitHub



Data Preprocessing

- ❖ Transforms raw data to an understandable format. Real data is often incomplete, inconsistent and noisy what makes the data dirty.
- **Incomplete:** missing values, i.e. gender = “ ”.
- **Inconsistent:** lack of compatibility, i.e. age = “42” , date of birth = “03/07/1997”
- **Noisy:** error or outliers, i.e. salary = “-10”

Kidney Disease Dataset.

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44	7800	5.2	yes	yes	no	good	no	no	ckd
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38	6000	NaN	no	no	no	good	no	no	ckd
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31	7500	NaN	no	yes	no	poor	no	yes	ckd
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35	7300	4.6	no	no	no	good	no	no	ckd
...
395	395	55.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	47	6700	4.9	no	no	no	good	no	no	notckd
396	396	42.0	70.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	54	7800	6.2	no	no	no	good	no	no	notckd
397	397	12.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	49	6600	5.4	no	no	no	good	no	no	notckd
398	398	17.0	60.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	51	7200	5.9	no	no	no	good	no	no	notckd
399	399	58.0	80.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	53	6800	6.1	no	no	no	good	no	no	notckd

Types of Data

1

Continuous

2

Discrete

3

Nominal

4

Ordinal

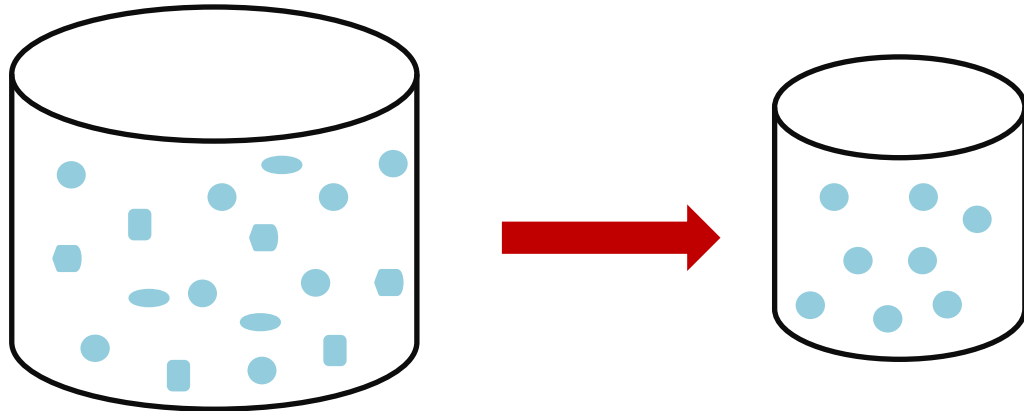
5

Binary

1. **Continuous:** Height or Weight of an individual, Rate of Interest on loans.
2. **Discrete:** Student count in a class, Color count in a Rainbow.
3. **Nominal:** States in a country, zip codes of areas.
4. **Ordinal:** Ratings for a restaurant (e.g. very good, good, bad, very bad), Level of Education of an individual (e.g. Doctorate, Post Graduate, Under Graduate), etc.
5. **Binary:** Gender (male or female), Fraudulent transaction (Yes or No), Cancerous Cell (True or False).

Data Cleaning

- ☐ Remove missing values.
- ☐ Filling missing Values.
- ☐ Remove duplicate values.
- ☐ Identify and remove outliers.

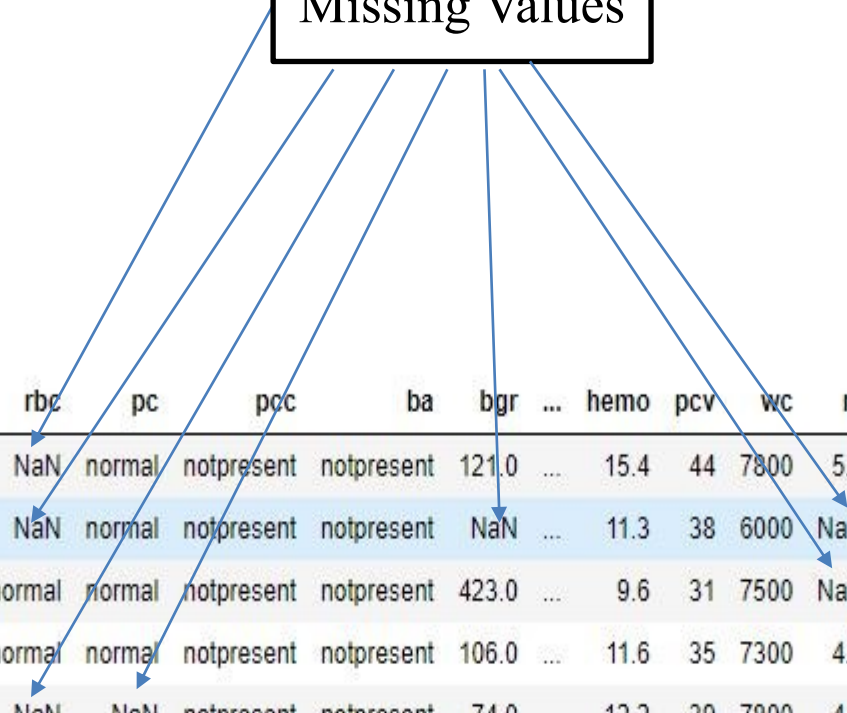




Remove Missing Values

- ☐ Deleting the rows or columns having null values.
- ☐ Drop columns having more than half of rows as null.
- ☐ Drop rows having one or more columns values as null.

Missing Values



The diagram illustrates the concept of missing values in a dataset. A central box labeled "Missing Values" has arrows pointing to specific cells in the table below that contain "NaN" (Not a Number), indicating missing data points.

	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane
0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	...	15.4	44	7800	5.2	yes	yes	no	good	no	no
1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	...	11.3	38	6000	NaN	no	no	no	good	no	no
2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	...	9.6	31	7500	NaN	no	yes	no	poor	no	yes
4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	...	11.6	35	7300	4.6	no	no	no	good	no	no
5	60.0	90.0	1.015	3.0	0.0	NaN	NaN	notpresent	notpresent	74.0	...	12.2	39	7800	4.4	yes	yes	no	good	yes	no



Filling Missing Values

- ❑ Replacing missing values with mean / median.
- ❑ Assigning An Unique Category.



Remove Duplicate Values

- ❑ Duplication can mean two slightly different things:
- ❑ More than one record that is exactly the same. This is what I call exact duplication.
- ❑ More than one record associated with the same observation, but the values in the rows are not exactly the same. This is what I call partial duplication.

Duplication

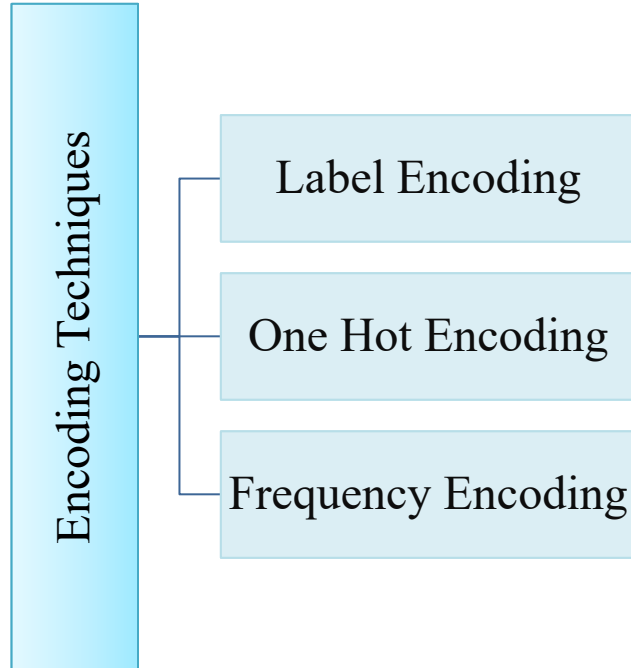
- Exact Duplication
- Partial Duplication

id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
1	51	80	1.02	0	0	normal	normal	notpresent	notpresent	94	15	1.2	144	3.7	15.5	46	9500	6.4	no	no	no	good	no	no	notckd
2	41	80	1.025	0	0	normal	normal	notpresent	notpresent	112	48	0.7	140	5	17	52	7200	5.8	no	no	no	good	no	no	notckd
3	52	80	1.025	0	0	normal	normal	notpresent	notpresent	99	25	0.8	135	3.7	15	52	6300	5.3	no	no	no	good	no	no	notckd
4	36	80	1.025	0	0	normal	normal	notpresent	notpresent	85	16	1.1	142	4.1	15.6	44	5800	6.3	no	no	no	good	no	no	notckd
5	57	80	1.02	0	0	normal	normal	notpresent	notpresent	133	48	1.2	147	4.3	14.8	46	6600	5.5	no	no	no	good	no	no	notckd
6	36	80	1.025	0	0	normal	normal	notpresent	notpresent	85	16	1.1	142	4.1	15.6	44	5800	6.3	no	no	no	good	no	no	notckd
7	57	80	1.025	0	0	normal	normal	notpresent	notpresent	133	48	1.2	147	4.3	14.8	46	6600	5.5	no	no	no	good	no	no	notckd
8	55	80	1.02	0	0	normal	normal	notpresent	notpresent	140	49	0.5	150	4.9	15.7	47	6700	4.9	no	no	no	good	no	no	notckd
9	42	70	1.025	0	0	normal	normal	notpresent	notpresent	75	31	1.2	141	3.5	16.5	54	7800	6.2	no	no	no	good	no	no	notckd
10	12	80	1.02	0	0	normal	normal	notpresent	notpresent	100	26	0.6	137	4.4	15.8	49	6600	5.4	no	no	no	good	no	no	notckd
11	17	60	1.025	0	0	normal	normal	notpresent	notpresent	114	50	1	135	4.9	14.2	51	7200	5.9	no	no	no	good	no	no	notckd
12	58	80	1.025	0	0	normal	normal	notpresent	notpresent	131	18	1.1	141	3.5	15.8	53	6800	6.1	no	no	no	good	no	no	notckd

Data Encoding

- ❑ Performance of a machine learning model depends highly on how you feed the variables.
- ❑ Most of the Machine Learning algorithm can process only numerical data.
- ❑ So, conversion of numerical from categorical is an important data cleaning process.

Data Encoding



dm	cad	appet	pe	ane	classification
yes	no	good	no	no	ckd
no	no	good	no	no	ckd
yes	no	poor	no	yes	ckd
no	no	poor	yes	yes	ckd
no	no	good	no	no	ckd
yes	no	good	yes	no	ckd
no	no	good	no	no	ckd
no	no	good	no	no	notckd
no	no	good	no	no	notckd
no	no	good	no	no	notckd

Data Encoding

Label Encoding: Assigns value from 1 to N either in an increasing or a decreasing order.
(Applies to Ordinal Attribute)

dm	cad	appet	pe	ane	classification	dm_encoded
yes	no	good	no	no	ckd	1
no	no	good	no	no	ckd	0
yes	no	poor	no	yes	ckd	1
no	no	poor	yes	yes	ckd	0
no	no	good	no	no	ckd	0
yes	no	good	yes	no	ckd	1
no	no	good	no	no	ckd	0
no	no	good	no	no	notckd	0
no	no	good	no	no	notckd	0
no	no	good	no	no	notckd	0

Data Encoding

One Hot Encoding: Maps attribute with a binary variable containing either 0 or 1.
(Applies to Nominal Attribute)

dm	cad	appet	pe	ane	classification	dm_no	dm_yes
yes	no	good	no	no	ckd	0	1
no	no	good	no	no	ckd	1	0
yes	no	poor	no	yes	ckd	0	1
no	no	poor	yes	yes	ckd	1	0
no	no	good	no	no	ckd	1	0
yes	no	good	yes	no	ckd	0	1
no	no	good	no	no	ckd	1	0
no	no	good	no	no	notckd	1	0
no	no	good	no	no	notckd	1	0
no	no	good	no	no	notckd	1	0

Data Encoding

Frequency Encoding: Assigns value as per the frequency of values in its total lot.

dm	cad	appet	pe	ane	classification	dm_freq_encode
yes	no	good	no	no	ckd	0.3
no	no	good	no	no	ckd	0.7
yes	no	poor	no	yes	ckd	0.3
no	no	poor	yes	yes	ckd	0.7
no	no	good	no	no	ckd	0.7
yes	no	good	yes	no	ckd	0.3
no	no	good	no	no	ckd	0.7
no	no	good	no	no	notckd	0.7
no	no	good	no	no	notckd	0.7
no	no	good	no	no	notckd	0.7

Data Transformation

- The range of the attributes in a dataset may differ a lot.
- This leads to giving more importance to a feature than the others.
- So, data transformation is meant to produce same scale to all the features.

rbc	pc	pcc	ba	bgr	bu	sc	sod
normal	normal	notpresent	notpresent	70	36	1	150
normal	normal	notpresent	notpresent	82	49	0.6	147
normal	normal	notpresent	notpresent	119	17	1.2	135
normal	normal	notpresent	notpresent	99	38	0.8	135
normal	normal	notpresent	notpresent	121	27	1.2	144
normal	normal	notpresent	notpresent	131	10	0.5	146
normal	normal	notpresent	notpresent	91	36	0.7	135



Data Transformation –Scaling

Scaling

Min-Max Scaling

Standard Scaling

Maximum Absolute
Scaling

Robust Scaling

Min-Max Scaling: Feature values are scaled in a way that range of attribute values would be between 0 and 1 (or -1 and 1).

Standard Scaling: Feature values are scaled in a way to ensure the mean and the standard deviation to be 0 and 1.

Standardisation (Z-score Normalization)

$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation}(x)}$$

Max-Min Normalization

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$$



Data Transformation - Scaling

Maximum Absolute Scaling: Rescales each feature between -1 and 1 by dividing every observation by its maximum absolute value.

$$x_{scaled} = \frac{x}{\max(|x|)}$$

Robust Scaling: Rescales each feature of the data set by subtracting the median and then dividing by the interquartile range.

$$x_{rs} = \frac{x_i - Q_2(x)}{Q_3(x) - Q_1(x)}$$



Data Transformation - DBSCAN

- ❑ Clustering algorithms is used classify the data points into specific groups for a given set of data points.
- ❑ Used to segregate data based on their properties/ features and group them into different clusters.
- ❑ Application: classifying diseases in medical science and classifying customers in the field of market research

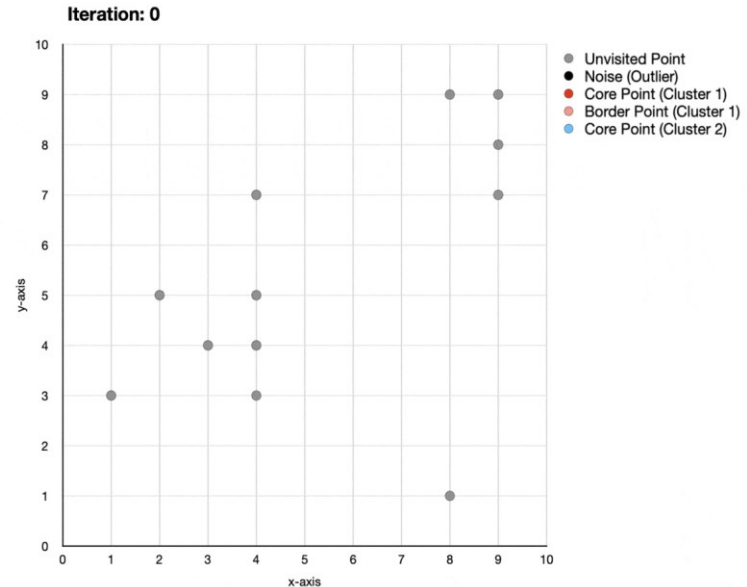


DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

1. It starts with a random unvisited starting data point. All points within a distance 'Epsilon - ϵ ' classify as neighborhood points.

2. You need a minimum number of points within the neighborhood to start the clustering process. Under such circumstances, the current data point becomes the first point in the cluster. Otherwise, the point gets labeled as 'Noise.' In either case, the current point becomes a visited point.

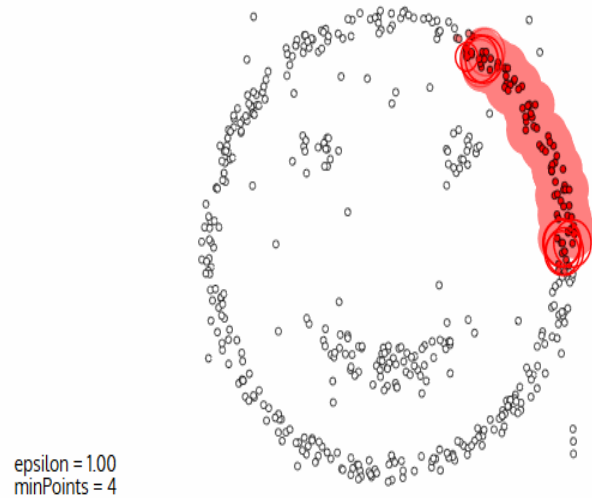
3. All points within the distance ϵ become part of the same cluster. Repeat the procedure for all the new points added to the cluster group.





DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

4. Continue with the process until you visit and label each point within the ϵ neighborhood of the cluster.
5. On completion of the process, start again with a new unvisited point thereby leading to the discovery of more clusters or noise. At the end of the process, you ensure that you mark each point as either cluster or noise.



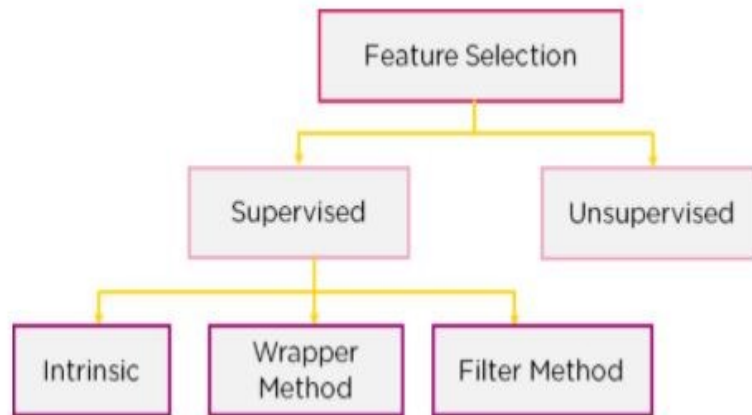
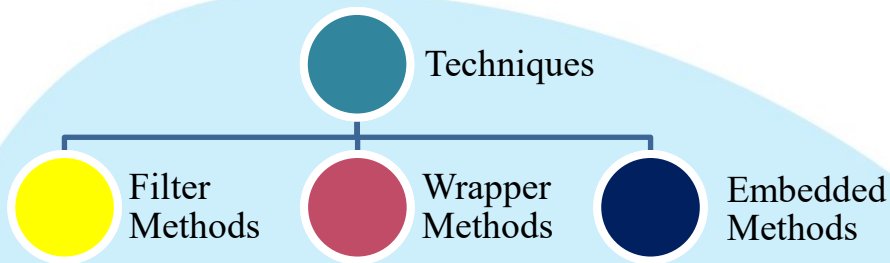
Restart

Pause



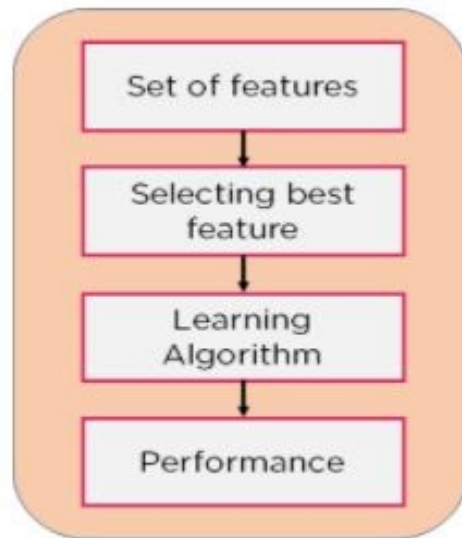
Feature Selection

- ❑ Used to reduce the number of attributes while building a predictive model.
- ❑ Reducing the redundant attributes would reduce time and computational complexity.
- ❑ Selects the attributes which have strong relationship with the target attribute.



Feature Selection – Filter Methods

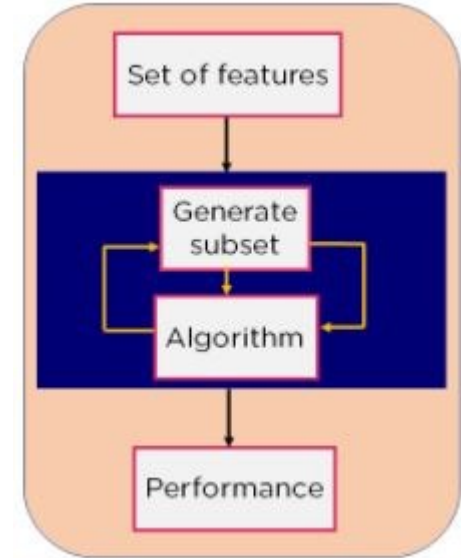
- ❑ Selects features based on statistical measure.
- ❑ Uses correlation to check if the features are positively or negatively correlated to the output labels and drop features accordingly.



Examples: Information Gain, Chi-square test, Fisher's Score, Correlation Coefficient, Variance Threshold, Mean Absolute Difference (MAD), Dispersion Ratio, Mutual Dependence , Relief.

Feature Selection – Wrapper Methods

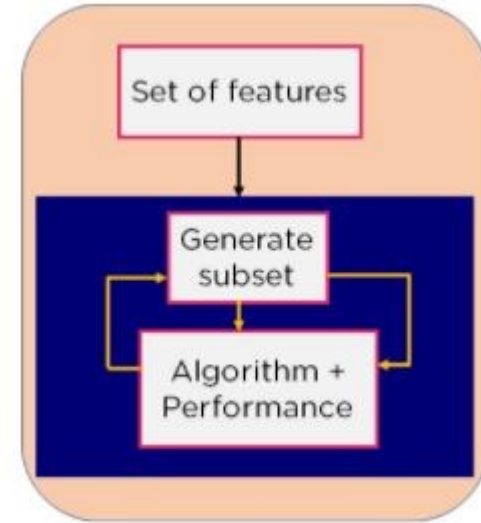
- ❑ Splits the data into several subsets.
- ❑ Feed the subsets to algorithm and observe the accuracy.
- ❑ Finally outputs the subset with higher accuracy.



Examples: Recursive Feature Elimination, Forward Selection , Backward Elimination, Exhaustive Selection, Bi-directional Elimination.

Feature Selection – Embedded Methods

- ❑ Combines the qualities of both the Filter and Wrapper method to create the best subset.



Examples: Ridge Regression, LASSO Regression , Elastic Net, Random Forest Importance.

Correlation in Feature Selection

1. Correlation with target attribute. E.g. Correlation of Breathing problem, Chest pain with presence of CORONA.
2. Inter-correlation among independent attributes. E.g. Correlation of age with Date of Birth.

Tools for Machine Learning



THANK YOU
FOR
LISTENING



Dataset Splitting

It is standard in ML to split data into training and test sets. This step is usually implemented after pre-processing.

Data Splitting Approaches:

- ❑ **Hold Out Validation Approach**
 - ❑ Train Test Split Method
- ❑ **Cross Validation Approach**
 - ❑ Leave One Out Cross Validation
 - ❑ K Fold Cross Validation
 - ❑ Stratified K Fold Cross Validation
 - ❑ Time Series Cross Validation

Train Test Split Method

Independent Variable(x)			Dependent Variable $y=F(x)$
45	34	56	1
67	45	65	1
33	45	23	0
45	67	34	1
34	34	56	1
27	23	67	0
45	45	53	1
34	36	34	1
45	75	34	1
65	34	23	0

x_train			y_train
34	34	56	1
65	34	23	0
45	45	53	1
33	45	23	0
67	45	65	1
34	36	34	1
45	34	56	1

x_test			y_test
45	67	34	1
27	23	67	0
45	75	34	1

Syntax:

```
from sklearn.model_selection import train_test_split  
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size, random_state)
```

Cross Validation Approach

Leave one out cross validation

	1	2	3	4	5	6	7	8	9	10
Iteration_1	Test	Train	Train	Train	Train	Train	Train	Train	Train	Train
Iteration_2	Train	Test	Train	Train	Train	Train	Train	Train	Train	Train
Iteration_3	Train	Train	Test	Train	Train	Train	Train	Train	Train	Train
Iteration_4	Train	Train	Train	Test	Train	Train	Train	Train	Train	Train
Iteration_5	Train	Train	Train	Train	Test	Train	Train	Train	Train	Train
Iteration_6	Train	Train	Train	Train	Train	Test	Train	Train	Train	Train
Iteration_7	Train	Train	Train	Train	Train	Train	Test	Train	Train	Train
Iteration_8	Train	Train	Train	Train	Train	Train	Train	Test	Train	Train
Iteration_9	Train	Train	Train	Train	Train	Train	Train	Train	Test	Train
Iteration_10	Train	Train	Train	Train	Train	Train	Train	Train	Train	Test

$$ACC = \frac{\sum \text{accuracy of each iteration}}{\text{total number of iteration}}$$

Cross Validation Approach(Cont.)

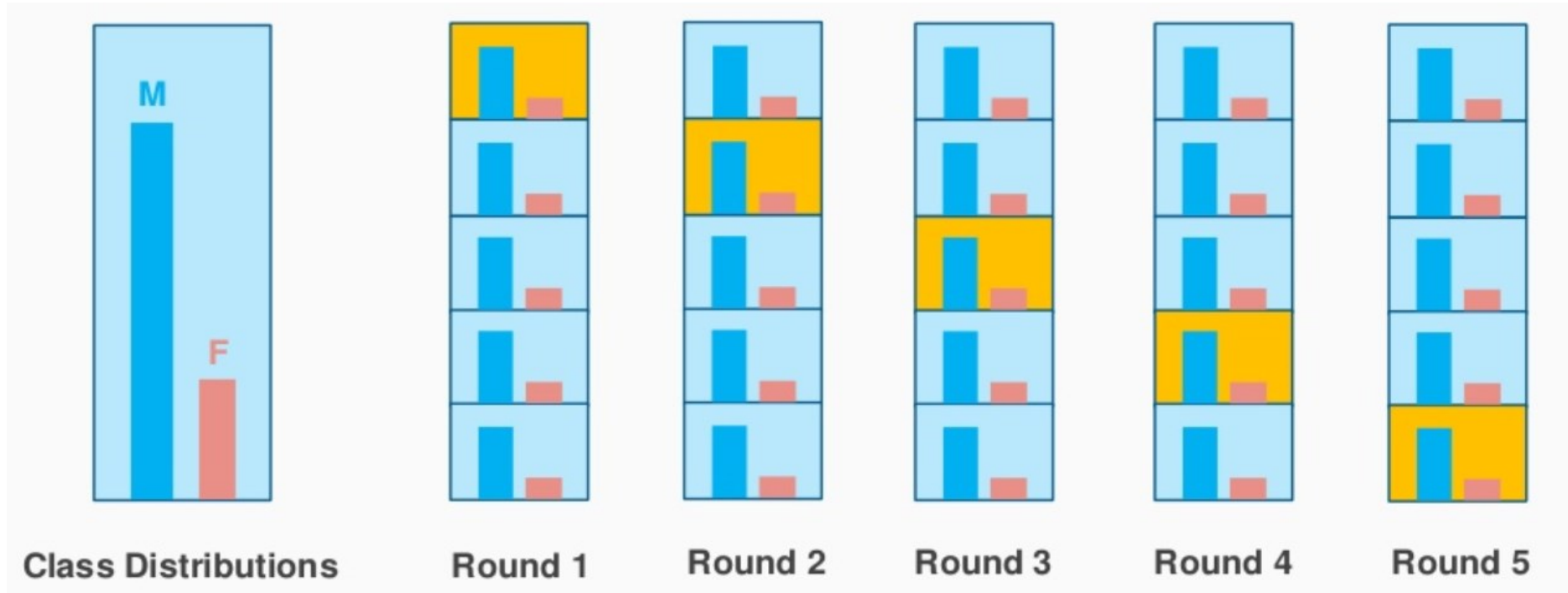
K Fold Cross Validation

10 Records					
	2	2	2	2	2
Iteration_1	Test	Train	Train	Train	Train
Iteration_2	Train	Test	Train	Train	Train
Iteration_3	Train	Train	Test	Train	Train
Iteration_4	Train	Train	Train	Test	Train
Iteration_5	Train	Train	Train	Train	Test

$$ACC = \frac{\sum \text{accuracy of each iteration}}{\text{total number of iteration}}$$

Cross Validation Approach(Cont.)

Stratified K Fold Cross Validation



$$ACC = \frac{\sum \text{accuracy of each round}}{\text{total number of round}}$$

Cross Validation Approach(Cont.)

Difference Between K Fold And Stratified K Fold Cross Validation (Example) :

100 Records	
-------------	--

80% Class_1
20% Class_2

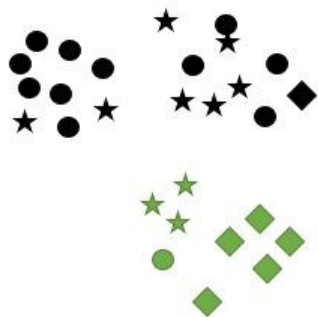
Block1: 25		Block2: 25		Block3: 25		Block3: 25	
Class_1	Class_2	Class_1	Class_2	Class_1	Class_2	Class_1	Class_2
8	17	24	1	24	1	24	1

100 Records	
-------------	--

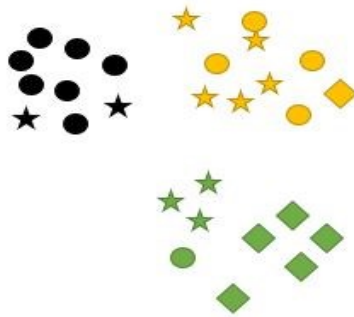
80% Class_1
20% Class_2

Block1: 25		Block2: 25		Block3: 25		Block4: 25	
Class_1	Class_2	Class_1	Class_2	Class_1	Class_2	Class_1	Class_2
20	5	20	5	20	5	20	5

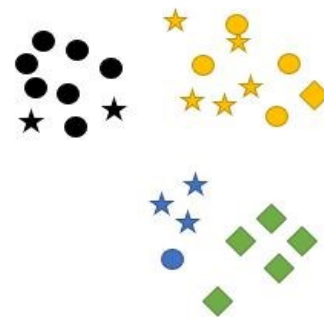
Classification and Clustering



2 Cluster



3 Cluster



4 Cluster

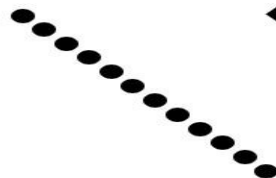
Train Class :

★ =Star; ● = Circle; ◆ = Dimond

Classification:



Star



Circle

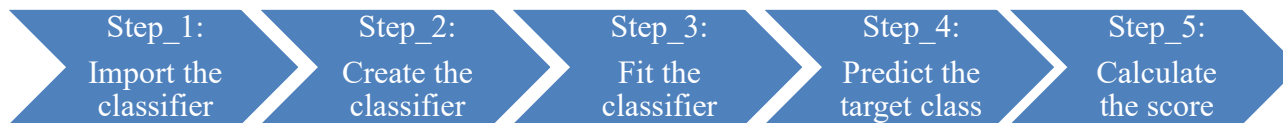


Dimond

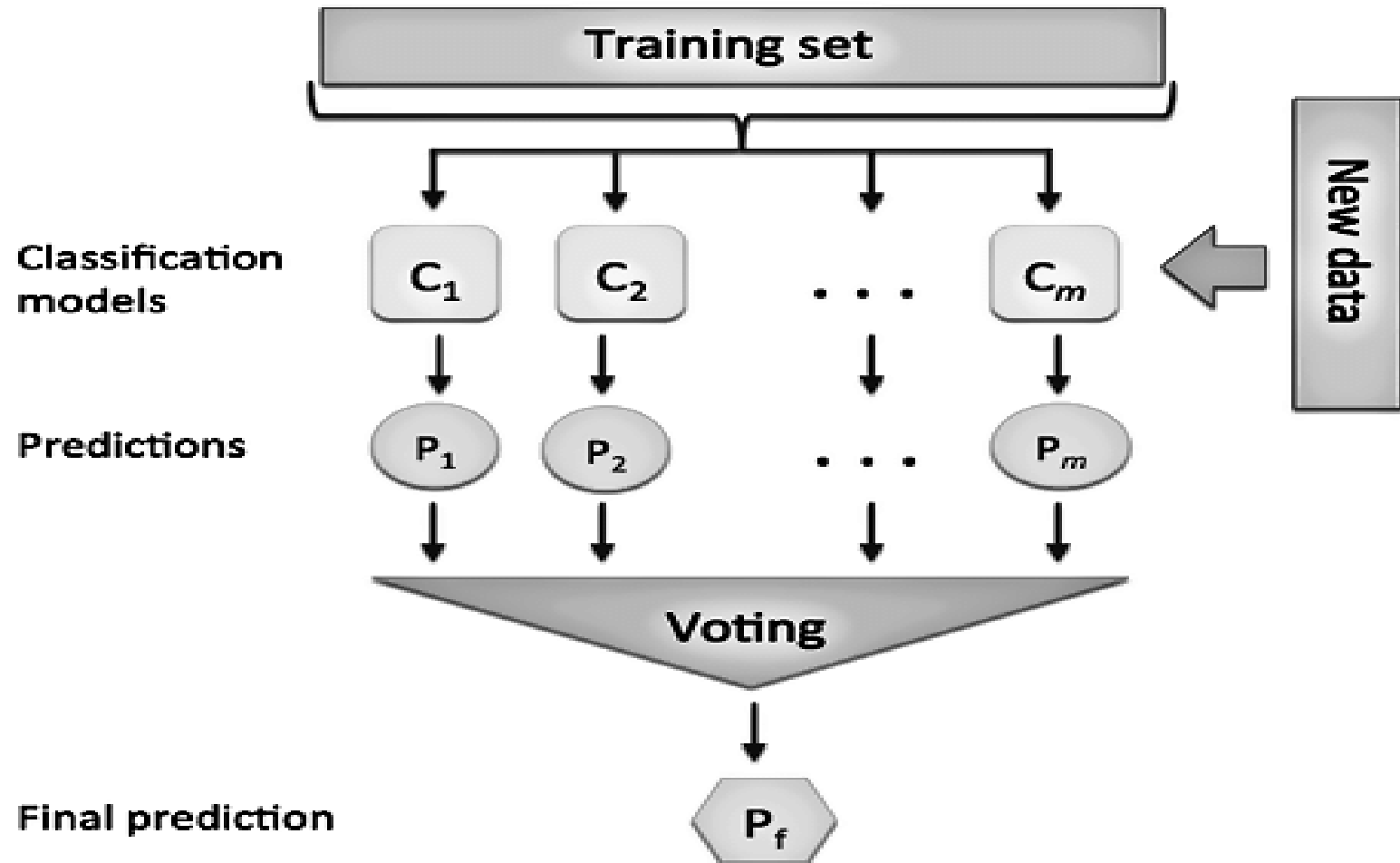
Classification

Classification With Train Test Split Method

Classifiers	Step_1	Step_2
Random Forest	<code>from sklearn.ensemble import RandomForestClassifier</code>	<code>RandomForestClassifier()</code>
Ada Boost	<code>from sklearn.ensemble import AdaBoostClassifier</code>	<code>AdaBoostClassifier()</code>
XgBoost	<code>from xgboost import XGBClassifier</code>	<code>XGBClassifier()</code>
Gradient Boosting	<code>from sklearn.ensemble import GradientBoostingClassifier</code>	<code>GradientBoostingClassifier()</code>
K Nearest Neighbor	<code>from sklearn.neighbors import KNeighborsClassifier</code>	<code>KNeighborsClassifier(n_neighbors)</code>
Logistic Regression	<code>from sklearn.linear_model import LogisticRegression</code>	<code>LogisticRegression()</code>
Naive Bayes	<code>from sklearn.naive_bayes import GaussianNB</code>	<code>GaussianNB()</code>
Decision Tree	<code>from sklearn.tree import DecisionTreeClassifier</code>	<code>DecisionTreeClassifier()</code>
Support Vector Machine	<code>from sklearn import svm</code>	<code>svm.SVC()</code>
Linear Regression	<code>from sklearn.linear_model import LinearRegression</code>	<code>LinearRegression()</code>

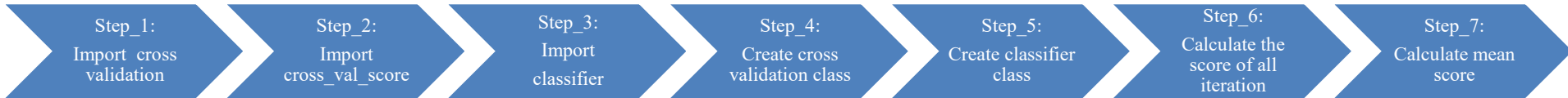


Voting Ensemble Classification With Train Test Split Method

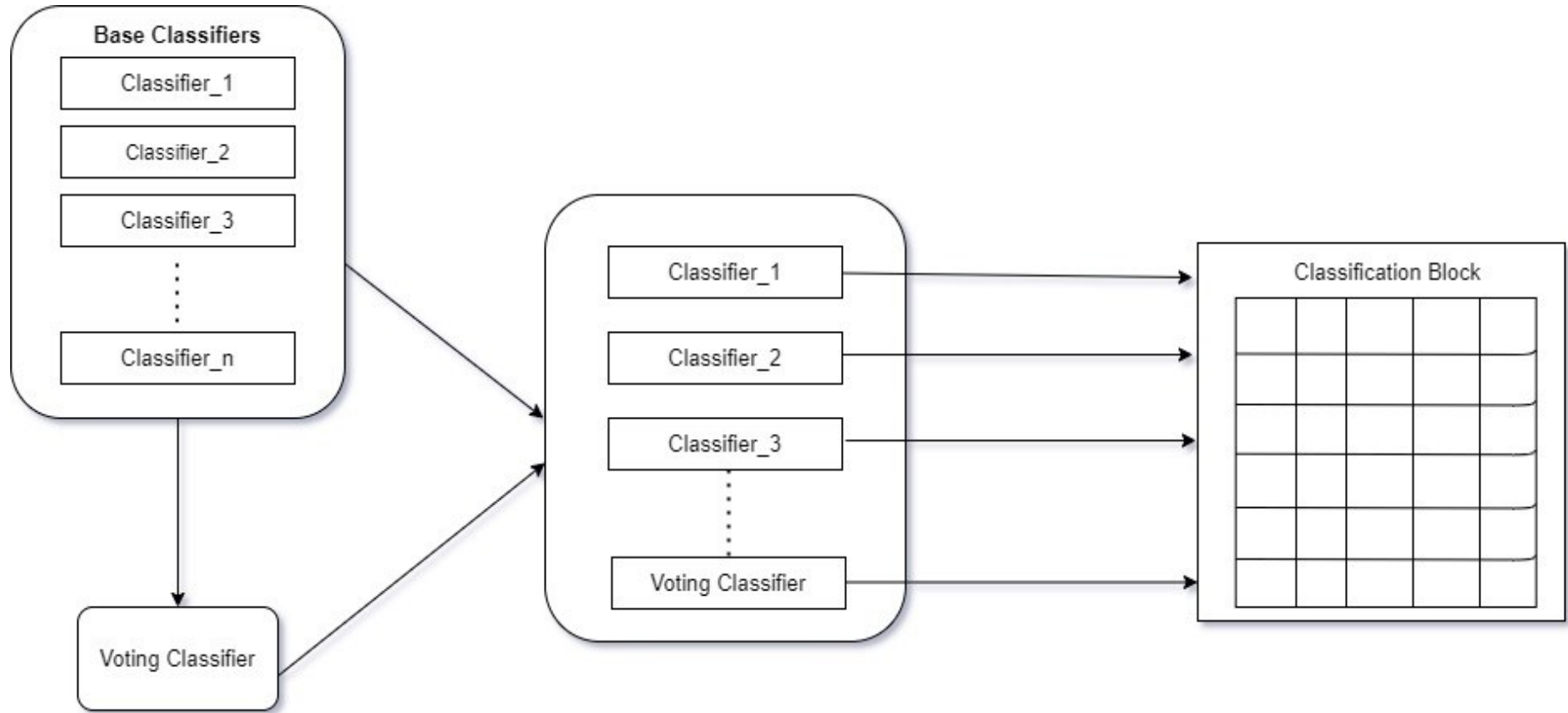


Classification With Cross Validation

Cross Validation	Step_1	Step_4
LeaveOneOut	<code>from sklearn.model_selection import LeaveOneOut</code>	<code>LeaveOneOut()</code>
KFold	<code>from sklearn.model_selection import KFold</code>	<code>Kfold()</code>
StratifiedKFold	<code>from sklearn.model_selection import StratifiedKFold</code>	<code>StratifiedKFold()</code>



Voting Ensemble Classification With Cross Validation



Confusion Matrix

		Predicted	
		0	1
Actual	0	TN	FP
	1	FN	TP

$$\text{Accuracy} = \frac{TN+TP}{TN+TP+FN+FP}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall / Sensitivity / True Positive Rate} = \frac{TP}{\text{Actual Positive}} = \frac{TP}{TP+FN}$$

$$\text{False Positive Rate} = \frac{FP}{\text{Actual Negative}} = \frac{FP}{FP+TN}$$

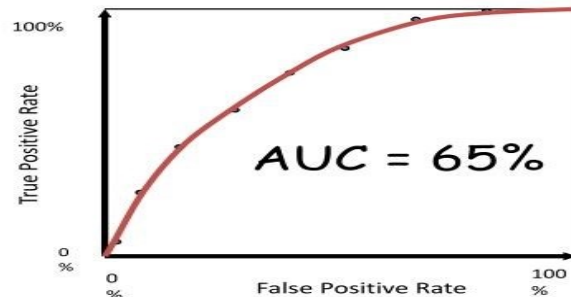
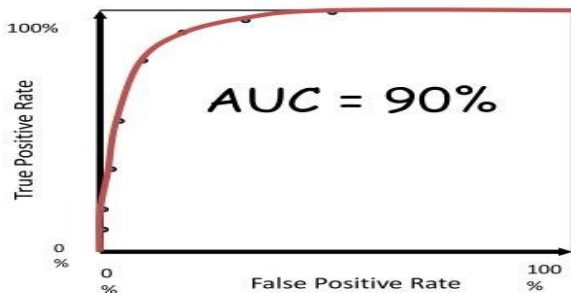
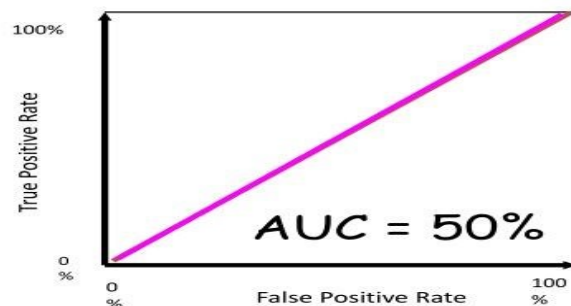
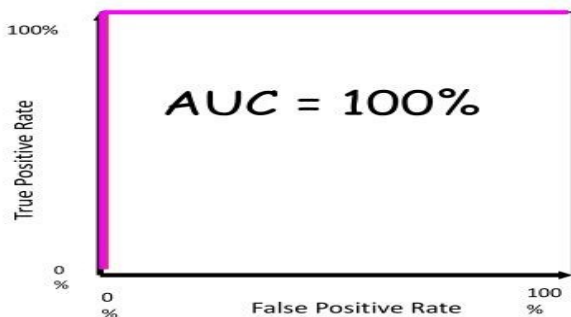
$$\text{Specificity / True Negative Rate} = \frac{TN}{\text{Actual Negative}} = \frac{TN}{TN+FP}$$

$$\text{False Negative Rate} = \frac{FN}{\text{Actual Positive}} = \frac{FN}{FN+TP}$$

ROC and AUC

AUC and ROC :

ROC(Receiver Operating Characteristic Curve) curve shows the performance of a classification model at all classification thresholds. ROC is a curve of probability. The ROC curve is plotted with TPR against the FPR



Multi Class Classification

Multiclassification Approach:

- ❑ One Vs Rest or One Vs All Classifier
- ❑ One Vs One Classifier

Confusion Matrix of Multi Class:

	0	1	2
0	TP	FN	FN
1	FP	TN	TN
2	FP	TN	TN

for class_0

Positive : 0 ; Negative : 1, 2

TP = [0,0]

TN = [1,1] + [1,2] + [2,1] + [2,2]

FP = [1,0] + [2,0]

FN = [0,1] + [0,2]

	0	1	2
0	TN	FP	TN
1	FN	TP	FN
2	TN	FP	TN

for class_1

Positive : 1 ; Negative : 0,2

TP = [1,1]

TN = [0,0] + [0,2] + [2,0] + [2,2]

FP = [0,1] + [2,1]

FN = [1,0] + [1,2]

	0	1	2
0	TN	TN	FP
1	TN	TN	FP
2	FN	FN	TP

for Class_2

Positive : 2 ; Negative : 0,1

TP = [2,2]

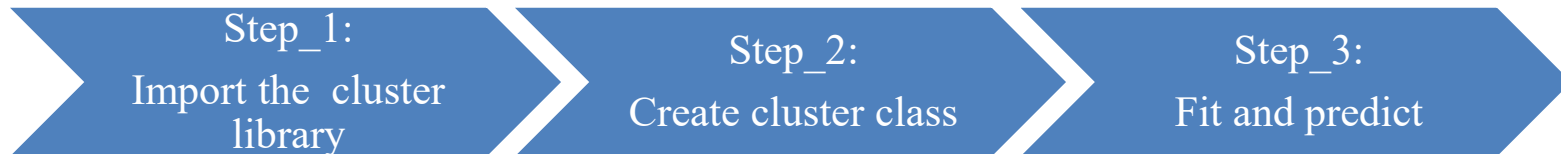
TN = [0,0] + [0,1] + [1,0] + [1,1]

FP = [0,2] + [1,2]

FN = [2,0] + [2,1]

Cluster Algorithm Implementation

Clustering Algorithm	Step_1	Step_2
Simple K means	<code>from sklearn.cluster import AgglomerativeClustering</code>	<code>KMeans(num_clusters)</code>
Agglomerative Clustering	<code>from sklearn.cluster import KMeans</code>	<code>AgglomerativeClustering(num_clusters)</code>



Result Analysis



THANK YOU
FOR
LISTENING

