**Department of Computer Engineering**

**Academic Year:** 2022-2023 **Semester:** VIII

**Subject:** Applied Data Science **Class / Division:** BE/CMPN/B

**Name:** Domingos J Pinto **Roll Number:** 007

**Date: Seat Number:** BE/CMPN/B

**Experiment No.: 3**

**To implement data cleaning techniques (e.g. Data Imputation).**

**Aim :** To implement data cleaning techniques (e.g. Data Imputation).

**I-THEORY**

## What is Data Cleaning?

Data cleaning is the process of removing incorrect, duplicate, or otherwise erroneous data from a dataset. These errors can include incorrectly formatted data, redundant entries, mislabeled data, and other issues; they often arise when two or more datasets are combined together. Data cleaning improves the quality of your data as well as any business decisions that you draw based on the data.

## What is Imputation?

Imputation is a technique used for replacing the missing data with some substitute value to retain most of the data/information of the dataset. These techniques are used because removing the data from the dataset every time is not feasible and can lead to a reduction in the size of the dataset to a large extend, which not only raises concerns for biasing the dataset but also leads to incorrect analysis.

## Identify the Missing Data Values

Most analytics projects will encounter three possible types of missing data values, depending on whether there’s a relationship between the missing data and the other data in the dataset:

* **Missing completely at random (MCAR):** In this case, there may be no pattern as to why a column’s data is missing. For example, survey data is missing because someone could not make it to an appointment, or an administrator misplaces the test results he is supposed to enter into the computer. The reason for the missing values is unrelated to the data in the dataset.
* **Missing at random (MAR):** In this scenario, the reason the data is missing in a column can be explained by the data in other columns. For example, a school student who scores above the cutoff is typically given a grade. So, a missing grade for a student can be explained by the column that has scores below the cutoff. The reason for these missing values can be described by data in another column.
* **Missing not at random (MNAR):** Sometimes, the missing value is related to the value itself. For example, higher income people may not disclose their incomes. Here, there is a correlation between the missing values and the actual income. The missing values are not dependent on other variables in the dataset.

## How to Handle Missing Data Values

Data teams can use a number of strategies to handle missing data. On one hand, algorithms such as random forest and KNN are robust in dealing with missing values.

On the other hand, you may have to deal with missing data on your own. The first common strategy for dealing with missing data is to **delete the rows with missing values.** Typically, any row which has a missing value in any cell gets deleted. However, this often means many rows will get removed, leading to loss of information and data. Therefore, this method is typically not used when there are few data samples.

You can also **impute the missing data.** This can be based solely on information in the column that has missing values, or it can be based on other columns present in the dataset.

Finally, you can use **classification or regression models** to predict missing values.

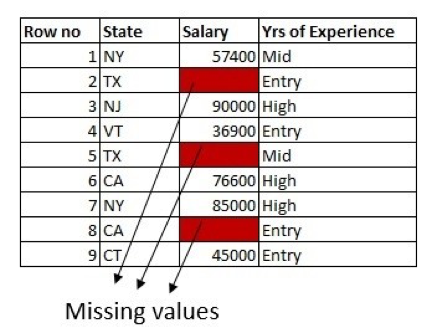
Let’s look at these three strategies in depth:

## *1. Missing Values in Numerical Columns*

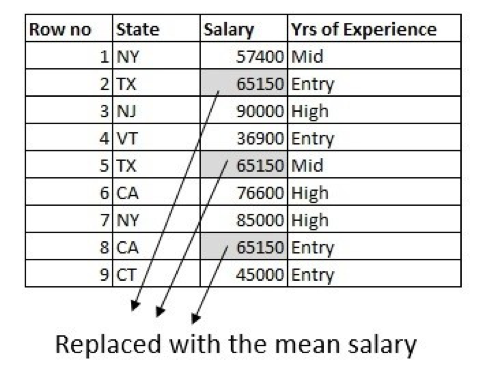
The first approach is to replace the missing value with one of the following strategies:

* Replace it with a constant value. This can be a good approach when used in discussion with the domain expert for the data we are dealing with.
* Replace it with the mean or median. This is a decent approach when the data size is small—but it does add bias.
* Replace it with values by using information from other columns.

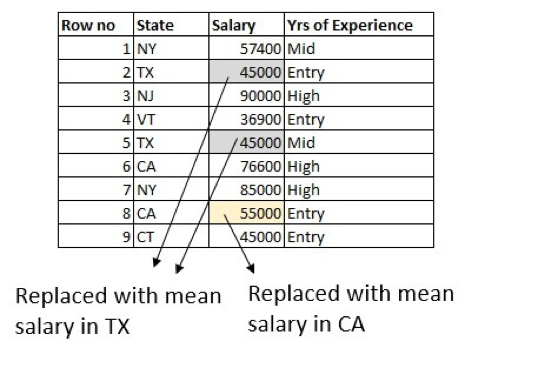
In the employee dataset subset below, we have salary data missing in three rows. We also have State and Years of Experience columns in the dataset:



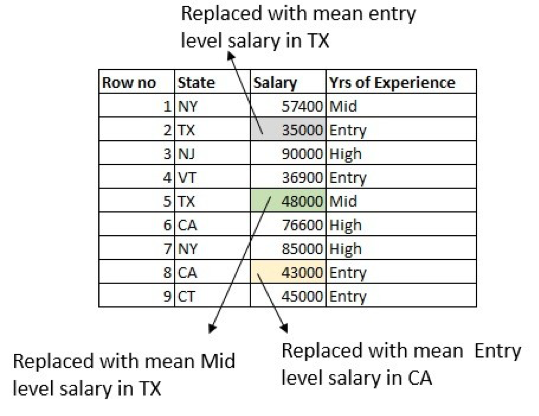
The first approach is to fill the missing values with the mean of the column. Here, we are solely using the information from the column which has missing values:



With the help of a domain expert, we can do little better by using information from other columns in the dataset. The average salary is different for different states, so we can use that to fill in the values. For example, calculate the average salary of people working in Texas and replace the missing data with an average salary of people who typically work in Texas:



What else can we do better? How about making use of the Years of Experience column as well? Calculate the average entry-level salary of people working in Texas and replace the row where the salary is missing for an entry-level person in Texas. Do the same for the mid-level and high-level salaries:



Note that there are some boundary conditions. For example, there might be a row that has missing values in **both** the Salary and Years of Experience columns. There are multiple ways to handle this, but the most straightforward is to replace the missing value with the average salary in Texas.

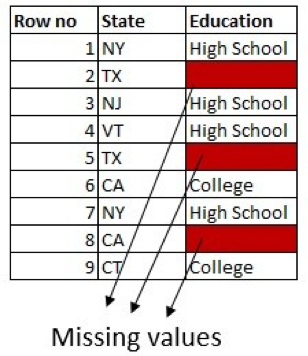
## *2. Predicting Missing Values Using an Algorithm*

Another way to predict missing values is to create a simple regression model. The column to predict here is the Salary, using other columns in the dataset. If there are missing values in the input columns, we must handle those conditions when creating the predictive model. A simple way to manage this is to choose only the features that do not have missing values, or take the rows that do not have missing values in any of the cells.

## *3. Missing Values in Categorical Columns*

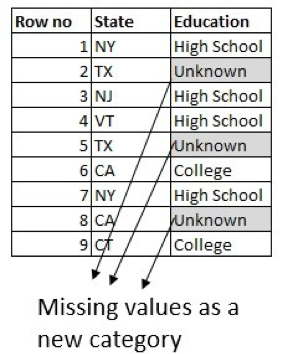
Dealing with missing data values in categorical columns is a lot easier than in numerical columns. Simply replace the missing value with a constant value or the most popular category. This is a good approach when the data size small, though it does add bias.

For example, say we have a column for Education with two possible values: High School and College. If there are more people with a college degree in the dataset, we can replace the missing value with College Degree:



We can tweak this more by making use of information in the other columns. For example, if there are more people from Texas with High School in the dataset, replace the missing values in rows for people from Texas with High School.

One can also create a classification model. The column to predict here is Education, using other columns in the dataset. But the most common and popular approach is to model the missing value in a categorical column as a new category called Unknown:



In summary, you’ll use different approaches to handle missing data values while data cleaning depending on the type of data and the problem at hand. If you have access to a domain expert, always incorporate their expert advice when filling in the missing values.

Most importantly, no matter the imputation method you choose, always run the predictive analytics model to see which one works best from the standpoint of data accuracy.

## Data Cleaning Tools

1. Microsoft Excel (Popular data cleaning tool)
2. Programming languages (Python, Ruby, SQL)
3. Data Visualizations (To spot errors in your dataset)
4. Proprietary software (OpenRefine, Trifacta, etc.,)
5. Openrefine
6. Trifacta Wrangler
7. TIBCO Clarity
8. Cloudingo
9. IBM Infosphere Quality Stage

**II Implementation-**

**Dataset :** Fashion Dataset

**Description :** It is a dataset of women fashion products. This dataset is a collection of 30000 women fashion products. It has 7 columns that contains Number, Brand Name, Details, Sizes, MRP, Sell Price and Category.

#Load libraries

import pandas as pd

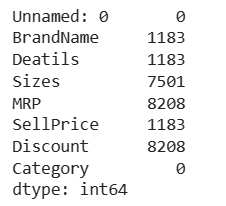
import numpy as np

fds0 = pd.read\_csv("FashionDataset.csv")

fds0.head()



fds0.isnull().sum()



1. **Imputing Missing Values**

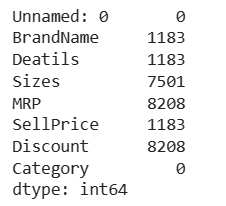
**#Mean Method**

fds1 = pd.read\_csv("FashionDataset.csv")

fds1.head()



fds1.isnull().sum()



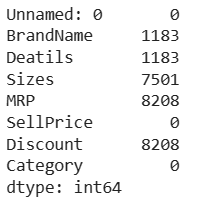
missing\_col = ['SellPrice']

#Using mean to impute the missing values

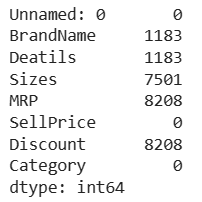
for i in missing\_col:

fds1.loc[fds1.loc[:,i].isnull(),i]=fds1.loc[:,i].mean()

fds1.isnull().sum()



fds1.isnull().sum()



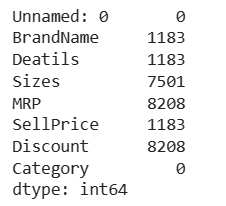
**#Median Method**

fds2 = pd.read\_csv("FashionDataset.csv")

fds2.head()



fds2.isnull().sum()



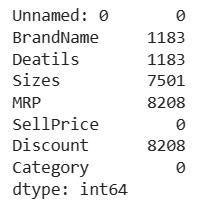
missing\_col\_1 = ['SellPrice']

#Using mean to impute the missing values

for i in missing\_col\_1:

fds2.loc[fds2.loc[:,i].isnull(),i]=fds2.loc[:,i].median()

fds2.isnull().sum()



**#Filling with a new category**

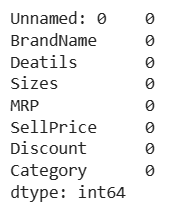
fds2['BrandName'] = fds2['BrandName'].fillna('No Brand')

fds2['Deatils'] = fds2['Deatils'].fillna('No Details')

fds2['Sizes'] = fds2['Sizes'].fillna('No Size')

fds2['MRP'] = fds2['MRP'].fillna('No MRP')

fds2['Discount'] = fds2['Discount'].fillna('No Discount')



1. **Removing Rows**

#Duplicate Values

len(fds1)



fds1= fds1.drop\_duplicates()

len(fds1)



#Missing Values

len(fds1)



fds1 = fds1.dropna()

len(fds1)



#Unnecessary Column

fds1.head()



fds1 = fds1.drop(columns=["Unnamed: 0"])

fds1.head()

