Handout 4
Need & Basics of Data Cleaning

Introduction

There are no shortcuts for data exploration. If you are in a state of mind, that machine learning can sail you away from every data storm, trust me, it won't. After some point of time, you'll realize that you are struggling at improving model's accuracy. In such situation, data exploration techniques will come to your rescue.

Why Data Cleaning?

Imagine that you have a room filled with dozens of sleeping cats, and you want to know how many cats there are. It would also be good to know some basic insights about your new cat colony — for example, what colors the cats are and whether any of them have extra long tails.

Not difficult, right? Just go around the room and check out each cat.

Now imagine that the room is also filled with dozens of birds and flying squirrels, and all the cats are hyped up on catnip. So counting the no of cats with specific insights would be a tedious task in the presence of the other animals and you will be still not sure of the count.

A dirty data set is like a crazy animal-filled room. It's possible to wrangle around your data points — and get insights from them, but it won't be fun and you'll still be pretty uncertain by the end.

Solution :

1. It's better to first bring order to the room.
2. Calm the animals down and remove unnecessary ones
3. Organize the remaining cats
4. Only then get the insights you need.

The more you care about your insights, the more you should care about data cleaning. After all, you don't want to make important decisions based on incorrect insights simply because your data had errors.

Points to ponder:

1. Quality of input decides the quality of output.
2. Data exploration, cleaning and preparation can take up to 60% of the total project time – for example: ML, Big Data

Basics of Data Cleaning

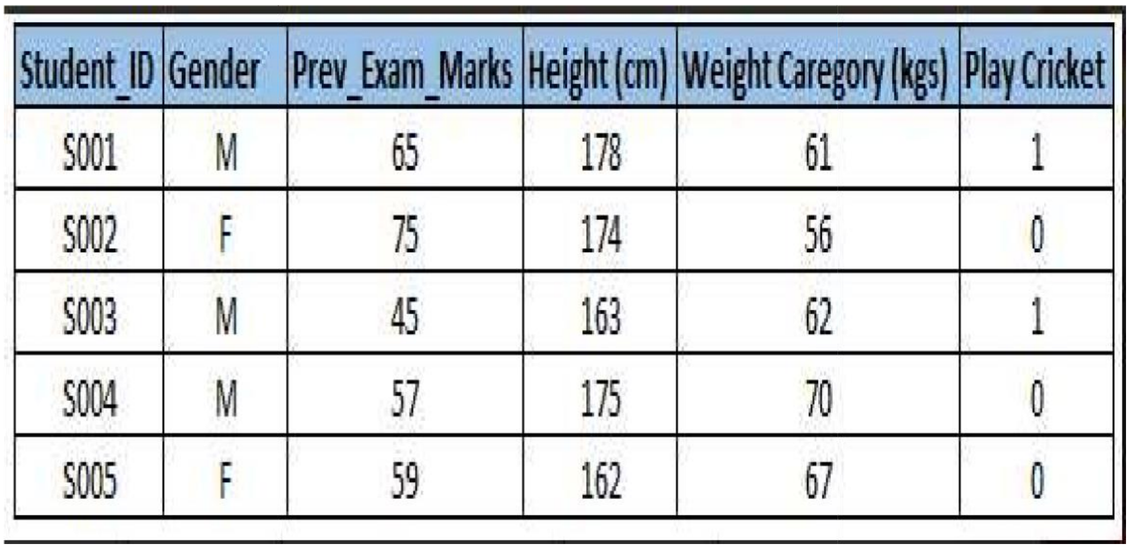
Below are the steps involved to understand, clean and prepare your data for building your predictive model:

1. Variable Identification
2. Univariate Analysis
3. Missing values treatment
4. Outlier treatment
5. Variable transformation
6. Variable creation

1. Variable Identification

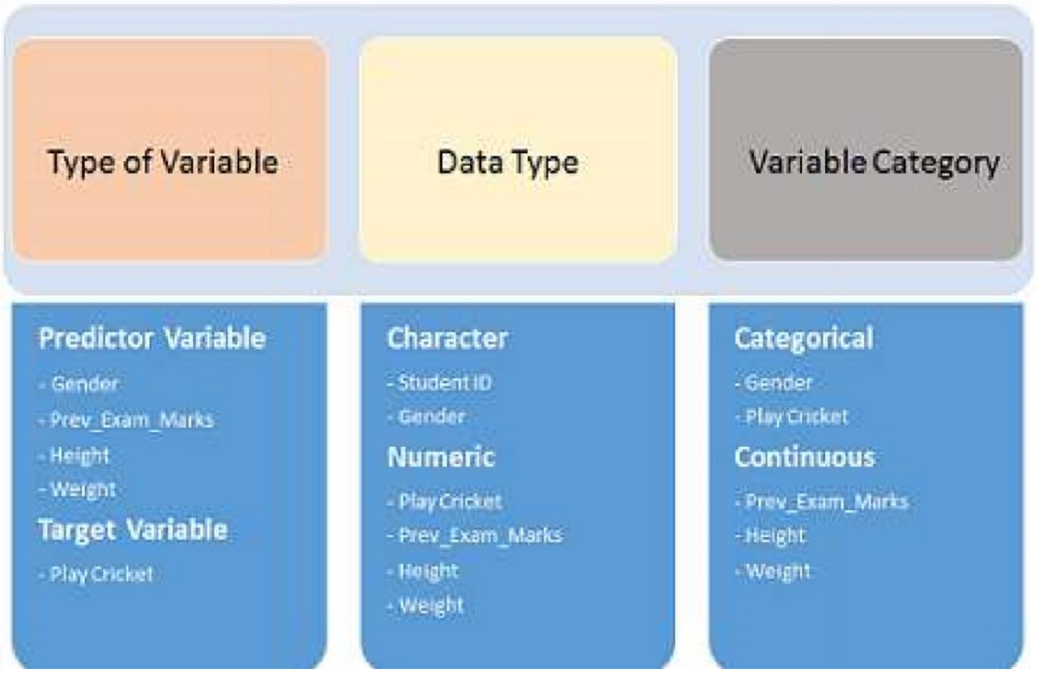
First, identify Predictor (Input) and Target (output) variables. Next, identify the data type and category of the variables.

Eg: Suppose, we want to predict, whether the students will play cricket or not (refer below data set). Here you need to identify predictor variables, target variable, data type of variables and category of variables.



Student_ID	Gender	Prev_Exam_Marks	Height (cm)	Weight Caregory (kgs)	Play Cricket
S001	M	65	178	61	1
S002	F	75	174	56	0
S003	M	45	163	62	1
S004	M	57	175	70	0
S005	F	59	162	67	0

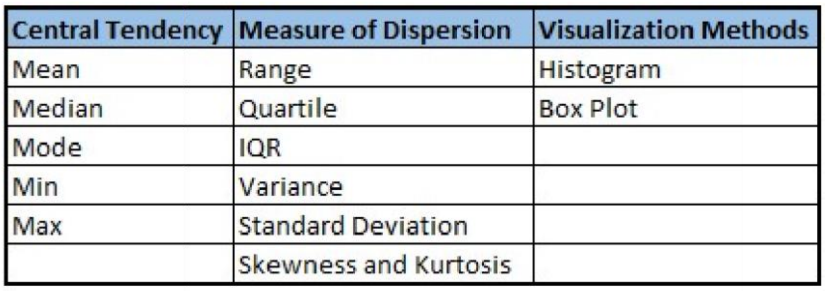
Below, the variables have been defined in different category:



2. Univariate Analysis

Depends on whether the variable type, i.e, Categorial and Continuous.

Continuous Variables: In case of continuous variables, we need to understand the central tendency and spread of the variable.

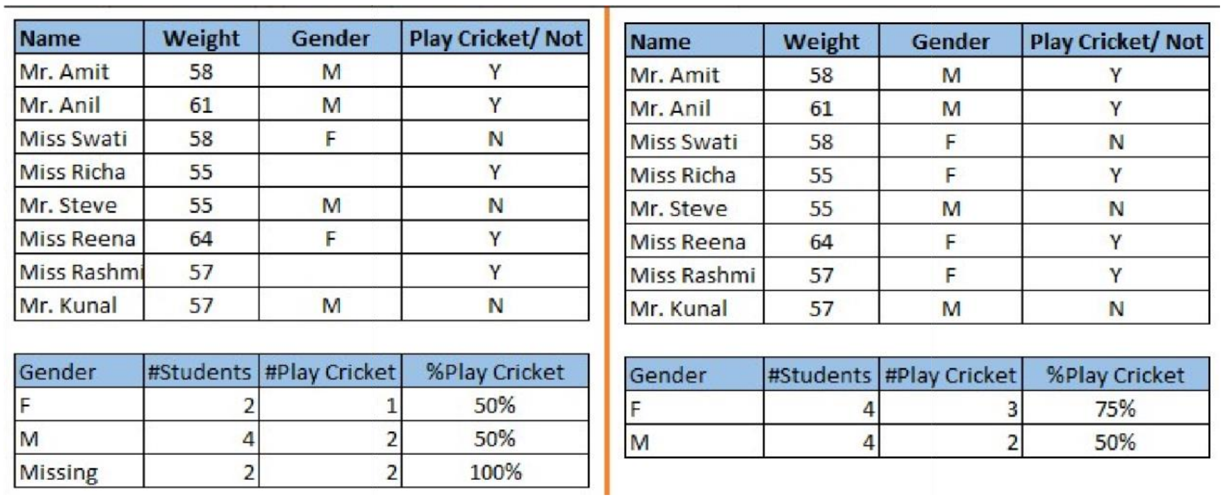


Central Tendency	Measure of Dispersion	Visualization Methods
Mean	Range	Histogram
Median	Quartile	Box Plot
Mode	IQR	
Min	Variance	
Max	Standard Deviation	
	Skewness and Kurtosis	

Categorical Variables: For categorical variables, we’ll use frequency table to understand distribution of each category. We can also read as percentage of values under each category. It can be measured using two metrics, Count and Count% against each category. Bar chart can be used as visualization.

3. Missing Value Treatment

Missing data in the training data set can reduce the power / fit of a model or can lead to a biased model because we have not analysed the behavior and relationship with other variables correctly. It can lead to wrong prediction or classification.



Name	Weight	Gender	Play Cricket/ Not
Mr. Amit	58	M	Y
Mr. Anil	61	M	Y
Miss Swati	58	F	N
Miss Richa	55		Y
Mr. Steve	55	M	N
Miss Reena	64	F	Y
Miss Rashmi	57		Y
Mr. Kunal	57	M	N

Gender	#Students	#Play Cricket	%Play Cricket
F	2	1	50%
M	4	2	50%
Missing	2	2	100%

Name	Weight	Gender	Play Cricket/ Not
Mr. Amit	58	M	Y
Mr. Anil	61	M	Y
Miss Swati	58	F	N
Miss Richa	55	F	Y
Mr. Steve	55	M	N
Miss Reena	64	F	Y
Miss Rashmi	57	F	Y
Mr. Kunal	57	M	N

Gender	#Students	#Play Cricket	%Play Cricket
F	4	3	75%
M	4	2	50%

Notice the missing values in the image shown above: In the left scenario, we have not treated missing values. The inference from this data set is that the chances of playing cricket by males is higher than females. On the other hand, if you look at the second table, which shows data after treatment of missing values (based on gender), we can see that females have higher chances of playing cricket compared to males.

Reasons for missing value data

1. Data Extraction: It is possible that there are problems with extraction process. In such cases, we should double-check for correct data with data guardians. Some hashing procedures can also be used to make sure data extraction is correct. Errors at data extraction stage are typically easy to find and can be corrected easily as well.

2. Data collection: These errors occur at time of data collection and are harder to correct. They can be categorized in four types:

   - Missing completely at random: This is a case when the probability of missing variable is same for all observations. For example: respondents of data collection process decide that they will declare their earning after tossing a fair coin. If an head occurs, respondent declares his / her
     earnings & vice versa. Here each observation has equal chance of missing value.

- Missing at random: This is a case when variable is missing at random and missing ratio varies for different values / level of other input variables. For example: We are collecting data for age and female has higher missing value compare to male.
- Missing that depends on unobserved predictors: This is a case when the missing values are not random and are related to the unobserved input variable. For example: In a medical study, if a particular diagnostic causes discomfort, then there is higher chance of drop out from the study. This missing value is not at random unless we have included "discomfort" as an input variable for all patients.
- Missing that depends on the missing value itself: This is a case when the probability of missing value is directly correlated with missing value itself. For example: People with higher or lower income are likely to provide non-response to their earning.

Solution:

- Deletion
- Mean/Median/Mode imputation
- Prediction model
- KNN imputation

Note: Solution not in syllabus.

4. Outliers Treatment

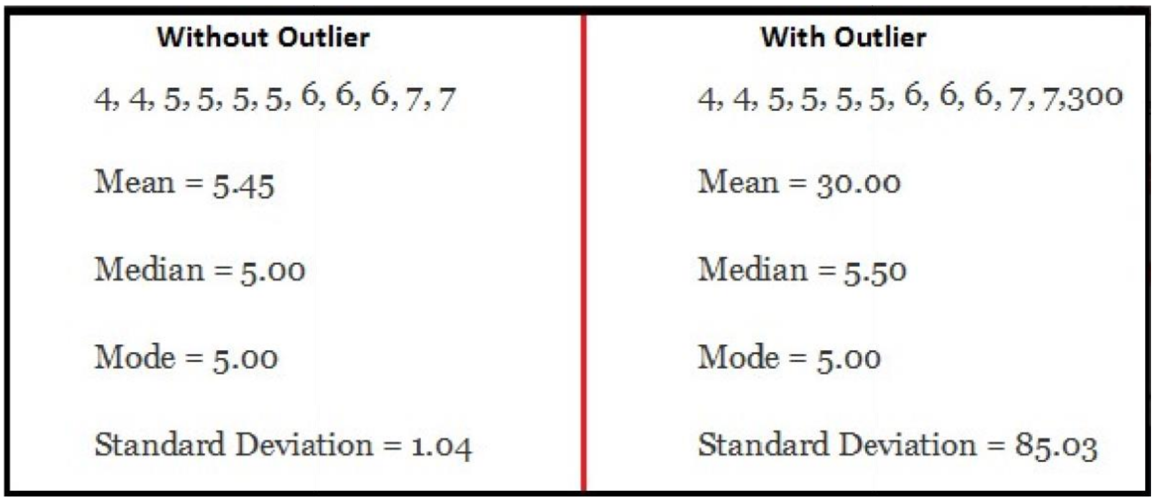
Outlier is an observation that appears far away and diverges from an overall pattern in a sample.

What is the impact of Outliers on a dataset?

Outliers can drastically change the results of the data analysis and statistical modelling.

- It increases the error variance and reduces the power of statistical tests
- If the outliers are non-randomly distributed, they can decrease normality
- They can bias or influence estimates that may be of substantive interest
- They can also impact the basic assumption of Regression, ANOVA and other statistical model assumptions.

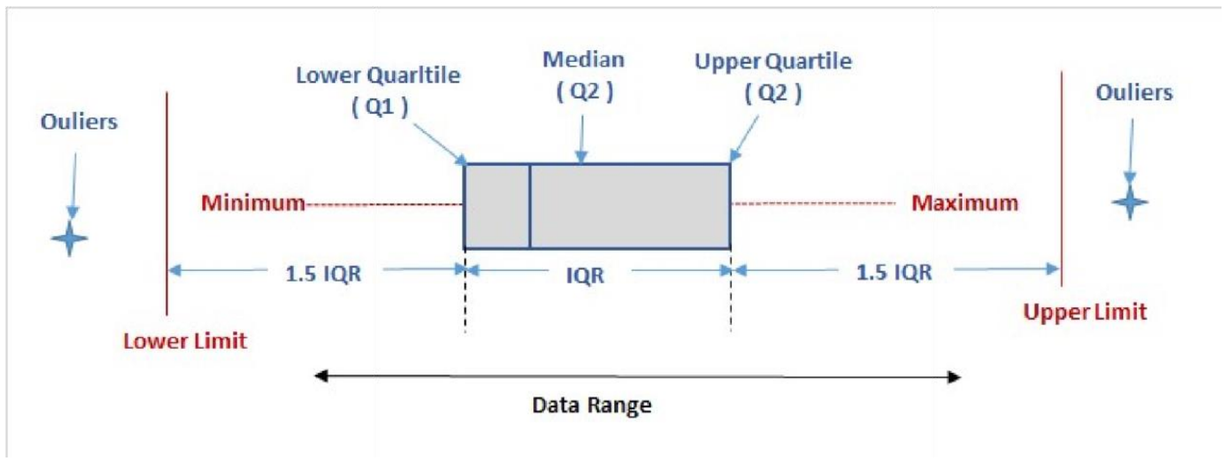
Example:



Without Outlier	With Outlier
4, 4, 5, 5, 5, 5, 6, 6, 6, 7, 7	4, 4, 5, 5, 5, 5, 6, 6, 6, 7, 7,300
Mean = 5.45	Mean = 30.00
Median = 5.00	Median = 5.50
Mode = 5.00	Mode = 5.00
Standard Deviation = 1.04	Standard Deviation = 85.03

Solution:

Most commonly used method to detect outliers is visualization. We use various visualization methods, like Box-plot, Histogram,Scatter Plot.



Q1 : 25th percentile Q2: 50th percentile

Q3: 75th percentile IQR : Q3 – Q1

Lower Limit: Q1 – (1.5*IQR) Upper Limit: Q3 + (1.5*IQR)

- Any data point in the interval of [Lower limit , Upper Limit] is valid point, else Outlier (Any value, which is beyond the range of -1.5 x IQR to 1.5 x IQR is outlier)

Outliers should not be excluded while doing data analysis.

Exercise:

- n=25 data points

126 132 138 140 141 141 142 143 144 144 144 145 146 147 148 148 149 149 150 150
150 154 155 158 158
How many outliers are there? Note them separately.

- Consider the following data set (representing scores in an examination)

67, 44, 60, 31, 15, 81, 77, 70, 84, 95, 91
1. Find mean
2. Find median
3. Find variance
4. Find standard deviation
5. Find range
6. Find first quartile
7. Find third quartile
8. Find IQR
9. Find 60th percentile

- True/False and justify the statement:

1. For any list of numbers, half of them will be below the mean?
2. Is the sample mean always the most frequently occurring value?
3. Is the sample mean always equal to one of the values in the sample?
4. Is the sample median always equal to one of the values in the sample?
5. Is it possible for standard deviation of a list of numbers to be equal to 0?
6. Is it possible for standard deviation to be greater than mean?

Data Cleaning using Python

Reference: https://www.tutorialspoint.com/python/python_data_cleansing

Missing data is always a problem in real life scenarios. Areas like machine learning and data mining face severe issues in the accuracy of their model predictions because of poor quality of data caused by missing values. In these areas, missing value treatment is a major point of focus to make their models more accurate and valid.

When and Why Is Data Missed?

Let us consider an online survey for a product. Many a times, people do not share all the information related to them. Few people share their experience, but not how long they are using the product; few people share how long they are using the product, their experience but not their contact information. Thus, in some or the other way a part of data is always missing, and this is very common in real time.

Let us now see how we can handle missing values (say NA or NaN) using Pandas.

```
# import the pandas library
import pandas as pd
import numpy as np
```

```
df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f',
'h'],columns=['one', 'two', 'three'])

df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])

print df
```

Its **output** is as follows −

```
        one        two      three
a   0.077988   0.476149   0.965836
b        NaN        NaN        NaN
c  -0.390208  -0.551605  -2.301950
d        NaN        NaN        NaN
e  -2.000303  -0.788201   1.510072
f  -0.930230  -0.670473   1.146615
g        NaN        NaN        NaN
h   0.085100   0.532791   0.887415
```

Using reindexing, we have created a DataFrame with missing values. In the output, **NaN**means **Not a Number.**

Check for Missing Values

To make detecting missing values easier (and across different array dtypes), Pandas provides the **isnull()** and **notnull()** functions, which are also methods on Series and DataFrame objects −

Example

```
import pandas as pd
import numpy as np

df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f',
'h'],columns=['one', 'two', 'three'])

df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])

print df['one'].isnull()
```

Its **output** is as follows −

```
a  False
b  True
c  False
d  True
```

```
e  False
f  False
g  True
h  False
Name: one, dtype: bool
```

Cleaning / Filling Missing Data

Pandas provides various methods for cleaning the missing values. The fillna function can “fill in” NA values with non-null data in a couple of ways, which we have illustrated in the following sections.

Replace NaN with a Scalar Value

The following program shows how you can replace "NaN" with "0".

```
import pandas as pd
import numpy as np
df = pd.DataFrame(np.random.randn(3, 3), index=['a', 'c', 'e'],columns=['one',
'two', 'three'])
df = df.reindex(['a', 'b', 'c'])
print df
print ("NaN replaced with '0':")
print df.fillna(0)
```

Its **output** is as follows −

```
        one        two     three
a  -0.576991  -0.741695  0.553172
b        NaN        NaN       NaN
c   0.744328  -1.735166  1.749580

NaN replaced with '0':
        one        two     three
a  -0.576991  -0.741695  0.553172
b   0.000000   0.000000  0.000000
c   0.744328  -1.735166  1.749580
```

Here, we are filling with value zero; instead we can also fill with any other value.

Fill NA Forward and Backward

Using the concepts of filling discussed in the ReIndexing Chapter we will fill the missing values.

Method	Action
pad/fill	Fill methods Forward
bfill/backfill	Fill methods Backward

Example

```
import pandas as pd
import numpy as np

df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f',
'h'],columns=['one', 'two', 'three'])
df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])

print df.fillna(method='pad')
```

Its **output** is as follows −

```
        one        two      three
a   0.077988   0.476149   0.965836
b   0.077988   0.476149   0.965836
c  -0.390208  -0.551605  -2.301950
d  -0.390208  -0.551605  -2.301950
e  -2.000303  -0.788201   1.510072
f  -0.930230  -0.670473   1.146615
g  -0.930230  -0.670473   1.146615
h   0.085100   0.532791   0.887415
```

Drop Missing Values

If you want to simply exclude the missing values, then use the **dropna** function along with the **axis** argument. By default, axis=0, i.e., along row, which means that if any value within a row is NA then the whole row is excluded.

Example

```
import pandas as pd
import numpy as np

df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f',
'h'],columns=['one', 'two', 'three'])

df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
print df.dropna()
```

Its **output** is as follows −

```
        one       two      three
a   0.077988   0.476149   0.965836
c  -0.390208  -0.551605  -2.301950
e  -2.000303  -0.788201   1.510072
f  -0.930230  -0.670473   1.146615
h   0.085100   0.532791   0.887415
```

Replace Missing (or) Generic Values

Many times, we have to replace a generic value with some specific value. We can achieve this by applying the replace method.

Replacing NA with a scalar value is equivalent behavior of the **fillna()** function.

Example

```
import pandas as pd
import numpy as np
df = pd.DataFrame({'one':[10,20,30,40,50,2000],
'two':[1000,0,30,40,50,60]})
print df.replace({1000:10,2000:60})
```

Its **output** is as follows −

```
   one  two
0   10   10
1   20    0
2   30   30
3   40   40
4   50   50
5   60   60
```