**Artificial intelligence**

Exploratory Data Analysis (EDA) is the bread and butter of anyone who deals with data. With information increasing by 2.5 quintillions bytes per day (Forbes, 2018), the need for efficient EDA techniques is at its all-time high.

So where is this deluge coming from? The amount of useful information is almost certainly not increasing at such a rate. When we take a closer look, we would realize that most of this increase is contributed by noise. There are so many hypotheses to test, so many datasets to mine, but a relatively constant amount of objective truth. With most data scientists, their key objective is to able to distinguish the signal from the noise, and EDA is the main process to do this.

Enter EDA

In this post, I shall introduce a Starter Pack to perform EDA on the Titanic dataset using popular Python packages: pandas, matplotlib, seaborn, and scikit-learn.

For code reference, you can refer to my GitHub repository here.

Data for Communication

Generically, we visualize data for two primary reasons:

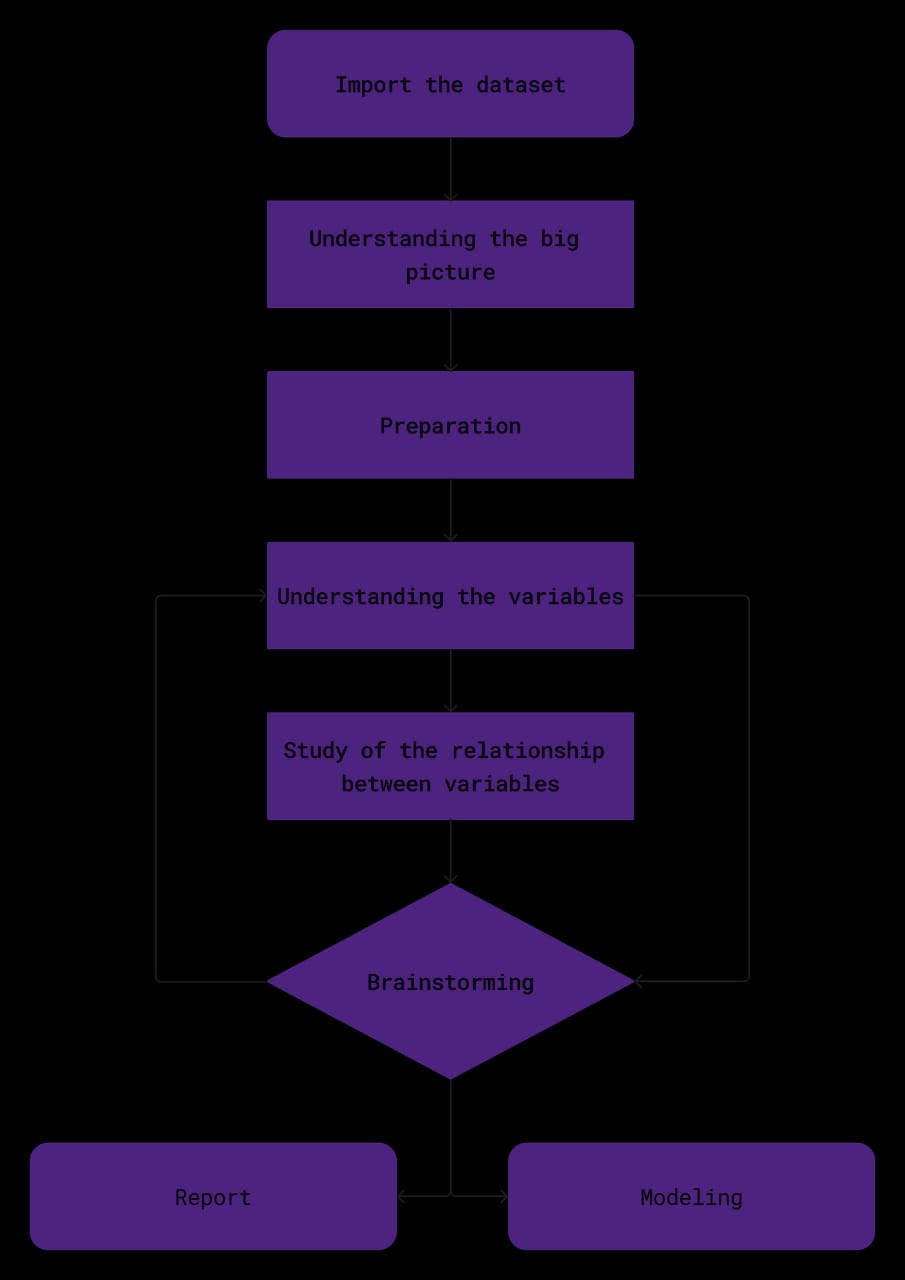
To understand (EDA)

To communicate

In the last section, I shall share some useful dashboarding guidelines that would aid you in convey the results of your analysis clearly and effectively.

Outline:

What is Data

Categorical Analysis

Quantitative Analysis

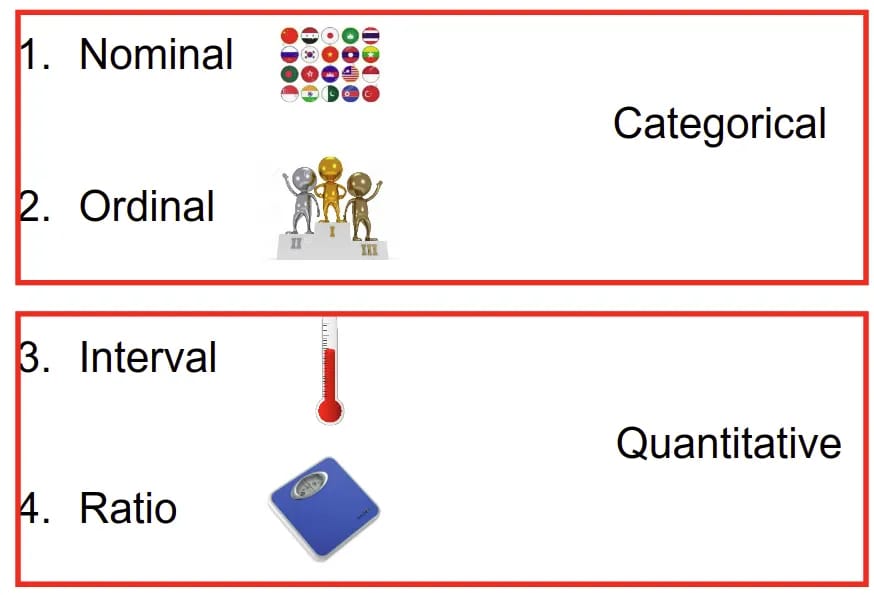
Clustering

Feature Importance by Tree-based Estimators

Dashboarding Techniques

1. What is Data

First and foremost, some theory. The word “data” was first used to mean “transmissible and storable computer information” in 1946 (source). At the highest level, the term data can be broadly categorized under two umbrellas: structured and unstructured. Structured data are pre-defined data models that normally reside in your relational database or data warehouse that has a fixed schema. Common examples include transaction information, customers’ information, and dates. On the other hand, unstructured data have no pre-defined data model and are found in NoSQL databases and data lakes. Examples include images, video files, and audio files.

In this post, we would focus on structured data, where I would propose a systemic approach to quickly show latent statistics from your data. Under the umbrella of Structured data, we can further categorize them to categorical and quantitative. For Categorical data, the rules of arithmetics do not apply. In the categorical family, we have nominal and ordinal data, while in the Quantitative family, we have interval and ratio. It is important that we take some time to clearly define and understand the subtle, yet significant differences each term is from the other as this would affect our analysis and preprocessing techniques later.

4 Different types of Data

Nominal data

The name “nominal” comes from the Latin word, nomen, which means name. Nominal data are objects which are differentiated by a simple naming system. An important thing to note is that nominal data may also have numbers assigned to them. This may appear ordinal (definition below), but they are not. Numbered nominal data are simply used to capture and reference. Some examples include:

a set of countries.

the number pinned on a marathon runner.

Ordinal data

Ordinal data are items in which their order matters. More formally, their relative positions on an ordinal scale provide meaning to us. This may indicate superiority or temporal positions, etc.. By default, the order of ordinal data is defined by assigning numbers to them. However, letters or other sequential symbols may also be used as appropriate. Some examples include:

the competition ranking of a race (1st, 2nd, 3rd)

the salary grade in an organization (Associate, AVP, VP, SVP).

Interval data

Similar to ordinal data, interval data is measured along a scale in which each object’s position is equidistant from one another. This unique property allows arithmetic to be applied to them. An example is

the temperature in degrees Fahrenheit where the difference between 78 degrees and 79 degrees is the same as 45 degrees and 46 degrees.

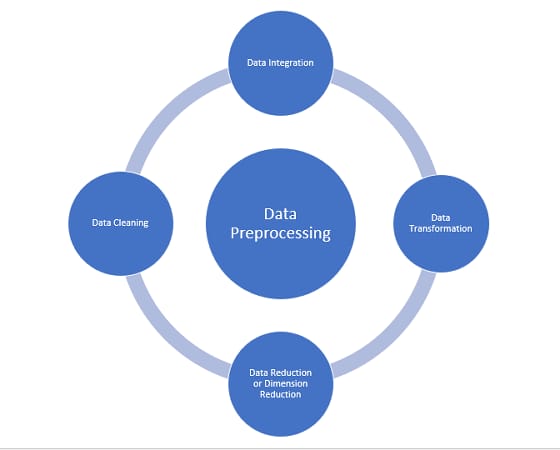
Ratio data

Like Interval data, the differences in Ratio data are meaningful. The Ratio data has an added feature which makes the ratios of the objects meaningful as well, and that is that they have a true zero point. Zero represents the absence of a certain property. So when we say something is of zero weight, we mean that thing has an absence of mass. Some examples include:

the weight of a person on a weighing scale

Interval vs Ratio

The difference between interval and ratio is just one does not have a true zero point while the other does have. This is best illustrated with an example: When we say something is 0 degrees Fahrenheit, it does not mean an absence of heat in that thing. This unique property makes statements that involve ratios such as “80 degrees Fahrenheit is twice as hot as 40 degrees Fahrenheit” not hold true.



Back to outline

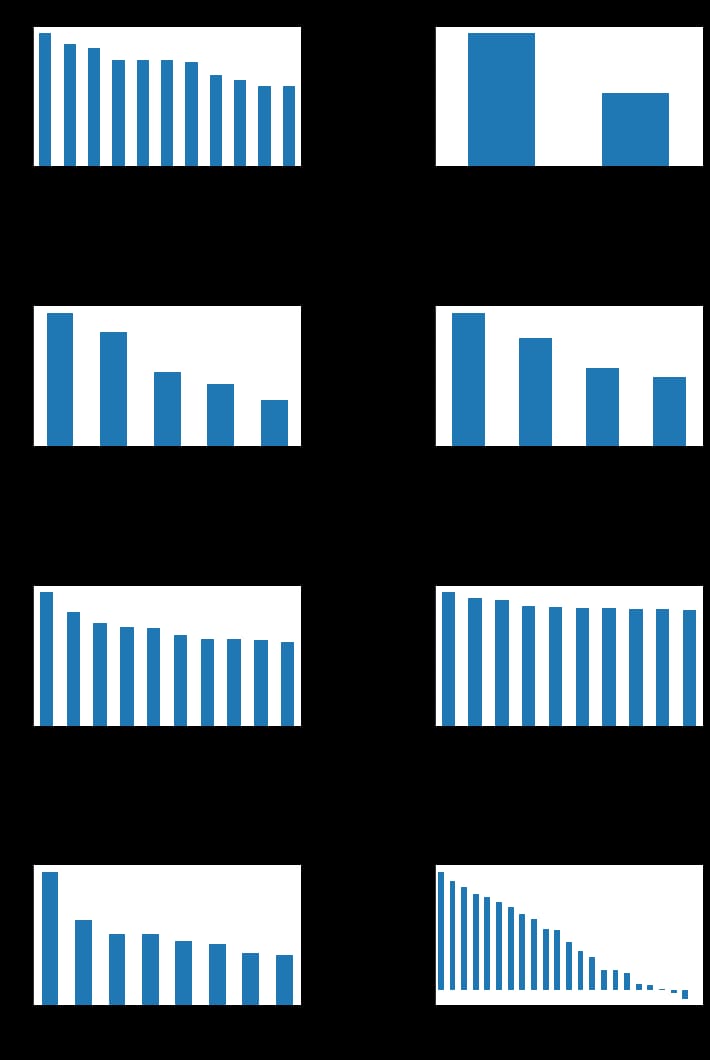
Before we delve into the other sections, I would like to formalize some concepts so you would be clear on the thinking process to why we are doing things shown below.

Let me begin by saying that the best way to quickly show summaries of your data is through 2D plots. Despite living in a 3D spatial world, humans find it difficult to perceive the 3rd dimension, e.g. depth, needless to say, a projection of a 3D plot on a 2D screen. Hence, in the subsequent sections, you would see that we only use bar graphs for Categorical data, and box plots for Quantitative data as they succinctly express the data distributions respectively. We would only be focusing on univariate analysis and bivariate analysis with the target variable. For more details on Dashboarding Techniques, please refer to Section 6 of the article.

We would be mainly using seaborn and pandas to accomplish this. As we all know, statistics is an essential part of any data scientist’s toolkit and seaborn allows quick and easy use of matplotlib to beautifully visualize the statistics of your data. matplotlib is powerful, but it can get complicated at times. Seaborn provides a high-level abstraction of matplotlib allowing us to plot attractive statistical plots with ease. To make the best use of seaborn, we would also need pandas as seaborn works best with pandas’ DataFrames.

Also, if you want to follow along with the coding, be sure to download the data and set up your environment right. You can find the instructions in the README.MD file in my GitHub repo.

With these being said, let’s begin and get our hands dirty!



2. Categorical Analysis

We can start reading the data using pd.read\_csv() . By doing a .head() on the data frame, we could have a quick peek at the top 5 rows of our data. For those who are not familiar with pandas or the concept of a data frame, I would highly recommend spending half a day going through the following resources:

Pandas documentation

What is a DataFrame?

Other useful methods are .desribe() and .info() where the former would show:

Output of .describe() method on a data frame

And the latter would show:

Output of .info() method on a data frame

We see now that,

Categorical data:

PassengerId,

Survived,

Pclass,

Name,

Sex,

Ticket,

Cabin,

and Embarked

while Qualitative data:

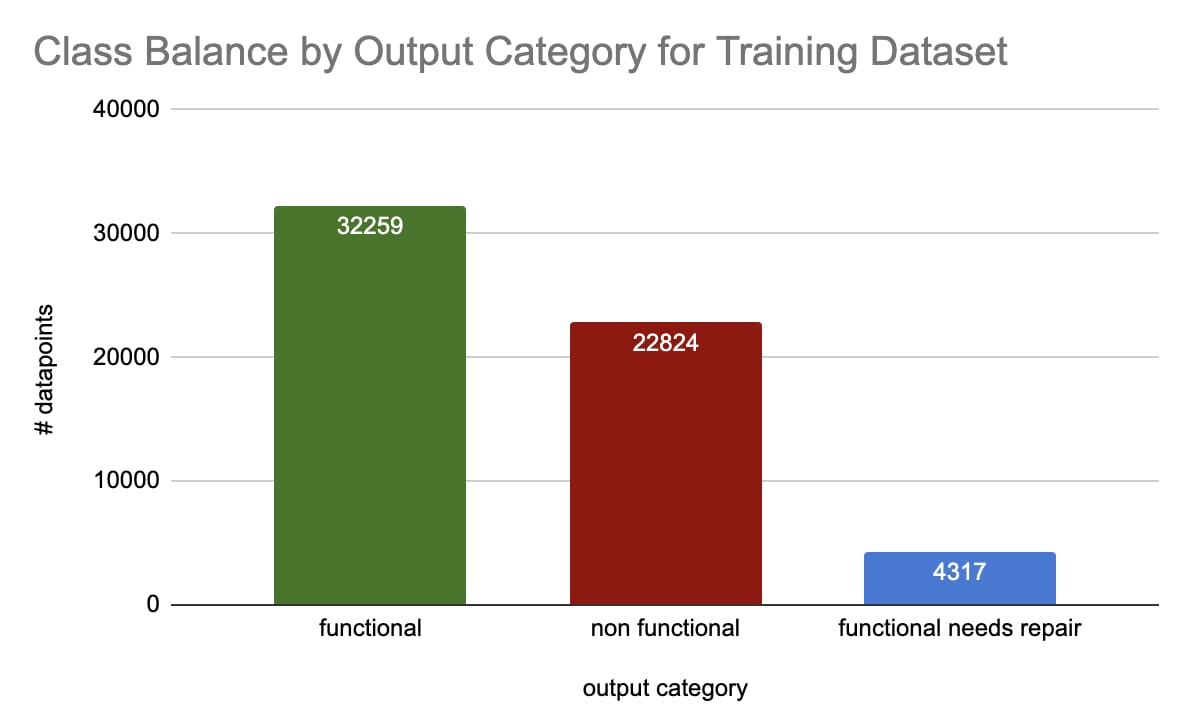
Age,

SibSp,

Parch,

and Fare

Now, with this knowledge and what we have learned in Section 1, let’s write a custom helper function that can be used to handle most kinds of Categorical data (or at least attempt to) and give a quick summary of them. We shall do these with the help of some pandas methods and seaborn’s .countplot() method. The helper function is called categorical\_summarized and is shown below.



But by looking at how many “unique” numbers come up per feature, we can guess whether or not a numerical feature may be categorical.

Df.select\_dtypes(include=’number’).nunique()

The number of unique values per numerical feature

Here, we can see that there are five features that have an abnormally low number of unique values. That is indicative that those numerical values actually stand for categorical values.

Looking at the dataset documentation, we find that three of the five features: ‘num\_private’, ‘region\_code’, ‘district\_code’ are geographic codes whose numerical values aren’t relevant for our challenge. Let’s convert these into strings so that we can play with them in the next sections.

Df[‘num\_private’] = df.num\_private.astype(str)

Df[‘region\_code’] = df.region\_code.astype(str)

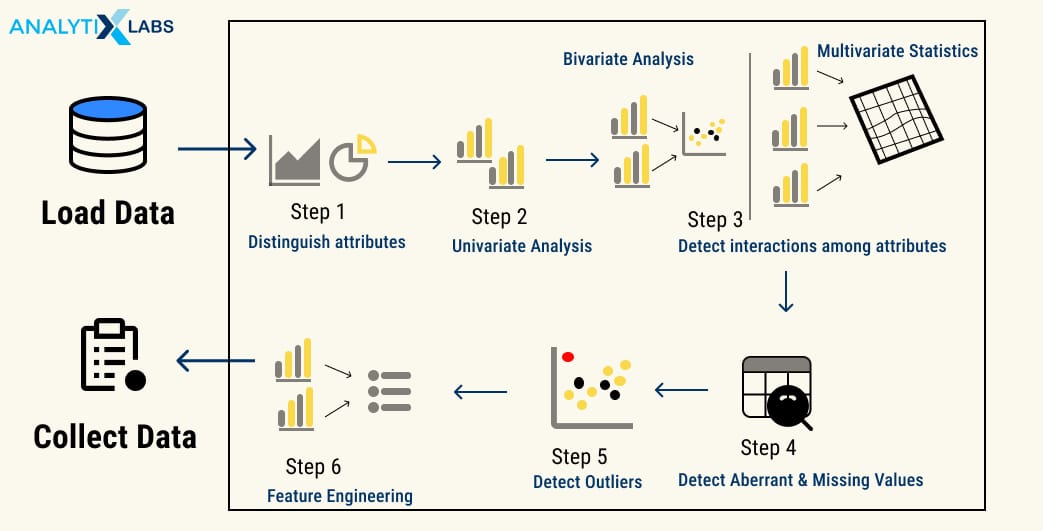
Df[‘district\_code’] = df.district\_code.astype(str)

Drop categorical features with too many categories

Categorical features with too many categories are usually not useful for predictive models. Let’s find how many categories each categorical features have.

Uniques = df.select\_dtypes(exclude=’number’).nunique()

Uniques



What is Exploratory Data Analysis (EDA)?

EDA is the first step when developing a data model or analyzing data. EDA encompasses all the various subtasks that allow you to understand the data you are dealing with. These tasks include describing, summarizing, visualizing, and understanding the relationship between data features.

Therefore, as the term suggests, EDA refers to all the steps undertaken by a data science professional to “explore” and consequently understand the traits of data and its variables.

EDA typically aims to answer specific preliminary questions before any significant changes or manipulation can be done to the data and be fed to a predictive model.

These questions include-

What kind of data am I dealing with?

Here the user tries to identify the fundamental characteristics of the data, such as its file format, volume, number of rows and columns, metadata, structure, type and data types of columns, etc.)

What is the complexity level of the data?

Data may comprise many files and be connected through primary and secondary keys. The user, therefore, needs to understand such relations and find if the data is nested or if some fields have nested data.

Does the data serve my purpose?

Data is extracted to serve a purpose. For example, if you want to develop a predictive model that forecasts Sales, then the data should have the sales number for the model to analyze patterns and extrapolate them. Therefore the data must be able to answer the business questions for which the data is being used.

Is my data clean?

Users need to check if the data is unclean or not. Unclean data refers to data that has missing values, outliers, incorrect data types, unconventional column names, etc. The identification of such drawbacks in the data allows users to mark them and solve these issues at a later stage.

What is the relationship between the features of the data?

Predictive models often have multicollinearity issues while seeking a strong relationship between the independent features and the dependent variables. Therefore, before starting, a user needs to understand the features’ relationship among themselves and how strong this relationship is.

While these are some of the critical questions that EDA tries to answer, it is not an exhaustive list. Let us look at the motivation behind performing EDA.

Motivation for Exploratory Data Analysis

EDA doesn’t describe the data just for a better understanding of it. Instead of just describing, summarizing, and visualizing the data, it achieves multiple goals crucial in the later steps of Data Preprocessing, Feature Engineering, Model Development, Model Validation, Model Evaluation, and Model Deployment.