Data Cleaning

# **Introduction**

Welcome to the next step in the process of EDA called ‘Data Cleaning’.

In the last session, you learnt about the data sourcing techniques. Once you source the data, it is essential to get rid of the irregularities in the data and fix it to improve its quality.

One can encounter different kinds of issues in a dataset. Irregularities may appear in the form of **missing values, anomalies/outliers, incorrect format** and **inconsistent** **spelling**, etc., These irregularities may propagate further and affect the assumptions and analysis based on that dataset and may hamper the further process of machine learning model building. Hence, data cleaning is a very important step in EDA.

Play Video

1569002

## In this session

In this session, you will learn the process of data cleaning using a case study on **‘Bank Marketing Campaign Dataset’**. Though data cleaning is often done in a somewhat haphazard manner, and it is difficult to define a ‘single structured process’, you will study data cleaning through the following steps:

* Identifying the data types
* Fixing the rows and columns
* Imputing/removing missing values
* Handling outliers
* Standardising the values
* Fixing invalid values
* Filtering the data

Before going any further, it is important for you to get familiar with the problem statement that you are going to solve in this module to understand the EDA practically.

**Problem statement**

The bank provides financial services/products such as savings accounts, current accounts, debit cards, etc. to its customers. In order to increase its overall revenue, the bank conducts various marketing campaigns for its financial products such as credit cards, term deposits, loans, etc. These campaigns are intended for the **bank’s existing customers.** However, the marketing campaigns need to be cost-efficient so that the bank not only increases their overall revenues but also the total profit. You need to apply your knowledge of EDA on the given dataset to analyse the patterns and provide inferences/solutions for the future marketing campaigns.

Download the CSV file of the bank marketing dataset from the following attachment.

Bank Telemarketing Campaign Dataset

Download

Play Video

1569002

A bank conducted a telemarketing campaign for one of its financial products called ‘Term Deposits’ to help foster long-term relationships with existing customers. The dataset contains information about all the customers who were contacted during a particular year to open term deposit accounts with the bank.

**What is a term deposit?**

Term deposits, also called fixed deposits, are the cash investments made for a specific time period ranging from 1 month to 5 years for predetermined fixed interest rates. The fixed interest rates offered for term deposits are higher than the regular interest rates for savings accounts. The customers receive the total amount (investment plus the interest) at the end of the maturity period. Also, the money can only be withdrawn at the end of the maturity period. Withdrawing money before that will result in penalty charges, and the customer will not receive any interest returns.

**Important Note:**

To enhance the learning outcome, you are expected code along with the instructor as you watch the videos. So, please pace yourself accordingly. To assist you, you are provided with a structured and blank Python notebook to code. This is a **must**-**do** task for you to answer certain in-segment questions, as it serves the purpose of practice. Also, the final notebook will act as a reference for you in the future as well.

**Please do not expect a complete solution notebook attached at the end of this module**

Jupyter Notebook\_Bank\_Telemarketing\_Campaign

Download

## Guidelines for In-Module Questions

The in-video and in-content questions for this module are not graded. Note that graded questions are given on a separate page labelled 'Graded Questions' at the end of each session. The graded questions in these sessions will adhere to the following guidelines:

|  | First Attempt Marks | Second Attempt Marks |
| --- | --- | --- |
| Question with 2 Attempts | 10 | 5 |
| Question with 1 Attempt | 10 | 0 |

## People you will hear from in this module

**Subject Matter Expert**

[Mirza Rahim Baig](https://www.linkedin.com/in/rahim-baig)

Analytics Lead, Flipkart

Flipkart is one of the leading e-commerce companies in India. It started with selling books and has now expanded its business to almost every product category, including consumer electronics, fashion and lifestyle products. Rahim is currently the Analytics Lead at Flipkart. He holds a graduate degree from BITS Pilani, a premier educational institute in India.

**Subject Matter Expert**

[S Anand](https://www.linkedin.com/in/sanand0)

CEO, Gramener

Gramener is one of the most prominent data analytics and visualisation companies in India. Anand, currently the CEO, was previously the Chief Data Scientist at Gramener and also has extensive experience in management consulting and equity research.

# **Data Types**

Now, let’s talk about a very important aspect of any dataset, i.e., data types. In a particular dataset, you have multiple types of variables with different kinds of data types such as integers, string, floats, etc.

For data analysis, you will use the following libraries through the entire module, which you must have already covered in prep content:

* **Pandas**: It is a library to deal with dataframes in Python. Pandas is an acronym derived from panel data. It is solely used for data analysis purposes in Python.
* **NumPy**: This library is used for performing numerical operations on a dataset.

Now, let’s go through the bank marketing dataset along with Rahim and try to find out the data types that are present in it.

Play Video

1569002

In general, any given data set is expected to have different types of data. Following are some examples with their data types.

| Example | Variable Type | Data Type |
| --- | --- | --- |
| Height, weight, age, temperature | Numerical variable | Int, float |
| Size of clothes, months, type of jobs, blood group. | Categorical variable | Object |
| Grades in exam, education level, months, integer ratings | Ordinal categorical type | Object, int, float |
| Date, time, timestamp | Date and time variable | Date and time |

Question 1/2

Mandatory

#### **Data Type**

Let's say there is a column in a dataset which essentially containing the numerical values in it. But its data type is an object, which needs to be rectified and should be converted into int or float type.

Choose the correct command to convert such object type column in a dataframe into integer or float type.

df["feature"]= df["feature"].astype(int)

✓ Correct

Feedback:

*This particular command is used to convert any type of variable into int.*

df.["feature"]= df.["feature"].astype(int)

df["feature"]= df["feature"].astype(float)

✓ Correct

Feedback:

*This particular command is used to convert any type of variable into float.*

df.["feature"]= df.["feature"].astype(float)

Your answer is Correct.

Question 2/2

Mandatory

#### **Data Type**

Find the average age of the customers in the bank marketing data set.

51

40

✕ Incorrect

Feedback:

*Convert the age variable into numeric type from object type, which can be an integer, and then find the mean.*

50

41

✓ Correct

Feedback:

*Convert the age variable into numeric type from object type, which can be an integer, and then find the mean.*

inp0["age"]= inp0["age"].astype(int)

inp0.age.mean()

Your answer is Wrong.

In the next segment, you will get an understanding of the steps in the data cleaning process, particularly fixing the rows and columns.

# **Fixing the Rows and Columns**

You learnt about some of the issues with raw data and understood the need for data cleaning. Now, let's listen to Anand to understand different cases in fixing the columns and rows of a given dataset.

Play Video

1569002

Now let’s summarise what you learnt with the help of the checklists below. Make sure you correctly identify these issues and resolve them before moving on to the next stage of data cleaning.

**Checklist for fixing rows:**

* Delete summary rows: Total and Subtotal rows
* Delete incorrect rows: Header row and footer row
* Delete extra rows: Column number, indicators, blank rows, page number

**Checklist for fixing columns:**

* if needed, merge columns for creating unique identifiers, for example, merge the columns State and City into the column Full Address.
* Split columns to get more data: Split the Address column to get State and City columns to analyse each separately.
* Add column names: Add column names if missing.
* Rename columns consistently: Abbreviations, encoded columns.
* Delete columns: Delete unnecessary columns.
* Align misaligned columns: The data set may have shifted columns, which you need to align correctly.

Data Cleaning Checklist

Download

Now, let’s listen to Rahim to learn how to fix the columns in the bank marketing dataset.

Play Video

1569002

You have seen in the above video that both heading rows have been deleted, as they have no use in our analysis. It is very important to note here that if you find anything irregular at the very glance of the data set then it is very essential to get rid of that at the very first process.

In the next video, Rahim will explain how to fix the columns in our bank marketing dataset.

Play Video

1569002

Now you have learnt to fix the following columns:

* **Customerid**: It has been dropped, as it has no specific use in the analysis.
* **Jobedu**: It has been separated to extract job and education. Job and education have to be analysed separately. You will understand in further sessions how education and job play a very important role in determining the customer segment who will respond positively to term deposits.
* **Month**: The month name will be extracted in the further segments based on the missing values imputation analysis.

In the next segment, you will learn how to treat the issue of missing values.

Question 1/1

Mandatory

#### **Data Cleaning**

What points should be kept in mind while handling the rows and columns for preparing the data for analysis and to fetch maximum information from it?

Simply remove the columns if you find more than 20% of data in that column as missing. It may reduce the bulkiness of the data set and make it simple to analyse.

Splitting a column into 2 or 3 columns increases the number of variables in the data set. However, one should remember that splitting is not preferred unless you get any unique information.

✓ Correct

Feedback:

*Columns should be split to get more variables if that yields any unique and useful information. Like in our data set, jobedu can be split into job and education.*

In the bank marketing dataset, you can bucket the data according to age group, for example, <30, 30-40, 40-50, 50-60, >70, It may yield better insights about the responses based on age groups.

✓ Correct

Feedback:

*Bucketing by age group may yield new insights about the perspective of different age groups.*

It is important to have information about the variables or instances, for example, whether they have null values, blank values or invalid values.

✓ Correct

Feedback:

*When you observe the dataset, it is necessary to look into the count of null values. You will learn more about it in further segments.*

Your answer is Correct.

# **Impute/Remove Missing Values**

You learnt how to fix columns and rows, and applied those learnings to the bank marketing dataset. Now, you will learn what missing values are and how they should be treated. Before working on the dataset, let’s listen to Anand as he explains the different methods to fix missing values in a dataset.

Play Video

1569002

The most important takeaway from this lecture is: good methods add information, bad methods exaggerate information. In case you can add information from reliable external sources, you should use it to replace missing values. But often, it is better to let missing values be and continue with the analysis rather than extrapolate the available information.

Let’s summarise the takeaways from the above video:

* **Set values as missing values:** Identify values that indicate missing data, for example, treat blank strings "NA", "XX", "999", etc., as missing.
* **Adding is good, exaggerating is bad:** You should try to get information from reliable external sources as much as possible, but if you can’t, then it is better to retain missing values rather than exaggerating the existing rows/columns.
* **Delete rows and columns:** Rows can be deleted if the number of missing values is insignificant, as this would not impact the overall analysis results. Columns can be removed if the missing values are significant in number.
* **Fill partial missing values using business judgement:** Such values include missing time zones, century, etc. These values can be identified easily.

In the next video, Rahim will explain the different types of missing values and how to delete or impute them.

Play Video

1569002

Following is a list of the major takeaways from the video.

Types of missing values:

* **MCAR**: It stands for Missing completely at random. The reason behind the missing value is not dependent on any other features.
* **MAR**: It stands for Missing at random. The reason behind the missing value may be associated with some other features.
* **MNAR**: It stands for Missing not at random. There is a specific reason behind the missing value.

Now, let’s apply all these concepts to the bank marketing campaign data set to tackle the issue of missing values in the age and month columns.

Play Video

1569002

There are various ways to deal with missing values. Either you can drop the entries that are missing if you find that the percentage of missing values in a column is very small, or you can impute the missing values with some other values. Let’s look into the various ways to impute the missing values.

**Imputation on categorical/numeric columns:**

* **Categorical column:**
* Impute the most popular category.
* Imputation can be done using logistic regression techniques.
* **Numerical column:**
* Impute the missing value with mean/median/mode.
* The other methods to impute the missing values involve the use of interpolation, linear regression. These methods are useful for continuous numerical variables.

In this video, you will go through the analysis of the 'pdays' variable to deal with its missing values.

Play Video

1569002

The major takeaway from the above video is that missing values does not always have to be null. So, now you must have a clear understanding of how to treat missing values in a dataset.

* Sometimes, it is good to just drop the missing values because they are missing completely at random.
* Sometimes, it is good to impute them with another value, maybe mean/median/mode, because they are not missing at random and have to be incorporated for further analysis.

You have gone through with the bank telemarketing data set. There is a 'response' variable which is basically the target variable of the data set. You have learnt about the missing values and the process to treat them. Based on your understanding of codes and process on missing values, answer the following questions.

In the next segment, you will learn how to deal with outliers.

Question 1/3

Mandatory

#### **Missing Values**

Implement the code in your blank Jupyter notebook and find the exact number of missing values in the response column. Choose the correct option from the below.

30

✓ Correct

Feedback:

*The code in Python to find the missing values of the response column is as follows:*

inp1.response.isnull().sum()

40

50

60

Your answer is Correct.

Question 2/3

Mandatory

#### **Missing Values**

Implement the code in your blank Jupyter notebook and find the percentage of missing values in the response column. Choose the correct option from the below.

1%

0.1%

0.06%

✓ Correct

Feedback:

*Correct! You need to write the code in Python to find the missing values in the response column.*

0.006%

Your answer is Correct.

Question 3/3

Mandatory

#### **Missing Values**

What do you think about the treatment of missing values in the response column?

They should not be dropped, as that may affect the data set.

✕ Incorrect

Feedback:

*You can drop the missing values, as the percentage of such values is very small.*

They can be removed because they are very small in number as compared to entries in the dataset.

✓ Correct

Feedback:

*You can drop it as it is very less in percentage.*

Create a separate category named 'missing' to deal with such values.

Your answer is Wrong.

# **Handling Outliers**

You have learnt what missing values are and how to treat them. Now, let’s move to the next concept of data cleaning, which is outliers.

The definition of outliers is as follows:

***Outliers are values that are much beyond or far from the next nearest data points.***

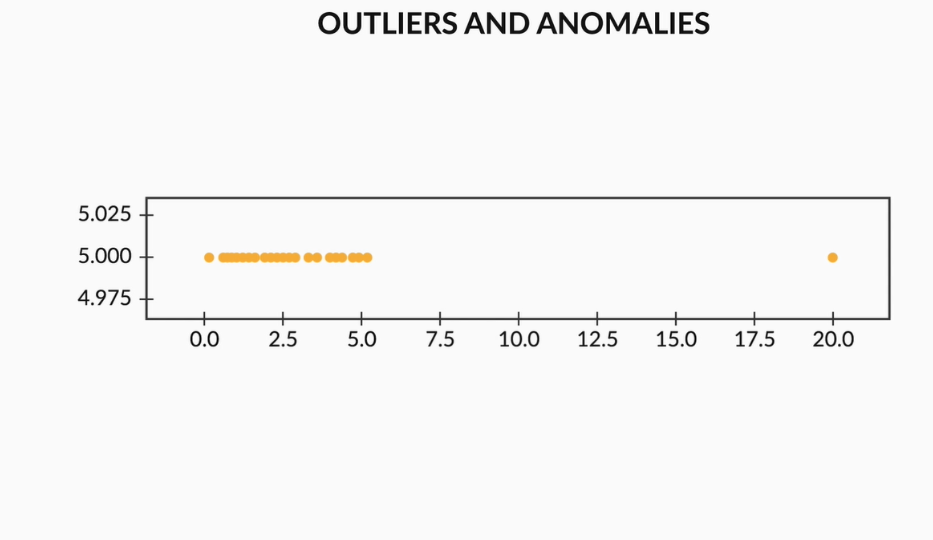
In this video, Rahim will help you understand the concept of outliers.

Play Video

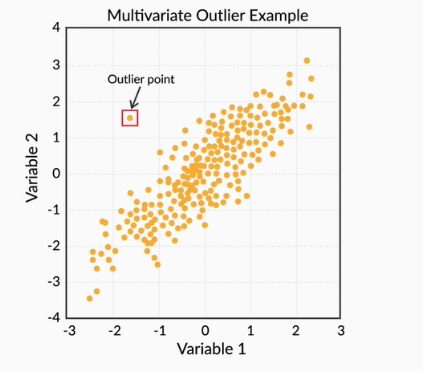
1569002

You learnt that there are two types of outliers. These are:

* **Univariate outliers:** Univariate outliers are those data points in a variable whose values lie beyond the range of expected values. You can get a better understanding of univariate outliers from the image below. Here, almost all the points lie between 0 and 5.0, and one point is extremely far away (at 20.0) from the normal norms of this data set.



* **Multivariate outliers:** While plotting data, some values of one variable may not lie beyond the expected range, but when you plot the data with some other variable, these values may lie far from the expected value. These are called multivariate outliers. You can refer to the image below to get a better understanding of multivariate outliers.



Now, let’s proceed to the next video, where you will learn about the reasons behind the appearance of outliers in data and how to treat them.

Play Video

1569002

From the video, you must have understood that outliers should be treated before investigating data and drawing insights from a dataset.

Now, the major approaches to the treatment of outliers can include:

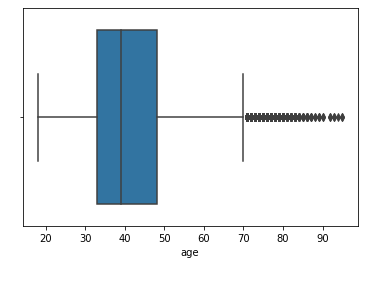
* Imputation
* Deletion of outliers
* Binning of values
* Capping the outliers

In the process of handling missing values and outliers of different columns, you are already performing univariate analysis. You will learn more about it in further sessions. In this video, you will learn how to implement all your learning on the bank marketing dataset.

Play Video

1569002

So, in the above video, you have seen that the age variable has outliers, but these can be treated as the normal values of age because any person can be over 70 or 80 years of age. Also, the 70-90 age group is sparsely populated and participate in opening the term deposit account, which is why these set of people fall out of the box plot but they are not outliers and can be considered as normal values.



Let’s listen to Rahim as he explains the variable 'balance'.

Play Video

1569002

An important aspect that has been covered in this video is **quantiles**. Sometimes, it is beneficial if you look into the quantiles instead of the box plot, mean or median. Quantile may give you a fair idea about the outliers. If there is a huge difference between the maximum value and the 95th or 99th quantiles, then there are outliers in the data set.

In the next segment, you will learn about the standardisation process in EDA.

Question 1/4

Mandatory

#### **Outliers**

Consider the following two statements:

* The difference between the maximum value of the balance variable and the 99th percentile is too high.
* The difference between the 99th percentile value and the 95th percentile value of the balance variable is in the normal range, meaning it is not too high.

Based on the above two statements, choose the correct option which concludes that the balance variable has outliers in it.

Statement 1 is alone sufficient to conclude that balance variables have outliers.

Both the statements are insufficient to conclude that the balance variable has outliers.

Both the statement are together sufficient to conclude that it has outliers.

✓ Correct

Feedback:

*Both the statements are simultaneously required for you to infer from statement 1 that if the maximum value of the variable is far from the 99th percentile, it gives a clear idea that there are outliers in the data set. In addition, from the 2nd statement, you can see that there is no huge difference between the quantile of 95th and 99th.*

Any of the statement is alone sufficient to conclude that the balance variable has outliers.

Your answer is Correct.

Question 2/4

Mandatory

#### **Outliers**

Which of the following methods can be used to identify the outliers (univariate/multivariate) in the dataset?

Box plot can be used to plot the single variable and find its interquartile range and quantiles.

✓ Correct

Feedback:

*Box plot gives a clear picture of all the points and visualises the quantiles to infer knowledge about the outliers.*

The difference of each point from the mean/median value in the dataset is alone sufficient to identify whether a point is an outlier or not.

Scatter plot can not be used to identify the multivariate outliers.

Your answer is Correct.

Question 3/4

Mandatory

#### **Outliers**

What is the mean and 75th percentile of the salary variable in bank marketing data set, respectively?

57004, 60000

57004, 70000

✓ Correct

Feedback:

*Just write the following code to describe the salary variable. You will find the mean and 75th*

*percentile.*

inp1.salary.describe()

70000, 57004

60000, 57004

Your answer is Correct.

Question 4/4

Mandatory

#### **Outliers**

After plotting the boxplot, can you write your opinion about the outliers in the salary variable? Are there any outliers in salary?

Word Count 9Word Limit 5 - 500

Suggested Answer

Once you plot the boxplot, you see that the salary variable does not have any outliers. All values lies within the range of 100th percentile always.

# **Standardising Values**

You learnt different techniques to handle outliers and also implemented the same in Bank marketing dataset. Now, you will learn the next important aspect, which is to standardise values in a dataset.

In this video, Anand will explain how to standardise quantitative values in a dataset.

Play Video

1569002

Scaling ensures that the values in a dataset have a common scale; this makes it easy to perform data analysis. Let's take a data set that contains the grades of students studying in different universities. Some of the universities assign grades on a scale of 4, whereas the others assign grades on a scale of 10. Hence, you cannot assume that a GPA of 3 on a scale of 4 is equal to a GPA of 3 on a scale of 10, even though they are the same quantitatively. Thus, for the purpose of analysis, these values need to be brought to a common scale, such as the percentage scale.

Now, let’s summarise what you learnt so far about standardising the variables in a dataset. Given below is a list of the points that we covered. You could use this as a checklist for future data cleaning exercises:

* **Standardise units:** Ensure all observations under one variable are expressed in a common and consistent unit, e.g., convert lbs to kg, miles/hr to km/hr, etc.
* **Scale values if required:** Make sure all the observations under one variable have a common scale.
* **Standardise precision** for a better presentation of data, e.g., change 4.5312341 kg to 4.53 kg.

Now that you have learnt how to standardise the numeric values in a data set, let's proceed to learn how to standardise text values, which is an equally important aspect of data analysis.

Play Video

1569002

Now, let’s summarise what you learnt about standardising text values in a dataset. Given below is a list of the points that were covered, you can use this as a checklist for future data cleaning exercises:

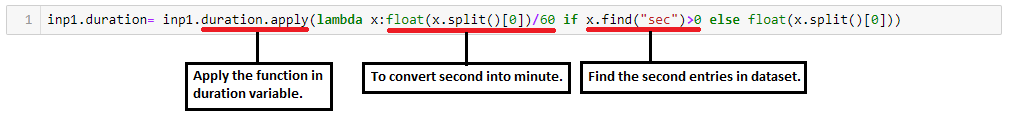
* Remove extra characters such as common prefixes/suffixes, leading/trailing/multiple spaces, etc. These are irrelevant to the analysis.
* **Standardise case:** String variables may take various casing styles, e.g., FULLCAPS, lowercase, Title Case, Sentence case, etc.
* **Standardise format:** It is important to standardise the format of other elements such as date, name, etc. For example, change 23/10/16 to 2016/10/23, “Modi, Narendra” to “Narendra Modi", etc.

In this video, Rahim will apply the concepts covered in this segment to the bank marketing data set and standardise some required values in it.

Play Video

1569002

In the videos, you saw the application of standardisation with a real-life example of the ‘duration’ variable in bank marketing data set. The duration variable has data in minutes as well as in seconds, which has to be converted into minute only. You can understand the entire code to convert the 'duration' variable into minutes in the image below.



In the next segment, you will learn how to fix invalid values in a data set.

# **Fixing Invalid Values and Filter Data**

In the previous segments, you learnt the concepts to deal with different kinds of irregularities in a data set. You also went through bank marketing dataset, where you saw the practical aspect of all the concepts covered. Datasets also have some other irregularities, which you need to get rid of. Though our bank marketing dataset does not have these kinds of irregularities, it is essential to deal with these as well.

Let’s watch the next video to gain more insight into fixing invalid values in a data set.

Play Video

1569002

If your data set has invalid values, and if you do not know which accurate values could replace the invalid values, then it is recommended that you treat these values as missing. For example, if the Contacts columns in a data set contain a string ‘tr8ml’, then it is recommended that you remove the invalid value and treat it as a missing value.

Now, let’s summarise what you learnt about fixing invalid values in a data set. Given below is a list of points that we covered. You could use this as a checklist for future data cleaning exercises:

* **Encode unicode properly**: In case the data is being read as junk characters, try to change the encoding, for example, use CP1252 instead of UTF-8.
* **Convert incorrect data types**: Change the incorrect data types to the correct data types for ease of analysis. For example, if numeric values are stored as strings, then it would not be possible to calculate metrics such as mean, median, etc. Some of the common data type corrections include changing a string to a number ("12,300" to “12300”), a string to a date ("2013-Aug" to “2013/08”), a number to a string (“PIN Code 110001” to "110001"), etc.
* **Correct the values that lie beyond the range**: If some values lie beyond the logical range, for example, temperature less than -273° C (0° K), then you would need to correct those values as required. A close look at the data set would help you determine whether there is scope for correction or the value needs to be removed.
* **Correct the values not belonging in the list**: Remove the values that do not belong to a list. For example, in a data set of blood groups of individuals, strings ‘E’ or ‘F’ are invalid values and can be removed.
* **Fix incorrect structure**: Values that don’t follow a defined structure can be removed from a data set. For example, in a data set containing the pin codes of Indian cities, a pin code of 12 digits would be an invalid value and would need to be removed. Similarly, a phone number of 12 digits would be an invalid value.
* **Validate internal rules**: Internal rules, if present, should be correct and consistent. For example, the date of a product’s delivery should definitely come after the date of purchase.

After you have fixed the missing values, standardised the existing values and corrected the invalid values in a data set, you would arrive at the last stage of data cleaning. Although you have a largely accurate dataset by now, you may not need all of it for your analysis. It is important for you to understand what you need in order to draw insights from the data, and then choose relevant parts of the dataset for your analysis. Thus, you need to filter the data in order to get what you need for your analysis.

Let’s watch Anand as he takes us through the various steps of data filtering in the next video.

Play Video

1569002

Now, let’s summarise what you learnt about filtering data. Given below is a list of the points we covered. You could use this as a checklist for future data cleaning exercises:

* **Deduplicate data:** Remove identical rows and the rows in which some columns are identical.
* **Filter rows**: Filter rows by segment and date period to obtain only rows relevant to the analysis.
* **Filter columns**: Filter columns relevant to the analysis.
* **Aggregate data**: Group by the required keys and aggregate the rest.

# **Graded Questions**

Suppose you are working as an analyst in an e-commerce company, and you have been given the two datasets containing the details of women's apparels sold during the last year.

* **Attribute Dataset:** This dataset contains the different features of women's apparels.
* **Dress Sales:** This data set contains the number of sales for a particular dress ID on a particular date.

Let’s look into the first data set.

You can download the attribute data set from the link provided below:

Dress Attribute Dataset

Download

In this data set, there are a total of 13 dress-related features.

* **Dress\_ID:** This represents the ID number of a particular dress. This is a unique identification number for different dresses.
* **Style:** This represents the style of a particular dress according to the occasion, like a party, a vintage event, etc.
* **Price:** Each dress ID can fall in a particular price bucket, which can be Low, Medium and High.
* **Rating:** This is the average rating given by the customers for a particular dress ID.
* **Size:** The size column represents the majority of the size bought by the customers for that particular dress ID in the previous sale.
* **Season:** This depicts the season in which particular dress is suitable, for example, summer, winter, etc.
* **Neckline:** This contains the type of neck in the dress, like V-neck, round-neck, etc.
* **SleeveLength:** This represents the type of sleeve of the dresses. Half sleeves, full sleeves, cap sleeves, etc.
* **Material:** This contains information regarding which material the dress has been made of, like cotton, nylon, polyester, silk, etc.
* **FabricType:** This contains information regarding the type of fabric of the dress, for example, chiffon, broadcloth, jersey, etc.
* **Decoration:** This represents the kind of decoration around the dress, like ruffles, bow, embroidery, etc.
* **PatternType:** This represents the type of pattern a particular dress has. Pattern may be solid colours, geometric designs, printed or patchwork.
* **Recommendation:** This is the target variable. 'Recommendation' is based on the features and sales of the dress in the previous year. This is either 1 (means yes) or 0 (means no). This represents whether a particular dress is suitable for sale to the customers or not.

Now, moving to the second data set.

You can download the Dress Sales data set from the link below:

Dress Sales Dataset

Download

This particular data set represents the number of sales of a particular dress ID on a certain date, where columns represent the dates on which a particular dress ID has been sold.

Now, based on the above two datasets, you are expected to perform the EDA and draw useful insights from that. Based on the EDA analysis, answer the graded questions for this module.

You have gone through the data cleaning part with an example of bank marketing data set in the previous segments. Now, let’s answer the following questions based on all that you learnt in this session.

You are provided with a blank Jupyter notebook with all comments to perform the operations.

Jupyter Notebook\_Graded Exercise

Download

Question 1/9

Mandatory

#### **Data Types**

You have “Attribute DataSet” which contains a column named “Price”. Choose the correct option for its data type and variable type.

Integer type and numerical variable

Object type and categorical ordinal variable

✓ Correct

Feedback:

*It is the categorical type with a specific order in it; hence, it is an ordinal type of data.*

Object type and categorical nominal variable

Float type and categorical variable

Your answer is Correct.

Question 2/9

Mandatory

#### **Data Types**

There is another column in “Attribute DataSet” named as “Recommendation”, choose the correct statement for its data type and variable type.

Integer type and categorical

✓ Correct

Feedback:

*It is an integer type and has categories in it like 1 and 0. You can check its type using the following code.*

inp0.info()

Object type and categorical

Integer type and continuous numerical

Object type only.

Your answer is Correct.

Question 3/9

Mandatory

#### **Data Types**

Which of the following columns do you think are of no use in the Attribute DataSet?

Dress\_ID

Price

Size and material

NeckLine

None of the above

✓ Correct

Feedback:

*All the columns mentioned above in the dataset are equally important.*

Your answer is Correct.

Question 4/9

Mandatory

#### **Fixing Rows and Columns**

As you can see, there is a column in the Attribute Dataset named 'Size’. This column contains the values in an abbreviated format. Write a code in Python to convert the followings:

M into “Medium”

L into “Large”

XL into “Extra large”

Free into “Free”

S, s and small into “Small”

Now, once you are done with changes in the dataset, what is the value of the lowest percentage, the highest percentage and the percentage of Small size categories in the column named “Size”, respectively?

7.5%, 2.9%, 35.7%

2.9%, 35.7%, 7.5%

✓ Correct

Feedback:

*You should code to convert the values as mentioned in the question and then count the percentages of the categories using value\_counts(). You can refer the following code to perform this operation.*

*Convert the abbreviations of size into words.*

inp0.Size= inp0.Size.replace(['S', 'small', 's'], "Small")

inp0.Size= inp0.Size.replace('free', "Free")

inp0.Size= inp0.Size.replace('M', "Medium")

inp0.Size= inp0.Size.replace('L', "Large")

inp0.Size= inp0.Size.replace('XL', "Extra large")

*Code to find the percentage of each category in the ‘Size’ column.*

inp0.Size.value\_counts(normalize=**True**)

7.5%, 35.7%, 2.9%

2.9%, 7.5%, 35.7%

Your answer is Correct.

Question 5/9

Mandatory

#### **Impute/Remove Missing Values**

You are given another dataset named “Dress Sales”. Now if you observe the datatypes of the columns using ‘inp1.info()’ command, you can identify that there are certain columns defined as object data type though they primarily consist of numeric data.

Now, if you try and convert these object data type columns into numeric data type(float), you will come across an error message. Choose the correct option with the reason for the error and also how to remove this error.

The error occurred because there are some blank entries in such specific columns, and when you convert these null values into float, it shows the error message. To remove this error, you need to remove such null values from the dataset.

Error occurred because there are string entries in such columns. You should replace these string values with null values and then convert it into float type.

✓ Correct

Feedback:

*Because there is one string value in each of these columns, it is showing the error. You cannot convert a string into float type, and once you replace string with a null value, you will be able to convert it into float type.*

*You can convert the string values into null using the following codes:*

inp1.loc[inp1['09-12-2013']== 'Removed',"09-12-2013"] = np.NaN

inp1.loc[inp1['14-09-2013']== 'removed',"14-09-2013"] = np.NaN

inp1.loc[inp1['16-09-2013']== 'removed',"16-09-2013"] = np.NaN

inp1.loc[inp1['18-09-2013']== 'removed',"18-09-2013"] = np.NaN

inp1.loc[inp1['20-09-2013']== 'removed',"20-09-2013"] = np.NaN

inp1.loc[inp1['22-09-2013']== 'Orders',"22-09-2013"] = np.NaN

The error occurred because object type cannot be converted into float type. This error cannot be removed, and you have to deal with the object data types in such columns.

Your answer is Correct.

Question 6/9

Mandatory

#### **Handling Missing Values**

When you see the null counts in “Dress Sales” dataset after performing all the operations that have been mentioned in Jupyter notebook, you will find that there are some columns in “Dress Sales” data where the number of missing values is more than 40%. Based on your understanding of dealing with missing values, select the most appropriate statement, with the reason.

You should impute such missing values with the mean or median of column data.

These columns can be removed because they have a huge number of missing values. Moreover, if you see, the number of sales (numerical values) in such columns is not so high as compared to other Date columns. Hence, it is safe to remove such columns.

✓ Correct

Feedback:

*The huge number of missing values in such columns is one of the criteria, but another major criterion is that these dates have very small sales values and it is safe to remove them. You can refer to the following code to perform this operation:*

inp1= inp1.drop(["26-09-2013"] , axis= **1**)

inp1= inp1.drop(["30-09-2013"] , axis= **1**)

inp1= inp1.drop(["10-02-2013"] , axis= **1**)

inp1= inp1.drop(["10-04-2013"] , axis= **1**)

inp1= inp1.drop(["10-08-2013"] , axis= **1**)

inp1= inp1.drop(["10-10-2013"] , axis= **1**)

Create a separate category named 'missing' for such blank entries in such columns.

✕ Incorrect

Feedback:

*In this scenario, creating a separate category in numerical columns is not a good way to deal with missing values.*

Your answer is Wrong.

Question 7/9

Mandatory

#### **Handling Missing Values**

You need to categorise the dates into seasons in “Dress Sales” data to simplify the analysis according to the following criteria:

* June, July and August: Summer.
* September, October and November: Autumn.
* December, January and February: Winter.
* March, April and May: Spring.

Which of the seasons has the lowest sales among all the seasons, and what is its value?

Spring, 135343

Spring, 143600

✓ Correct

Feedback:

*Write the code to club the dates into seasons according to the criteria and then find the sum of each season in “Dress Sales”. You can refer to the following code to perform such an operation:*

inp1['Spring'] = inp1.apply(**lambda** x: x['09-04-2013'], axis=**1**)

inp1['Summer'] = inp1.apply(**lambda** x: x['29-08-2013'] + x['31-08-2013']+ x['09-06-2013']+ x['09-08-2013']+ x['10-06-2013'], axis=**1**)

inp1['Winter'] = inp1.apply(**lambda** x: x['09-02-2013'] + x['09-12-2013']+ x['10-12-2013'], axis=**1**)

inp1['Autumn'] = inp1.apply(**lambda** x: x['09-10-2013'] + x['14-09-2013']+ x['16-09-2013']+ x['18-09-2013']+ x['20-09-2013']+ x['22-09-2013']+ x['24-09-2013']+ x['28-09-2013'], axis=**1**)

Autumn, 135343

Autumn, 143600

Your answer is Correct.

Question 8/9

Mandatory

#### **Missing values handling**

You can see that there are two types of variables: one with a large number of missing values and another with very less number of missing values. These two columns can be categorised as:

**Type-1**: Missing values are very less (around 2 or 3 missing values): **Price, Season, NeckLine, SleeveLength, Winter and Autumn**

**Type-2**: Missing values are large in numbers (more than 15%): **Material, FabricType, Decoration and Pattern Type**

Based on your understanding, which is the best method to deal with missing values in Type-1 and Type-2 columns?

The missing values of both types of columns can be dropped.

You should create a separate category named as **Missing category** in both types of columns to deal with missing values.

You should create a separate category named as **Missing category** in Type-2 columns and just drop the missing values of Type-1 columns.

✓ Correct

Feedback:

*This is the best way to deal with missing values. You can refer to the following code:*

*Drop the missing values of Type-1 columns:*

inp0 = inp0[~inp0.Price.isnull()]

inp0 = inp0[~inp0.Season.isnull()]

inp0 = inp0[~inp0.NeckLine.isnull()]

inp0 = inp0[~inp0.SleeveLength.isnull()]

inp0 = inp0[~inp0.Winter.isnull()]

inp0 = inp0[~inp0.Autumn.isnull()]

*Create a separate category ‘Missing’ in Type-2 column:*

inp0.Material= inp0.Material.replace(np.nan, "Missing")

inp0.FabricType= inp0.FabricType.replace(np.nan, "Missing")

inp0.Decoration= inp0.Decoration.replace(np.nan, "Missing")

inp0['Pattern Type']= inp0['Pattern Type'].replace(np.nan, "Missing")

Your answer is Correct.

Question 9/9

Mandatory

#### **Standardising Values**

In the given dataset, there are certain discrepancies with the categorical names such as irregular spellings. Choose the correct option of columns with irregular categories and update them.

Season, NeckLine, Price

Price, Material, Rating

fabricType, Decoration, Price

Season, SleeveLength

✓ Correct

Feedback:

*Season and sleeve length contain some spelling mistakes in the categorical names, which need to be rectified. You can refer to the following code to perform this operation.*

*#correcting the season spellings.*

inp0.Season= inp0.Season.replace('Automn', "Autumn")

inp0.Season= inp0.Season.replace('spring', "Spring")

inp0.Season= inp0.Season.replace('winter', "Winter")

*#correcting the SleeveLength.*

inp0.SleeveLength= inp0.SleeveLength.replace(['cap-sleeves', 'capsleeves'], "cap sleeves")

inp0.SleeveLength= inp0.SleeveLength.replace('full', "full sleeves")

inp0.SleeveLength= inp0.SleeveLength.replace(['half','halfsleeve'], "half sleeves")

inp0.SleeveLength= inp0.SleeveLength.replace(['sleevless', 'sleeevless', 'sleeveless', 'sleveless'], "sleeve less")

inp0.SleeveLength= inp0.SleeveLength.replace(['threequarter','threequater', 'thressqatar'], "three quater")

inp0.SleeveLength= inp0.SleeveLength.replace(['turndowncollor','urndowncollor'], "turn down collar")

Your answer is Correct.

# **Summary**

Having completed this session, you must be clear about the various irregularities that can be present in a data set. They can be unfixed rows/columns, missing values, outliers or may even be in the form of un-standard/un-scaled data, etc.

Let’s summarise the steps in Data Cleaning:

* **Fixing the rows and columns:** You need to remove the irrelevant columns and heading lines from the dataset. The irrelevant columns or rows are those that are of absolutely no use for analysis on the data set. Like in the Bank Marketing Dataset, the headers and customer ID columns are of absolutely no use.
* **Remove/impute the missing values:** There are different types of missing values in the dataset. Based on their type and origin, you need to take a decision regarding whether they can be removed if their percentage is too low, or whether they can be considered as a separate category. There is an important possibility where you need to impute missing values with some other value. While doing imputation, one should be very careful because it should not add any wrong information into the dataset. The imputation can be done using mean, median, mode or using quantile analysis.
* **Outlier handling:** Outliers are those points which are beyond the normal trend. There are two types of outliers:
  + **Univariate**
  + **Multivariate**

An important aspect that has been covered is that outliers should not always be treated as anomalies in the dataset. You can understand this using the Bank Marketing Dataset itself, where age has outliers, but these high values of age are as relevant as other values.

* **Standardising values:** Sometimes, there are many entries in the dataset which are not in the correct format. Like you have seen in the Bank Marketing dataset itself, the duration of the call is in seconds and minutes. It has to be in the same format. The other standardisation involves the unit and precision standardisation.
* **Fixing invalid values:** Sometimes, there are some values in the dataset that are invalid, maybe in the form of their unit, range, data type, format, etc. It is essential to deal with these types of irregularities before processing the dataset.
* **Filter data:** Sometimes, filtering out certain details can help you get a clearer picture of the dataset.

It is very important to get rid of such irregularities to be able to analyse a dataset. Otherwise, it may hamper further analysis of the dataset, either while building a machine learning model or in EDA itself.

Now that you are done with the process of data cleaning, the next important step is data analysis. This is covered in the following two sessions:

* Univariate analysis
* Bivariate/multivariate analysis.