ML Assignment - 1

Name **:** Isha Girish Kanade

Roll No. **:** 33135

Batch **:** K9

Title: Data Preparation

Problem Statement:

Perform following operation on given dataset:

a) Find Shape of Data
b) Find Missing Values
c) Find data type of each column
d) Finding out Zero's
e) Find Mean age of patients
f) Now extract only Age, Sex, ChestPain, RestBP, Chol. Randomly divide dataset in training (75%) and testing (25%).
g) Through the diagnosis test I predicted 100 report as COVID positive, but only 45 of those wereactually positive. Total 50 people in my sample were actually COVID positive. I have total 500 samples.

Create confusion matrix based on above data and find

i. Accuracy
ii. Precision
iii. Recall
iv. F-1 score

Objective: This assignment will help the students to realize what is need of data preparation

Theory:

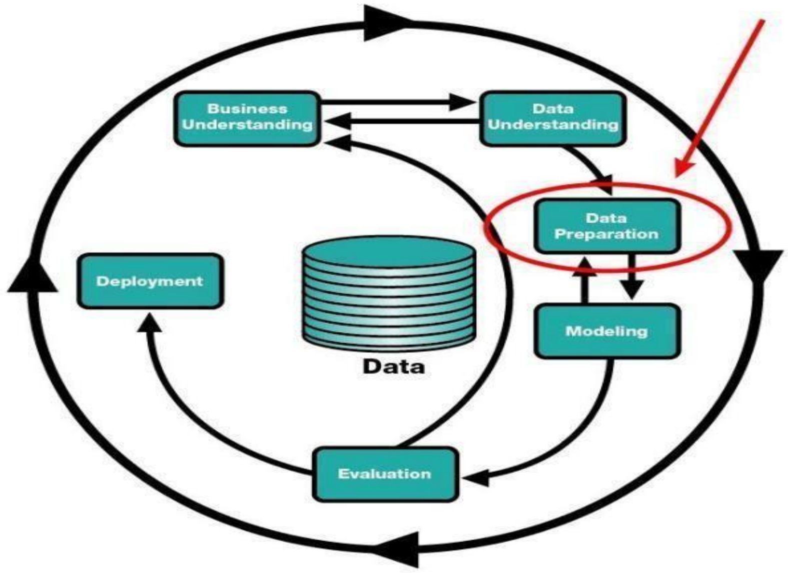
Data Preparation

Data preparation (also referred to as "data preprocessing") is the process of transforming raw data so that data scientists and analysts can run it through machine learning algorithms to uncover insights or make predictions.

Why is Data Preparation Important?

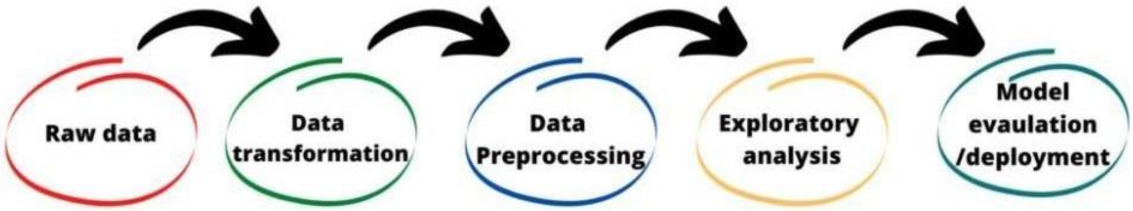
Most machine learning algorithms require data to be formatted in a very specific way, so datasets generally require some amount of preparation before they can yield useful insights. Some datasets have values that are missing, invalid, or otherwise difficult for an algorithm to process. If data is missing, the algorithm can't use it. If data is invalid, the algorithm produces less accurate or even misleading outcomes. Some datasets are relatively clean but need to be shaped (e.g., aggregated or pivoted) and many datasets are just lacking useful business context (e.g., poorly defined ID values), hence the need for feature enrichment. Good data preparation produces clean and well-curated data which leads to more practical, accurate model outcomes.

It is the most required process before feeding the data into the machine learning model. The reason behind that the data set needs to be different and specific according to the model so that wehave to find out the required features of that data. The data preparation process offers a method via which we can prepare the data for defining the project and also for the project evaluation of ML algorithms. Different many predicting machine learning models are there with a different process but some of the processes are common that are performed in every model, and also it allows us to find out the actual business problem and their solutions. Some of the data preparationprocesses are:



Data Preparation [3]

1. **Determine the problems**
2. **Data cleaning**
3. **Feature selection**
4. **Data transformation**
5. **Feature engineering**
6. **Dimensionality reduction**



1. **Determine the problems:**

This step tells us about the learning method of the project to find out the results for future prediction or forecasting. For example, which ML model suitable for the data set regression or classification or clustering algorithms. This includes data collection that is useful for predicting the result and also involving the communication to project stakeholders and domain expertise. We use classification and regression models for categorical and numerical data respectively.

It includes determining the relevant attributes with the stied data in form of .csv, .html, .json, .doc, and many, also for unstructured data in a form for audio, video, text, images, etc for scanning and detect the patterns of data with searching and identifying the data that have taken from external repositories.

2. **Data cleaning:**

After collecting the data, it is very necessary to clean that data and make it proper for the ML model. It includes solving problems like outliers, inconsistency, missing values, incorrect, skewed, and trends. Cleaning the data is very important as the model learning from that data only, so if we feed inconsistent, appropriate data to model it will return garbage only, so it is required to make sure that the data does not contains any unseen problem. For example, if we have a data set of sales, it might be possible that it contains some features like height, age, that cannot help in the model building so we can remove it. We generally remove the null values columns, fill the missing values, make the data set consistent, and remove the outliers and skewed data in data cleaning.

3. **Feature selection:**

Sometimes we face the problem of identifying the related features from the set of data and deleting the irrelevant and less important data without touching the target variables to get the better accuracy of the model. Features selection plays a wide role in building a machine learning model that impacts the performance and accuracy of the model. It is that process which contributes mostly to the predictions or output that we need by selecting the features automatically or manually. If we have irrelevant data that would cause the model with overfitting and underfitting.

The benefits of feature selection:

1. Reduce the overfitting/underfitting
2. Improves the accuracy
3. Reduced training/testing time
4. Improves performance

4. Data transformation:

Data transformation is the process that converts the data from one form to another. It is required for data integration and data management. In data transformation, we can change the types of data, clear the data removing the null values or duplicate values, and get enrich data that depends on the requirements of the model. It allows us to perform data mapping that determines how individual features are mapped, modified, filtered, aggregated, and joined. Data transformation is needed for both structured and unstructured data, but it is time consuming, costly, slow.

5. Feature engineering:

All ML algorithms use some input data for giving required output and this input required some features which are in a structured form. To get the proper result the algorithms required features with some specific characteristics which we find out with feature engineering. we need to perform different feature engineering on different datasets, and we can observe their effect on model performance. Here I am listing out the techniques of feature engineering.

1. Imputation
2. Handling outliers
3. Binning
4. Log transform
5. one-hot encoding
6. Grouping operations
7. Feature split
8. Scaling

6. **Dimensionality reduction:**

When we use the dataset for building an ML model, we need to work with 1000s of features that cause the curse of dimensionality, or we can say that it refers to the process to convert a set of data.For the ML model, we have to access a large amount of data and that large amount of data can lead us in a situation where we can take possible data that can be available to feed it into a forecasting model to predict and give the result of the target variable. It reduced the time that is required for training and testing our machine learning model and also helps to eliminate over- fitting. It is kind of zipping the data for the model.

Implementation : Attached below.

Conclusion:

Data preparation is recognized for helping businesses and analytics to get ready and prepare the data for operations.

Name: Isha Kanade
Roll no.: 33135
Batch: K9

LP1 ML - Assignment 1

Exercise - Part A

Download heart.csv

Perform the following operations on the dataset:

1. Find shape of data
2. Find missing values
3. Find datatype of each column.
4. Finding out Zero's
5. Find mean age of patients.
6. Extract only Age,Sex,ChestPain,RestBP,Chol.
7. Randomly divide the dataset in training(75%) and testing(25%)

In [1]:
```
#importing required libraries

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
from sklearn.model_selection import train_test_split #splitting the dataset
```

In [2]:
```
#importing dataset

heart = pd.read_csv('heart.csv')
heart
```

Out[2]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal	AHD
0	1	63	1	typical	145	233	1	2	150	0	2.3	3	0.0	fixed	No
1	2	67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.0	normal	Yes
2	3	67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.0	reversable	Yes
3	4	37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.0	normal	No
4	5	41	0	nontypical	130	204	0	2	172	0	1.4	1	0.0	normal	No
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	299	45	1	typical	110	264	0	0	132	0	1.2	2	0.0	reversable	Yes
299	300	68	1	asymptomatic	144	193	1	0	141	0	3.4	2	2.0	reversable	Yes
300	301	57	1	asymptomatic	130	131	0	0	115	1	1.2	2	1.0	reversable	Yes
301	302	57	0	nontypical	130	236	0	2	174	0	0.0	2	1.0	normal	Yes
302	303	38	1	nonanginal	138	175	0	0	173	0	0.0	1	NaN	normal	No

303 rows × 15 columns

In [3]:
```
#shape of the data
print("Shape of the dataset : ",heart.shape)
```

```
Shape of the dataset :  (303, 15)
```

In [4]:
```
#check whether any missing value in data
heart.isnull().sum()
```

Out[4]:
```
Unnamed: 0    0
Age           0
Sex           0
ChestPain     0
RestBP        0
Chol          0
Fbs           0
RestECG       0
MaxHR         0
ExAng         0
Oldpeak       0
Slope         0
Ca            4
Thal          2
AHD           0
dtype: int64
```

There are 4 missing values in "Ca" column and 2 missing values in "Thal" column

In [5]:
```
#finding datatype of each column
heart.dtypes
```

Out[5]:
```
Unnamed: 0      int64
Age             int64
Sex             int64
ChestPain      object
RestBP          int64
Chol            int64
Fbs             int64
RestECG         int64
MaxHR           int64
ExAng           int64
Oldpeak       float64
Slope           int64
Ca            float64
Thal           object
AHD            object
dtype: object
```

In [6]:
```
#Finding out zeros
(heart == 0).sum()
```

Out[6]:
```
Unnamed: 0      0
Age             0
Sex            97
ChestPain       0
RestBP          0
Chol            0
Fbs           258
RestECG       151
MaxHR           0
ExAng         204
Oldpeak        99
Slope           0
Ca            176
Thal            0
AHD             0
dtype: int64
```

In [7]:
```
#Find mean age of patients
print("Mean age of patients is :",heart['Age'].mean())
```

```
Mean age of patients is : 54.43894389438944
```

In [8]:
```
#Extracting only Age,Sex,ChestPain,RestBP,Chol without changing the initial dataset
heart_extract = heart.filter(['Age','Sex', 'ChestPain', 'RestBP', 'Chol'])
heart_extract
```

Out[8]:

	Age	Sex	ChestPain	RestBP	Chol
0	63	1	typical	145	233
1	67	1	asymptomatic	160	286
2	67	1	asymptomatic	120	229
3	37	1	nonanginal	130	250
4	41	0	nontypical	130	204
...	...	...	...	...	...
298	45	1	typical	110	264
299	68	1	asymptomatic	144	193
300	57	1	asymptomatic	130	131
301	57	0	nontypical	130	236
302	38	1	nonanginal	138	175

303 rows × 5 columns

Test-Train Split

In [9]:
```
X = heart_extract
```

In [10]:
```
#splitting the data set with test size = 25% and train = 75%
X_train,X_test = train_test_split(X,test_size=0.25 ,random_state=1)
```

In [11]:
```
X_train
```

Out[11]:

	Age	Sex	ChestPain	RestBP	Chol
170	70	1	nonanginal	160	269
192	43	1	asymptomatic	132	247
168	35	1	asymptomatic	126	282
42	71	0	nontypical	160	302
90	66	1	asymptomatic	120	302
...	...	...	...	...	...
203	64	0	nonanginal	140	313
255	42	0	nonanginal	120	209
72	62	1	asymptomatic	120	267
235	54	1	asymptomatic	122	286
37	57	1	asymptomatic	150	276

227 rows × 5 columns

In [12]:
```
X_test
```

Out[12]:

	Age	Sex	ChestPain	RestBP	Chol
204	43	1	asymptomatic	110	211
159	68	1	nonanginal	118	277
219	59	1	asymptomatic	138	271
174	64	1	asymptomatic	145	212
184	60	0	asymptomatic	158	305
...	...	...	...	...	...
131	51	1	nonanginal	94	227
234	54	0	nonanginal	160	201
107	57	1	nonanginal	128	229
285	58	1	asymptomatic	114	318
17	54	1	asymptomatic	140	239

76 rows × 5 columns

Exercise - Part B

Through diagnosis test I predicted 100 report as COVID positive, but only 45 of those were actually positive. Total 50 people in my sample were actually COVID positive. I have total 500 samples.

Create confusion matrix based on above data and find

1. Accuracy
2. Precision
3. Recall
4. F1 Score

In [13]:
```
#Getting the values from data
tp = 45 #true positive
fp = 55 #false positive
tn = 395 #true negative
fn = 5 #false negative
```

In [14]:
```
#User defined Confusion matrix
conf_m = np.matrix([[tp, fp], [fn, tn]])
print('Confusion Matrix :\n', conf_m)
```

```
Confusion Matrix :
 [[ 45  55]
 [  5 395]]
```

In [15]:
```
accuracy = (tp + tn) / (tp + fp + tn + fn)
print("Accuracy : ",accuracy)
```

```
Accuracy :  0.88
```

Precision tells us how many of the correctly predicted cases actually turned out to be positive.

In [16]:
```
precision = tp / (tp + fp)
print("Precision : ",precision)
```

```
Precision :  0.45
```

Recall tells us how many of the actual positive cases we were able to predict correctly with our model.

In [17]:
```
recall = tp / (tp + fn)
print("Recall : ",recall)
```

```
Recall :  0.9
```

F1-score is a harmonic mean of Precision and Recall, and so it gives a combined idea about these two metrics. It is maximum when Precision is equal to Recall

In [18]:
```
f1_score = 2 / ((1/recall) + (1/precision))
print("F1 Score : ",f1_score)
```

```
F1 Score :  0.6
```