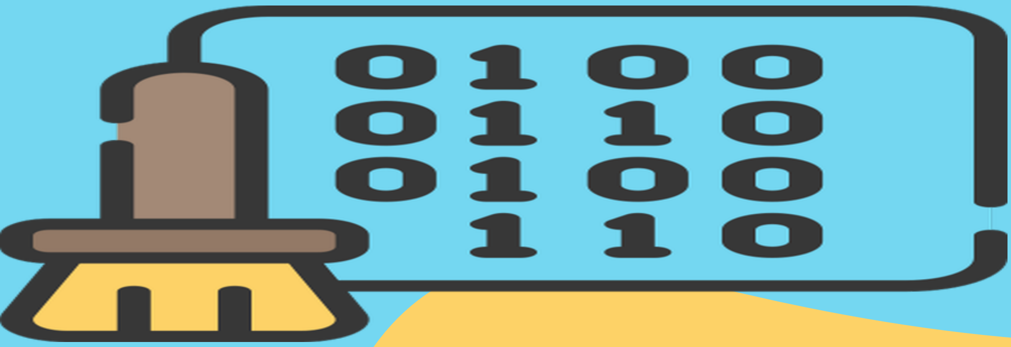
DATA CLEANING IN PHYTON

Key Tasks

- Dropping Columns
- Removing Duplicate Values
- Renaming Columns
- Cleaning Text (Strings)
- Changing Datatypes
- Handling Missing Values
- Feature Engineering
- Merging Columns
- Seperating Data and Rearranging Columns



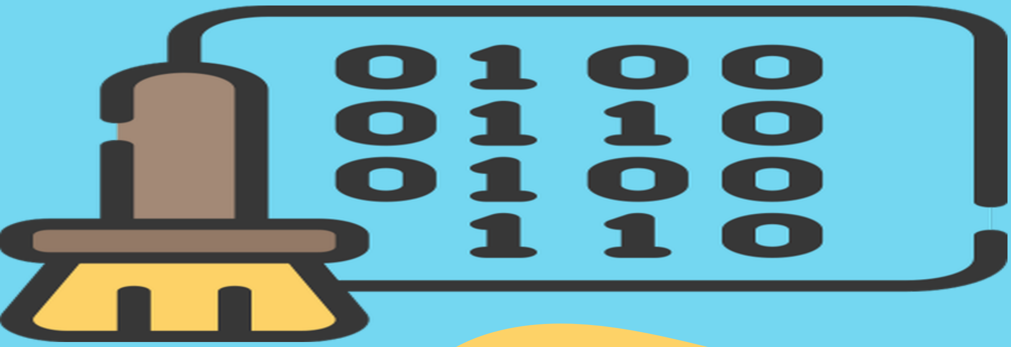
DROPPING COLUMNS

Unnecessary column(s) are dropped from the dataset using; df.drop() method.

In the case of the dirty_data datafame, the Active_Customers column was dropped because it provided no further information for analysis.

```
 dirty_data.drop(['Active_Customer'], axis = 1, inplace = 
True)
```

*axis = 1 : implies actions are to be performed on columns

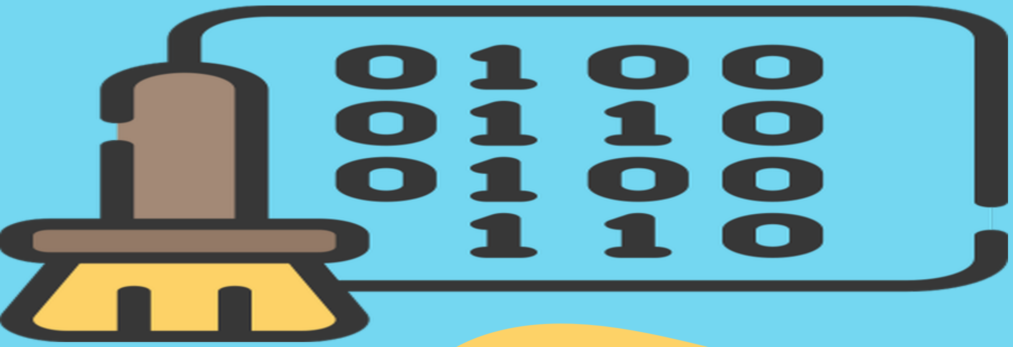


REMOVING DUPLICATES

Duplicate values are usually erroneous repetitions that need to be treated before carrying out analysis.

The df.drop_duplicates() method is used to remove duplicate values. It has 2 arguments
keep: This specifies which of the repeated value to be retained
inplace = True: ensures changes to the df are permanent

```
dirty_data.drop_duplicates(keep = 'first', inplace = True)
```

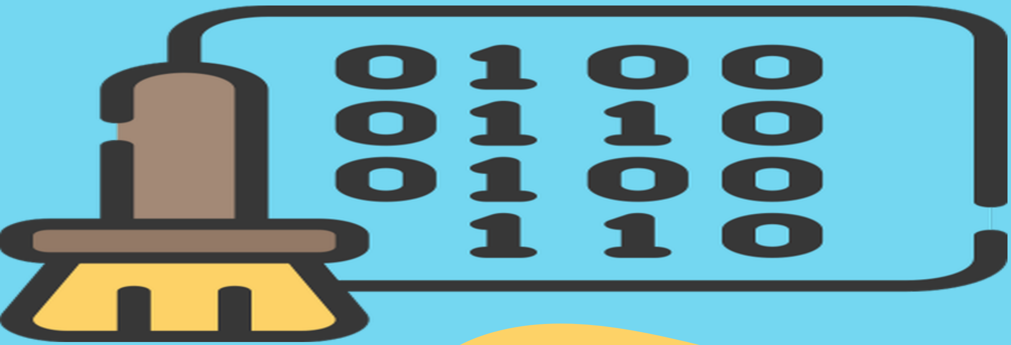


RENAMING COLUMNS

It is good practice to replace column names with easily recalled/appropriate names during the data cleaning phase.

The Number, Number_Purchased, and Method columns were changed to Customer_Id, Quantity, and Payment_Method respectively using the df.rename() method.

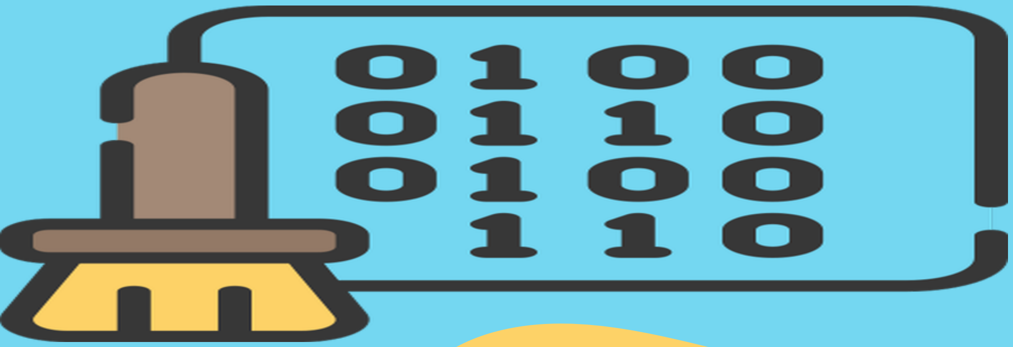
```
dirty_data.rename(columns =
{'old_col_name':'new_col_name'}, inplace = True)
```



CLEANING TEXT (STRINGS)

Texts from columns are cleaned using various string methods str(). String methods help in manipulating text in Python and there are 47 of them.
Another way of cleaning text is using the regular expression module re().

The following str() methods were used in the project
* str.strip() method is used to remove unwanted strings
* str.replace() method is used to change values in columns to allow for uniformity
* str.title() method is used to standardize the format of values in columns

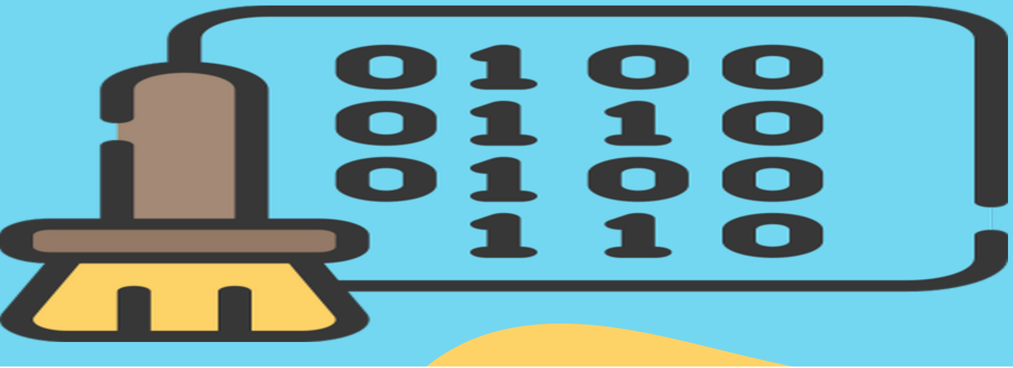


CHANGING DATATYPES

Oftentimes columns are not in the appropriate format This error can be spotted in the EDA phase using the df.info() or df.dtypes methods.

In this project, the Price and Date columns were converted from object to float and datetime datatypes respectively.

```
dirty_data['Price'] = pd.to_numeric(dirty_data['Price'])
dirty_data['Date'] pd.to_datetime(dirty_data['Date'])
```

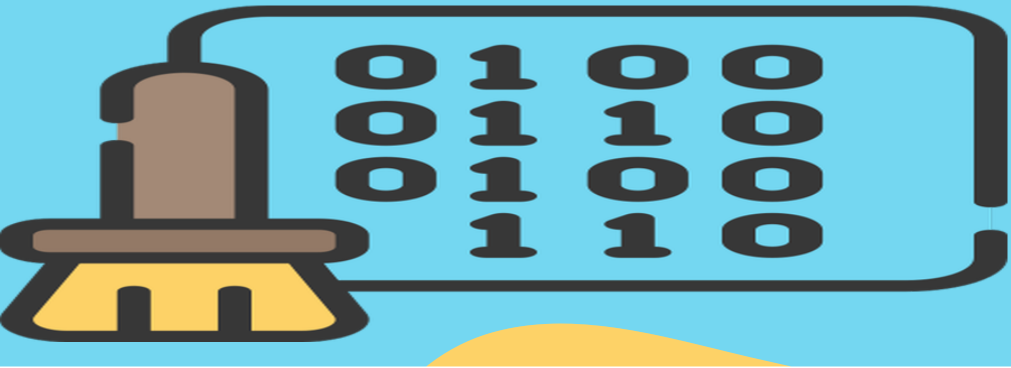


HANDLING MISSING VALUES

One of the first tasks during EDA is checking for missing data in columns using the df.isnull() or df.info() method.

Once missing data is ascertained, the next step is determining how it will be treated. In this project, missing data in the Price column was replaced using the fillna() method. A displot was used in determining the shape (skewness) of the column then the median value was used to replace the missing value.

```
dirty_data['Price'] =
dirty_data['Price'].fillna(dirty_data['Price'].median())
```

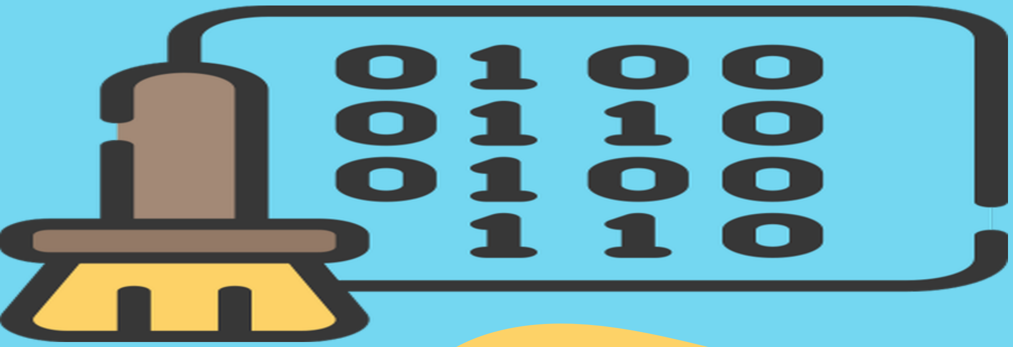


FEATURE ENGINEERING

Feature Engineering in Data Science is the process of creating new features from the dataset by manipulating existing features so as to improve the accuracy of analysis.

For this project, Quantity and Price columns were combined to create the Revenue column. This new column helps in better analyzing sales.

```
df['Revenue'] = dirty_data['Quantity'] * dirty_data['Price']
```



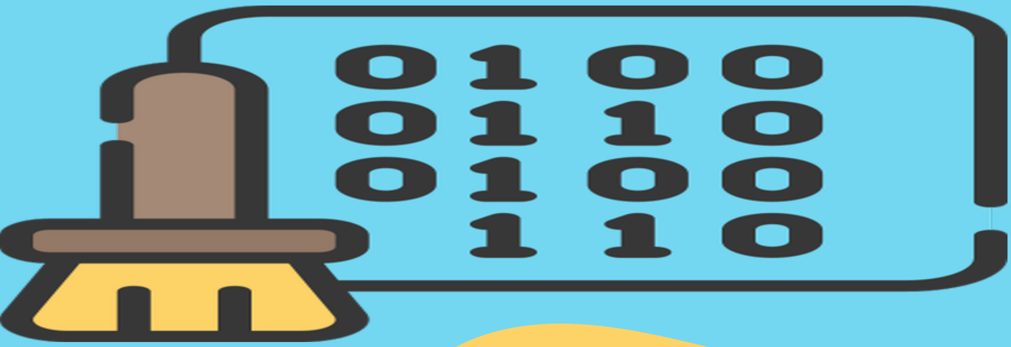
MERGING COLUMNS

Data Cleaning also involves merging columns together so they can create better insights during analysis.

In this project the First_Name and Last_Name columns were merged to form the Customer_Name column using the map() function.

```
df['Customer_Name'] = dirty_data['First_Name'].map(str) +
' ' + dirty_data['Last_Name'].map(str)
```

```
df = dirty_data.drop(columns = ['First_Name', 'Last_Name' ]
axis = 1)
```

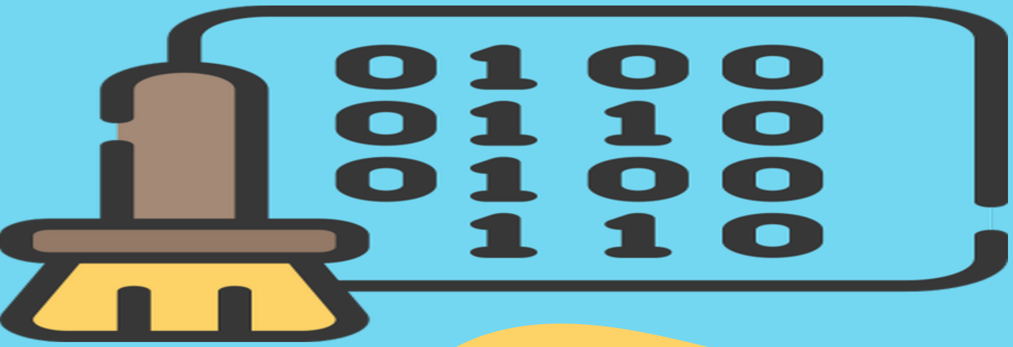


SEPERATING DATA

Sometimes columns contain information that needs to be split into separate columns during the data cleaning process.

The Address column was split into Street Address, City, State_Code and Zip_Code using the str.split() method. This makes it possible to analyse sales on a more detailed level.

```
df[['Street_Address', 'City', 'State_Code, Zip_Code']] =
dirty_data['Address'].str.split(',',3, expand = True)
```

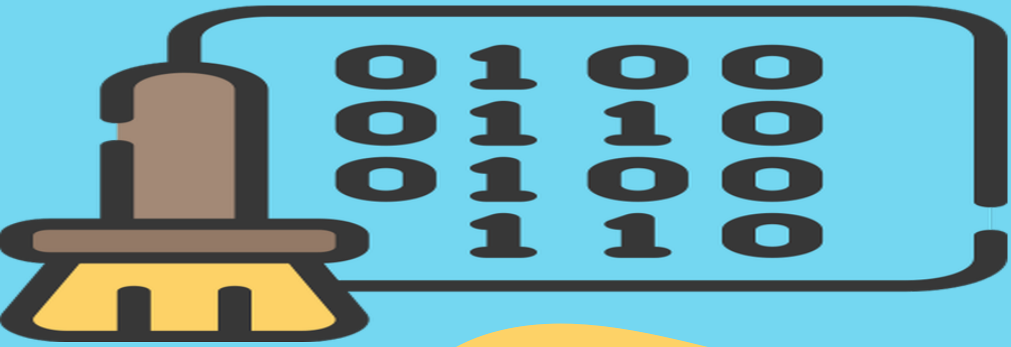


REARRANGING COLUMNS

The last step taken in this project is to rearrange columns closer to their original order and also drop some unneccesary columns.

This step will be carried out by creating a new df called clean_data and selecting only required columns from the old df (dirty_data)

```
clean_data = dirty_data[['col1', 'col2', 'col3',........, 'coln']]
```



QUALITY DATA

After performing these cleaning tasks, the new dataset should have the core attributes of Data Quality stated below:

- Accuracy
- Completeness
- Consistency
- Uniqueness
- Validity

The next stage is then Data Analysis